ELSEVIER

Image and Vision Computing 20 (2002) 359–368

image AND vision COMPUTING

www.elsevier.com/locate/imavis

Composite support vector machines for detection of faces across views and pose estimation

Jeffrey Ng*, Shaogang Gong

Department of Computer Science, Queen Mary and Westfield College, University of London, London E1 4NS, UK

Received 16 October 2000; accepted 18 December 2001

Abstract

Support vector machines (SVMs) have shown great potential for learning classification functions that can be applied to object recognition. In this work, we extend SVMs to model the appearance of human faces which undergo non-linear change across multiple views. The approach uses inherent factors in the nature of the input images and the SVM classification algorithm to perform both multi-view face detection and pose estimation. © 2002 Published by Elsevier Science B.V.

Keywords: Support vector machine; Face detection; Pose estimation

1. Introduction

Tracking people across a variety of views is becoming increasingly important in computer vision systems. Apart from traditional applications such as segmenting faces for identity recognition [4,8,15], multi-view face detection and tracking are also being used in smart-systems for visually mediated interaction [14], inferring user intention in human–computer interaction [1] and incident monitoring [13]. The ability to extract visual cues such as gait or head orientation allows advanced vision systems to better extract information about their contextual situation. A better perception of their prevailing operating conditions allows such systems to interact more intelligently with their environment.

Head pose, in particular, provides good cues about the general focus of attention of people. However, the appearance of the human head can change drastically across different viewing angles, mainly caused by non-linear deformations during in-depth rotations of the head. Existing template-matching and neural network systems would be hard pressed to learn the whole gamut of multi-view face appearances. Systems based on similarity measures to prototypes, on the other hand, are to some extent still restricted to the available views for the prototypes selected in the training database [4]. Similarity measures can be noisy and sensitive to the choice of representation. In order to perform accurate and robust face tracking across views, its generalisation to novel views requires prior pose information to be available.

In this work, we exploit the potential of support vector machines (SVMs) [16] for generalising and transforming a generic 2D facial appearance model across the view sphere [3,4]. In particular, we investigate the ability of SVMs to identify important prototypes across different face poses to provide a plausible solution for effective face detection at different views, tracking across views and pose estimation at no extra cost. We offer a viable solution for addressing the needs for both multi-view face detection and pose estimation at near-frame rate.

2. Support vector machines

SVMs are based on a generic learning framework that has shown unique potential in resolving some computer vision problems [7,10–12,16]. SVMs have been applied to learn wavelet coefficients for detecting human faces and pedestrians [7,8]. Mohan et al. [6] used hierarchical SVMs to combine separate body-part SVM classifiers into an object detector that can deal with partial occlusion and a certain degree of in-depth object rotation. SVMs have also been applied to the task of event detection by learning object sizes and trajectories [9]. Let us first outline the basic concept of this approach to learning classification functions for object recognition.

* Corresponding author. Tel.: +44-20-7882-5214; fax: +44-20-7882-6533.

0262-8856/02/$ - see front matter © 2002 Published by Elsevier Science B.V.
PII: S0262-8856(02)00008-2

2.1. Structural risk minimisation

Previous approaches to statistical learning have tended to be based on finding functions to map vector-encoded data to their respective classes. The conventional minimisation of the empirical risk over training data does not, however, imply good generalisation to novel test data. Indeed, there could be a number of different functions which all give a good approximation to a training data set. It is nevertheless difficult to determine a function which best captures the true underlying structure of the data distribution. Structural risk minimisation (SRM) aims to address this problem and provides a well-defined quantitative measure for the *capacity* of a learned function to generalise over unknown test data. Due to its relative simplicity, Vapnik–Chervonenkis (VC) dimension [16] in particular has been adopted as one of the more popular measures for such a capacity. By choosing a function with a low VC dimension and minimising its empirical error to a training data set, SRM can offer a guaranteed minimal bound on the test error.

Perhaps the notion of VC dimension can be more clearly illustrated through hyperplane classifiers. Given a data set $\{\mathbf{x}_i, y_i\}$, $i = 1, \ldots, l$, $\mathbf{x} \in R^N$, $y \in \{+1, -1\}$, a hyperplane such as

$$(\mathbf{w}\cdot\mathbf{x}) + b = 0, \qquad \mathbf{w} \in R^N, \ b \in R, \tag{1}$$

can be oriented across the input space to perform a binary classification task, minimizing the empirical risk of a hyperplane decision function $f(\mathbf{x}) = \text{sign}((\mathbf{w}\cdot\mathbf{x}) + b)$. This is achieved by changing the normal vector $\mathbf{w}$, also known as the weight vector. There is usually a margin on either side of the hyperplane to the two classes. The VC dimension of the decision function decreases, and therefore improves, with an increasing margin. To obtain a function with the smallest VC capacity and the optimal hyperplane, one has to maximise the margin:

$$\text{Maximise } W(\boldsymbol{\alpha}) = \sum_{i=1}^{l} \alpha_i - \frac{1}{2} \sum_{i,j=1}^{l} \alpha_i \alpha_j y_i y_j (\mathbf{x}_i \cdot \mathbf{x}_j) \tag{2}$$

$$\text{Subject to } \alpha_i \geq 0, \qquad i = 1, \ldots, l \text{ and } \sum_{i=1}^{l} \alpha_i y_i = 0 \tag{3}$$

The optimal hyperplane is mainly defined by the weight vector $\mathbf{w}$ expressed in terms of the data elements with non-zero Lagrange multipliers (α_i) in Functional (2). Those data elements lie on the margins of the hyperplane. They therefore define both the hyperplane and the boundaries of the two classes. The decision function of the optimal hyperplane is thus:

$$f(\mathbf{x}) = \text{sign}\left(\sum_{i=1}^{l} y_i \alpha_i (\mathbf{x}\cdot\mathbf{x}_i) + b\right) \tag{4}$$

2.2. Support vector machines using kernel functions

A hyperplane classification function attempts to fit an optimal hyperplane between two classes in a training data set, which will inevitably fail in cases where the two classes are not linearly separable in the input space. Therefore, a high dimensional mapping

$$\phi : R^N \mapsto F$$

is used to cater for non-linear cases. As both the objective function and the decision function is expressed in terms of dot products of data vectors $\mathbf{x}$, the potentially computation-intensive mapping $\phi(\cdot)$ does not need to be explicitly evaluated. A kernel function, $k(\mathbf{x}, \mathbf{z})$, satisfying Mercer's condition can be used as a substitute for $(\phi(\mathbf{x})\cdot\phi(\mathbf{z}))$ which replaces $(\mathbf{x}\cdot\mathbf{z})$ [16].

For noisy data sets where there is a large overlap between data classes, error variables $\varepsilon_i > 0$ are introduced to allow the output of the outliers to be locally corrected, constraining the range of the Lagrange multipliers α_i from 0 to C. C is a constant which acts as a penalty function, preventing outliers from affecting the optimal hyperplane. Therefore, the non-linear objective function is

$$\text{Maximise } W(\boldsymbol{\alpha}) = \sum_{i=1}^{l} \alpha_i - \frac{1}{2} \sum_{i,j=1}^{l} \alpha_i \alpha_j y_i y_j (k(\mathbf{x}_i, \mathbf{x}_j)) \tag{5}$$

$$\text{Subject to } 0 \leq \alpha_i \leq C, \ \ i = 1, \ldots, l \text{ and } \sum_{i=1}^{l} \alpha_i y_i = 0 \tag{6}$$

with corresponding decision function given by

$$f(\mathbf{x}) = \text{sign}\left(\sum_{i=1}^{l} y_i \alpha_i k(\mathbf{x}, \mathbf{x}_i) + b\right) \tag{7}$$

There are a number of kernel functions which have been found to provide good generalisation capabilities, e.g. polynomials. Here we explore the use of a Gaussian kernel function (analogous to RBF networks) as follows:

$$\text{Gaussian Kernel } k(\mathbf{x}, \mathbf{y}) = \exp\left(\frac{|\mathbf{x} - \mathbf{y}|^2}{2\sigma^2}\right) \tag{8}$$

3. The nature of face pose distribution

Detecting human faces across views involves the recognition of a whole spectrum of very different face appearances. The pose of the head reveals some details about the 3D structure of the face while the prominent ridges can mask others. Head rotations introduce non-linear deformations in captured face images while the rotation can occur in two axes outside the view plane of the camera. A face's main direction of reflection of light also changes and affects the illumination conditions of the captured image. For instance, ambient

Fig. 1. A sample view sphere image-array with calibrated elements varying horizontally from 0 to 180° yaw and vertically from 60 to 120° tilt.

daytime lighting conditions in normal office environments are hardly symmetric for the top and bottom hemispheres of the face, while the bias towards the upper hemisphere is exacerbated by ceiling-fixed light sources during the night.

The view sphere provides a framework for analysing face pose distribution and for training SVMs over the infinite number of possible pose angles of human faces. For collecting training data, a 3D iso-tracking machine can be used to capture human faces at preset yaw (lateral) and tilt (vertical) angles. The tracking mechanism can also provide semi-automatic segmentation facilities for cropping the face. The result is an array of accurately calibrated and cropped images as shown in Fig. 1.

A face rotating across views forms a smooth trajectory as can be seen in Fig. 2. In fact, faces form continuous manifolds across the view sphere in a pose eigenspace (PES). It is plausible to suggest that head rotations describe a continuous function in PES. This can be seen more clearly in Fig. 3. In particular, a pattern appears for the vertical positioning (from the selected view angle) of the groups of trajectories across the view sphere. The volume enclosed by the entire view sphere is more visible when the nodes of the sphere are plotted individually as in Fig. 4. The distribution appears to be a convex hull.

Given the correlations of the lateral bands of the face sphere, we group the whole distribution into 19 different clusters according to their yaw orientation (0–180°). We observed that the trajectory of the mean positions of the clusters, which are indeed their centroids in PES, structures the distribution across a main axis of variation. This notion is further supported by the tangentiality of the main axes of local variation inside the clusters across the mean trajectory as shown in the lower right picture in Fig. 4. The above observations strongly

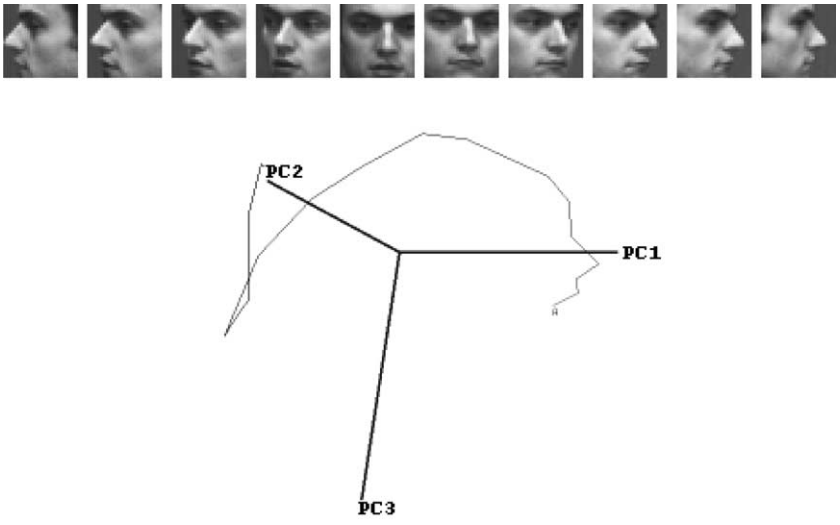

Fig. 2. Face rotation in depth forms a smooth trajectory in a 3D pose eigenspace.

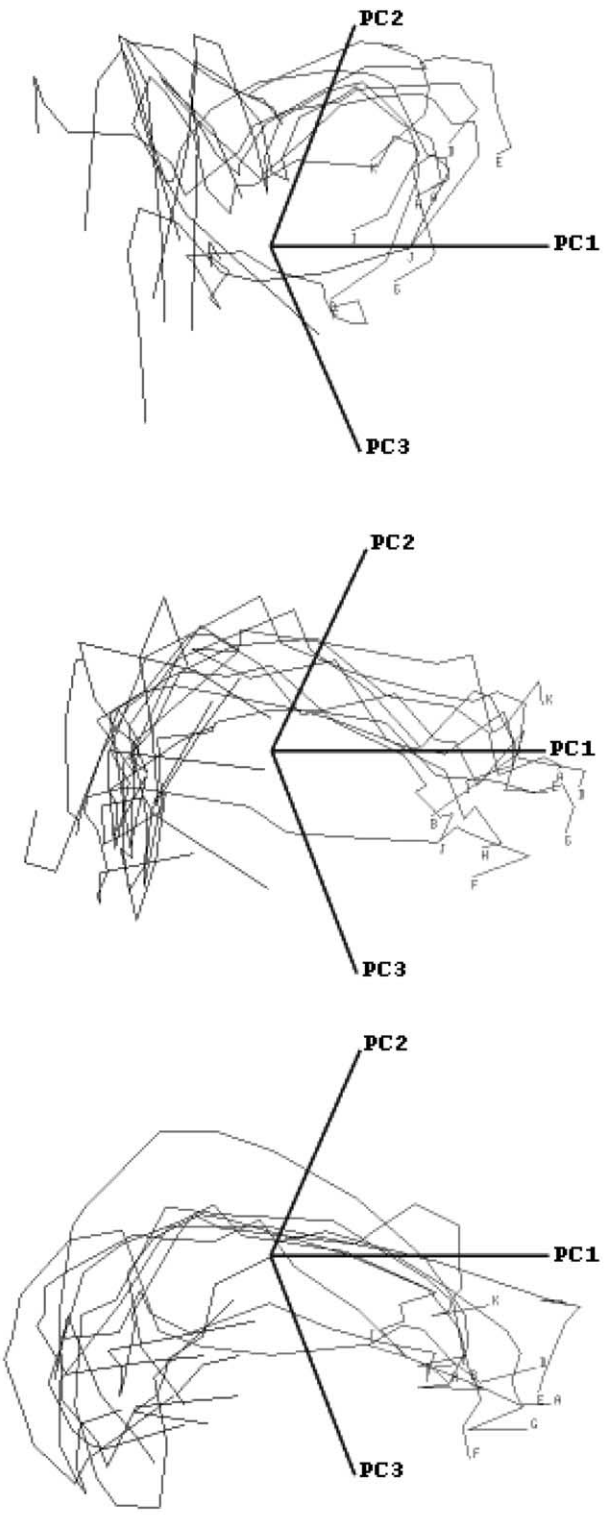

Fig. 3. From top to bottom: the graphs show the PES trajectories for a set of 10 people rotating their heads from profile to profile, at 60°, 90° and 120° tilt, respectively.

suggest that the convex hull is more akin to a 'tube', a volume function, through which data elements 'flow' from one end to the other as their yaw angles increase from 0 to 180°.

4. Learning a face model across views using SVMs

SVMs perform automatic feature extraction and enable the construction of complex non-linear decision boundaries for learning the distribution of a given data set. As the task to be learned is intrinsically a detection task, the bias introduced for a particular subset of the feature space by projection into eigenspace is deemed to be detrimental to the learning process. Therefore, pre-processing of the images is limited to masking and intensity normalisation only and the resulting images are learned as data vectors. The learning process and the number of support vectors for a data set are determined in a principled way by only a few customisable parameters which define the characteristics of the learned function. In our case, the parameters are limited to two: C, the penalty value for the Lagrange multipliers to distinguish between noisy data and, σ for determining the effective range of the Gaussian kernels. Effective values for the two parameters have already been reported for frontal view face detection [7]. A value of $C = 197$ was adopted for the training process and alternate values of C were found to have minimal impact on classification accuracy. On the other hand, the parameter σ controls the 'shape' of the agglomerated Gaussian kernels and a value of $\sigma = 158$ was empirically found to provide good results while other values did not appear to have any direct correlation with classification accuracy.

We adopt a semi-iterative approach for obtaining good examples of negative training data as proposed in Ref. [8]. The ideal negative images chosen by SVM training algorithms for negative support vectors have been reported to be naturally occurring non-face patterns that possess a strong degree of similarity to a human face [7]. Given the highly complex distribution of the view sphere described in Section 3, it is crucial to find good examples of these to allow the training algorithm to construct accurate decision boundaries.

We first extend the training of a single frontal-view SVM face model to the use of face images across the view sphere. The process uses an iterative refinement methodology to find important negative training samples from a database of randomly selected scenery pictures. This process is shown in Fig. 5. The resulting single SVM cannot cope with the degree of view generalisation required when applied to face detection across a significantly large range of views away from the frontal view. However, the model is very useful for iteratively collecting negative training samples beyond the near frontal view. Such negative samples are then used to train a multi-view face model based on a set of local component SVMs along the view sphere.

Given the face distribution in PES as shown in Fig. 4, the view sphere can be divided into smaller, more localised yaw segments as in Table 1. The observed asymmetry of the view sphere distribution and the greater complexity of the left portion are reflected into the selection of smaller segments for that region.

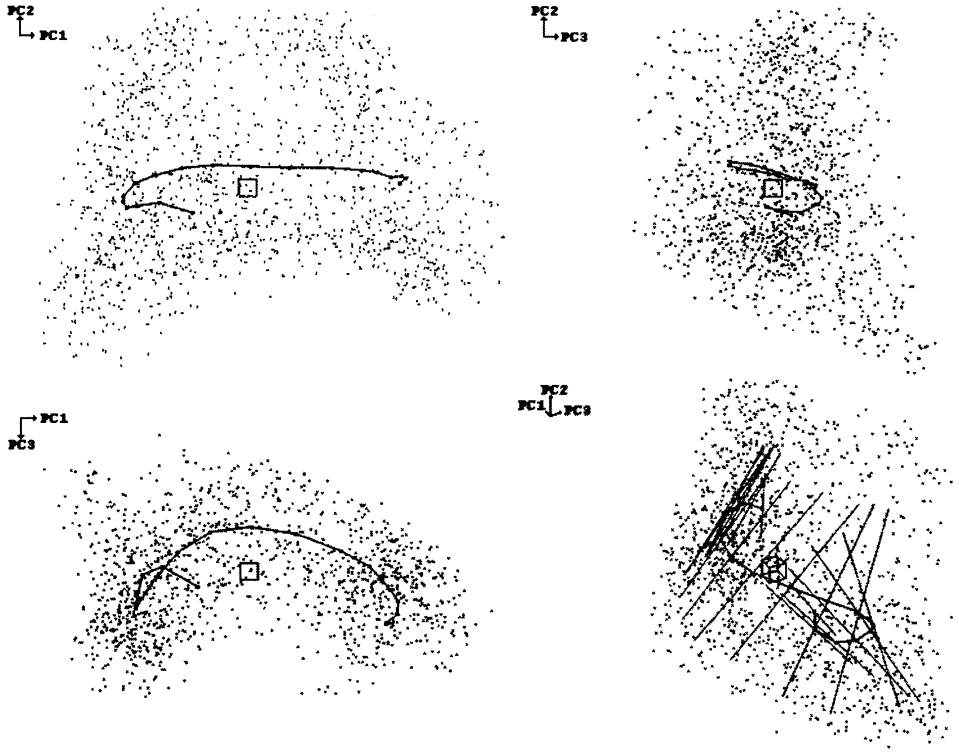

Fig. 4. Counter-clockwise from the upper right image: side, front and top views of the distribution of the face sphere, with the trajectory of the mean yaw clusters. The lower right image uses a special angle to show the direction of biggest variance of the yaw clusters (by the tangential lines) across the mean yaw positions.

All the component SVMs were trained on the same global negative data set. The size of the negative training data is about 6000 images and of those, the SVMs selected 1666 as negative support vectors in total, with only 36 shared between two or more component SVMs. This shows that the negative support vectors are well localised to the subspace of each yaw segment.

The modelling capabilities of the component SVMs and their tendency to overflow to the neighbouring segments corroborated with the previous observations of the structure of the distribution of the view sphere in PES. In general, the component SVMs could detect faces at yaw angles of 10° on either side of their training ranges. In some cases, the overlap was as much as 30°. The observed phenomenon also shows that support vectors are localised in a composite distribution such as the view sphere. They can be used to detect either the whole distribution or smaller segments in that distribution.

For face detection across the view sphere, the component SVMs can be arranged into a linear array to form a

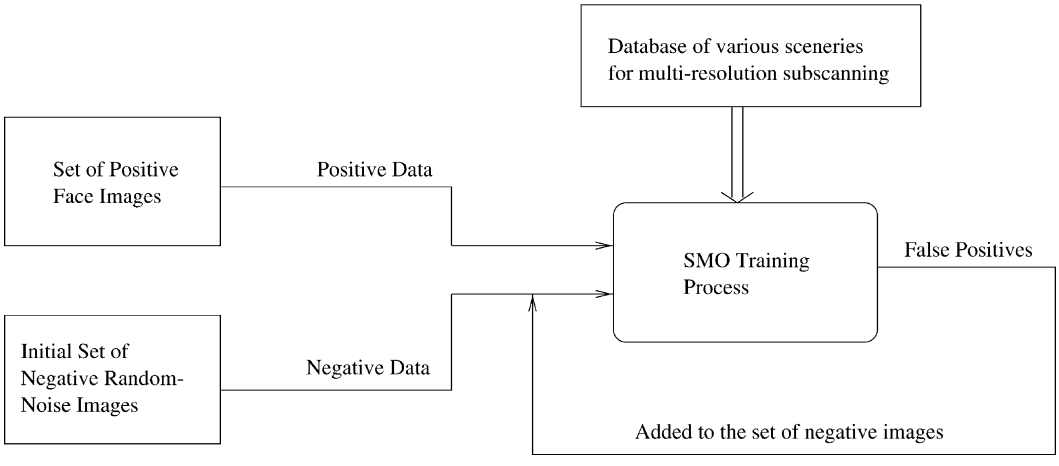

Fig. 5. Boot-strapping technique for obtaining negative support vectors.

Table 1
The division of the view sphere for learning multi-view SVMs

Segment	1	2	3	4	5
Yaw angles (°)	0–10	20–40	50–80	90–130	140–180
No. of Elems	140	210	280	350	350
No. of Pos SVs	107	139	176	190	203

composite SVM classier as follows:

$$\text{Composite SVM}(\mathbf{x}) = \text{sign}\left(\sum_{i=1}^{n} \text{SVM}(i,\mathbf{x}) + 1\right) \quad (9)$$

where SVM(i,$\mathbf{x}$) is the decision function $f(\mathbf{x})$ for SVM number i and n is the number of component functions used.

This multi-view face model can also be applied to pose estimation across the view sphere. Fig. 4 shows the correspondence of the yaw angles to the data elements' positions along the mean trajectory of the yaw clusters. A similar correspondence of the tilt angles to their 'vertical position' from the selected viewing angles, with the variation lying approximately perpendicular to the mean yaw trajectory, can also be observed in Fig. 3.

Since the support vectors define the boundaries of the face pose distribution, they lie on the 'walls' of the 'tube'. Furthermore, they are also localised with regard to the pose sphere. Therefore, they can be effectively used to perform pose estimation by using nearest-neighbour matching, as shown in Fig. 6. Conceptually, SVMs are believed to attempt to identify key support vectors which can solely represent the structure of the face distribution across views. The relative accuracy of the nearest-neighbour-based pose-estimation technique when applied to support vector prototypes, as shown in Section 5, lends a certain degree of credence to the theory. In fact, this process of pose estimation is retrieved at no extra computational cost to the calculation of the decision function. Pose-estimation has previously been performed separately from the face detection process by learning the variations of point distribution shape models [5] and active appearance models [2]. Wang et al. [17] uses multiple cameras and hairline based features to estimate the pose.

Table 2
Face detection on training data across the view sphere, grouped by human subject

Training subsets	Full detection (%)	Multi-scaling (%)
1	100	100
2	97.7	100
3	94.7	100
4	92.5	97.0
5	82.7	85.7
6	88.7	99.2
7	94.7	97.7
8	100	100
9	99.2	100
10	97.0	98.4

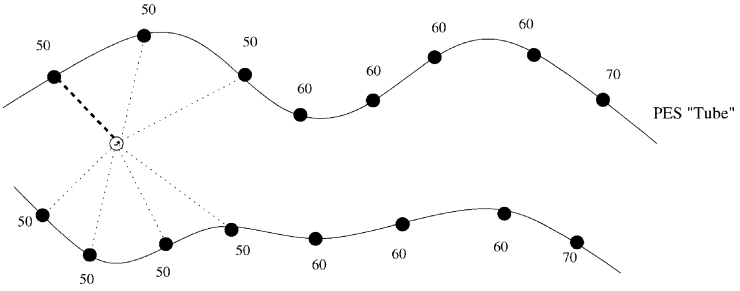

Fig. 6. Top view of the face manifold across the pose eigenspace with pan angles labelled to each support vector (dark circles). The pose orientation of the classification image (white circle) is retrieved from that of the closest support vector.

5. Multi-view face detection and pose estimation

We have applied the multi-view SVM-based face model to perform both multi-view face detection and pose estimation across views. First, we show the performance of the multi-view face detection system on training data given in Table 2.

It is important to point out that the accuracy in the alignment of face images plays a crucial role in the learning process. Most of the misclassified elements of the view sphere were correctly recognised after multi-scale scanning of the images. The multi-scale scanning is performed on the input images with a bias in each of the four directions to correct misalignment of the face images.

Our previous work reported that the variation of the view sphere distribution along the second principle component

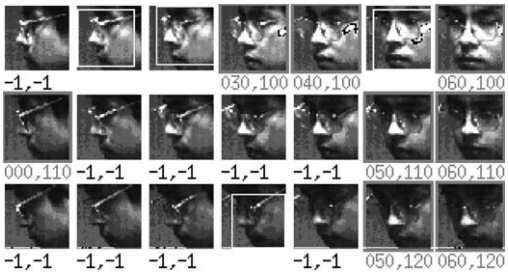

Fig. 7. Misclassification in lower hemisphere of the view sphere (shown by $-1, -1$). Image multi-scaling is shown with white rectangles.

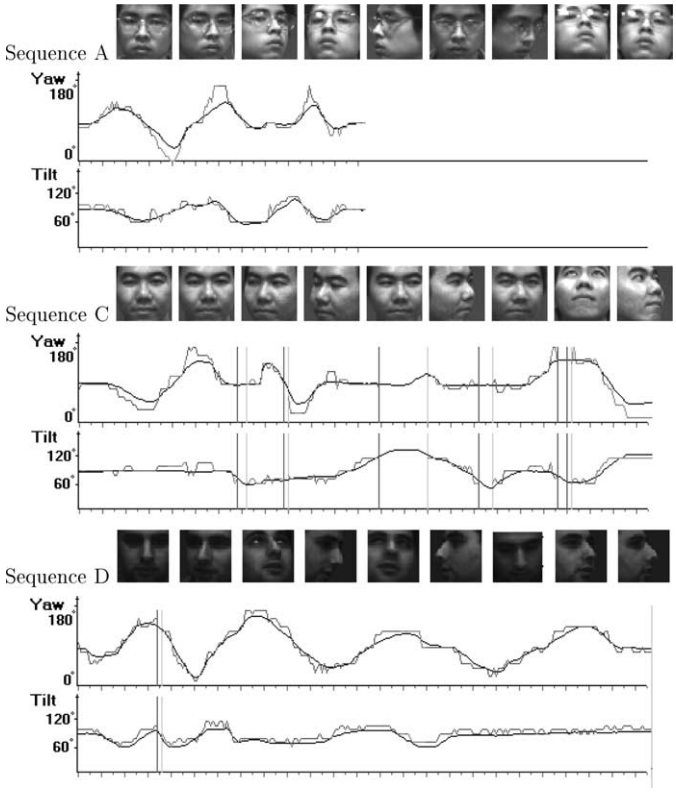

Fig. 8. Examples of detected and tracked moving faces. The graphs also show the estimated face pose (in grey) over time and their corresponding ground-truths (in black), measured by electro-magnetic sensors. The vertical lines indicate moments in time where no face was detected.

axis was highly related to the level of local lighting in the image [3]. Using an overhead light source can yield such an effect on the captured images. The lighting conditions must therefore help in the determination of the tilt orientation of the faces. However, it also makes down-facing poses very poorly illuminated and therefore, very difficult to detect by the system as shown in Fig. 7.

The multi-view system was tested over a number of test sequences of human subjects freely turning their heads in 3D space, with the ground-truths of the pose information measured for comparison. The system was also connected to an iso-tracking device, allowing face detection (alignment) and pose estimation to be independently evaluated. Experiments on three subjects are given here for illustration: the subject with the worst training detection results (test sequences A and B) and two novel subjects unknown to the training process (test sequences C and D). They were selected to test the generalisation capabilities of the system. Figs. 8–10 show example frames from different test sequences in which novel faces were detected in multi-views and tracked across views with their pose estimated simultaneously. Table 3 provides a summary of the results for the test sequences and it can be noticed that the detection rate and the average pose estimation error do not vary significantly between the sequences of known subjects and those of unknown subjects, namely sequences A and B against C and D.

6. Real-time performance

SVMs use kernel functions to learn and classify non-linearly separable data distributions. With typical support vector sets of SVMs ranging in thousands and a kernel evaluation

Table 3
Test results of the multi-view face detector and pose estimator from a total of over 1000 images from a set of test sequences

Test sequence	Detection rate (%)	Mean yaw error (°)	Mean tilt error (°)
A	100	11.07	6.62
B	84.9	11.467	6.32
C	82.9	13.57	7.29
D	99.6	8.73	8.67
E	99.2	8.90	8.21

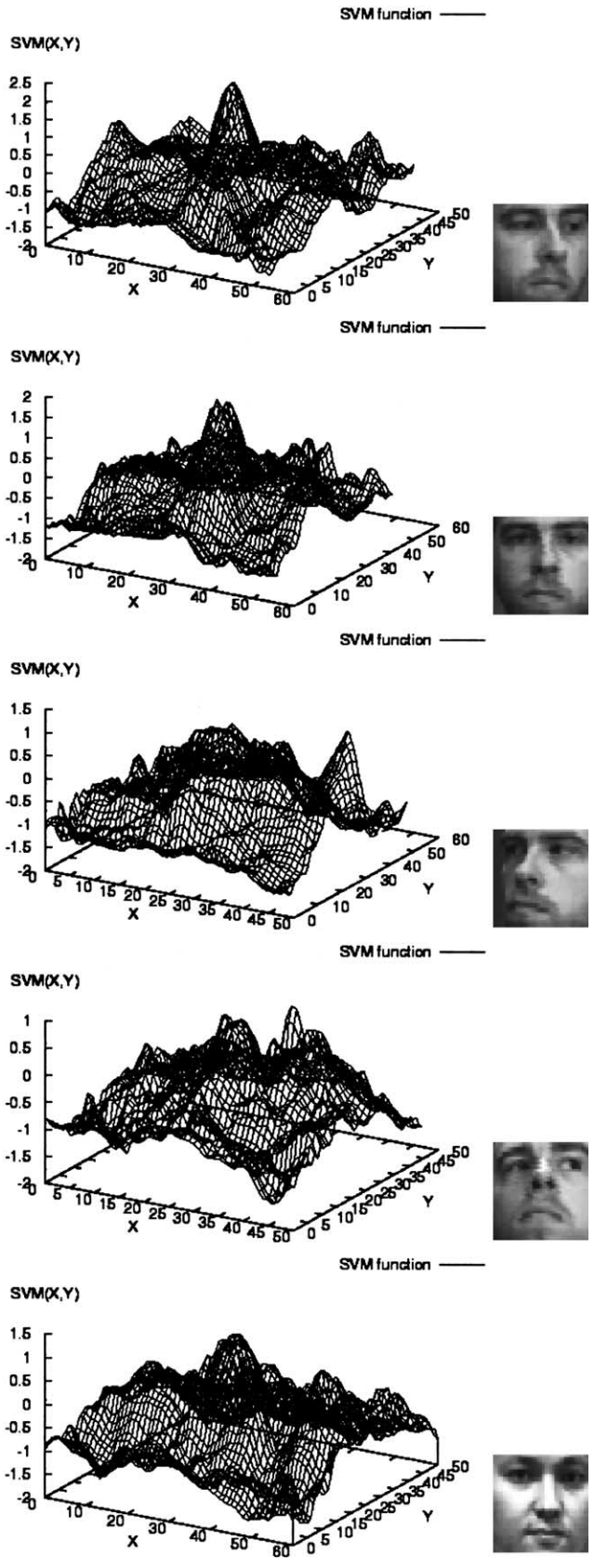

Fig. 9. Examples of topographical outputs from a multi-view composite SVM classifier when applied to face detection at different views. The detected faces are also shown.

required for each support vector, classification can become computationally expensive. The problem can be further exacerbated by the type of hierarchical multi-resolution image scanning required for online face detection and tracking. SVM optimisation techniques such as the reduced or virtual support vector set methods cannot be easily implemented in the multi-view SVM system because of the importance of each original support vector for determining the pose of new images.

The performance of the SVM tracker was improved, however, by continuously tracking objects, restricting the range of resolution and image regions to be searched according to previous tracking results. The active set of component SVMs used for tracking can also be constrained to the local sub-spaces where objects were previously detected.

Furthermore, the output of the multi-view SVM classifier exhibits positive peaks at regions of faces in the image, as can be seen in Fig. 9. The peaks are at their strongest at the centre of face regions where the best detection occurred. Threshold filtering was found to give best detection locations more quickly than uniformly scanning the image. However, noise exists due to translational misalignment in the training face images at some poses. This can be observed in some of the examples shown in Fig. 9. It was again observed, however, that translational noise occurred mostly along the vertical axis of the image with each scanned line maintaining a perfectly distinguishable peak. This still enabled us to implement a peak detection mechanism for each scan line in order to avoid redundant scanning. By also performing temporal prediction, we achieved a multi-view face tracking and pose estimation at a frame rate greater than 1 Hz on a standard 330 MHz Pentium PC running Linux without utilising any special hardware. An example of this multi-view tracking process is shown in Fig. 9.

7. Conclusion

In this work, we have shown that the complex distribution of face poses can be modelled by a collection of view-based component SVMs. Pose estimation can also be automatically performed by Gaussian kernel functions used in the multi-view SVMs, allowing both tasks to be performed by a single integrated process, thereby, greatly reducing computation. More accurate pose estimation can be achieved by using a better-aligned training set. Future research into using the non-linear mapping learned by the SVM classifier can also provide an improvement in pose estimation accuracy over the simple nearest-neighbour matching we adopted for retrieving pose information from support vectors.

On the matter of real-time performance, multi-view SVM classification is still not computationally attractive for real-time use. However, image-scanning using global optima search methods provides a promising future for

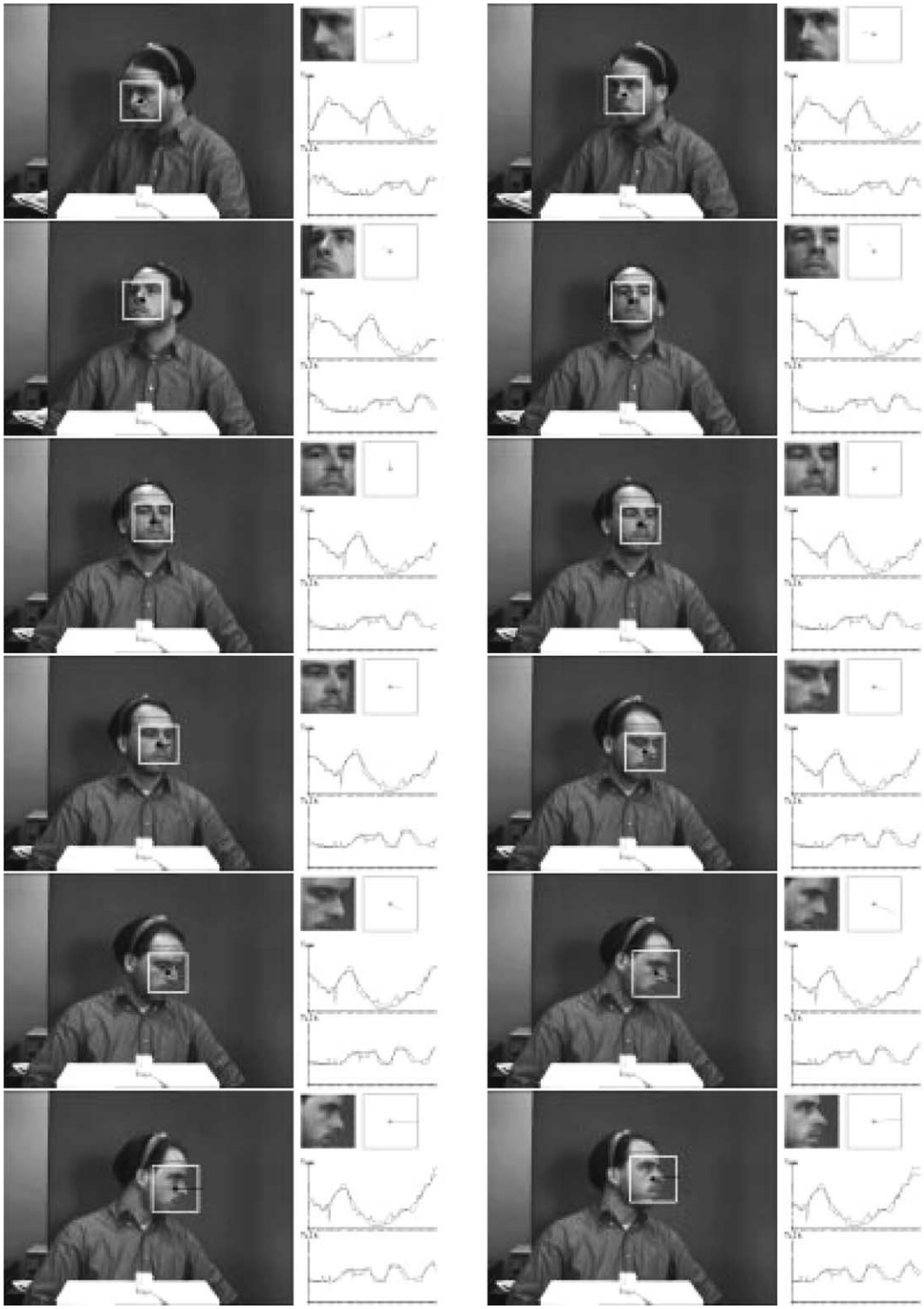

Fig. 10. Examples of near-frame rate face detection, tracking and pose estimation using a multi-view composite SVM. On the right of each picture are detected faces in each image frame, its pose estimated in a dial, and the estimated pose versus the ground-truth from the Polhemus sensor over time.

faster tracking. Combined with motion prediction techniques, an example of which is CONDENSATION, multi-view SVM tracking and pose estimation have been shown to possess great potential for real-time systems.

References

[1] C. Colombo, A. Del Bimbo, Gaze tracing by eye pupil remapping using computer vision, International Workshop on Visual Form, Singapore, 1997, pp. 89–98.
[2] T.F. Cootes, K. Walker, C.J. Taylor, View-based active appearance

models, IEEE International Conference on Automatic Face and Gesture Recognition, Los Alamitos, CA, USA, 2000, pp. 227–232.
[3] S. Gong, S. McKenna, J. Collins, An investigation into face pose distributions, IEEE International Conference on Automatic Face and Gesture Recognition, Vermont, 1996, pp. 265–270.
[4] S. Gong, E. Ong, S. McKenna, Learning to associate faces across views in vector space of similarities to prototypes, British Machine Vision Conference, UK, vol. 1, 1998, pp. 54–64.
[5] A. Lanitis, P.D. Sozou, C.J. Taylor, T.E. Cootes, E.C. Di Mauro, A general non-linear method for modelling shape and locating image objects, International Conference on Pattern Recognition, 1996, pp. 266–270.
[6] A. Mohan, C. Papageorgiou, T. Poggio, Example-based object detection in images by components, IEEE PAMI 23 (4) (2001) 349–361.
[7] E. Osuna, R. Freund, F. Girosi, Training support vector machines: an application to face detection, IEEE Conference on Computer Vision and Pattern Recognition, Los Alamitos, CA, USA, 1997, pp. 130–136.
[8] C.P. Papageorgiou, M. Oren, T. Poggio, A general framework for object detection, IEEE International Conference on Computer Vision, New Delhi, India, 1998, pp. 555–562.
[9] M. Pittore, C. Basso, A. Verri, Representing and recognizing visual dynamic events with support vector machines, Conference on Image Analysis and Processing, 1999, pp. 18–23.
[10] J. Platt, Fast Training of Support Vector Machines using Sequential Minimal Optimisation, Microsoft Research Technical Report MSR-TR-98-14, 1998.
[11] B. Schölkopf, C. Burges, A. Smöla, Advances in Kernel Methods—Support Vector Learning, MIT Press, Cambridge, MA, 1998.
[12] B. Schölkopf, C. Burges, V. Vapnik, Incorporating invariances in support vector learning machines, International Conference on Artificial Neural Networks, 1996, pp. 47–52.
[13] J. Sherrah, S. Gong, Exploiting context in gesture recognition, Second International Interdisciplinary Conference on Modelling and Using Context, Trento, Italy, September 1999, pp. 515–518.
[14] J. Sherrah, S. Gong, Fusion of perceptual cues using covariance estimation, British Machine Vision Conference, Nottingham, England, September 1999, pp. 564–573.
[15] R. Sukthankar, R. Stockton, Argus: the digital doorman, IEEE Intelligent Systems 16 (2) (2001) 14–19.
[16] V. Vapnik, The Nature of Statistical Learning Theory, Springer, New York, 1995.
[17] C. Wang, M. Brandstein, Head pose estimation for video-conferencing with multiple cameras and microphones, International Conference on Advances in Multimodal Interfaces, Berlin, Germany, 2000, pp. 111–118.