Tracking Faces

Stephen McKenna and Shaogang Gong
Machine Vision Laboratory,
Department of Computer Science
Queen Mary and Westfield College,
Mile End Road,
London,
England

e-mail stephen@dcs.qmw.ac.uk

Tracking Faces[*]

Stephen McKenna and Shaogang Gong
Machine Vision Lab., Dept. of Computer Science
Queen Mary and Westfield College,
Mile End Road, London, England
stephen@dcs.qmw.ac.uk

Abstract

Robust tracking and segmentation of faces is a prerequisite for face analysis and recognition. In this paper, we describe an approach to this problem which is well suited to surveillance applications with poorly constrained viewing conditions. It integrates motion-based tracking with model-based face detection to produce segmented face sequences from complex scenes containing several people. The motion of moving image contours was estimated using temporal convolution and a temporally consistent list of moving objects was maintained. Objects were tracked using Kalman filters. Faces were detected using a neural network. The essence of the system is that the motion tracker is able to focus attention for a face detection network whilst the latter is used to aid the tracking process.

1 Introduction

In order to analyse and recognise peoples' faces in realistically unconstrained environments, robust tracking and segmentation is a prerequisite. This provides a sequence of face images normalised with respect to scale and image-plane translation. Although such a *normalisation process* is often treated as a separate preprocessing step, it is of course an inherent part of face recognition. The ability of a system to produce a normalised face sequence implies that it recognises faces as a unique class of objects and in a manner which exhibits invariance under many possible transformations. These transformations include changes in illumination, changes of a face's orientation and position in 3D space relative to the camera, and changes in the motion of the face. The need for such invariances is what makes consistent face recognition difficult in surveillance applications.

Two broad approaches to the representation and tracking of moving objects are motion-based and model-based. Both methods have their relative strengths and weaknesses and seem to be complementary [9]. Motion-based approaches depend on a robust method for grouping visual motions consistently over time [10]. They tend to be fast but do not guarantee that the tracked regions have any meaning. Model-based approaches, on the other hand, can impose high-level semantic knowledge more readily but suffer from being computationally expensive due to the need to cope with scaling, translation, rotation and deformation.

In this paper, we propose a system for tracking faces which combines motion-based and model-based representations. We suggest an integrated face detection-tracking system with a *closed-loop* in which a motion-based tracker is used to reduce the search space for a model-based face detection whilst the latter is used to aid the motion tracking and resolve ambiguities in the grouping of visual motion. Our system outputs segmented face sequences suitable for face recognition from scenes containing several people. The availability of face sequences as opposed to isolated images then allows the use of temporal information to help constrain the recognition task [20].

It is worth noting here that in applications such as H.C.I. and access control, face detection and tracking are often less demanding (see e.g. [19]). There is usually a single user whose face fills a large proportion of the field of view at a resolution sufficient for localisation of facial features. In contrast, applications in surveillance complicate the task with poor spatial resolution and the need to handle multiple moving objects in real-time sequences [4], some or all of which may possess visible faces.

The remainder of this paper is arranged as follows. Section 2 describes a motion-based tracking system. Section 3 briefly reviews techniques for face detection in static scenes and describes the model-based matching method used by our system. Section 4 discusses the integration of the two techniques and describes the performance of the overall system. Finally, in section 5 we draw some conclusions for use in future work.

[*] The research was in part funded by EPSRC Grant GR/K44657 and EC Grant CHRX-CT94-0636.

2 Tracking multiple motions

In surveillance applications, we should like to track several people against a complex background using a stationary monochrome camera. This section describes such a tracker comprised of motion detection, grouping, temporal matching and Kalman filters.

Visual motion can be best estimated at moving image contours where such estimates are likely to be most relevant and reliable [3, 7]. This can be effectively achieved by convolving the intensity "history" of each image pixel $I(x, y, t)$ with the second order temporal derivative of a Gaussian function $G(t)$ which yields an image of temporal zero-crossings $S(x, y, t)$:

$$S(x,y,t) = \frac{\partial^2 G(t)}{\partial t^2} * I(x,y,t)$$

The motion of an edge in the image sequence then produces such a temporal zero-crossing in $S(x, y, t)$ at the location of the edge in the middle frame of the "history" used for the temporal convolution [7]. Global illumination changes and even changes in the intensity level of static objects do not result in such zero-crossings. The normal components of visual motion can be estimated from the partial derivatives of $S(x, y, t)$ [3, 7]. Figure 1 shows an image from a sequence of two people walking through our laboratory along with its temporally filtered image (in the middle) and the detected temporal zero-crossings (on the right).

For each frame, the detected temporal zero-crossings are clustered. Each cluster should correspond to a moving object although in practice this is not always the case with objects occasionally splitting into several clusters or separate objects merging into a single cluster. In the current implementation, clustering is performed using only the Euclidean distances between the temporal zero-crossings. Each cluster is modelled by its centroid and the standard deviations of its zero-crossings in the directions of the x and y image axes. Obviously, modelling the spatial extent of an object by assuming a Gaussian distribution of moving edge pixels is a crude approximation. A more detailed shape model could be used (e.g. [24]). A temporally consistent list of clusters is maintained by performing time-symmetric matching [25]. Forward matching selects the nearest cluster in the current frame whilst reverse matching selects the oldest candidate cluster. In this manner, clusters are consistently tracked even if they are sometimes erroneously split into several smaller clusters. Given a sufficiently high ratio of frame rate to image-plane velocity, normal components of visual motion can be used to improve the clustering process and in particular to help segment occluding objects with differing visual motions.

In our system, Kalman filters are used to robustly track the clusters in the cluster list based on measurements of their position, motion and shape. The coordinates of a cluster's centroid are modelled by a system model describing its position, velocity and acceleration. For the x coordinate (and similarly for y), an update condition is given by [11, 15, 27]:

$$\begin{bmatrix} x \\ \dot{x} \\ \ddot{x} \end{bmatrix}_{k+1} = \begin{bmatrix} 1 & \Delta t & \Delta t^2/2 \\ 0 & 1 & \Delta t \\ 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} x \\ \dot{x} \\ \ddot{x} \end{bmatrix}_{k} +$$

$$\begin{bmatrix} 0 \\ 1/2 \\ 1 \end{bmatrix} \mathbf{v_{acc}} + \begin{bmatrix} 1 \\ 0 \\ 0 \end{bmatrix} \mathbf{v_{pos}}$$

where $\mathbf{v_{acc}}$ and $\mathbf{v_{pos}}$ are noise covariances for assumptions about constant acceleration and position estimation. A measurement model is given by

$$\mathbf{z_{k+1}} = \mathbf{x_{k+1}} + \mathbf{w}$$

where $\mathbf{x_{k+1}}$ is the actual state vector at time t_{k+1} and $\mathbf{w}$ is a noise covariance vector for measurement errors. $\mathbf{z_{k+1}}$ is the observation vector at time t_{k+1}. Clusters can be simply modelled as having constant height (h_0) and width (w_0). An appropriate system model is

$$h_{k+1} = h_k + v_h, \quad w_{k+1} = w_k + v_w$$

where v_h and v_w are system noise, and the corresponding measurement model is then given by

$$z_{k+1} = h_{k+1} + u, \quad z_{k+1} = w_{k+1} + u$$

where u is measurement noise.

The bounding box of a cluster is only initialised after the cluster has been tracked for four frames. A cluster is then assigned a "persistence" parameter, p, and will be maintained even in the absence of a matching cluster for up to p frames[1]. This allows objects to be tracked for short periods of time despite clustering errors or an absence of detected motion. The left-most image in Figure 1 shows an example of estimated bounding boxes for the tracked objects and their "heads". The system successfully tracks multiple moving objects in the absence of occluding motions. It has been implemented using a Datacube with MaxVideo250 board for acquisition, image sampling, spatial and temporal convolution. The system can currently track objects at approximately 5Hz with a spatial resolution of 189×144 pixels.

Tracking faces based on motion information alone is often insufficient and computationally under-constrained, especially with multiple objects moving closely or under occlusions. If it is assumed that a tracked object is a person, the location of the head can be estimated relative to the tracked

[1] Currently, p is set to 10. This is largely dependent on the frame rate of the video signals.

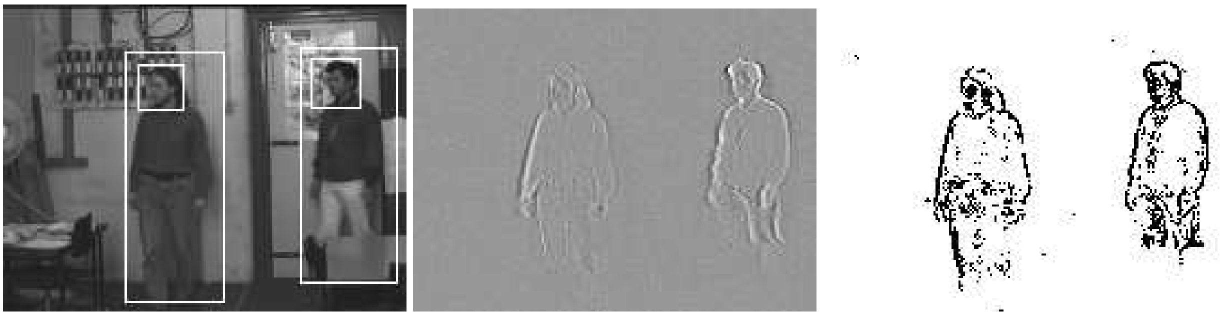

Figure 1. The motion-based tracker. Left: An image from a sequence taken in our lab with bounding boxes for the tracked people and their heads overlaid. Centre: The image after temporal convolution. Right: Detected temporal zero-crossings.

bounding box using some simple heuristics, as shown in Figure 1. However, such a crude approach can be easily broken down under most operational conditions in surveillance. An additional "face model" has to be present in order to constrain the problem.

3 Model-based face detection

Ideally, techniques for face detection are desired to perform across a range of lighting conditions, spatial scales, head poses and at least small changes in image-plane orientation. They should also detect faces irrespective of hairstyle, facial expression and the presence or absence of spectacles and make-up. Here we give a brief review of methods employed for face detection in *static* scenes before describing the method used in our system which is designed to find faces in *dynamic* scenes.

Static faces can be detected based upon simple shape information by using ellipse fitting or eigen-silhouettes [14, 22]. However, a robust method should incorporate information regarding the internal structure of faces. The property of facial symmetry has been used to align faces [14]. Colour provides a useful cue through the detection of natural skin tones [23, 29] and texture measures have also been incorporated [5]. The detection of local facial features (e.g. eyes, nose, lips) using photometric measurements appears to be unreliable and must be coupled with a model of the spatial arrangement of these features [2, 12]. At low spatial resolution such an approach would be even less robust.

A more promising approach for our purposes is the use of photometric representations which model faces as points in large multi-dimensional hyperspaces. The spatial arrangement of facial features is then implicitly encoded within an "holistic" representation of the internal structure of a face. In this detection framework, image patches are classified as either 'face' or 'non-face' vectors. A naive example of this approach could be the use of template matching of raw image intensities. Turk and Pentland [28, 18] suggested matching linear combinations of "eigenfaces" as a more robust alternative. In a refinement, probability density estimates for a "face class" were obtained using either Gaussian or Mixture-of-Gaussians density models in a principal sub-space of the image hyperspace [17]. Several classifiers have also been trained using images that contain non-face as well as face patterns in order to more accurately estimate the distribution boundaries of the face class. For example, Burel and Carel [1] used a vector quantization network to cluster face and non-face data while Sung and Poggio [26] used 12 Gaussian clusters (6 face and 6 non-face) to model the face class distribution. Both used a multi-layer perceptron (MLP) to classify the clustered data. In order to obtain a low false-positive rate[2], non-face patterns lying near the true decision boundary were needed. These patterns lie "close" to the distribution of face patterns and are therefore easily confused. The intuitive idea is that a classifier with decision boundaries trained to cope with these difficult non-face patterns will also be able to correctly classify easier non-face patterns lying "further" from face patterns in representation space. Such confusable non-face patterns can be selected using an iterative training method in which patterns incorrectly classified as faces are included in the training set for future training iterations [16, 1, 26, 21].

A weakness inherent in all these "holistic" methods is the representation of images as raster vectors without any coding for 2D topology or local spatial arrangement. Neural networks with locally connected receptive fields (RF's) have been suggested as a way to build in such knowledge. Fogelman-Soulie *et al.* [8] employed weight-sharing RF's as used in time-delay neural networks to extract local features irrespective of position on the input image. Rowley *et al.* [21] trained an MLP with square and horizontally elongated RF's obtaining good results.

[2] False-positive rate is defined as the fraction of non-face patterns incorrectly classified as face patterns.

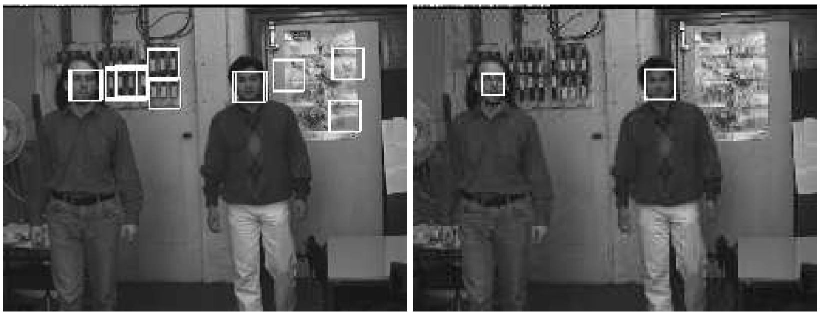

Figure 2. Left: Example output from a neural network based static face detector at a given scale. Erroneous detections occur due to "face-like" patterns. Right: Example output when the motion-based tracker and the model-based face detector are combined.

Our system uses a similar approach to that of Rowley *et al.*. A network was trained to detect frontal or near-frontal views. Other networks could be similarly trained to detect faces at different pose angles in order to build a hierarchical view-based detector exhibiting pose invariance. Face training images were normalised with respect to orientation and scale as described in [21]. An oval mask was applied to the resulting images (see Figure 3). The remaining windows consisted of 320 pixels. Both Sung and Poggio [26] and Rowley *et al.* [21] used a linear shading correction and histogram equalisation. Since we wished to minimise the amount of preprocessing needed during network operation we merely subtracted the mean intensity and divided by the standard deviation of the intensity values in the window.

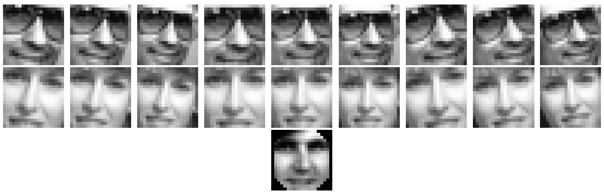

Figure 3. Examples of the face images used for training the neural network. The lower image shows the mask used.

One thousand face images were assembled from various face databases most of which were publicly available (see Acknowledgements). The majority of faces came from the Usenix face database. Non-face patterns were extracted from a database containing 70 images of indoor and outdoor scenes. The training set was expanded by rotating the faces through 10° and scaling them to 90% and 110%. This forced the network to learn tolerance to small amounts of scaling and rotation in the image-plane. Invariance to larger changes in scale was obtained by using the network to scan images in a pyramid. A total of 9000 face images were used during training. Some examples can be seen in Figure 3. An MLP was trained using back-propagation with iterative selection of false-positive non-face patterns.

The left image in Figure 2 shows the output of a neural network when used to scan an image at one particular scale. This example illustrates some of the weaknesses inherent in this form of static scene detector. Since an image contains a very large number of network sized patches, the detector must achieve extremely low false-positive rates. The number of false-positives can be reduced by discarding isolated detections and merging overlapping ones [21]. However, there will always exist non-face image patches which when taken out of their spatial and temporal context appear "face-like".

4 Tracking faces

Face detection by matching all possible sub-images with a 2D model is computationally infeasible in dynamic scenes. In our system, motion-based tracking is used to focus attention for this matching process. Motion prediction provides both the scale and the location for the face matching. Face pose could also be estimated using prediction since this also varies smoothly over time. The matching process need not localise the face correctly in every frame in order to track it successfully.

A further benefit of combining motion-based tracking with model-based matching is the ability to provide feedback from the matching process to the tracking process. A good face match could be used to stabilise the tracker by, for example, estimating the measurement noise on-line. The matching process can also help to resolve ambiguities which arise during motion grouping. For example, a cluster which is consistently found to possess two heads should probably be split into two clusters.

Once a person is being tracked, their face is searched for within the estimated head region. Once detected, the face

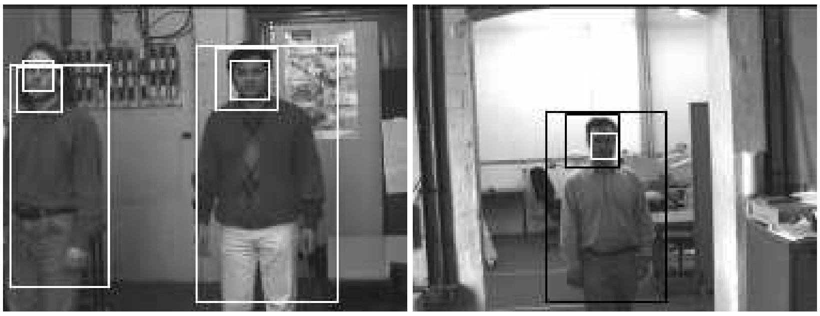

Figure 4. Example output frames from our system. The three bounding boxes are the object and head bounding boxes estimated by the motion-based tracker and the face detected in combination with the model-based face detector.

is tracked using the neural network until such time as several frames elapse without a strong face match occurring. At this point, the face is again searched for within the estimated head region of the associated motion cluster.

Figure 4 shows an example of output from the system. Three bounding boxes are overlaid on the sequence. The outermost two are the object and head bounding boxes as estimated by the motion-based tracker. The innermost box is produced in combination with the model-based face detector. Once a face is detected it can be tracked despite inaccurate motion clustering and resulting erroneous head region estimation.

5 Conclusions

This paper has dealt with the task of tracking faces in complex scenes such as arise in surveillance applications. A real-time multi-motion tracker was implemented using Kalman filters to track objects as groups of temporal zero-crossings. This was combined with a model-based neural network face detector. Motion-based and model-based representations provide two complementary approaches to the tracking problem which when combined ultimately as a closed-loop system yield more robust solutions than either approach in isolation. Such a system has been described and demonstrated.

Darrell *et al.* describe a face tracking system for use in an interactive room which is similar in some respects [6]. Our system is designed with a different set of applications in mind and as such is based upon a rather different set of assumptions and objectives. It relies on a single, fixed, uncalibrated, wide-angle, monochrome camera. It does not build any explicit model of the background and it can track more than one person at a time.

We are currently investigating the performance of face recognition based upon the segmented face sequences output by the tracker. Howell and Buxton report some preliminary results [13].

Future work will also be concerned with closer integration of the motion and model-based approaches as well as improving their performances in isolation. A number of avenues for improvement of the face detector suggest themselves. These include the use of weight-sharing to enforce symmetry and determination of good receptive field (RF) structures. Certain RF configurations could lend themselves to real-time implementation on our pipeline hardware. Models for other face views such as 3/4 and profile would increase the robustness of the tracking system by providing good pattern matches in a higher proportion of frames. Better mechanisms for providing feedback from the model-based matching to the Kalman filters and the grouping processes also merit further investigation.

6 Acknowledgements

This work used databases from ORL, Usenix and Universities of Essex, Manchester, Bern and Bochum. We thank those who helped provide data including Rolf Würtz, Henry Rowley, Woody Yang, Libor Spacek, Andreas Lanitis and Bernard Achermann.

References

[1] G. Burel and D. Carel. Detection and localization of faces on digital images. In *Patt. Recog. Letts.*, volume 15, pages 963–967, 1994.

[2] M. C. Burl, T. K. Leung, and P. Perona. Face localization via shape statistics. In *IWAFGR*, pages 154–159, 1995.

[3] B. F. Buxton and H. Buxton. Monocular depth perception from optic flow by space time signal processing. *Proc. Royal Soc. of London*, B-218, 1983.

[4] H. Buxton and S. Gong. Visual surveillance in a dynamic and uncertain world. *Artificial Intelligence*, 78:431–459, 1995.

[5] Y. Dai and Y. Nakano. Extraction of facial images from complex background using color information and sgld matrices. In *IWAFGR*, pages 238–242, 1995.

[6] T. Darrell, B. Moghaddam, and A. P. Pentland. Active face tracking and pose estimation in an interactive room. In *CVPR*, 1996.

[7] J. H. Duncan and T.-C. Chou. On the detection of motion and the computation of optical flow. *IEEE PAMI*, 14(3), 1992.

[8] F. Fogelman-Soulie, E. Viennet, and B. Lamy. Multimodular neural network architectures: applications in optical character and human face recognition. *Int. J. of Patt. Recog. and A. I.*, 1994.

[9] S. Gil, R. Milanese, and T. Pun. Combining multiple motion estimates for vehicle tracking. In *ECCV*, volume II, pages 307–320, 1996.

[10] S. Gong and H. Buxton. Bayesian nets for mapping contextual knowledge to computational constraints in motion segmentation and tracking. In *BMVC*, 1993.

[11] S. Gong, A. Psarrou, I. Katsoulis, and P. Palavouzis. Tracking and recognition of face sequences. In *Eur. Workshop on Combined Real and Synthetic Image Processing for Broadcast and Video Production*, Hamburg, 1994.

[12] H. P. Graf, T. Chen, E. Petajan, and E. Cosatto. Locating faces and facial parts. In *IWAFGR*, pages 41–46, 1995.

[13] A. J. Howell and H. Buxton. Towards unconstrained face recognition from image sequences. In *ICAFGR*, 1996.

[14] A. Jacquin and A. Eleftheriadis. Automatic location tracking of faces and facial features in video sequences. In *IWAFGR*, 1995.

[15] S. J. McKenna, S. Gong, and H. Liddell. Real-time tracking for an integrated face recognition system. In *Proc. 2nd PAMONOP Workshop, Faro*, 1995.

[16] S. J. McKenna, I. W. Ricketts, A. Y. Cairns, and K. A. Hussein. Cell searching with a neural net. In *Neural computing - research and applications II*, pages 205–216. IOP, 1993.

[17] B. Moghaddam and A. Pentland. Maximum likelihood detection of faces and hands. In *IWAFGR*, pages 122–128, 1995.

[18] A. Pentland, B. Moghaddam, and T. Starner. View-based and modular eigenspaces for face recognition. In *IEEE CVPR*, 1994.

[19] C. Ponticos. A robust real time face location algorithm for videophones. In *BMVC*, pages 449–458, 1993.

[20] A. Psarrou, S. Gong, and H. Buxton. Modelling spatio-temporal trajectories and face signatures on partially recurrent neural networks. In *ICNN*, Perth, Australia, 1995. IEEE.

[21] H. A. Rowley, S. Baluja, and T. Kanade. Human face detection in visual scenes. Technical Report CMU-CS-95-158R, CMU, 1995.

[22] A. Samal and P. A. Iyengar. Human face detection using silhouettes. *Int. J. of Patt. Recog. and A. I.*, 9(6):845–867, 1995.

[23] B. Schiele and A. Waibel. Gaze tracking based on face-color. In *IWAFGR*, pages 344–349, 1995.

[24] S. M. Smith. Asset-2: Real-time motion segmentation and shape tracking. In *IEEE ICCV*, 1995.

[25] S. M. Smith and J. M. Brady. A scene segmenter: visual tracking of moving vehicles. *Engineering applications of Artificial Intelligence*, 7(2):191–204, 1994.

[26] K. Sung and T. Poggio. Example-based learning for view-based human face detection. Technical Report AI Memo 1512, CBCL 103, MIT, 1995.

[27] P. Torr, T. Wong, D. Murray, and A. Zisserman. Cooperating motion processes. In *BMVC*, 1991.

[28] M. Turk and A. Pentland. Eigenfaces for recognition. *J. Cog. Neuroscience*, 3(1):71–86, 1991.

[29] H. Wu, Q. Chen, and M. Yachida. An application of fuzzy theory: face detection. In *IWAFGR*, pages 314–319, Zurich, June 1995.