Tracking of Facial Features Using Deformable Triangles

P. P. Pradeep and P. F. Whelan

Vision Systems Laboratory
School of Electronic Engineering
Dublin City University, Dublin 9, Ireland.

ABSTRACT

In this paper we present a novel algorithm to track the facial features using a deformable triangle model. The face is modeled as an isosceles triangle connecting the eyes and lips which are called as *features of interest* (FoI). A maximum likelihood estimator is used to estimate the position of such a triangle in the image sequences by maximizing the correlation value of the feature template in the image and also the probability that the resulting structure represents a face structure. A method is proposed to remove the noise in the obtained structure by projecting it into a shape subspace of isosceles triangles. Burst error in tracking is removed by using a Kalman filter. The algorithm can successfully locate and track the facial features on two sets of video sequences obtained in the laboratory under normal lighting condition with a cluttered background.

Keywords: face tracking, deformable triangle model, maximum likelihood estimators

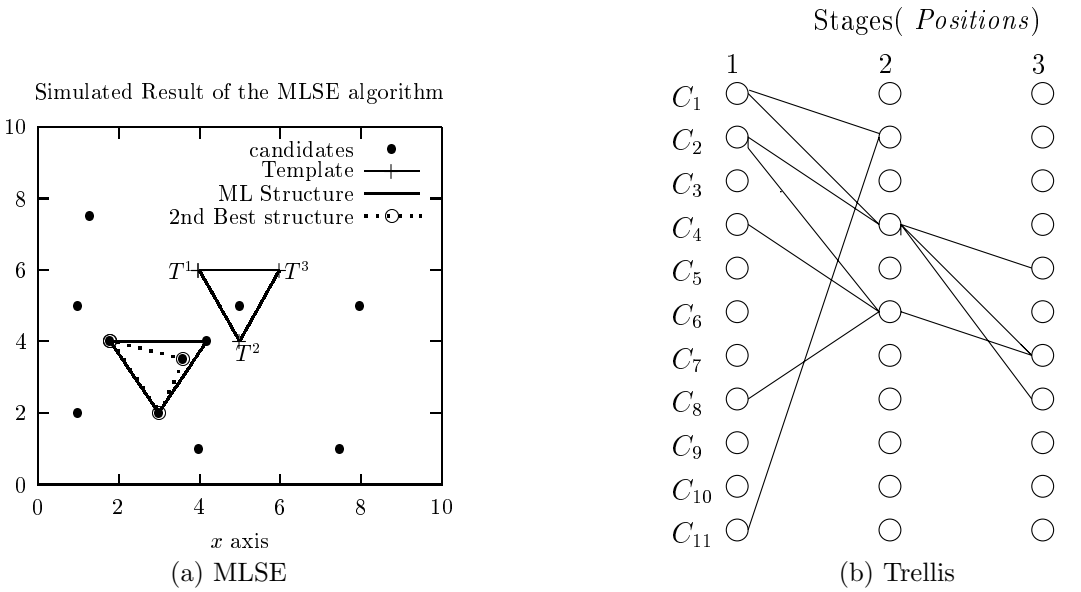
1. INTRODUCTION

Tracking of faces and its features is an important requirement for many applications such as real-time face recognition, facial gesture understanding, face analysis etc. The existing techniques for face and feature detection include correlation, *Principal Component Analysis* (PCA) ,[1] *Hidden Markov Model* (HMM) [2]and *Support Vector Machine* (SVM) .[3] These methods require an initial set of face or features to learn the pattern and then detect such a pattern in the image. When the face is in motion its pattern vary considerably due to the change in size, perspective projection, illumination and also due to various facial expressions. Hence the above mentioned methods fail to track the face efficiently. Other methods such as elliptical tracker [4] and active shape models [5] work either by fixing a predefined model on the face or by tracking a manually extracted shape of the features.

In this paper, a correlation based method is used for tracking the FoI. These features are manually initialized in the first frame. The templates extracted from the first frame are correlated with the subsequent images to estimate the position of FoI in them. The error in these estimates will drift away from zero as there may be changes in the pattern of FoI due to the face motion, lighting etc. The algorithm described in this paper increases the accuracy of these estimates by integrating the structural information of face and the information obtained from the correlation. Here the structure face is defined as a triangle connecting the FoI. Such a triangle should be isosceles due to the symmetry of the face. A likelihood function is derived to integrate both the informations. The maximum value of the likelihood function gives the three points which has a high value of correlation as well as a high probability of maintaining the facial structure. The obtained triangle is projected on to a subspace of isosceles triangles to remove noise. The smoothed structure is then tracked over time using a Kalman filter. The term *deformable* is used in the sense that the triangle obtained in the first frame is allowed to deform from frame to frame while maintaining a high value of correlation for the feature templates.

This paper is organized as follows. In Section. 2 the maximum-likelihood estimator for structure detection is derived. Section. 3 and Section. 4 deal with smoothing and tracking of the obtained structure.

Further author information: (Send correspondence to Pradeep)
Pradeep: E-mail: pradeepp@eeng.dcu.ie
Paul F. Whelan: E-mail: paul.whelan@eeng.dcu.ie, Telephone: +353(1)7005869.



(a) MLSE (b) Trellis

Figure 1. (a) The estimation of ML structure for a randomly generated data set. (b) The trellis constructed for finding the ML structure.

2. MAXIMUM-LIKELIHOOD STRUCTURE ESTIMATION(MLSE)

In this section a probabilistic framework is derived to determine the closeness of the two triangles. The $\triangle^{le}$ obtained from any frame at a time t serves as a template $\triangle^{le}$ for the next frame at time $t+1$. Let the template $\triangle^{le}$ be denoted as $T = \{T^1, T^2, T^3\}$, where T^1, T^2, T^3 are shown in Fig. 1(a). The indices i of T^i are called the *positions*. The points where the correlation is high are referred to as *candidate* points and the i^{th} candidate point is denoted as C_i. The candidate set C is defined as a set of all candidate points and N_c is the number of elements in C. The problem can now be stated as finding a subset C_f of C containing three elements $\{C_i, C_j, C_k\}$ in such a way that the resulting triangle is approximately equivalent to T. If a candidate C_j is selected at the k^{th} *position* then it is denoted as C_j^k. $P(C_j^k)$ is the probability of C_j being selected for the k^{th} position. Given three candidate points C_i, C_j, C_k and the vertices of the template $\triangle^{le}$, the closeness can be determined by measuring the angle and difference in length between the corresponding sides of the two triangles. This can be modeled as a Gaussian pdf as in Eq.1.

$$\begin{aligned} P(C_i^k | C_j^{k-1}, T) &\propto \exp - \left[\frac{(D_C(i,j) - D_T(k,k-1))^2}{2\sigma_D^2} \right. \\ &+ \left. \frac{(\theta_C(i,j) - \theta_T(k,k-1))^2}{2\sigma_\theta^2} \right] \quad (1) \\ i,j &\in \{1,2,\ldots N_c\} \quad (2) \end{aligned}$$

where, $\theta_C(i,j) = \angle(\overrightarrow{C_i - C_j})$ and $\theta_T(i,j) = \angle(\overrightarrow{T^i - T^j})$, the angles of the corresponding vectors. $D_C(i,j) = ||(\overrightarrow{C_i - C_j})||$ and $D_T(i,j) = ||(\overrightarrow{T^i - T^j})||$, the norms of the corresponding vectors.

The variance σ_D and σ_θ defines the acceptable level of deformation in the triangle with respect to the template $\triangle^l e$. Given the candidate point C_i^1 for the first position, the path probability at the second position is calculated using the Bayes' rule.

$$\begin{aligned} P(C_i^2, C_j^1 | T) &= P(C_i^2 | C_j^1, T) P(C_j^1 | T). \quad (3) \\ i,j &\in \{1,2,\ldots,N_c\} \end{aligned}$$

Eq. 3 gives the joint probability of j^{th} candidate point being selected for the first position and i^{th} candidate point being selected for the second position. The term $P(C_j^1|T)$ is the probability of C_j being selected for the first position. This value is assumed to be same for all the candidate pints in the region of the first point and zero for the rest of the candidate points.

Proceeding in a similar fashion and using a Markovian assumption,[6]

$$\begin{aligned} P(C_i^3, C_j^2, C_k^1|T) &= P(C_i^3|C_j^2, T) \times P(C_j^2|C_k^1, T) \\ &\times P(C_k^1|T). \end{aligned} \tag{4}$$

The Eq. 4 gives the likelihood of the three points C_i, C_j and C_k maintaining the structure close to that of the template. A Viterbi like algorithm is used to find the maximum value of this likelihood function.

The algorithm starts by initializing all states with equal probabilities. The trellis can be constructed as in Fig. 1(b) with *candidates* as its states and *positions* as its depth. The transition probabilities of the trellis is given by Eq.1. A Viterbi algorithm is used to maximize the Eq.4. The high probability paths in the trellis refers to the closest triangles to the template $\triangle^{le}$. The best triangle is the one which maximizes the cost function given in Eq.5

$$P(C_i^3, C_j^2, C_k^1|T) + Corr(C_i) + Corr(C_j) + Corr(C_k) \tag{5}$$

where $Corr(C_i^k)$ is the cross correlation of the k^{th} feature template with the image at C_i .

3. RESAMPLING

The maximum-likelihood (ML) triangle obtained in Section. 2 may not be exactly isosceles. This error can be removed by projecting the ML triangle to a shape sub-space of isosceles triangles. A shape sub-space can be created in active shape models [7] using a set of isosceles triangles obtained from the edges in the area around the vertices of the ML triangle. The algorithm for smoothing is described below.

- Using the vertices $\{(x_1, y_1), (x_2, y_2), (x_3, y_3)\}$ of the triangle construct a vector $\mathbf{x}$

$$\mathbf{x} = (x_1, y_1, x_2, y_2, x_3, y_3)^T$$

- Translate $\mathbf{x}$ such that its center of gravity is at the origin.
- Construct N training isosceles triangles by randomly sampling the points on the edges in a small area around the vertices with a uniform probability.
- Compute the mean of the training set,

$$\overline{\mathbf{x}} = \frac{1}{N} \sum_{i=1}^{N} \mathbf{x}_i$$

- Compute the covariance of the training set,

$$\mathbf{S} = \frac{1}{N-1} \sum_{i=1}^{N} \left(\mathbf{X}_i - \overline{\mathbf{X}}\right) \left(\mathbf{X}_i - \overline{\mathbf{X}}\right)^T$$

- Compute the eigen vectors, ϕ_i and corresponding eigenvalues λ_i of $\mathbf{S}$. Arrange the eigen vectors and eigen values so that $\lambda_i \geq \lambda_{i+1}$.
- Construct a matrix $\Phi = (\phi_1|\phi_2|\ldots|\phi_t)$ with the eigen vectors corresponding to the t largest eigen values. Φ represents the shape-space

- The original vector $\mathbf{x}$ can now be projected on the shape subspace using the equation

$$\hat{\mathbf{x}} = \overline{\mathbf{x}} + \Phi\mathbf{b}$$

where $\mathbf{b}$ is a t dimensional vector given by

$$\mathbf{b} = \Phi^T(\mathbf{x} - \overline{\mathbf{x}})$$

Φ^T is called as the projection matrix.

The number of eigen vectors to retain, t, are chosen so that the residual terms can be considered as noise. This is done by eliminating the eigen vectors corresponding to the eigen values whose contribution to the total variance is less than 10%.

4. TRACKING

In a video sequence, a further improvement in the performance of the algorithm can be achieved by tracking the features through the frames. Instead of using the smoothed data as a template for the next time step, a better approach is to predict the data for the next time step which may be used as a template. This predicted data has to be modified once the measurement is made for the next time step.

Kalman filtering is an effective mechanism for implementing such a prediction-updation strategy. The Kalman filter has been extensively used for tracking in many areas of control theory, signal processing, computer vision, etc. The fundamental elements of a Kalman filter are *states* and *measurement*. The state vector $\mathbf{s}(t)$ is defined here as,

$$\mathbf{s}(t) = \left(\mathbf{x}(t)^T, \Delta\mathbf{x}(t)^T\right)^T \tag{6}$$

where,

$$\Delta\mathbf{x}(t) = (\Delta x_1(t), \Delta y_1(t), \Delta x_2(t), \Delta y_2(t), \Delta x_3(t), \Delta y_3(t))$$

$\mathbf{x}(t)$ is the data (ML structure) at time t and $\Delta\mathbf{x}(t)$ is the displacement of the vertices of the triangle per unit frame. With the assumption that the face moves with a constant acceleration, a state transition matrix can be written as

$$\mathbf{A}(t) = \begin{pmatrix} \mathbf{I}_{6\times6} & \Delta t\mathbf{I}_{6\times6} \\ \mathbf{0}_{6\times6} & \mathbf{I}_{6\times6} \end{pmatrix} \tag{7}$$

Similarly, the measurement vector and the measurement matrix are formed as,

$$\mathbf{m}(t) = (\mathbf{x}(t)) \qquad \mathbf{B}(t) = \begin{pmatrix} \mathbf{I}_{6\times6} & \mathbf{0}_{6\times6} \end{pmatrix} \tag{8}$$

With the system model and the measurement model as defined above, a Kalman filter algorithm can be applied to obtain the expected position of the features.

5. RESULTS AND DISCUSSION

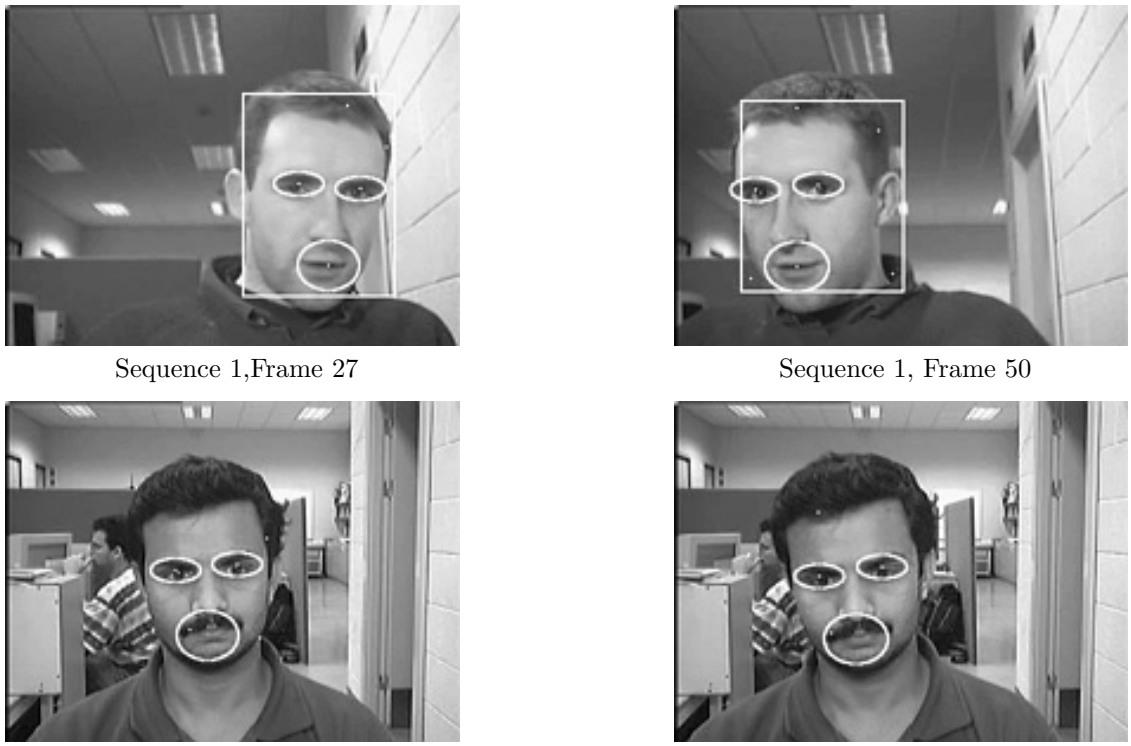
This algorithm was initially tested with a randomly generated set of *candidates* and typical template of face structure . The probabilities calculated are shown in the Table 1. The corresponding trellis with the successful paths are shown in Fig. 1(b). In the actual implementation, the eyes and mouth were segmented manually in the first frame. The resulting triangle was made isosceles using the projection method.

Preliminary tests for tracking produced good results(Fig. 2). It was tested on two video sequence obtained in the laboratory under normal lighting condition with a cluttered background. The system tracked the facial features efficiently at a rate of 22 frames per minute on a PC with a processor speed of 233 MHz and a RAM of 128Mb. Although the tracking system produced good result against variation in lighting, expressions, pose change, size change and orientation, it failed to track when one of the feature is missing.

The active nature of this algorithm is due to σ_D and σ_θ. These values determine the amount of deformation a template can undergo to match with the given structure. In our implementation σ_D was set to 50% of the length of one side of the template while σ_θ was kept at 10°.

	Position 1	Position 2	Position 3	$P(C_i, C_j, C_k \mid T)$
Path 1	2	4	5	0.4581
Path 2	2	4	8	0.1247
Path 3	1	4	5	0.0179
Path 4	1	4	7	0.0110

Table 1. These numbers represent the indexes of the *candidate* points giving the highest likelihood paths for the randomly generated data set in Fig. 1(a,b).



Sequence 1,Frame 27 Sequence 1, Frame 50

Sequence 2, Frame 120 Sequence 2, Frame 170

Figure 2. The extracted features from a human face.

6. CONCLUSION

This paper presents a computationally efficient algorithm that extracts and tracks the facial features from a sequence of images without the requirement of user intervention. The problem of facial feature detection was formulated as a problem of finding the best matching structure to the template face-structure. The algorithm works by estimating the most likely structure which is close to the template structure. It is smoothed by projecting the detected structure in to the shape subspace to remove noise accumulated during the feature extraction stage. The tracking is done using Kalman filtering based on a constant acceleration model. The algorithm was applied to different sequences and proved to be robust to changes such as background variation and lighting conditions.

REFERENCES

1. M. Turk and A. Pentland, "Face recognition using eigen faces," in *Proceedings of International Conference on Pattern Racognition*, pp. 586–591, 1991.

2. A. V. Nefian and M. H. Hayes, “Hidden markov models for face detection and recognition,” in *International Conferance on Image Processing*, 1998.
3. E. Osuna, R. Freund, and F. Girosi, “Training support vector machines:an application to face detection,” in *CVPR*, 1997.
4. S. Birchfield, “Elliptical head tracking using intensity gradients and color histograms,” in *Proceedings of CVPR*, pp. 232–237, 1998.
5. T. Cootes and C. Taylor, “Active shape models - smart snakes,” in *Proc. British Machine Vision Conferance*, pp. 266–275., Springer-Verlag, 1992.
6. P. R. Kumar and P. Varaiya, *Stochastic Systems*, Prentice-Hall, Engle-wood Cliffs, NJ, 1986.
7. T. Cootes and C. Taylor, “Statistical models of appearance for computer vision,” in *Technical report*, University of Manchester, Wolfson Image Analysis Unit, Imaging Science and Biomedical Engineering., september 1999.

P. P. Pradeep, P. F. Whelan (2002), "Tracking of facial features using deformable triangles", OPTO-Ireland: SPIE's Regional Meeting on Optoelectronics, Photonics and Imaging, Galway, 5 - 6 September 2002, Proc SPIE Vol. 4877 pp 138-143