Learning Support Vector Machines for A Multi-View Face Model

Jeffrey Ng Sing Kwong and Shaogang Gong

Department of Computer Science, Queen Mary and Westfield College,
London E1 4NS, UK
{jeffng,sgg}@dcs.qmw.ac.uk

Abstract

Support Vector Machines 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 nonlinear 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.

1 Introduction

Support Vector Machines (SVMs) have recently been shown to be effective learning mechanisms for object recognition. By defining hyperplanes in a high-dimensional feature space, SVMs build complex decision boundaries to learn the distribution of a given data set. Their capabilities to learn a function approximation have been successfully applied in the field of handwritten digit recognition [5] and face detection [2]. The handwriting recognition task involved constrained two-dimensional variations in the input data for each recognition class. Osuna's face detection experiment limited the operational parameters of the SVM classifier to almost full-frontal views of human faces, with a small degree of tolerance to variations in the pose of detected faces.

The 3D pose of a face greatly influences the 2D images captured by a camera. Three-dimensional head rotations perpendicular to the camera view plane introduce complex deformations into the appearance of the face. Changes in the lateral and vertical orientation, i.e. yaw and tilt, of a person's head reveal more details of the 3D structure of the head, as other details are occluded. The rotation of the reflective planes of a face can also cause large fluctuations in the local lighting conditions of captured images. Such transformations are highly nonlinear but the distribution of faces across poses have been shown to form smooth trajectories in low dimensional pose eigenspace [1]. In this paper, we investigate both the problem of performing multi-view face detection and the task of using Support Vector Machines to learn a model of the face pose distribution. In addition, we extend SVMs to perform pose estimation by enriching support vectors with extra pose information.

BMVC 1999 doi:10.5244/C.13.50

2 Support Vector Machines

SVMs are based on a generic learning framework that have exhibited useful potentials in resolving some computer vision problems [6, 5, 2, 3, 4]. Let us first outline the basic concept of this approach to learning classification functions for object recognition.

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 the 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 [6] 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 $\{\mathbf{x}_i, y_i\},\ i = 1, ..., l,\ \mathbf{x} \in R^N, y \in \{+1, -1\}$, a hyperplane such as

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

can be oriented across the input space to perform a binary classification task, minimising 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 **w**, also known as the weight vector. There is usually a margin on either side of the hyperplane between 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:

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

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

The optimal hyperplane is mainly defined by the weight vector **w** which consists of all the data elements with non-zero Lagrange multipliers (α_i) in Functional (2), those 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 nonlinear cases. As both the objective function and the decision function is expressed in terms of dot products of data vectors $\mathbf{x}$, the potentially computational intensive mapping $\phi(.)$ 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 $(\mathbf{x} \cdot \mathbf{z})$ [6].

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 nonlinear objective function is

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

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

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 the view sphere involves the recognition of a whole spectrum of very different face appearances. The pose of the head reveals some details about the 3-dimensional structure of the face while masking others. Head rotations introduce nonlinear deformations in captured face images while the rotation can occur in two axes outside the view plane of the camera. The face's main direction of light reflection also changes and affects the illumination conditions of the captured image. Ambient day-time 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 support vector machines 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 Figure 1.

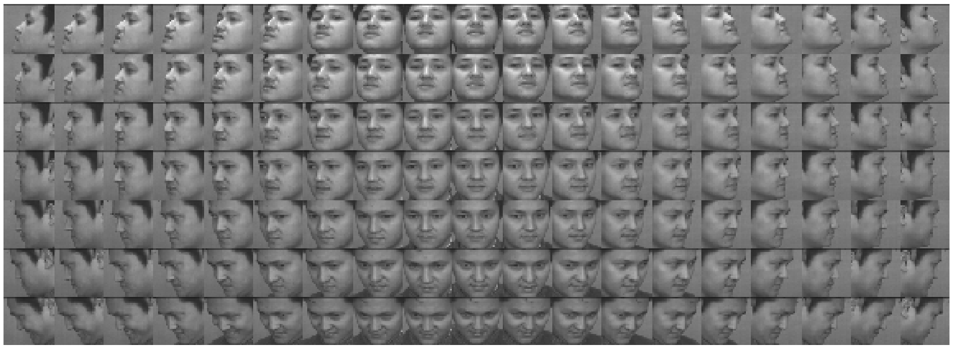

Figure 1: A sample view sphere image-array with calibrated elements varying horizontally from 0° to 180° yaw and vertically from 60° to 120° tilt.

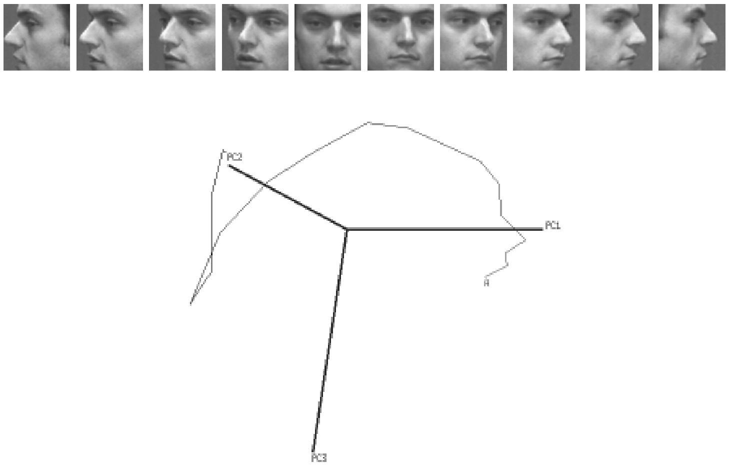

Figure 2: Face rotation in depth forms a smooth trajectory in a 3D pose eigenspace.

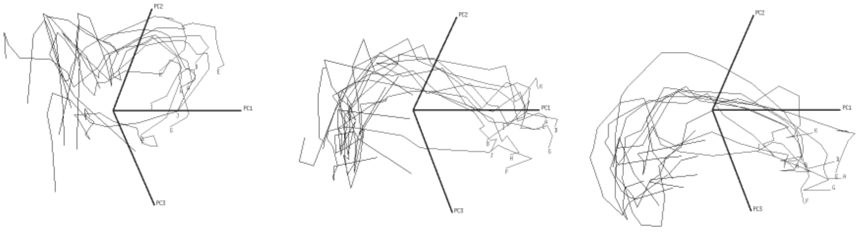

Figure 3: From left to right: The graphs show the PES trajectories for a set of 10 people rotating their heads from profile to profile, at 60° tilt, 90° tilt and 120° tilt respectively.

A face rotating across views forms a smooth trajectory as can be seen in Figure 2. In fact, faces form continuous manifolds across the view sphere in a Pose Eigen-Space (PES). It is plausible to suggest that head rotations describe a continuous function in PES.

This can be seen more clearly in Figure 3. It can also be observed in Figure 3 that an emerging pattern exists for the vertical positioning (from the selected view angle) of the groups of trajectories. Considering that the two images on either sides are made up of the extreme tilt angles of the view sphere, the middle image indeed corresponds to the middle tilt band. The volume enclosed by the entire view sphere is more visible when the nodes of the sphere are plotted individually as in Figure 4. The distribution appears to be a convex hull.

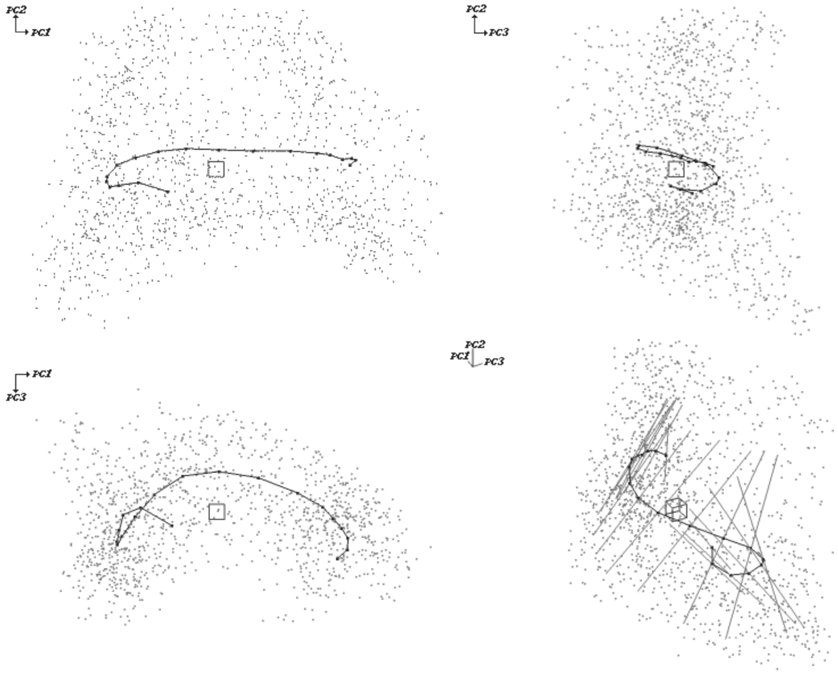

Figure 4: Counter-clockwise from the upper right image: Side, front and top views of the distribution of the face sphere, with the trajectory of the mean yaw clusters. The lower right image uses a special angle to show the direction of biggest variance of the yaw clusters (by the tangential lines) across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

Given better correlation of the lateral bands of the face sphere, the whole distribution can be grouped into 19 different clusters according to their yaw orientation (0° to 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 Figure 4. The above observations strongly 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

Support Vector Machines perform automatic feature extraction and enable the construction of complex nonlinear decision boundaries for learning the distribution of a given data set. 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 Gaussian Support Vectors. Effective values for the two parameters have already been reported for frontal view face detection [2].

We adopt a semi-iterative approach for obtaining good examples of negative training data. 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 [2]. Given the highly complex distribution of the view sphere described in the previous section, it is crucial to find good examples of these to allow the training algorithm to construct accurate decision boundaries. It must be stressed that training is performed on masked vectors consisting of normalised pixel intensity values of face and non-face images of some 300 dimensions. PCA was only used for investigating the nature of the face-pose distribution.

We extended a training process for frontal-view SVM face detection to use the images of the view sphere. The process uses an iterative refinement methodology to find important negative pattern examples in a database of big scenery pictures. This process is illustrated in Figure 5. Although the resulting SVM did not show any potential for robustly detecting faces across views, its training process yielded a good database of negative examples for training such a system. This shows that a single Support Vector Machine cannot learn a unique model of the human face across all views. A multi-view face model must be broken down into component models which form better localised clusters in the distribution and therefore is easier for each SVM to learn a view-based subspace.

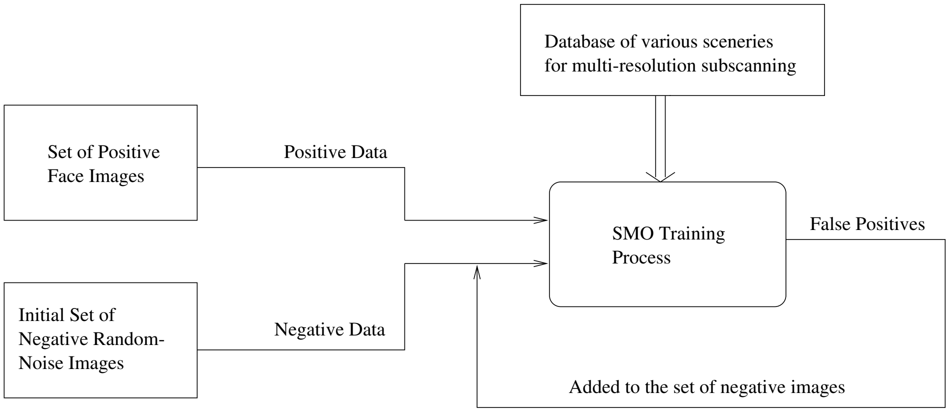

Figure 5: Boot-strapping technique for obtaining negative support vectors.

Based on the nature of the face distribution in PES (Figure 4), the view sphere is 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.

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

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

All the component SVMs were trained on the same global negative data set. The size of the negative training data is about 6,000 images and of those, the SVMs selected 1,666 as negative support vectors, with only 36 shared between two or more component SVMs. This shows that the negative support vectors are well localised to the sub-space 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 pose eigenspace. 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 composite SVM classifier as follows:

$$\text{Composite SVM}(\mathbf{x}) = \text{sign}\left(\sum_{i=1}^{5} SVM(i, \mathbf{x}) + 1\right) \tag{9}$$

where $SVM(i, \mathbf{x})$ is the decision function $f(\mathbf{x})$ for SVM number i.

The multi-view face model can also be applied to pose estimation across the view sphere. Figure 4 shows the correspondence of the yaw angles to the data elements' position 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 Figure 3.

Support vectors in fact define the boundaries of respective classes and should therefore lie on the "walls" of the "tube". Knowing the correspondence between their position in input space to their pose orientation, nearest neighbour matching should enable estimation of the pose for each classified image. The pose estimation is retrieved at no extra computational cost to the calculation of the decision function and is illustrated in Figure 6.

5 Experiments

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.

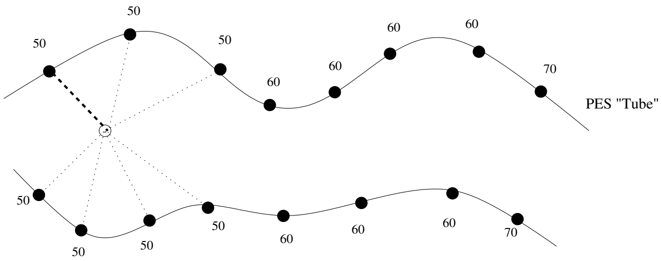

Figure 6: Top view of the face manifold across the pose eigenspace with pan information shown next to each support vector (dark circles). The pose orientation classification image (white circle) is retrieved from that of the closest support vector.

Training subsets	1	2	3	4	5
Full detection	100%	97.7%	94.7%	92.5%	82.7%
Multi-scaling	100%	100%	100%	97.0%	85.7%

Training subsets	6	7	8	9	10
Full detection	88.7%	94.7%	100%	99.2%	97.0%
Multi-scaling	99.2%	97.7%	100%	100%	98.4%

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

The quality of alignment of the input images played an important role in the learning process. Most of the misclassified elements of the view sphere were correctly recognised after multi-scaling the images. Multi-scaling is performed on the input images with a bias in each of the four directions to correct misalignments of the face images.

It is worth pointing out that our previous work reported that the variation of the view sphere distribution along the second principle component axis was highly related to the level of local lighting in the image [1]. Using an overhead light source yields 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 makes down-facing poses very poorly illuminated and therefore, very difficult to detect by the system as shown in Figure 7.

The Multi-View SVM face detector and pose estimator was tested over a number of test sequences of human subjects freely turning their heads around, with the ground-truths of the pose information measured for comparison. For test sequences A-D, the system has been coupled to the iso-tracker to test its classification and pose-estimation accuracy. In sequence E, the system is used to detect, track and estimate the pose of the human subject without the iso-tracker. Experiments on three subjects are given here for illustration: the subject with the worst detection results during training (test sequences A, B and E) and two novel subjects unknown to the training process (test sequences C and D). The latter were selected to test the generalisation capabilities of the system.

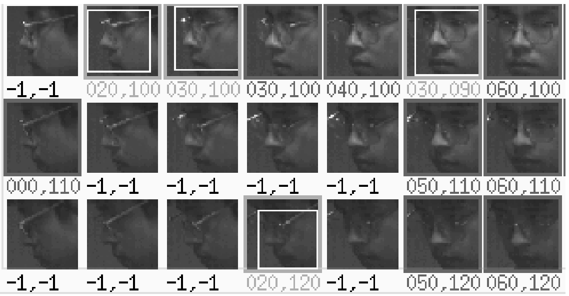

Figure 7: Misclassification in lower hemisphere of the view sphere (shown by -1,-1). Image multi-scaling is shown with white rectangles.

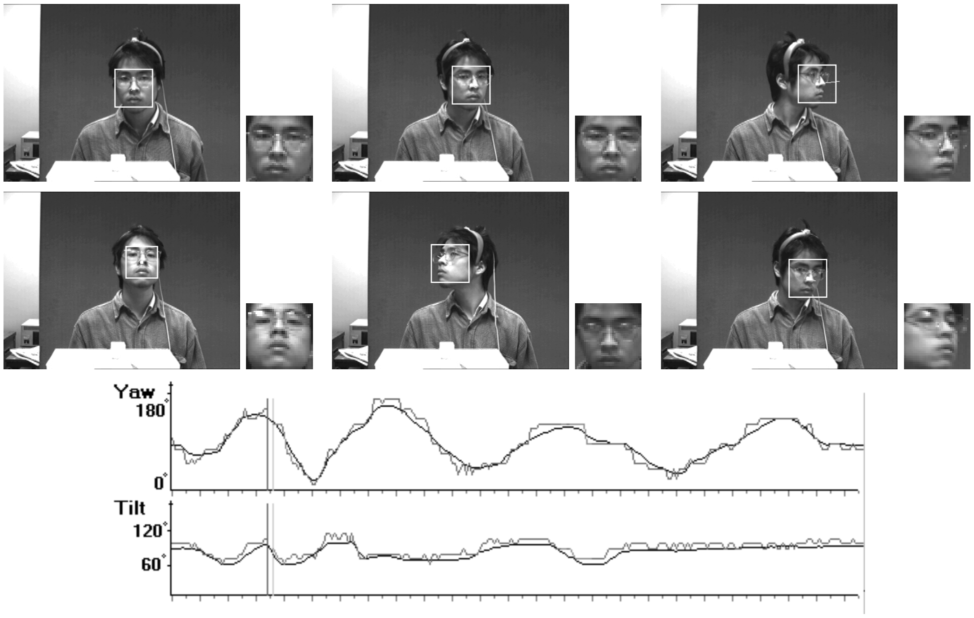

Figure 8: Selected frames from an example sequence (E) of detected and tracked moving faces. The graphs also show the estimated face pose (in grey) over time and their corresponding ground-truths (in black), measured by electro-magnetic sensors. The vertical lines indicate moments in time where no face was detected.

6 Discussion

In this work, we have shown that a well structured distribution of a face training image data set allows a collection of view-based component Support Vector Machines to be locally trained on segments of that distribution. The outputs of the component Support Vector Machines can then be integrated into a composite SVM function, which effectively gives a generic face model across the entire view sphere. The model enables multi-view face detection across the view sphere in our case without any gap in the detection of faces at the “seams” of the segments. The technique has also been extended to use the inherent

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°

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.

structure of the data to perform pose estimation at no extra computational cost to the detection process. In particular, support vectors have been tagged with pose information to allow the retrieval of pose orientation by nearest neighbour matching to the support vectors. The results show that the support vectors obtained from the view sphere make good prototypes for pose estimation by nearest neighbour matching.

The accuracy of the face alignment and orientation calibration of some of the training images were not perfect. A better training set could have better defined the decision surface and allow nearest neighbour matching to yield more accurate results. We believe that pose estimation can still be further refined by using high-dimensional mapping and the weighted decision function of the SVMs to perform nonlinear pose estimation.

References

[1] S. Gong, S. McKenna, and J. Collins. An investigation into face pose distributions. In *IEEE Int. Conf. on Automatic Face and Gesture Recognition*, pages 265–270, Vermont, 1996.

[2] E. Osuna, R. Freund, and F. Girosi. Training support vector machines: an application to face detection. In *CVPR*, 1997.

[3] J. C Platt. *Fast Training of Support Vector Machines using Sequential Minimal Optimisation*. Microsoft Research Technical Report MSR-TR-98-14, 1998.

[4] B. Schölkopf, C. Burges, and A. Smöla. *Advances in Kernel Methods - Support Vector Learning*. MIT Press, 1998.

[5] B. Schölkopf, C. Burges, and V. Vapnik. Incorporating invariances in support vector learning machines. In *International Conference on Artificial Neural Networks*, 1996.

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