Model-Based Face Tracking for View-Independent Facial Expression Recognition

Salih Burak Gokturk[1] Jean-Yves Bouguet[3] Carlo Tomasi[2] Bernd Girod[1]

[1]Electrical Engineering,[2]Computer Science
Stanford University
Stanford, CA 94305, USA

[3]Microprocessor Research Labs
Intel Corporation - SC12-303
Santa Clara, CA 95052, USA

Abstract

Facial expression recognition is necessary for designing any realistic human-machine interfaces. Previous published facial expression recognition systems achieve good recognition rates, but most of them perform well only when the user faces the camera and does not change his 3D head pose. In this study, we propose a new method for robust, view-independent recognition of facial expressions that does not make this assumption. The system uses a novel 3D model-based tracker to extract simultaneously and robustly the pose and shape of the face at every frame of a monocular video sequence. There are two main contributions of this paper. First, we demonstrate that the 3D information extracted through 3D tracking enables robust facial expression recognition in spite of large rotational and translational head movements (up to 90 degrees in head rotation). Second, we show that Support Vector Machine is a suitable engine for robust classification. Recognition rates as high as 91 percent are achieved at classifying 5 distinct dynamic facial motions (neutral, opening/closing mouth, smile, raising eyebrow).

1 Introduction and Previous work

In search for the "perfect" human-machine interface, many computer vision researchers have been working on automatic detection, tracking and recognition of the whole or parts of the face [20, 14]. Most facial expression recognition systems developed so far require that the subject faces the camera and does not change his 3D pose. In this paper, we propose a novel scheme for facial expression recognition that is coupled with a novel 3D model-based face tracker in a monocular video sequence. This technique enables classification of dynamic facial expressions while the subject is free to move his head in front of the camera. Following this approach, pose and shape characteristics are naturally factored into two separated signature vectors through tracking, leading to good facial expression classification rates even under extreme head pose configurations.

The initial 2D methods for facial expression recognition suffer from a high degree of dependence upon camera viewing angle [16, 3]. In [5], Chen *et al.* use learning subspace method on features obtained from images of subparts of the face. In [22], Wang *et al.* use 19 point 2D feature tracker for the recognition of three emotional expressions. In [14], Lien *et al.* use feature tracking with partial affine transformation compensation in order to recognize the movement directions of action units. In [17], Sako and Smith use color matching and template matching to find the positions of important 2D features on the face and use the dimension and position information about these features for expression recognition. Another 2D method is due to Hara and Kobayashi in [12] where they use scanline brightness distribution to detect six different expressions. One of the main motivations behind 3D techniques for face or expression recognition is to be able to succeed in a broader range of camera viewing angles. In [9], Essa and Pentland developed a system for observing dynamic facial motions using model based optical flow method. They show that their system is suitable for coding, analysis and recognition of facial expressions.

Tracking of the face can provide information regarding the pose and shape of the face, which can further be used for expression recognition. A large number of researchers [13, 10, 15, 19, 1, 2, 11] have studied optical flow tracking. Black and Yacoob uses the flows of local 2-D face features for recognition purposes. The method does not explicitly cover a large angular degree of head rotations.

0-7695-1602-5/02 $17.00 © 2002 IEEE

More recently, DeCarlo and Metaxas [7, 6] 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 [8] tracked the human face in a video sequence using a similar model-based approach and used the tracking results for object based compression. These 3D methods [8, 7, 11] have potential to produce view-independent recognition systems.

We introduced a model-based, markerless face tracker in [11]. Here, we propose to apply this tracker for the purpose of classifying facial expressions. The recognition system consists mainly of two components: a training stage and a testing stage. In the training stage, the three dimensional deformable model of the subject's face is built using a stereo system. The shape vectors obtained through this stage are used to train the support vector machine classifier. In the testing stage, the face of the subject is tracked through a monocular sequence in three dimensions using the compact deformable shape model computed after training. The 3D deformation of the face is then encoded in a form of a small vector of scalars (also called shape vector) that is used by the support vector machine classifier to recognize the facial expression at every frame of the sequence.

The rest of the paper is organized as follows. In Section 2, we describe the components of the system. In Section 3, two set of experiments, one with five dynamic facial motions and the other with three emotional expressions are presented. In Section 4, we discuss possible directions for future work and conclude.

2 Description of the System

Figure 1 shows the flowchart of the system. Similar to other statistical models, the system is composed of training and testing stages. Stereo tracking serves as a basis for building the three dimensional deformable model of the subject's face. Inherently, the learned models serve as the training data for support vector machine classifier. Monocular tracking uses the learned model for robust tracking of face pose and shape. The shape vector is subsequently used by the support vector machine classifier to assign particular probability of each expression.

An overview of the face model used for tracking is given in Section 2.1. We describe the details of the tracking system in Sections 2.2 and 2.3. Section 2.4 describes the Support vector machine classifier, and presents its application to our multi-class problem.

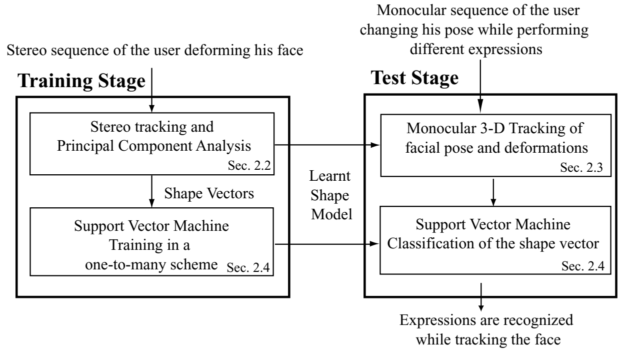

Stereo sequence of the user deforming his face
Training Stage
Stereo tracking and
Principal Component Analysis
Sec. 2.2
Shape Vectors
Support Vector Machine
Training in a
one-to-many scheme
Sec. 2.4
Learnt
Shape
Model
Monocular sequence of the user
changing his pose while performing
different expressions
Test Stage
Monocular 3-D Tracking of
facial pose and deformations
Sec. 2.3
Support Vector Machine
Classification of the shape vector
Sec. 2.4
Expressions are recognized
while tracking the face

Figure 1. The main components of the algorithm.

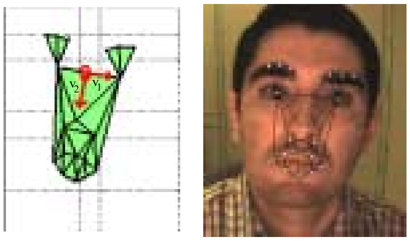
2.1 Deformable F aceModel

The face is modeled by a collection of $N = 19$ points P_i $(i = 1, \ldots, N)$. See Figure 2. We define the face reference frame as a reference frame attached to the head of the user. Let $\vec{X}^i(n)$ and $\vec{X}^i_c(n)$ be the coordinate vectors of a generic point P_i at frame n in the face and camera reference frames respectively. Those two 3-vectors are related to each other through a rigid body transformation characterizing the pose of the user's face with respect to the camera: ${}^i_c(n) = \vec{R}(n)\vec{X}^i(n) + \vec{t}(n)$, where $\vec{R}(n)$ and $\vec{t}(n)$ are the rotation matrix and translation vector defining respectively the orientation and the absolute position of the center of the face in the camera reference frame. Of course, since $\vec{R}(n)$ is a rotation matrix, it is uniquely parameterized by a 3-vector $\omega(n)$ also known as rotation vector(see [10]). The problem of tracking the face in a monocular sequence corresponds then to estimating the quantities $\vec{X}^i(n)$ (shape), and $\omega(n)$ and $\vec{t}(n)$ (pose) for all points P_i $(i = 1, \ldots, N)$ for all frame numbers n.

As expressed in its most general form, it is easy to show that this estimation problem is unsolvable from a monocular observation. For example, one may pick any rigid pose parameters $\{\omega(n), \vec{t}(n)\}$, and there will always exist a shape $\{\vec{X}_i(n)\}$ that will result into the same projected points on the image. However, we can assume some more constraints on the shape unknown in order to make the problem solvable. Let $\vec{X}(n)$ be the resulting $3N \times 1$ vector after stacking the coordinates $\vec{X}^i(n)$ of all of the points at time n such that $\vec{X}(n) = [\vec{X}^1(n) \ldots \vec{X}^N(n)]^T$. The key constraint that makes monocular tracking possible is to assume that at any time n in the sequence, the whole shape coordinate vector $\vec{X}(n)$ is a linear combination of a small number of (known) $3N-$vectors $\vec{X}_0, \vec{X}_1, \ldots, \vec{X}_p$ $(p \ll 3N)$:

$$\vec{X}(n) = \vec{X}_0 + \sum_{k=1}^{p} \alpha_k(n)\vec{X}_k, \qquad (1)$$

where the vectors $\vec{X}_k$, $(k = 0, \ldots, p)$ are not functions of the frame number n. The p scalar coefficients



(a) (b)

Figure 2. (a) The $N = 19$ point mesh and the two vectors used to align the mesh across subjects. (b) The mesh projected on the image of the face.

$\alpha_k(n)$ are the only entities that allow for non-rigidity of the $3D$ shape across time. As a result, the $p-$vector $\alpha(n) = [\alpha_1(n)\ \alpha_2(n)\ \dots\ \alpha_p(n)]^T$ is called the shape vector. The integer p is referred to as the dimensionality of the deformation space. The shape $\vec{X}_0$ is referred to as the 'neutral shape' (at resting position) and the other p vectors $\vec{X}_k$ as the principal movement directions. The shape vector $\alpha(n)$ carries the facial expression information independent of 3D pose, thus will be our main resource for facial expression recognition.

In this present work, we propose to use a different mean shape vector $\vec{X}_0$ per subject to account for intrinsic geometric variations between faces, but the same set principal movement direction vectors $\vec{X}_1, \dots, \vec{X}_p$ for every individual. This corresponds to assuming that after subtraction of the mean shape, every individual has approximately the same modes of facial deformation. As a result, when dealing with a generic user u, we will sometimes denote his corresponding mean shape $\vec{X}_0^u$.

Given this new formalism, the monocular tracking algorithm answers the problem of estimating the deformation vector $\alpha(n)$, and the pose parameters $\omega(n)$ and $\vec{t}(n)$ at every frame. This procedure is described in Section 2.3. However, prior to monocular tracking, it is necessary to compute the average shape ($\vec{X}_0$), and the principal movement direction vectors ($\vec{X}_k$'s) as discussed in Section 2.2.

2.2 Stereo T raking

We propose to estimate the principal movement direction vectors $\vec{X}_k$, $k = 0, \dots, p$ from real tracked stereo data. Here, a stereo sequence is captured while the user makes a variety of facial expressions without changing his head pose. Having initialized the 19 point mesh given in figure 2, the points are tracked on the stereo image streams using standard optical flow techniques [15, 19] constrained by space energy functions as described in [11]. The outcome of this procedure is the 3D trajectory of each point P_i throughout the entire sequence.

After stereo tracking, the p shape basis vectors $\vec{X}_k$, $k = 0, \dots, p$ are computed using the Singular Value Decomposition (SVD) [4] of the 3D shape trajectory matrix Here, we choose to compute the basis shape vectors from several individuals. For this purpose, each subject is tracked in separate stereo sequences, and then all shape trajectories are registered (or aligned) in a consistent reference frame (figure 2). Once the shape sequences of K users are aligned, each neutral shape X_0^u ($u = 1, \dots, K$) is estimated as the initial face shape of the sequence belonging to individual u: $\vec{X}_0^u = \vec{X}^u(1)$. where $\vec{X}^u(n)$ is now the 3D mesh that belongs to individual u at frame n (after 3D alignment). Observe that this way, we assume that every stereo sequence starts with the user being at rest position. Next, the neutral shape is subtracted from the whole aligned trajectory of user u. Each resulting residual shape trajectory $\tilde{\vec{X}}^u(n)$ for user u becomes $\tilde{\vec{X}}^u(n) = \vec{X}^u(n) - \vec{X}_0^u$. The new shape trajectory $\tilde{\vec{X}}^u(n)$ is then used to build the following matrix:

$$\vec{M} = \left[\tilde{\vec{X}}^1(1)\ \ \tilde{\vec{X}}^1(2)\ \ \cdots\ \ \tilde{\vec{X}}^1(N_1)\ \ \tilde{\vec{X}}^2(1)\ \ \cdots\ \ \tilde{\vec{X}}^K(N_K)\right],$$

where N_u is the length of the tracked sequence corresponding to user u. Next, applying Singular Value Decomposition (SVD) on $\vec{M}$, we obtain $\vec{M} = \vec{U}\,\vec{S}\,\vec{V}^T$ where $\vec{U}$ and $\vec{V}$ are two unitary matrices and $\vec{S}$ is the diagonal matrix of the positive and monotonically decreasing singular values σ_k. The first p column vectors of U give the principal movement directions $\vec{X}_k$ ($k = 1, \dots, p$). Following this decomposition, we approximate the generic shape $\vec{X}$ of user u as the sum of its neutral shape $\vec{X}_0^u$ and a linear combination of the p principal movement direction vectors

$$\vec{X} = \vec{X}_0^u + \sum_{k=1}^{p} \alpha_k \vec{X}_k. \qquad (2)$$

Once the basis shape vectors are computed, it is straightforward to compute the blending coefficients $\alpha_k(n)$ corresponding to every frame number n. This is done by projecting orthogonally each residual shape $\tilde{\vec{X}}^u(n)$ onto the basis vectors $\vec{X}_k$ ($k = 1, \dots, p$) by a standard scalar product operator:

$$\alpha_k(n) = \langle \tilde{\vec{X}}^u(n), \vec{X}_k \rangle,$$

The sequence of shape vectors $(\alpha_k(i)'s)$ associated with every frame i of the stereo sequence is then used to train the Support Vector Machine classifier (Section 2.4).

2.3 Model-Based Monocular T raking

In its original form, optical flow tracking computes the translational displacement of a particular point in the image given two successive frames (see [15, 19]). 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 (1)), and the parameters defining the model configuration are estimated all at once from image measurements. In our case, those parameters are $\alpha(n)$ for shape and $\{\omega(n), \vec{t}(n)\}$ for pose. Assuming 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 to estimate the optimal pose $\{\omega(n) = \omega(n-1) + d\omega(n), \vec{t}(n) = \vec{t}(n-1) + d\vec{t}(n)\}$ and deformation $\alpha(n) = \alpha(n-1) + d\alpha(n)$ of the face model that best fit the subsequent frame I_n. For that purpose, let us define a cost function $\vec{C}_n$ whose minimum is achieved at the tracking solution

$$\vec{C}_n = \sum_{i,ROI} \eta^i \left\{ \begin{array}{c} (1-\epsilon)\left(I_n(\vec{x}_n^i) - I_{n-1}(\vec{x}_{n-1}^i)\right)^2 \\ +\epsilon\left(I_n(\vec{x}_n^i) - I_1(\vec{x}_1^i)\right)^2 \end{array} \right\} \quad (3)$$

$$\vec{x}_n^i = \pi_i\left(\alpha(n), \omega(n), \vec{t}(n)\right), \quad (4)$$

where π_i is the model-based image projection map of the face mesh vertex P_i (pinehole model) that is function only of the shape and pose parameters $\{\alpha(n), \omega(n), \vec{t}(n)\}$. The summation in Equation (3) is done over small pixel windows (ROI) around every image point $\vec{x}_n^i$, $\vec{x}_{n-1}^i$ and $\vec{x}_1^i$. The scalar η^i is the cosine of the angle between the point surface normal and camera viewing direction. As a result, points that face away from the camera are not weighted as much as the frontal points. Observe that the first term in Equation (3) is the standard matching cost used in the Shi-Tomasi-Kanade feature tracker [15, 19] (tracking cost). The second term however measures the image mismatch between the current image I_n and the first image I_1 in the sequence enforcing every facial feature to appear the same from the beginning to the end of the sequence (monitoring cost). 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 $d\omega(n)$, $d\vec{t}(n)$ and $d\alpha(n)$. This is done by setting the derivative of $\vec{C}_n$ with respect to $d\alpha(n)$, $d\omega(n)$ and $d\vec{t}(n)$ to zero

$$\frac{\partial \vec{C}_n}{\partial \vec{s}} = 0, \qquad \text{where} \qquad \vec{s} = \begin{bmatrix} d\alpha(n) \\ d\omega(n) \\ d\vec{t}(n) \end{bmatrix}. \quad (5)$$

The solution to Equation (5) is obtained by gradient descent iterations as discussed in detail in [11]. The vector $[d\alpha(n) \;\; d\omega(n) \;\; d\vec{t}(n)]^T$ provides a solution that characterizes the differential of pose and deformation between frame $n-1$ and frame n. It has to be noted that this tracking method solves for the shape vector α independently from the pose of the subject. This makes the 3D tracking approach very suitable for facial expression recognition.

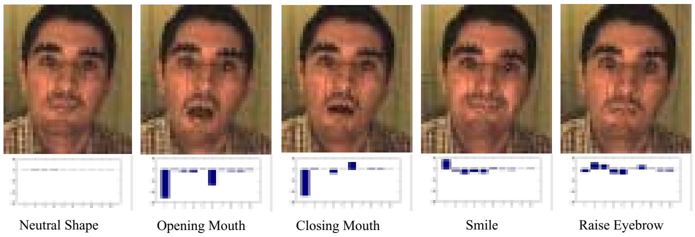


Figure 3. Five different dynamical motions with the corresponding shape vectors.

Input Expression	*Decision*				
	N.	O. M.	C. M.	S.	R. E.
Neutral(N.)(44)	32	6	3	0	3
Opening Mouth(O.M.)(80)	0	76	4	0	0
Closing Mouth(C.M.)(50)	0	1	49	0	0
Smile(S.)(87)	2	0	0	81	4
Raise Eyebrow(R.E.)(21)	3	0	0	0	18

Table 1. The input expression vs the decision of the system

2.4 Statistical Classification Using Support Vector Machine Classification

For recognizing dynamic expressions (e.g. opening/closing month), it is necessary to augment the shape vector $\alpha(n)$ with its first temporal derivative $\dot{\alpha}(n)$. Numerically, a finite difference equation is used to approximate the derivative operator. The resulting $2p \times 1$ feature vector at frame n is:

$$x(n) = \begin{bmatrix} \alpha(n) \\ \dot{\alpha}(n) \end{bmatrix} = \begin{bmatrix} \alpha(n) \\ \alpha(n) - \alpha(n-3) \end{bmatrix}. \quad (6)$$

Observe that, in order to remain robust with respect to noise in tracking, the difference is calculated using a baseline of three frames. In practice, other kernels for differentiation could be applied. The resulting vectors $x(n)$ are the feature vectors for support vector machine classifier. For clarity purposes, we will sometimes drop the frame number n in this section to denote the feature vector.

The first step of statistical classification is to identify the optimum classifier that corresponds to the facial feature vectors $x(n)$ calculated from the stereo sequences (training data). The goal is then to find a separation function that can be induced from the known data points (feature vectors from stereo tracking) and generalizes well on the unknown examples (feature vectors from monocular tracking). Without loss of generality, let us first consider two-class classification problem, i.e. classification between a particular expression vs. all other expressions. Proposed first by Vapnik [21], the SVM classifier aims to find the optimal differentiating hyperplane between the two classes. The optimal hyperplane is the one that not only correctly classifies the data, but also maximizes the margin of the closest data

	SVM with kernel erbf	SVM with kernel rbf	Clustering	N-Nearest with N=9	N-Nearest with N=5
Same Person	176/182	170/182	161/182	173/182	173/182
Total Performance	256/282	253/282	242/283	255/282	253/282

Table 2. The total number of correctly classified frames for five different classification algorithms.

points to the hyperplane.

Mathematically, we consider the problem of separating the training set S of points $x_i \in R^n$ with $i = 1, 2, \ldots, N$. Each data point x_i belongs to either class and thus is given a label $y_i \in \{-1, 1\}$. Support vector machines (SVM) [21, 18] implicitly transform the given feature vectors x_i into new vectors $\phi(x)$ in a space with more dimensions, such that the hypersurface that separates the x becomes a hyperplane in the space of $\phi(x)$'s. Finding the optimal hyperplane is then an optimization problem where the distance of the margin points to the hyperplane is maximized. In this optimization problem, only inner products of the form $K(x_i, x_j) = \phi(x_i)\phi(x_j)$ ever need to be computed, rather than the high dimensional vectors $\phi(x)$ themselves [18]. In our study, we used exponential radial basis functions (erbf) and radial basis functions (rbf) which are explicitly given by:

$$K_{erbf}(x_i, x_j) = e^{-\frac{|x_i - x_j|}{2\sigma^2}}, \quad K_{rbf}(x_i, x_j) = e^{-\frac{(x_i - x_j)^2}{2\sigma^2}}$$

where σ^2 is the variance parameter. In the classification process, only the vectors that are very close to the separating hypersurface need to be considered when computing kernels. These vectors are called the support vectors. Once the support vectors, x_k's are computed, the distance of a vector x from the optimal classifier has the following form:

$$z(x) = \frac{\sum_{x_k \in SV's} \beta_k \, y_k \, K(x_k, x) \, + \, b}{|w|} \qquad (7)$$

Observe that the summation is done over the support vectors only. Computing the coefficients β_k, b is a relatively expensive procedure (see [21]), but needs to be performed only once during training. During the classification process, only Equation (7) needs to be computed.

In order to apply support vector machines to expression recognition problem, we need to generalize the method to more than two classes. Let C_i be the class belonging to the ith expression. Using SVM, the best differentiating hypersurface can be deduced for each class. This hypersurface is the one that optimally differentiates the data belonging to the particular class C_i , from the rest of the data belonging to any C_j where $j \neq i$.

Having obtained the hypersurface for each class, a test shape vector (coming from monocular face tracking) is classified. First, the location of the new data is determined with respect to each hypersurface. For this, the learnt SVM for the particular hyperplane is used to find the distance of the new data to that hypersurface using the distance measure in equation (7). Let z_i be the distance of the new data point to the ith class's hyperplane. The probability that the new data belongs to the ith class is then given by P_i:

$$P_i = \frac{e^{z_i}}{\sum_j e^{z_j}}. \qquad (8)$$

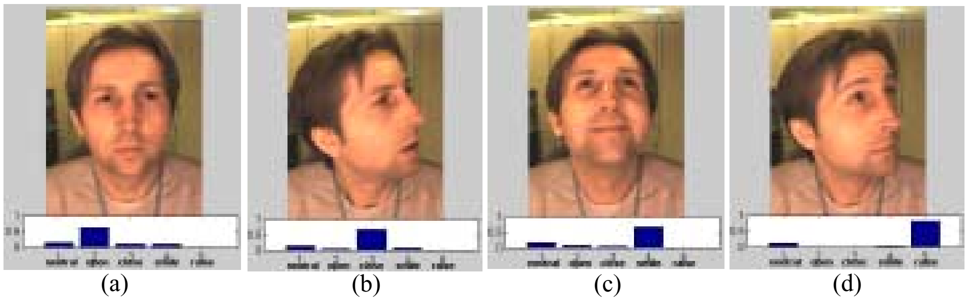
Once the probability function is obtained for each class, the most probable expression is given as the final decision of the system.

3 Experiments

In this section, we present and discuss results achieved on two sets of experiments: recognition of dynamical motions and recognition of emotional expressions. Three subjects were used in both experiments. Two of these subjects were included in the training set (stereo tracking) and all three were included in the test set (monocular tracking).

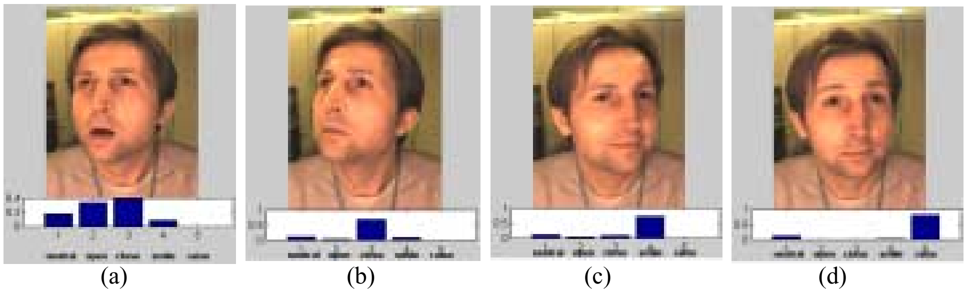
In the first set of experiments, we aimed to differentiate between the five distinct facial movements: Opening mouth, closing mouth, smile, raise eyebrow and neutral shape. The training set (sequences) included a total of 235 frames from the stereo sequences of two subjects. After training of the SVM, the number of support vectors were 82(35%),51(22%),54(23%),53(23%) and 74(31%) for neutral shape, opening mouth, closing mouth, smiling and raising eyebrow respectively. The observation that more support vectors are needed for characterizing the neutral shape class reflects the fact that this class is the most 'difficult' to separate from the other four. Intuitively, this comes from the fact that in the training sequences, the face often takes its neutral shape during the transitions between the other expressions.

The test data included monocular sequences of the three subjects. Figure 3 shows examples of the five dynamical motions along with the corresponding shape vectors $x(n)$. Here, the first five components correspond to the shape vector $\alpha(n)$ and the remaining five components correspond to its derivative ($p = 5$ in Equation 2). Table 1 summarizes the results of the experiments in categories of different expressions. Exponential radial basis function (erbf) with standard deviation of $\sigma = 4$ was used in this experiment. These results show that all the movements are easy to distinguish except for the neutral shape. The neutral shape is sometimes confused with the expression that occurs just before or just after in the sequence.



(a) (b) (c) (d)

Figure 4. (a) An example of missclassification (b-d) Examples of correct classifications with various head poses.



(a) (b) (c) (d)

Figure 5. (a-d) Examples of correct classifications with various head poses.

Figures 4-7 show several examples of facial expression classifications. The probabilities assigned to each expression are also given in a form of a bar plot below each picture with the following order of expressions(left to right): neutral, opening mouth, closing mouth, smiling, raising eyebrows. The set of experiments demonstrates that our recognition system is robust to changes in head pose. Indeed, Figure 4b shows that the system classifies correctly even when the subject faces away from the camera (rotation of nearly 90 degrees). The system is also able to recognize facial expressions in the case where some of the features are not fully visible. For example, in Figure 4d, in spite of the fact that the right corner of the mouth and half of the eyebrow are not visible, the raised eyebrow movement is still correctly identified. In Figure 4a, however, a misclassification occurs mainly since a transition of expressions occurs at this frame. In Figures 5a and 6d, two examples of transitions between two expressions are given. In these cases, the system assigns nearly equal probabilities to the two expressions. On Figures 7bcd the same correctly classified expression (smile) is shown when the head of the user is at different locations (depths).

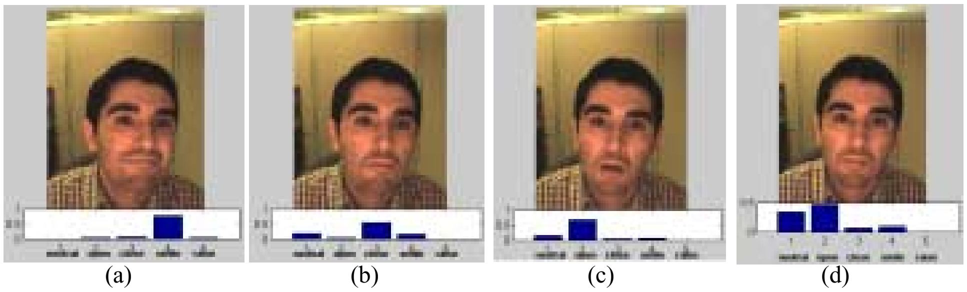
Table 2 gives a comparison of recognition performances of five different classification algorithms. The performance criteria are also divided into two groups with test subject in the training set (same person row), and total performance (including an additional unfamiliar test subject). We observe that SVM performs considerably better than the clustering algorithm, but is rather equivalent to the N-nearest neighbor algorithm. Although the N-nearest classification and the SVM classification produce similar results for this particular set of experiments, SVM has two major advantages over N-nearest classification. First, SVMs minimize the structural risk in a classification problem therefore potentially scale better to new data (different subjects and expressions). Second, classification with SVM takes less time since SVM uses only the support vectors.

4. Conclusion and Future Work

Facial expression recognition is a necessary application for many future human-computer interaction scenarios. In this paper, we proposed a new method for robust recognition of facial expressions. The system uses the 3-D monocular, markerless face tracker to extract a shape vector that is demonstrated to be a robust feature for classification purposes.

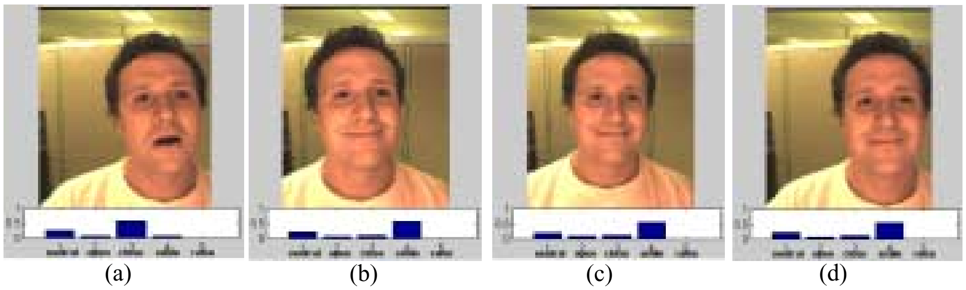
There are two main contributions of this paper. First, we demonstrated that the 3D information extracted through 3D tracking enables robust facial expression recognition in spite of large rotational and translational head movements (up to 90 degrees in head rotation). Second, we showed that Support Vector Machine is a suitable engine for robust classification. Recognition rates as high as 98 percent were achieved at classifying 3 distinct emotional expressions and 91 percent at classifying 5 distinct dynamic facial motions.

In the future, we would like to perform another set of experiments with more subjects and expressions. One important objective is then to build a generic parameterized static face model and use it along with the principal movement directions for tracking and expression recognition of any generic person. Another direction is to investigate applications of our tracking approach to different recognition problems such as face recognition, and lip reading.



(a) (b) (c) (d)

Figure 6. (a-d) Examples of correct classifications with various head poses.



(a) (b) (c) (d)

Figure 7. (a) An example of opening mouth, correctly classified. (b-d) Correct classifications with smile expressions at various depths with respect to the camera.

References

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

[2] M.J. Black and Y. Yacoob. Tracking and recognizing rigid and non-rigid facial motions using local parametric models of image motion. *International Conference on Computer Vision*, pages 374–381, 1995.

[3] W.W. Bledsoe. Man-machine facial recognition. *Panoramic Research Inc.,Palo Alto, CA,*, 1966.

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

[5] X. Chen, S. Kwong, and Y. Lu. Human facial expression recognition based on learning subspace method. *IEEE International Conference on Multimedia and Expo*, 1:403–406, 2000.

[6] 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.

[7] 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.

[8] 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.

[9] I.A. Essa and A.P. Pentland. Coding, analysis, interpretation and recognition of facial expressions. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 19 No.7:757–763, 1997.

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

[11] S.B. Gokturk, J.Y. Bouguet, and R. Grzeszczuk. A data-driven model for monocular face tracking. *International Conference on Computer Vision*, 2001.

[12] F. Hara and H. Kobayashi. A face robot able to recognize and produce facial expression. *Proceedings of the 1996 lEEE/RSJ International Conference on Intelligent Robots and Systems '96, IROS 96*, 3:1600–1607, 1996.

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

[14] J.J. Lien, T. Kanade, J.F. Cohn, and C.C. Li. Automated facial expression recognition based on FACS action units. *Proceedings of Third IEEE International Conference on Automatic Face and Gesture Recognition*, pages 390–395, 1998.

[15] 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.

[16] T. Sakaguchi and S. Morishima. Face feature extraction from spatial frequency for dynamic expression recognition. *Proceedings of the 13th International Conference on Pattern Recognition, ICPR'96*, 3:451–455, 1996.

[17] H. Sako and A.V.W. Smith. Real-time facial expression recognition based on features' positions and dimensions. *Proceedings of the 13th International Conference on Pattern Recognition*, 3:643–648, 1996.

[18] B. Schölkopf. *Support Vector Learning*. R. Oldenbourg Verlag, Munich, 1997.

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

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

[21] V.N. Vapnik. *The Nature of Statistical Learning Theory*. Springer, New York, 1995.

[22] M. Wang, Y. Iwai, and M. Yachida. Expression recognition from time-sequential facial images by use of expression change model. *Proceedings of Third IEEE International conference on Automatic Face and Gesture Recognition*, pages 324–329, 1998.