View-Based Adaptive Affine Tracking

Fernando de la Torre[1], Shaogang Gong[2], and Stephen McKenna[3]

[1] Department of Signal Theory, Universitat Ramon Llull, Spain
ftorre@salleURL.edu
[2] Department of Computer Science, Queen Mary and Westfield College, England
sgg@dcs.qmw.ac.uk
[3] Department of Applied Computing, University of Dundee, Scotland
stephen@dcs.qmw.ac.uk

Abstract. We propose a model for view-based adaptive affine tracking of moving objects. We avoid the need for feature-based matching in establishing correspondences through learning landmarks. We use an effective bootstrapping process based on colour segmentation and selective attention. We recover affine parameters with dynamic updates to the eigenspace using most recent history and perform predictions in parameter space. Experimental results are given to illustrate our approach.

1 Introduction

Object recognition in dynamic scenes using view-based representation requires establishing image correspondences in successive frames of a moving object which may undergo both affine and viewpoint transformations [1]. However, to obtain consistent dense image correspondence is both problematic and expensive since changes in viewpoint result in self-occlusions which prohibit complete sets of image correspondences from being established. Practically, only sparse correspondence can be established quickly for a carefully chosen set of feature points [2]. To realise near real-time performance, however, an entirely different approach is preferable which does not depend on reliable feature detection and tracking. Holistic texture-only-templates can be used [3]. This assumes that the object of interest is approximately rigid and therefore permits a relatively simplistic parametric model to be used. Furthermore, if the model is also built based on data from a large set of viewpoints, it can in theory recover pose change as well. Likewise, if it is trained under different illuminations, it can perform in changing lighting conditions (e.g. [4]).

The EigenTracking approach proposed by Black [5] essentially attempted to establish such holistic, appearance-based correspondence of a moving rigid object by recovering a parameterised affine transformation in an eigenspace, constructed from object images of different views. However, due to its rigidity assumption and the use of texture-only-templates, EigenTracking fails to capture changes which are not sufficiently affine. In particular, it copes poorly with flexible objects such as human faces. Furthermore, to be able to establish image correspondence across

different viewpoints, EigenTracking requires a training image set that spans the view sphere. This is impractical and computationally expensive.

In this work, we propose an integrated scheme for view alignment which takes the following into consideration: (a) the use of both shape and texture in eigenspace in a simple manner relaxes the rigidity assumption without introducing too much computational cost, (b) a process for effective bootstrapping, (c) parameter recovery with selective attention, (d) affine parameter estimation using a dynamically updated, viewpoint centred eigenspace, and (e) parameter prediction.

The rest of this paper is arranged as follows. We first introduce the shape augmented affine tracking model in Section 2. In Section 3, we describe a colour based bootstrapping process for the affine tracking. We then introduce the concept of view dependent model adaptation in affine tracking in Section 4. Section 5 presents model prediction before experiments and discussion are given in Sections 6 and 7.

2 Affine Tracking with Shape Constraints

A set of p images with N dimensions, forming an $N \times p$ matrix $\mathbf{A}$, can be represented by the eigenspace of their covariance matrix $\mathbf{C}$ where usually $p < N$. The image matrix $\mathbf{A}$ can be decomposed using Singular Value Decomposition (SVD) which gives

$$\mathbf{A} = \mathbf{U}\boldsymbol{\Lambda}\mathbf{V}^{\mathrm{T}}$$

where $\mathbf{U}$ is an orthogonal matrix of eigenvectors of $\mathbf{C}$ and $\boldsymbol{\Lambda}$ is a diagonal matrix of its eigenvalues. The matrix $\mathbf{V}$ defines the nullspace of $\mathbf{C}$ (since $p < N$). For view alignment, an image $\mathbf{I}$ is represented by projection onto eigenvectors $\mathbf{u}_j$, i.e.

$$\mathbf{I} \approx \sum_{j=1}^{k} c_j \mathbf{u}_j = \mathbf{U}\,\mathbf{c}$$

where $k < p$ is the number of eigenvectors actually used and $\mathbf{c}$ gives projection coefficients. Note that throughout this paper, we use $\mathbf{I}$ to represent an image vector rather than the Identity matrix. A pre-filtering process is often necessary if global illumination is unstable [3].

2.1 Template-Only Affine Tracking: EigenTracking

If image changes are approximately affine, correspondence for alignment can be achieved by treating an image $\mathbf{I}(\mathbf{f}(\mathbf{x}, \mathbf{a})) = [I(\mathbf{f}(\mathbf{x}_1, \mathbf{a})), I(\mathbf{f}(\mathbf{x}_2, \mathbf{a})) \ldots I(\mathbf{f}(\mathbf{x}_N, \mathbf{a}))]^{\mathrm{T}}$ as a function of an affine transformation given by parameters $\mathbf{a} = (a_0, a_1, a_2, a_3, a_4, a_5)^{\mathrm{T}}$, where

$$\mathbf{f}(\mathbf{x}, \mathbf{a}) = \begin{bmatrix} a_0 \\ a_3 \end{bmatrix} + \begin{bmatrix} a_1 & a_2 \\ a_4 & a_5 \end{bmatrix} \begin{bmatrix} x - x_c \\ y - y_c \end{bmatrix} \tag{1}$$

and $\mathbf{x}_c = (x_c, y_c)^{\mathrm{T}}$ is the centre position of the object template. Alignment can then be accomplished by recovering both the affine parameters $\mathbf{a}$ and the projection coefficients $\mathbf{c}$ by minimising a cost function $\min_{\mathbf{c},\mathbf{a}} \rho\left[\left(\mathbf{I}(\mathbf{f}(\mathbf{x},\mathbf{a})) - \mathbf{U}\,\mathbf{c}\right), \sigma\right]$, where ρ is a robust error norm and σ is a scale factor that controls the convexity of the norm [5,6]. In our scheme, we use the Geman-McClure error norm [7] given by

$$\rho(x, \sigma) = \frac{x^2}{\sigma^2 + x^2}$$

Outliers will be considered values from the inflexion point of the norm, which are residuals with $x_i > \sigma/\sqrt{3}$. In general, the above cost function is non-convex and minimisation can result in local minima. A minimisation algorithm found to be effective in this case is gradient descent with the continuation method of graduated non-convexity [8]. It begins with a large value of σ where all the points are inliers. Then σ is successively lowered, reducing the influence of outliers. While it is not guaranteed to converge to a global minimum, the method is effective for visual tracking since continuity of motion provides good starting points.

A dramatic reduction in computational cost is achieved by avoiding image warping in every iteration. This is done by adopting the following linear approximation to the above cost function:

$$\min_{\mathbf{c},\mathbf{a}} \rho\left[\left(\nabla\mathbf{I}^{\mathrm{T}}\,\mathbf{f}(\mathbf{a}) + (\mathbf{I} - \mathbf{U}\mathbf{c})\right), \sigma\right] \tag{2}$$

where $\nabla\mathbf{I}$ is the image gradient $[I_x,\, I_y]^{\mathrm{T}}$ [5].

2.2 Encoding Landmarks

The difficulty in encoding shape is to be able to compute correspondences quickly and sufficiently robustly. To achieve such a purpose, we encode the coordinates of known landmarks in the training images which are used for constructing the eigenspace. Let $\mathbf{A} = [\mathbf{I}_1\, \mathbf{I}_2 \ldots \mathbf{I}_p]$ be the matrix of training images and $\mathbf{X} = [\mathbf{x}_1\, \mathbf{x}_2 \ldots \mathbf{x}_p]$ be the coordinates of the landmarks in these images. The landmarks are assumed to have been located by hand. The landmarks $\mathbf{X}$ in Fig. 1 are the positions of the eyes.

We first took the approach of constructing a concatenated matrix:

$$\mathbf{A}^* = [\{\mathbf{I}_1, \mathbf{x}_1\}\, \{\mathbf{I}_2, \mathbf{x}_2\} \ldots \{\mathbf{I}_p, \mathbf{x}_p\}]$$

The new matrix $\mathbf{A}^*$ is a modification of $\mathbf{A}$ with additional feature vectors concatenated to the tail of each training image vector. However, the large scale difference in variances of $\mathbf{A}$ and $\mathbf{X}$ causes numerical problems. To obtain comparable variance, we scale each shape vector $\mathbf{x}_i$ by the largest norm. A more considered approach to achieve comparable variance between the texture and shape

Fig. 1. Examples of landmarks used in the training images.

vectors in eigenspace can be adopted [9]. It is worth pointing out though that better results were actually achieved with modelling the texture and shape vectors in independent eigenspaces rather than with the concatenated eigenspace. This seems to re-confirm the evidence reported elsewhere [10].

Once the landmarks have been learned from the training set, we can recall the landmarks during the tracking process. If a new image is aligned with the eigenspace, the reconstruction from the coefficients in the eigenspace is given by $\mathbf{Uc}$. To recall the most likely landmark positions for the new image based on what has been learned in training, an inverse transformation between the eigenspace and the training set (related by the SVD) is performed on the shape components only:

$$\mathbf{x}_{new} = \mathbf{X}\left(\mathbf{V}\Sigma^{-1}\mathbf{c}\right) \tag{3}$$

$\mathbf{V}\Sigma^{-1}\mathbf{c}$ are the coefficients which best reconstruct the new image in the least-square sense from the training data. Note that $\mathbf{V}\Sigma^{-1}$ can be pre-computed off-line in order to speed up the tracking process. Given sufficient training examples with known landmarks, incorporating shape with texture in eigenspace enables previously learned feature positions to be recalled during tracking, avoiding the need to perform on-line feature detection and correspondence which are computationally both expensive and problematic.

3 Initialisation

Colour-based segmentation can provide robust and very fast focus-of-attention for the initialization of the affine parameters [11]. Here we adopt multi-colour Gaussian mixture models to perform real-time object detection and focus of attention. The mixture models were estimated in two-dimensional hue-saturation colour space. Such representations are chosen to permit some level of robustness against brightness change. Probabilities are computed for pixels in an image search space and the size and position of the object are estimated from the resulting probability distribution in the image plane [11]. An example of colour-based, real-time, coarse segmentation using a mixture of four Gaussians can be seen in Fig. 2.

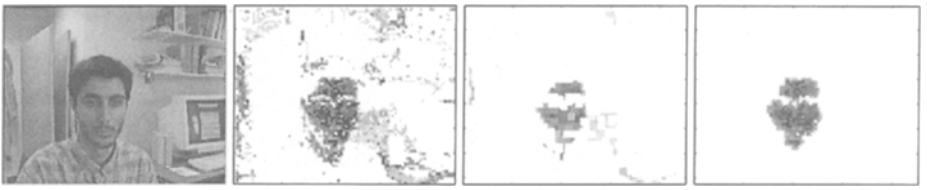

Fig. 2. From left to right: An image frame from a sequence; the object foreground colour probabilities in the image plane; results of segmentation with multi-resolution relaxation after 1 and 4 iterations.

3.1 Attentional Window

The most computationally expensive operation in recovering affine parameters is to recursively warp the image relative to its center in order to minimise the cost function of Equation (2). To address this problem, the affine transformation is only computed within an attentional window. The size of this window adapts to the size of the object (see Fig. 3). Affine transforms are performed relative to the center of the window which must coincide with the centroid of the object in order to minimize the errors in estimated rotation and scale parameters. The centre of the attentional window is estimated using prediction (see Section 5).

Fig. 3. Tracked face images with adaptive attentional windows shown by the larger bounding boxes. Affine transforms are applied only within these windows. The small bounding boxes give the tracked face with the recovered affine parameters.

3.2 Feature Extraction

Colour provides only a crude initial estimate for an attentional window. An improved estimate is obtained in a computationally efficient way by applying recursive, non-linear morphological operations at multiple resolutions. The method can be seen as a combination of relaxation and something similar in spirit to geodesic reconstruction in morphology [12]. This "geodesic relaxation" algorithm is as follows:

1. Compute log probabilities in a 1/4 sub-sampled image and normalise these probabilities to give a low resolution grey-scale "probability image" $\mathbf{I}^{o}_{N/16}$.
2. Apply grey-level morphological erosion to $\mathbf{I}^{o}_{N/16}$. This reduces noise and erroneous foreground and yields an image $\mathbf{I}^{er}_{N/16}$.
3. Let $\mathbf{I}^{*}_{N/16} = \mathbf{I}^{er}_{N/16}$, then apply the following operation for a fixed number of times:
   $\mathbf{I}^{*}_{N/16} = \frac{1}{2}(\mathbf{I}^{*}_{N/16} \otimes \texttt{low-pass-filter} + \mathbf{I}^{o}_{N/16})$, where $\otimes$ denotes convolution.

The resulting image $\mathbf{I}^{*}_{n/16}$ (see Fig. 2) is used to fit a bounding box which is then used as an initial attentional window. The iterative process is fast because good results are obtained in a few iterations using low resolution images. Morphological operators were also used to estimate the region within the initial attentional window occupied by the main facial features (see Fig. 4). The process was as follows:

1. Perform vertical erosion on the 1/2 sub-sampled attentional window $\mathbf{I}^{win}_{N/4}$ to give $\mathbf{I}^{er}_{N/4}$.
2. Perform geodesic reconstruction of $\mathbf{I}^{er}_{N/4}$ with $\mathbf{I}^{win}_{N/4}$ as the reference image to give $\mathbf{I}^{rec}_{N/4}$.
3. Compute $\mathbf{I}^{end}_{N/4} = \texttt{opening}(\mathbf{I}^{win}_{N/4} - \mathbf{I}^{rec}_{N/4})$

The extent of the estimated facial feature region was used to estimate initial affine scale parameters (a_1, a_5).

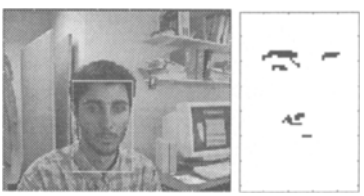

Fig. 4. Left: The attentional window estimated using colour cues. Right: The main facial feature region extracted using morphological operators.

3.3 Parameter Initialisation

Colour and morphological operators provide an initial attentional window and approximate estimates of translational and scale parameters. These initial parameter estimates are further refined by recursively applying Successive-Over-Relaxation [8] in order to minimize the cost function (2). Robust norms with a continuation method and a multi-resolution representation were used in order to avoid local minima. At first, only the translational parameters (a_0, a_3) were optimised in order to align the centre of the attentional window with the eigenspace. Subsequently, the scale parameters (a_1, a_5) were optimised. It was assumed that affine rotation was negligible in the initial frame. An example of this parameter initialisation process is shown in Fig. 5. In this example, the initial estimates provided by the colour model were unusually poor.

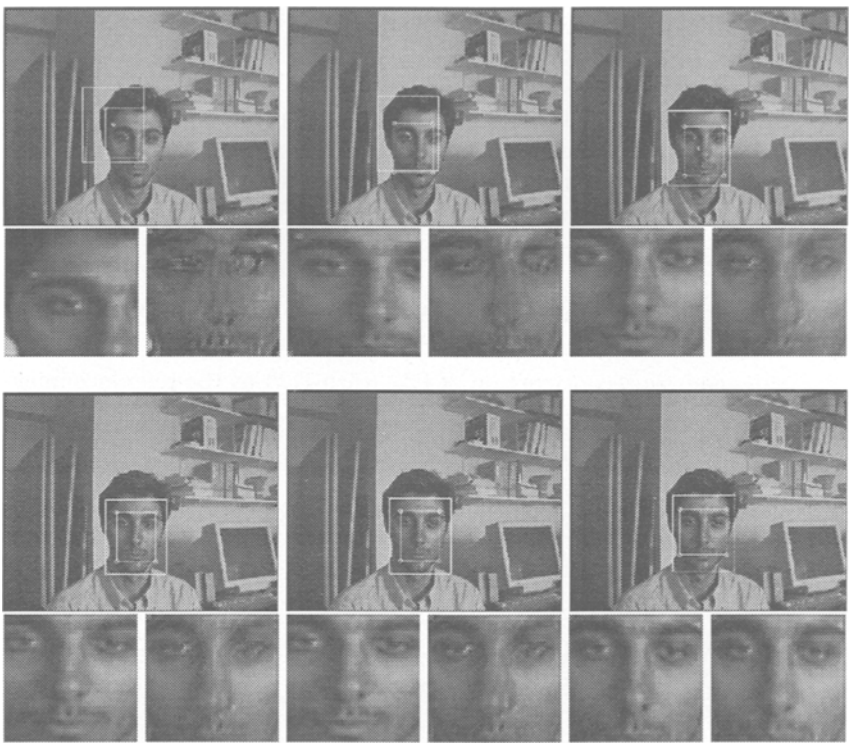

Fig. 5. Affine parameter initialisation: The attentional window is overlaid with a smaller box indicating estimated translation and scale parameters. Below each frame are the located region (left) and its reconstruction (right).

4 Model Adaptation

The EigenTracking method made use of a fixed, global eigenspace (GES) representation for reconstruction and tracking. This eigenspace was built by performing SVD on a relatively large, fixed training set. The alternative method described in this section yields faster and more accurate reconstruction. It involves the use of local eigenspace (LES) representations built using subsets of the original training set. In particular, at each time frame t, the k training images which are "closest" to the previous affine-normalised, tracked image $\mathbf{I}_{t-1}$, were selected. A new LES is then computed from these k images along with $\mathbf{I}_{t-1}$. The inclusion of $\mathbf{I}_{t-1}$ helps to compensate for temporary changes not represented in the original training set (e.g. unusual facial expressions).

The computation of a new LES in (potentially) every frame might seem prohibitively expensive. However, the iterative matching algorithm typically converges more quickly when reconstructions are performed using an LES. In practise, this faster convergence more than compensates for the expense of computing the LES. The overall result is faster and gives more robust tracking. The k selected training images are usually images with similar 3D pose and facial expression and can therefore be accurately represented using only a few eigenvectors. In order to achieve sufficiently good reconstruction, enough eigenvectors are retained to account for 95% of the variance in the training set.

In order to compute an LES, the k "closest" training images must be selected. An obvious way in which to perform this selection is to measure the Euclidean distance between $\mathbf{I}_{t-1}$ and each of the training images and to select the k nearest images. These distance measurements can be efficiently approximated using projections onto the precomputed GES [13]. Further efficiency can be obtained using a multi-resolution scheme in which $\mathbf{I}_{t-1}$ is sub-sampled and projected onto a precomputed GES of equally low resolution.

If an ordering can be imposed on the training set, however, an alternative scheme becomes possible. The closest match in the training set is then used to index into this ordered set and the k images for the LES are selected using the predetermined ordering. There are other strategies that have been suggested for performing such a nearest neighbour search. See [14] for a detailed discussion. In our case, if the training set consists of a sequence of a head rotating from left to right then time imposes a natural ordering. The nearest match then yields an estimation of head pose and the LES is computed from images of similar poses. Another way to derive the same matching could be done using the information of the projection coefficients and their relation with the training set. The p coefficients $\mathbf{y} = [y_1\, y_2\, \ldots\, y_p]^\mathsf{T}$ whose linear combination of the training set minimises the Euclidean distance, $\min_{\mathbf{y}} \| \mathbf{I}_{t-1} - \mathbf{A}\mathbf{y} \|^2$, are given by:

$$\mathbf{y} = \mathbf{V}\boldsymbol{\Sigma}^{-1}\mathbf{c}, \quad \mathbf{y} = (\mathbf{A}^\mathsf{T}\mathbf{A})^{-1}\mathbf{A}^\mathsf{T}\mathbf{I} \tag{4}$$

As we are working with normalized images, minimising the Euclidean distance is equivalent to maximising the dot product. Note that we can obtain the dot products if the coefficients in the eigenspace $\mathbf{c}$ are known, that is:

$$\mathbf{A}^T\mathbf{I} = (\mathbf{A}^T\mathbf{A})\mathbf{y} = \mathbf{V}\mathbf{\Sigma}\mathbf{c} \tag{5}$$

The position of the maximum component of $\mathbf{A}^T\mathbf{I}$ corresponds to the "closest" image in the training set. It can be easily shown that this is computationally equivalent to [13]. However, our approach establishes correspondence between the eigen-coefficients and the training images directly and is less expensive for computing the GES. For example, computing $\mathbf{C} = \mathbf{c}_1\mathbf{c}_2...\mathbf{c}_p = \mathbf{U}^T\mathbf{A}$ requires kpN operations. With SVD of A, computing $\mathbf{C} = \mathbf{\Sigma}\mathbf{V}^T$ needs just k^2p operations which is more efficient since $k \ll N$.

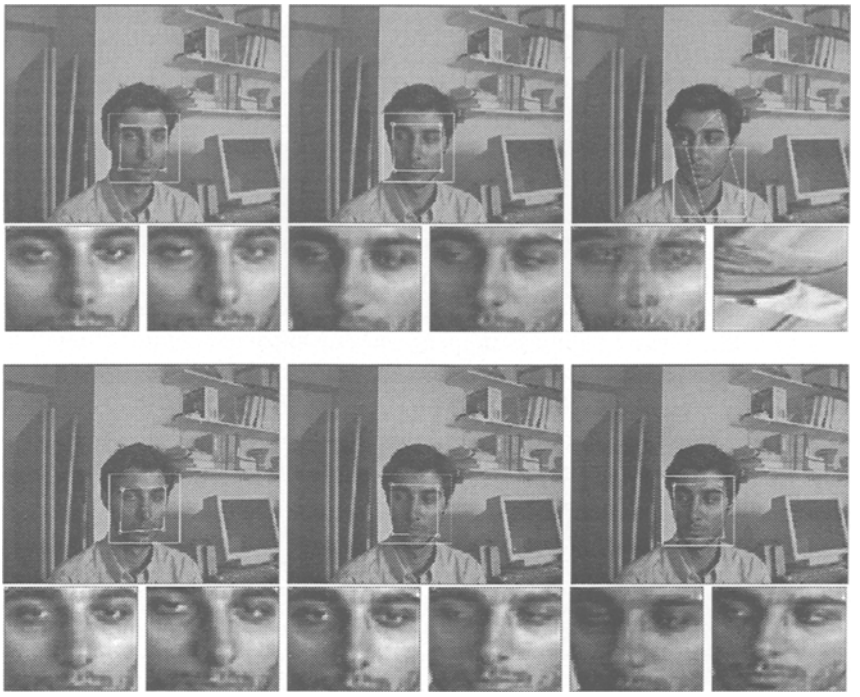

Fig. 6. Top row: EigenTracking without dynamic updating fails due to insufficient training views. Bottom row: The result from using the adaptive scheme.

5 Parameter Prediction

In order to cope with displacements of more than a few (2-3) pixels between frames, it was necessary to use prediction. In each frame, the affine parameters

were predicted and these predictions were used to initialise the iterative optimisation algorithm. Each of the six affine parameters was predicted using a Kalman filter [15]:

$$\mathbf{x}_{k+1} = \mathbf{\Gamma}\mathbf{x}_k + \mathbf{B}\mathbf{n}_k, \tag{6}$$

$$\mathbf{z}_k = \mathbf{H}\mathbf{x}_k + \mathbf{r}_k \tag{7}$$

where $\mathbf{x}$ and $\mathbf{z}$ were 12-dimensional state and measurement vectors and $\mathbf{n}_k$ and $\mathbf{r}_k$ were zero-mean white noise with covariance matrices $\mathbf{Q}_k$ and $\mathbf{R}_k$. The dynamic system model used assumed constant velocity. This seemed to be reasonable due to the fact that pairwise plots of the affine parameter measurements typically revealed trajectories in the 6-dimensional affine parameter space which were approximately linear or piecewise linear (see Fig. 7). The elements of the vectors $\mathbf{x}$ and $\mathbf{z}$ corresponded to the affine parameters and their velocities.

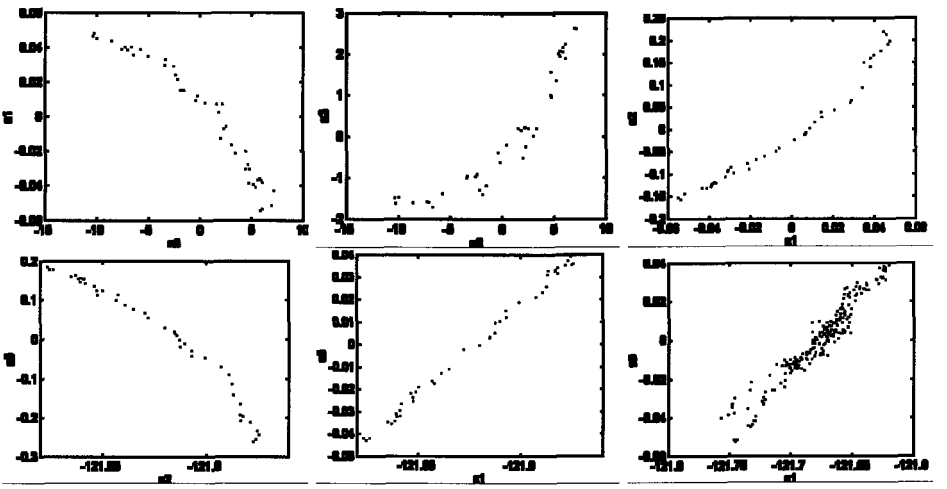

Fig. 7. Pairwise 2D affine parameter subspaces in our case are mostly linear. The example subspaces shown are *a0-a1*, *a0-a3*, *a1-a2*, *a1-a4*, *a1-a5* and *a1-a6*, from top-left to bottom-right.

Appropriate noise covariance matrices were estimated using the EM algorithm with the following least-squares approximations of observation noise and state noise:

$$\mathbf{r}_k \approx \mathbf{z}_k - \mathbf{H}\mathbf{x}_k^-, \tag{8}$$

$$\mathbf{n}_k \approx (\mathbf{B}^\mathrm{T}\mathbf{B})^{-1}\mathbf{B}^\mathrm{T}(\mathbf{x}_{k+1}^+ - \mathbf{\Gamma}\mathbf{x}_k^+) \tag{9}$$

where $^+$ denotes *a posteriori* estimation and $^-$ denotes *a priori* estimation. The estimated covariance noise between the affine parameters was also verified visually by plotting pairwise parameter observations. An alternative approach is to

derive an optimal estimate based on the Kalman filter's underlying cost function. Estimation of the state vector $\mathbf{x}_{k+1}$ based upon previous measurements is equivalent to minimising cost function:

$$E(\mathbf{x}_{k+1}^{+}) = \tfrac{1}{2}(\mathbf{x}_{k+1}^{+} - \mathbf{x}_{k-1}^{-})^{\mathrm{T}}(\mathbf{P}^{-1})(\mathbf{x}_{k+1}^{+} - \mathbf{x}_{k-1}^{-}) + \tfrac{1}{2}(\mathbf{z}_k - \mathbf{H}\mathbf{x}_{k+1}^{+})^{\mathrm{T}}(\mathbf{R}^{-1})(\mathbf{z}_k - \mathbf{H}\mathbf{x}_{k+1}^{+})$$

The first term specifies the temporal constraint while the second expresses the data conservation, where all errors are measured using the Mahalanobis distance. We apply one robust norm to the second term in order to derive a robust Kalman filter. Efficient minimization of this function could be performed using a technique such as Iteratively Recursive Least Squares (IRLS) [16].

The human head will inevitably move in ways which are not predictable using these simple dynamic models. An effective way in which to detect this "unpredictability" is to run one iteration of the optimisation algorithm used for tracking. Only if the direction in affine parameter space from this iteration "agrees" with that of the Kalman prediction are the predictions utilised. This works well if not many outliers are present.

6 Experiments

The system was initially implemented in Matlab and takes an average of 14 sec/frame. In C, it could run at near 2 sec/frame on a standard 200MHz PC. Applying IRLS to the minimisation process can solve an approximation of the robust formulation in near real-time.

Fig. 8 shows the ability of the new adaptive scheme with shape encoded to align a face undergoing non-rigid expression change. Similar problems occur when the assumption of affine transformation is no longer valid. The adaptive scheme was shown to be able to overcome the problem when changes in pose were not sufficiently captured by the training data (see examples in Fig. 6).

Fig. 9 shows view alignment from a 260 frame sequence with both affine and pose variations. The training set had 100 images and 54 eigenvectors were used in a GES capturing 95% of the variance. However, with the adaptive scheme using LES, only 6 eigenvectors were needed to recover sufficiently accurate parameters for alignment.

Fig. 10 gives an indication of savings in computational cost when the adaptive scheme is applied. It shows the time taken in seconds to perform the minimisation (Equation (2)). The first plot shows the time required for each frame using GES with 54 eigenvectors. The second plot shows the time required for each frame with LES but without the previous history at $t-1$. The third plot shows the result from augmented LES using $t-1$ tracked data. The mean frame rate were: 29 sec/frame for GES, 15 sec/frame for LES and 12 sec/frame for $t-1$ augmented LES.

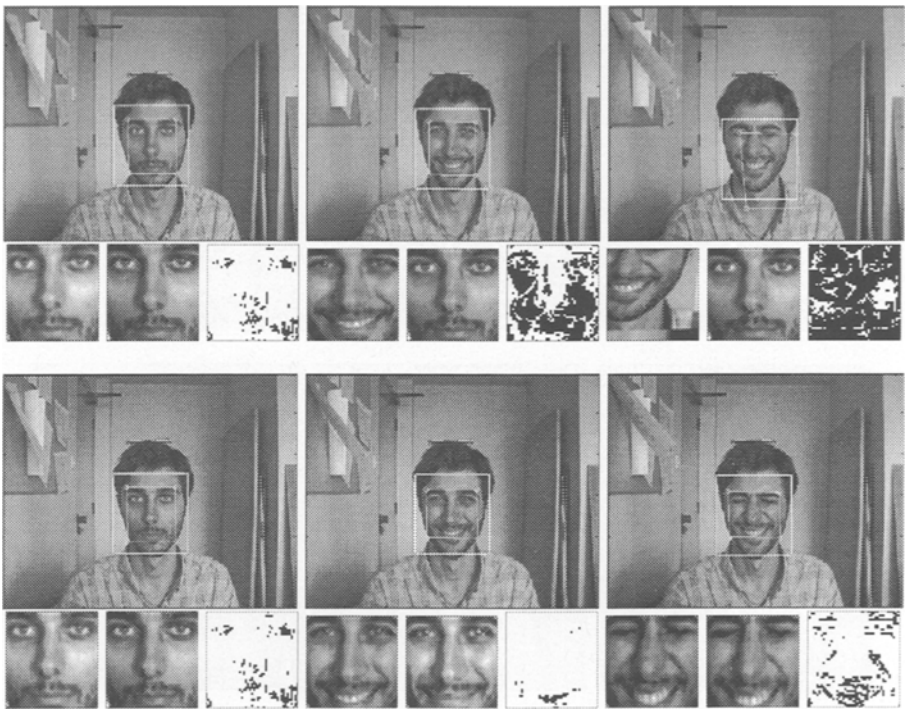

Fig. 8. This sequence demonstrates the advantage in using shape with non-rigid expression changes. The first two frames show the overlaid results from EigenTracking without shape. The corresponding three small images are, from left to right, the aligned object image, the reconstructed object image using the estimated affine parameters, and the outliers. The last two frames show the results on the same sequence from the adaptive scheme with shape encoded.

7 Conclusion

In this paper we presented an integrated scheme for view alignment. We exploited the transformation between the training set and the eigenspace in a computationally inexpensive manner in order to establish the correspondence between the landmarks in the training set and the image.

A dynamically adaptive scheme was adopted to compensate the small changes in illumination and the nonlinear transformations that cannot be recovered with global affine transformations. One advantage of our scheme is that the adaption does not change essentially the basis dataset of the eigenspace since the current tracked frame is only used once as an added bias in constructing the eigenspace at that time frame. It is not used to replace any initial sample images in the training set therefore the tracker is unlikely to eventually adapt and track the background.

Fig. 9. Every 10th frame from a sequence with attentional windows and aligned face images overlaid. Below each frame are the located region (left) and its reconstruction (right). The affine parameters were recovered using LES.

Our current scheme did not include a pre-filtering process to address changes in global illumination. This can be addressed by estimating the illumination in the new image dynamically with homomorphic filtering. An alternative approach to resolve the problem is to use an additional basis to represent all the probable illumination situations [4]. Gabor wavelets representation can also be employed which gives a certain degree of invariance to global illumination change [2].

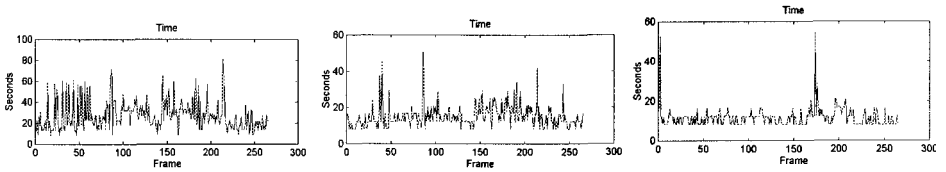

Fig. 10. Convergence times required by the minimisation process of different affine tracking schemes.

Currently, the parameters are only predicted independently. Prediction can therefore be rather under-constrained and allows for too much freedom. On the other hand, Fig. 7 indicates strongly that all the parameters should be tracked with a prediction model simultaneously since they are highly correlated and mostly linear in their correlations. This will enable the prediction and tracking much better constrained and consequently more robust and consistent.

Although the segmentation process in the initialisation stage utilises colour information, it still relies heavily upon morphological operations which can be relatively slow and inconsistent. Future work includes incorporating the adaptive affine tracking scheme developed here with a robust adaptive multi-colour model [17] in order to cope with changes in illumination over time.

References

1. D. Beymer and T. Poggio, "Image representations for visual learning," *Science*, vol. 272, pp. 1905–1909, June 1996.
2. S. McKenna, S. Gong, R. Wurtz, J. Tanner, and D. Banin, "Tracking facial feature points with gabor wavelets and shape models," in *IAPR International Conference on Audio-Video Based Biometric Person Authentication*, Crans-Montana, Switzerland, March 1997.
3. S. McKenna and S. Gong, "Real-time face pose estimation," *Real Time Imaging*, 1998, To appear in the Special Issue on Real-time Visual Monitoring and Inspection.
4. G. Hager and P. Belhumeur, "Real-time tracking of image region with changes in geometry and illumination," in *IEEE Conference on Computer Vision and Pattern Recognition*, 1996.
5. M. Black and Y.Yacoob, "Eigen tracking: Robust matching and tracking of articulated objects using a view-based representation," in *European Conference on Computer Vision*, Cambridge, England, April 1996.
6. P. Huber, *Robust statistics*, John Wiley and Sons, 1981.
7. S. Geman and D. McClure, "Statistical methods for tomographic image reconstruction," *Bull. Int. Statis. Inst.*, pp. 5–21, 1987.
8. M.J. Black and P. Anandan, "The robust estimation of multiple motions: Parametric and piecewise-smooth flow fields," *CVIU*, vol. 63, no. 1, pp. 75–104, 1996.
9. N. Sumpter, R. Boyle, and R. Tillett, "Modelling collective animal behaviour using extended point distribution models," in *British Machine Vision Conference*, Colchester, September 1997, pp. 242–251.

10. I. Craw, "A manifold model of face and object recognition," in *Cognitive and Computational Aspects of Face Recognition*, T. R. Valentine, Ed., pp. 183–203. Routledge, 1995.
11. Y. Raja, S. McKenna, and S. Gong, "Tracking and segmenting people in varying lighting conditions using colour," in *IEEE International Conference on Automatic Face and Gesture Recognition*, Nara, Japan, April 1998.
12. F. De la Torre, E. Martinez, E. Santamaria, and J.A Morin, "Moving object detection and tracking system: a real-time implementation," in *GRETSI*, Grenoble, 1997, pp. 375–378.
13. H. Murase and S. Nayar, "Detection of 3d objects in cluttered scenes using hierarchical eigenspace," *Pattern Recognition Letters*, vol. 18, pp. 375–384, 1997.
14. S. Nene and S. Nayar, "A simple algorithm for nearest neighbor search in high dimensions," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 19, no. 9, 1997.
15. S. Kay, *Fundamentals of statistical signal processing: Estimation theory*, Prentice Hall, 1993.
16. Z. Zhang, "Parameter estimation techniques: A tutorial with application to conic fitting," *Image and Vision Computing*, 1996.
17. Y. Raja, S. McKenna, and S. Gong, "Colour model selection and adaptation in dynamic scenes," in *European Conference on Computer Vision*, Freiburg, Germany, June 1998.