Available at
www.ElsevierComputerScience.com
POWERED BY SCIENCE DIRECT®

Image and Vision Computing 22 (2004) 413–427

www.elsevier.com/locate/imavis

Support vector machine based multi-view face detection and recognition

Yongmin Li[a,*], Shaogang Gong[b], Jamie Sherrah[c], Heather Liddell[b]

[a]*Department of Information Systems and Computing, Brunel University, Uxbridge, Middlesex UB8 3PH, UK*
[b]*Department of Computer Science, Queen Mary, University of London, London E1 4NS, UK*
[c]*Safehouse Technology Pty Ltd, 2a/68 Oxford Street, Collingwood, Victoria 3066, Australia*

Received 19 October 2003; received in revised form 15 December 2003; accepted 18 December 2003

Abstract

Detecting faces across multiple views is more challenging than in a fixed view, e.g. frontal view, owing to the significant non-linear variation caused by rotation in depth, self-occlusion and self-shadowing. To address this problem, a novel approach is presented in this paper. The view sphere is separated into several small segments. On each segment, a face detector is constructed. We explicitly estimate the pose of an image regardless of whether or not it is a face. A pose estimator is constructed using Support Vector Regression. The pose information is used to choose the appropriate face detector to determine if it is a face. With this pose-estimation based method, considerable computational efficiency is achieved. Meanwhile, the detection accuracy is also improved since each detector is constructed on a small range of views. We developed a novel algorithm for face detection by combining the Eigenface and SVM methods which performs almost as fast as the Eigenface method but with a significant improved speed. Detailed experimental results are presented in this paper including tuning the parameters of the pose estimators and face detectors, performance evaluation, and applications to video based face detection and frontal-view face recognition.

Keywords: Face recognition; Multi-view face detection; Head pose estimation; Support vector machines

1. Introduction

Over the past decade, face recognition has emerged as an active research area in computer vision with numerous potential applications including biometrics, surveillance, human–computer interaction, video-mediated communication, and content-based access of images and video databases.

1.1. Background

Statistical methods have been widely adopted in face detection. Moghaddam and Pentland [27–29,34] introduced the Eigenface method, where the probability of face patterns is modelled by the 'distance-in-feature-space' (DIFS) and 'distance-from-feature-space' (DFFS) criteria. Osuna et al. [32,33] presented an SVM-based approach to frontal-view face detection. Unlike the Eigenface method where only the positive density is estimated, this approach seeks to learn the boundary between face and non-face patterns. After learning, only the 'important' examples located on the boundary are selected to build the decision function. Soulie et al. [44] described a system using neural networks (NNs) for face detection. They implemented a multi-modal architecture where various rejection criteria are employed to trade-off false recognition against false rejection. Sung and Poggio [46] also presented a NN based face detection system. They designed six positive prototypes (faces) and six negative prototypes (non-faces) in the hidden layer. Supervised learning is performed to determine the weights of these prototypes to the output node. Rowley et al. [37] introduced a NN based upright frontal face detection system. A retinally connected NN examines small windows of an image and decides whether each window contains a face. This work was later extended to rotation invariant face detection by designing an extra network to estimate the rotation of faces in the image plane [38,39]. Gong et al. [9] used general and modified hyper basis function (HBF) networks with Gaussian mixture models to estimate the density function of face space with large pose variation. As a result, face recognition can be performed more successfully

* Corresponding author.
E-mail address: yongmin.li@brunel.ac.uk (Y. Li).

0262-8856/$ - see front matter © 2004 Elsevier B.V. All rights reserved.
doi:10.1016/j.imavis.2003.12.005

than with either of the linear models. McKenna et al. [23,24] presented an integrated face detection-tracking system where a motion-based tracker is used to reduce the search space and Multi-Layer Perceptron (MLP) based face detection is used to resolve ambiguities in tracking.

Feature-based methods have also been extensively addressed in previous work. For example, Brul and Perona [3] proposed a framework for recognising planar object classes, such as recognising faces from images, based on local feature detectors and a probabilistic model of the spatial configuration of the features. Yow and Cipolla [52] proposed a face detection framework that groups image features into meaningful entities using perceptual organisation, assigns probabilities to each of them, and reinforces the probabilities using Bayesian reasoning. They claimed that the framework can be applied to face detection under scale, orientation and viewpoint variations [51].

Recently, Viola and Jones [13] presented an approach to fast face detection using simple rectangle features which can be efficiently computed from the so-called Integral Image. AdaBoost and cascade methods are then used to train a face detector based on these features. Li et al. [17,53] adopted similar but more general features which can be computed from block differences. Also, FloatBoost is proposed to overcome the monotonicity of the sequential AdaBoost learning.

1.2. Difficulties

Face detection is normally formulated as a classification problem to separate face patterns from non-face patterns. From the statistical point of view, there are mainly three obstacles for this problem:

(1) The dimensionality of patterns is usually high;
(2) The possible number of non-face patterns is extremely large and their distribution is very irregular;
(3) It may also be difficult to model the probability distribution of face patterns, especially the multi-view face patterns, with a unimodal density function.

Most of the previous work is limited to the frontal view. The problem of dealing with rotation in depth and hence being able to detect faces across multiple views remains difficult. Many researchers addressed this problem by building multiple view-based face detectors, i.e. to divide the view sphere into several small segments and to construct one detector on each of the segments [9,17,28,31]. Nevertheless, a new problem is normally introduced in these view-based approaches: since the pose of a face is unknown before detection, which detector should we choose to determine if it is a face? A common solution to the problem is to apply all view-based detectors to an input image (or sub-image) and to make a decision based on the one with maximal response. Undoubtedly, it is computationally inefficient.

1.3. Our approach

In this research, an approach to multi-view face detection based on pose estimation is presented. Similar to Refs. [9,17,28,31], we also decompose the problem into a set of sub-problems, each of them for a small range of views. However, by using the pose information, only one of the view-based detectors is chosen to determine if a pattern is a face. Selective attention by motion/skin colour detection and background subtraction is used to bootstrap Regions of Interenst (ROI) which make the face detectors focus on small sub-images only.

This paper includes and significantly extends the work we have published in Refs. [18,19]. The former presented the approach to multi-view face detection based on pose information, and the latter introduced the combined method of SVM and Eigenface for face detection.

In developing our new approach, we have mainly benefited from the previous work of Moghaddam and Poggio [27–29,34], Rowley et al. [38,39] and Osuna et al. [32,33].

Moghaddam and Poggio [27–29,34] tried to address the problem of high dimensionality using Principal Component Analysis (PCA) to linearly extract the most significant modes of face patterns. They established a statistical density model based on these abstract features (Eigenfaces). However, non-face patterns are not modelled in their approach.

Rowley et al. [38,39] developed a NN based system which is capable of rotation invariant face detection. An extra network is designed to estimate the rotation of a face in the image plane. However, they have not addressed the rotation out of the image plane.

The SVM-based approach presented by Osuna et al. [32,33] seems to be a promising method to solve this problem. Instead of estimating the densities of face and non-face classes, it seeks to model the boundary of the two classes. Moreover, the generalisation performance, or the capacity of the learning machine, is automatically maintained by the principle of Structural Risk Minimisation [49]. However their work is only for frontal-view face detection.

The rest of this paper is arranged as follows: We discuss in Section 2 the overall framework of our approach to multi-view face detection and the methodology of constructing the pose estimators and multi-view face detectors. Implementation issues and experimental results are presented in Section 3, including parameter tuning in constructing the pose estimators and face detectors, performance evaluation, video-based face detection and frontal face recognition. The conclusions of this paper are presented in Section 4.

2. Multi-view face detection based on pose estimation

As discussed before, detecting face across multiple views is more challenge than from a fixed view as the appearance

of faces can be very different from different views. A straightforward method for multi-view face detection is to build a single detector which deals with all views of faces. The second approach is to build several detectors, each of them corresponding to a specific view. In run-time, if one or more of the detectors give positive output for a given pattern, a face is considered as detected. Previous studies showed that the first method led to poor performance as it fails to deal with the irregular variations of faces across multi-views [10,28]. The second approach usually performs better than the first one but the computation is expensive since all the multi-view face detectors need to be computed for a given pattern.

In this work, we present a novel approach to the problem. We build several view-based face detectors as in the second method described above. But when detecting faces, we use the pose information explicitly, i.e. we estimate the 'pose' of a given image pattern first, then use the pose information to choose *only one* of the view-based face detectors to determine whether the targeted image pattern is a face.

The process of multi-view face detection is described as follows (Fig. 1):

(1) Perform motion estimation, skin colour detection or background subtraction on input images or an image sequence to locate ROIs which may contain faces;
(2) Exhaustively scan these image regions at different scales;
(3) For each image patch from the scan, estimate the 'pose' (tilt and yaw) using pre-trained pose estimators;
(4) Choose an appropriate face detector according to the estimated pose to determine if the pattern is a face;
(5) Refine the results of detection.

It seems that extra computation is applied in pose estimation for each image patch. However, the estimated pose can be used to choose the appropriate face detector, so further computation is only applied to one of the detectors, therefore computation is actually saved in face detection. Otherwise, one has to compute all the detector outputs and combine them for a final detection. For example, with our method we only need to construct 4 view-based face detectors as in Fig. 3, and need two computations (one pose estimation and one face detection) to detect a face. If we do not use the pose information, eight computations (one for each detector) have to be performed for the same view space segmentation.

The issue of selective attention (first block in Fig. 1) is beyond the topic of this paper. Interested readers may refer to the following studies: motion [23,24], skin colour [14,25,26,35,36], and background modelling and subtraction [4,45,47].

In the rest of this section, we will mainly discuss two related problems: pose estimation and multiview face detection.

2.1. Estimating head pose

Pose and gaze estimation can be performed intrusively by using active and contacted sensing such as the early work of Hutchinson et al. [12]. Alternatively it can be performed using non-contact and passive methods directly from images. A geometrical approach based on facial features such as eyes, nose and mouth has been reported by Gee and Cipolla [6] and Horprasert et al. [11]. Meanwhile, some researchers proposed to use stereo input for head pose and gaze estimation [21,50]. The feature points chosen for modelling pose can also be those with the most significant pictorial characteristics such as edge, valley and ridge, or those preprocessed by image filtering. For example, Gabor wavelet jets have been adopted for tracking and pose estimation [5,16,22].

For both facial-feature-based and image-feature-based methods, the performance crucially depends on successful feature location. As opposed to the feature-based approach, some researchers tried to solve the problem using holistic facial appearance matching. Previous studies in this category include the Radial Basis Function network estimator by Beymer et al. [1], the NN based system by Rowley et al. which is capable of detecting faces

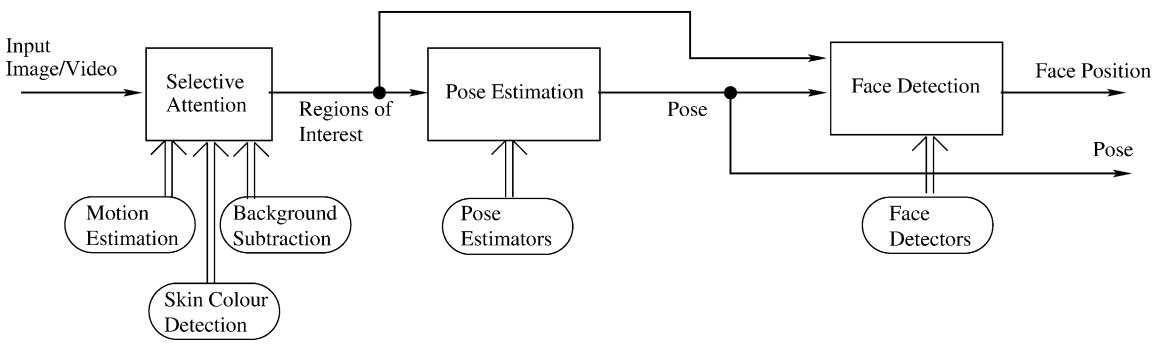


Fig. 1. The framework for multi-view face detection. Motion estimation, skin-colour detection and background subtraction are adopted for selective attention to obtain the ROIs that may contain faces. Pose estimation is performed first for each image patch of the search on the ROIs, regardless of whether it contains a face. The pose information is used to select an appropriate face detector to determine if it contains a face.

with rotation in the image plane [40], and the method based on Gabor wavelet transform and PCA presented by Gong et al. [7].

In this paper, we present an SVM-based approach to pose estimation. Like [1,7,40], the algorithm performs directly from image appearance.

2.1.1. Pre-processing and representation of face images

Instead of working on the high-dimensional raw images, we pre-process them in the following way:

(1) Two Sobel operators (horizontal and vertical) are used to filter the face images. The two filtered images are combined together as the composite patterns (see Fig. 2).
(2) PCA [30,34] is performed on the filtered image patterns in order to reduce the dimensionality of the training examples. Fig. 2 illustrates sample face images from a subject in different views, the filtered composite patterns, the reconstructed patterns from the first 20 PCs, and the first 10 Principal Components (PCs).

Note that we have experimented with applying PCA directly on the original images, i.e. representing face patterns without the first step of Sobel filtering. The results are not as good as those of the method mentioned above. One possible reason is that the filters capture the changes both in horizontal and vertical directions so that the filtered images are more representative for pose changes.

Determining the dimensionality of the PCA subspace, i.e. the number of PCs to represent face patterns, is a tricky problem. Usually it is a trade-off between estimation accuracy and computational efficiency. It is important to point out that, when dealing with data with noise such as imagery noise or misalignment of face patterns, keeping a smaller number of PCs may result in filtering out the noise, while a larger number of PCs does not necessarily lead to performance improvement. This is illustrated by experimental results shown in Fig. 6, and will be discussed in more detail in Section 3.2.2.

2.1.2. Estimating head pose using SVM regression

Two tasks need to be performed for head pose estimation: constructing the pose estimators from face images with known pose information, and applying the estimators to a new face image. We adopt the method of SVM regression to construct two pose estimators, one for tilt (elevation) and the other for yaw (azimuth). The input to the pose estimators is the PCA vectors of face images discussed in Section 2.1.1. The dimensionality of PCA vectors can be reasonably small in our experiments (20, for example). More on this will be discussed in Section 3.2.2. The output is the pose angles in tilt and yaw.

The SVM regression problem can be solved by maximising

$$W(\alpha^*, \alpha) = -\frac{1}{2}\sum_{i,j=1}^{l}(\alpha_i^* - \alpha_i)(\alpha_j^* - \alpha_j)k(\mathbf{x}_i, \mathbf{x}_j) - \varepsilon\sum_{i=1}^{l}(\alpha_i^* + \alpha_i) + \sum_{i=1}^{l} y_i(\alpha_i^* - \alpha_i) \quad (1)$$

$$\text{st} \sum_{i=1}^{l}(\alpha_i^* - \alpha_i) = 0 \quad (2)$$

$$0 \le \alpha_i^*, \ \alpha_i \le C \quad (3)$$

Fig. 2. Representation of face patterns "(a) From top to bottom are the original face images, the filtered patterns with horizontal and vertical Sobel operators, and recontstructed patterns from the first 20 PCs". "(b) The first 10 significant PCs".

which provides the solution

$$f(\mathbf{x}) = \sum_{i=1}^{l} (\alpha_i^* - \alpha_i) k(\mathbf{x}, \mathbf{x}_i) + b \quad (4)$$

where $\mathbf{x}$ is the PCA feature vector of a face image, k is the kernel function used in the SVM pose estimator, y_i is the ground-truth pose angle in yaw or tilt of pattern $\mathbf{x}$, C is the upper bound of the Lagrange multipliers α_i and α_i^*, and ε is the tolerance coefficient. More details about SVM regression can be found in Ref. [49].

Two pose estimators in the form of Eq. (4), f_t for tilt and f_y for yaw, are constructed. The Quadratic Programming problem is solved by a decomposition algorithm based on the LOQO algorithm [48]. The decomposition algorithm can be briefly described as follows: At each iteration, only a small set of training patterns are processed by the LOQO algorithm. The support vectors (SVs) and patterns with the largest error from the previous iteration have higher priority for selection. The algorithm is stopped when no significant improvement is achieved.

Compared with other learning methods, the SVM-based method has distinguishing properties such as:

(1) No model structure design is needed. The final decision function can be expressed by a set of 'important examples' (SVs).
(2) By introducing a kernel function, the decision function is implicitly defined by a linear combination of training examples in a high-dimensional feature space.
(3) The problem can be solved as a Quadratic Programming problem, which is guaranteed to converge to the global optimum of the given training set.

2.2. Multi-view face detection

Like other previous work, we define face detection as a classification problem, i.e. to build one or more classifiers to separate faces from non-faces. As described at the beginning of this section, we construct several view-based face detectors (classifiers), each responsible for a segment of the view space. When detecting faces from images or image sequences, we estimate the pose information first, and then use the pose information to choose an appropriate classifier to perform detection.

2.2.1. View space segmentation

The view space is divided into eight segments: left profile, left frontal, right frontal, right profile in the horizontal direction (yaw), and upper and lower in the vertical direction (tilt), as shown in Fig. 3. When constructing the view based piece-wise face detectors, we adopt the following strategies.

(1) Faces are symmetrical along the vertical line across the nose-bone, so the faces on the right half of the view plane can be converted to the left half without losing the general facial characteristics. Based on this, one only needs to model the multi-view faces either in the left or the right view. As illustrated in Fig. 3, only four detectors are constructed.
(2) The view space is divided at 0° in tilt to separate inclined faces from declined faces, and at 0° and ± 50° in yaw to separate left-view faces from right-view faces and one-eye faces (profile) from two-eye faces (frontal) effectively.
(3) Neighbouring segments overlap with each other by 10°. This is to make sure each view-based detector is

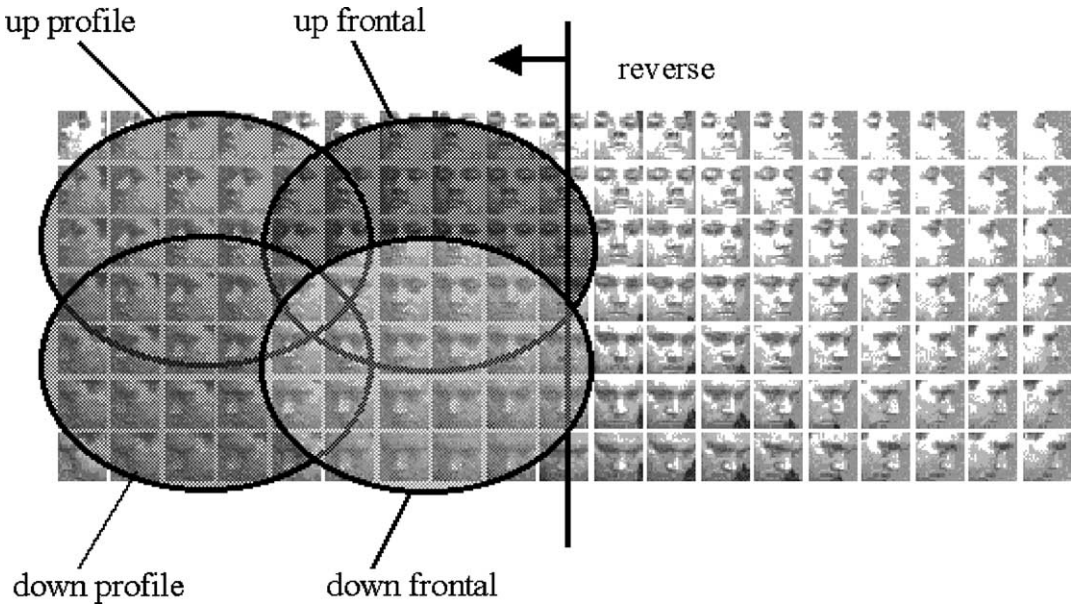


Fig. 3. Modelling multi-view faces. Only four detectors need to be constructed based on the symmetry property of human face: up profile, up frontal, down profile, down frontal. When detecting faces, only one of the detectors is chosen if pose information is available.

responsible for a larger view space. This helps to improve the robustness of face detectors.

2.2.2. Algorithms of face detection

The image patterns are pre-processed in the same way as in Section 2.1.1, i.e. filtered with Sobel operators and projected with PCA. We implemented three algorithms for multi-view face detection, the Eigenface algorithm [28,29], the SVM-based algorithm [32,33], and a novel algorithm—a hybrid method of Eigenface and SVM.

Moghaddam and Pentland [28,29] introduced the Eigenface method, where the probability $P(\mathbf{x})$ of a pattern $\mathbf{x}$ being a face is modelled by the DIFS and DFFS criteria.

$$P(\mathbf{x}) = \left[\frac{\exp\left(-\frac{1}{2}\sum_{i=1}^{M}\frac{u_i^2}{\lambda_i}\right)}{(2\pi)^{M/2}\prod_{i=1}^{M}\lambda_i^{1/2}}\right]\left[\frac{\exp\left(-\frac{\varepsilon^2(\mathbf{x})}{2\rho}\right)}{(2\pi\rho)^{(N-M)/2}}\right] \tag{5}$$

where λ_i is the ith eigenvalue, u_i is the projection onto the ith eigenvector, N is the total number of eigenvectors, M is the number of significant eigenvectors selected in the model, and ρ is an approximation factor.

Usually the histograms of $P(\mathbf{x})$ from face patterns (positive) and non-face (negative) look like the two curves in Fig. 4. One can choose a threshold based on these curves to separate faces from non-faces.

Alternatively, an SVM-based face detector [32,33] can be constructed by maximising

$$W(\alpha) = \sum_{i=1}^{l}\alpha_i - \frac{1}{2}\sum_{i,j=1}^{l}\alpha_i\alpha_j y_i y_j k(\mathbf{x}_i, \mathbf{x}_j) \tag{6}$$

$$\text{st} \sum_{i=1}^{l}\alpha_i y_i = 0 \tag{7}$$

$$0 \le \alpha_i \le C, \quad i = 1, 2, \ldots, l \tag{8}$$

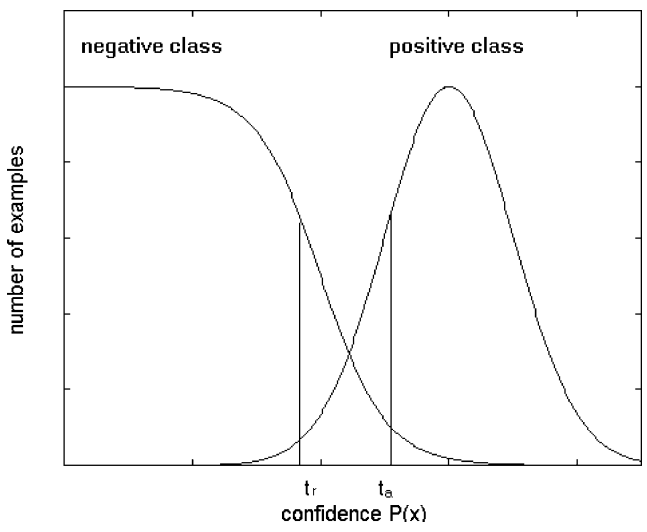


Fig. 4. The hybrid method of Eigenface and SVM.

where y_i is the label of a training example $\mathbf{x}_i$ which takes value 1 for face and -1 for non-face, k is a kernel function, and C is the upper bound of the Lagrange multiplier α_i. For a new pattern $\mathbf{x}$, the trained face detector gives an output

$$f(x) = \sum_{i=1}^{l} y_i\alpha_i k(\mathbf{x}, \mathbf{x}_i) + b \tag{9}$$

where b is the bias.

The Eigenface method actually models the probability of face patterns, while the importance of non-face patterns in this method is not significant except when choosing the threshold. On the other hand, an SVM-based face detector makes use of both face and non-face patterns: it estimates the boundary of positive and negative patterns instead of estimating the probabilities. Generally speaking, the Eigenface method is computationally efficient but less accurate, while the SVM method is more accurate but slow.

In order to achieve improved overall performance in terms of both speed and accuracy, a novel approach which combines the Eigenface and the SVM methods is presented. A schematic illustration of the classification criterion of the hybrid method is given in Fig. 4.

The whole process consists of a coarse detection phase by the Eigenface method followed by a fine SVM phase. In the first phase, the probability density of each class is estimated as simply as possible. Two thresholds, a rejection threshold (t_r) and an acceptance threshold (t_a), are defined. For a test sample $\mathbf{x}$, if the value of $P(\mathbf{x})$ given by Eq. (5) is less than t_r, it is rejected as a negative example. If the value is larger than t_a, it is accepted as positive. Otherwise, if the value falls between t_r and t_a, it is considered as ambiguous and left to the SVM classifier in the next phase. The values of the two thresholds should be determined by the acceptable false positive and false negative rates which are usually application dependent.

An SVM-based classifier is trained using the examples in the middle region of Fig. 4. The classifier is only activated when an ambiguous pattern emerges. Usually the SVM-based classifier is computationally more expensive than the Eigenface method, but more accurate. However, since the proportion of the examples in the ambiguous region is relatively small, a significant improvement of the classification speed can be achieved.

Furthermore, owing to the fact that the SVM classifier is trained only on the examples in the ambiguous region and not on the whole training set, the SVM classification problem is simplified to some degree. A more precise and compact set of SVs are obtained.

3. Experiments and discussions

This section is organised as follows: We first describe in Section 3.1 the database used in the experiments. Results on constructing the pose estimators and face detectors are

presented in Sections 3.2 and 3.3, respectively, together with parameter tuning, performance evaluation and implementation issues. Implementation and results on video based face detection are presented in Section 3.4. Based on the detection results, we demonstrate the results of frontal-view face recognition in Section 3.5.

3.1. Database descriptions

In our previous work, a system was designed to capture the multi-view face images and measure facial pose and positions of key facial features such as eyes and mouth. The system utilises a magnetic sensor rigidly attached to a subject's head and a camera calibrated to the sensor's transmitter. The sensor provides the 3D coordinates and orientation relative to its transmitter. In the initialisation stage, the positions of mouth and eyes are manually located on the screen. These positions are usually adjusted at different views to make sure they are rigidly 'attached' to the facial features in the images. More details about the multi-view face acquisition system are described in Ref. [8]. The system provides the positions of the key feature points and pose information as well as the multi-view face images.

The faces captured in the images are about 50×50 pixels. The range of pose of these face images is $[-90, +90°]$ in yaw and $[-30, +30°]$ in tilt. We have collected a set of multi-view face images from 31 subjects. We used part of them in the experiments presented here.

3.2. Pose estimation

When constructing the pose estimators described in Section 2.1, one needs to choose an appropriate kernel function, and determine the parameters of the kernel. We have tried different kernel functions such as linear kernel, Gaussian kernel and polynomial kennels with different orders. Experimental results indicate that they provide similar performance for this problem. However, the best results are achieved when a Gaussian kernel is chosen, Also, the performance is not very sensitive to small changes of the SVM parameters. In our experiments, it is found that the Gaussian kernel usually performs well when its parameter is set as $2\sigma^2 = 1$ and the patterns are normalised to unit vectors. The parameters of the algorithm used for the experiments below are listed in Table 1.

Table 1
Parameters of the SVM based algorithm for pose estimation

Kernel	Gaussian
$2\sigma^2$	1
C	1000
Image dimension	400 (20×20)
Range of tilt	$[-30, +30°]$
Range of yaw	$[-90, +90°]$
Total number of training images	1596 (12 subjects, 133 images of each)
Total number of validation images	1283 (from four sequences)

3.2.1. Tolerance coefficient ε

The tolerance coefficient ε is used to define the ε-insensitive loss function [49] in SVM regression.

$$|f(\mathbf{x}) - y| = \begin{cases} 0, & \text{if } |f(\mathbf{x}) - y| \le \varepsilon \\ |f(\mathbf{x}) - y|, & \text{otherwise} \end{cases} \tag{10}$$

where f is the regressed function, and y is the known label of pattern $\mathbf{x}$. By introducing the loss function defined in Eq. (10), the SVM can provide a sparse solution to a regression problem, i.e. the number of SVs can be far less than the number of the training examples.

Normally, ε can be used to control the accuracy of a SVM regressor. A large value of ε may lead to a regression function with poor accuracy and good real-time performance since a larger error is acceptable by the loss function (10) and a smaller number of SVs can be obtained from training. However, it is important to point out that one cannot expect to achieve a perfect result by setting ε to 0 or near 0. The maximal accuracy of a regression problem is determined by its VC-dimension [49]. Too small a value of ε may lead to over-fitting, i.e. the results are perfect on the training set but deteriorate on the validation set. Scholkopf et al. [41–43] have discussed this problem extensively. They also presented a method for automatic accuracy control in SVM regression.

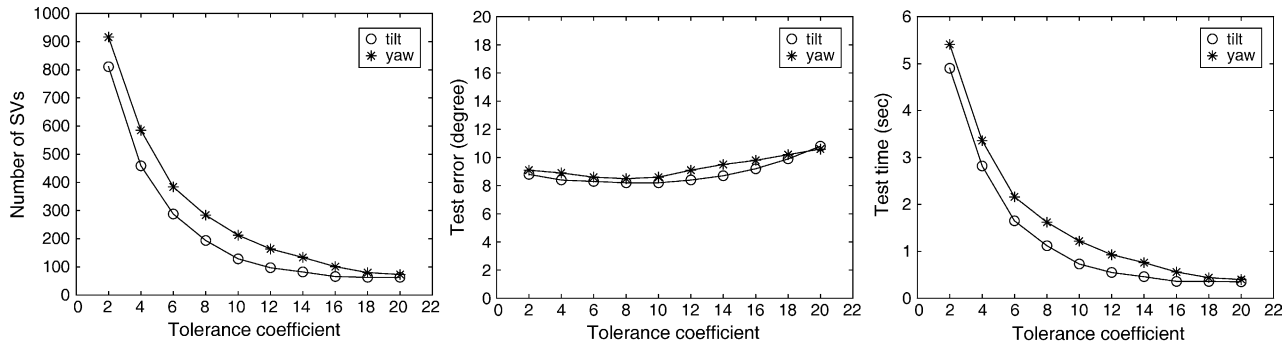
To investigate the influence of ε on the performance of a pose estimator, we designed the following experiments where the value of ε changes from 2 to 20. The PCA dimension is fixed to 20, and other parameters are chosen as listed in Table 1. Fig. 5 shows the results of SV numbers, errors in tilt and yaw, and test time.

The experimental results indicate:

(1) The number of SVs increases steeply with the decrease of ε. The number when $\varepsilon = 2$ is over eight times as large as that when $\varepsilon = 20$.
(2) Lowering the value of ε does not always improve the accuracies. Actually, the optimal accuracies are obtained when ε is chosen around 10, which may reflect the intrinsic precision of the training examples.
(3) Better real-time performance is achieved when increasing ε so that fewer SVs are obtained. This is because the estimation speed is determined by the number of SVs.

3.2.2. PCA dimension

We designed the following experiment to evaluate the performance of the SVM based pose estimators with different PCA dimensions. ε is fixed to 10, and other parameters are chosen as listed in Table 1 in this experiment. The performance is evaluated in terms of the number of



(a) SV number (b) Error on validation images (c) Running on validation images

Fig. 5. Pose estimation performance vs. tolerance coefficient ε. Results of (b,c) are computed on validation images.

SVs, the estimation error on the validation set, and validation time. The results are shown in Fig. 6.

From the experimental results, we have the following observations:

(1) Except for the very low dimensional case, the number of SVs remains constant with the increase of the PCA dimension. This reflects the underlying characteristics of SVMs since the number of SVs corresponds to the VC-dimension of the problem. A very high PCA dimension does not provide further improvement to the performance. When the PCA dimension is below 15, the number of SVs is considerably higher since the representation with very low dimension is incapable of capturing sufficient information for pose estimation.
(2) Estimation errors are approximately stable except for the case of very low dimensions. This indicates that a relatively low PCA dimension can provide sufficient accuracy. The stable error rates also indicate that the SVM based pose estimators correctly reflect the intrinsic precision of the training examples and they are not over-fitted even when the PCA dimension is high.
(3) The estimation speed is related to the number of SVs and the PCA dimension. When the PCA dimension is below 15, a poor real-time performance is observed owing to the large number of SVs. Above that, the test time increases nearly linearly with the increase of dimension.

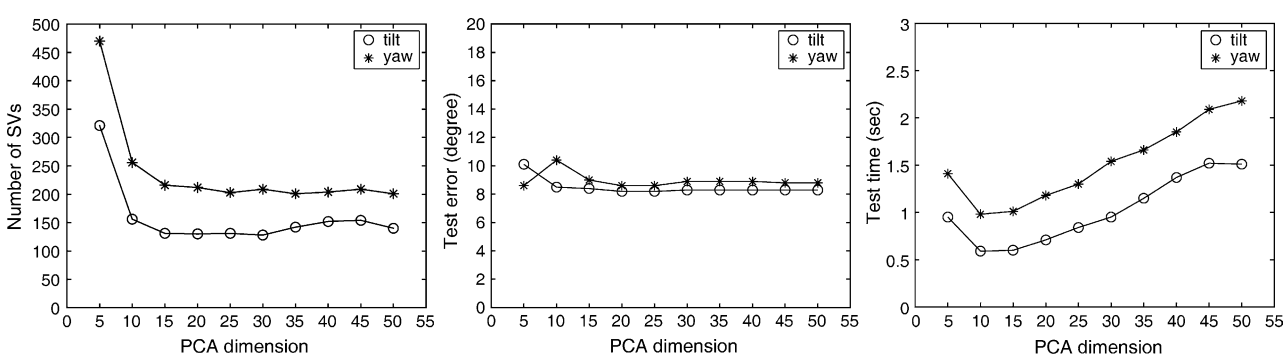
These experiments result indicate that a relatively low dimensional representation, for example, 20 in PCA dimension, can provide satisfactory performance in terms of accuracy and run-time speed in pose estimation.

3.2.3. Pose estimation results

Fig. 7 shows estimated pose from a test sequence. The parameters used for SVM training are listed in Table 1. The dimension of the PCA vector of face patterns is chosen as 20. Over the whole sequence, the estimation errors in both yaw and tilt are around 10°, which are sufficiently accurate for the purpose of multi-view face detection.

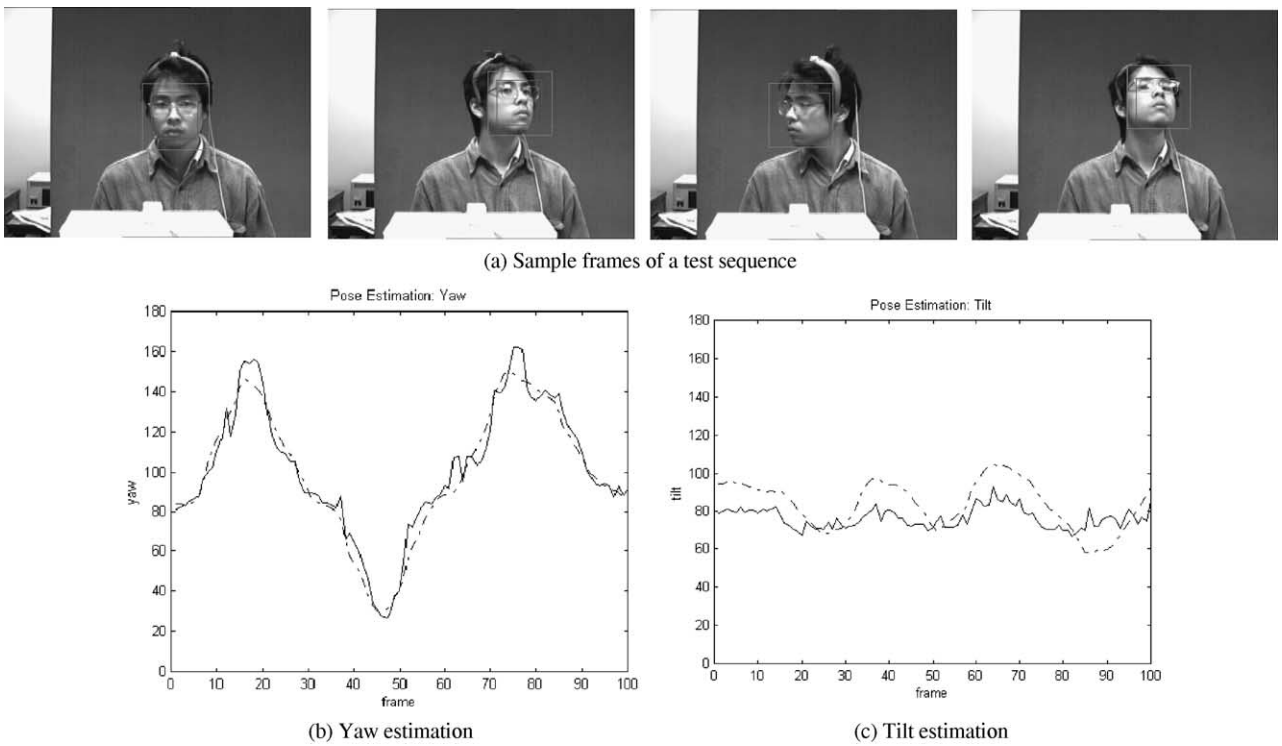
3.3. Multi-view face detection

When training each multi-view face detector, the face images corresponding to the specific view range are selected as positive examples (faces). In the experiments, 2660 face images of 20 subjects were selected as positive examples (faces) from the same database, with pose changing from -90 to $+90°$ in yaw and from -30 to $+30°$ in tilt. As described in Section 2.2.1, the face images with yaw angle in the range of $[0, +90°]$ were reversed to $[0, -90°]$ along the central vertical line of the images. This produced 140



(a) SV number (b) Test error (c) Test time

Fig. 6. Pose estimation performance vs. PCA dimension



(a) Sample frames of a test sequence

(b) Yaw estimation

(c) Tilt estimation

Fig. 7. Pose estimation on a test sequence. In (b) and (c), the solid curves are the estimated pose in yaw and tilt and the dotted curves are the ground-truth pose which is measured by the data acquisition system.

face images for each subject, and 2800 face images in total. These face images were then divided into four segments: up-profile, up-frontal, down-profile and down-frontal. The separating pose angles are 0° in tilt and $-50°$ in yaw which are included into both neighbouring segments.

Since the number of non-face patterns is very large, it is impossible to collect all non-face patterns before training the face detectors. A boot-strapping method [46] is adopted for non-face pattern selection and iterative training. In our experiments, negative examples were collected with the SVM face detector. The Eigenface and hybrid detectors were then constructed using the same set of training examples.

An arbitrary collection of non-face patterns, which can be cropped randomly from scenery pictures which do not contain any faces, were chosen as the first set of negative examples for training. Then we applied the resulting detector to the scenery pictures. If positive outputs (false positives) are reported, save these detections for further training. This process was repeated iteratively until satisfactory results were achieved. All example images were scaled to 20×20 pixels when training the face detectors. The images were then normalised to unit vectors and projected into the 20-dimensional PCA space as described in Section 2.1.1.

The parameters of the face detectors are listed in Table 2. A decomposition algorithm based on the LOQO [48] algorithm was developed to train the SVM face detectors (the SVM method and the hybrid method). This algorithm is similar to that discussed in Section 2.1.2. A Gaussian was chosen as the kernel function with parameter $2\sigma^2 = 1$.

The results from the three methods on a test sequence are illustrated in Fig. 9, while some sample frames of the sequence is shown in Fig. 8. The ground-truth position of the face on each frame is obtained from the multi-view

Table 2
Parameters used to train the multi-view face detectors

	Up-profile	Up-frontal	Down-profile	Down-frontal
Image dimension	400 (20×20)			
Number of subjects	20			
Images of each subject	140			
Total number of face images	2800			
Range of tilt	$[-30,0°]$	$[-30,0°]$	$[0, +30°]$	$[0, +30°]$
Range of yaw	$[-90, -50°]$	$[-50,0°]$	$[-90, -50°]$	$[-50,0°]$
Number of face images	800	960	800	960
Number of non-face images	2320	1243	1733	1208

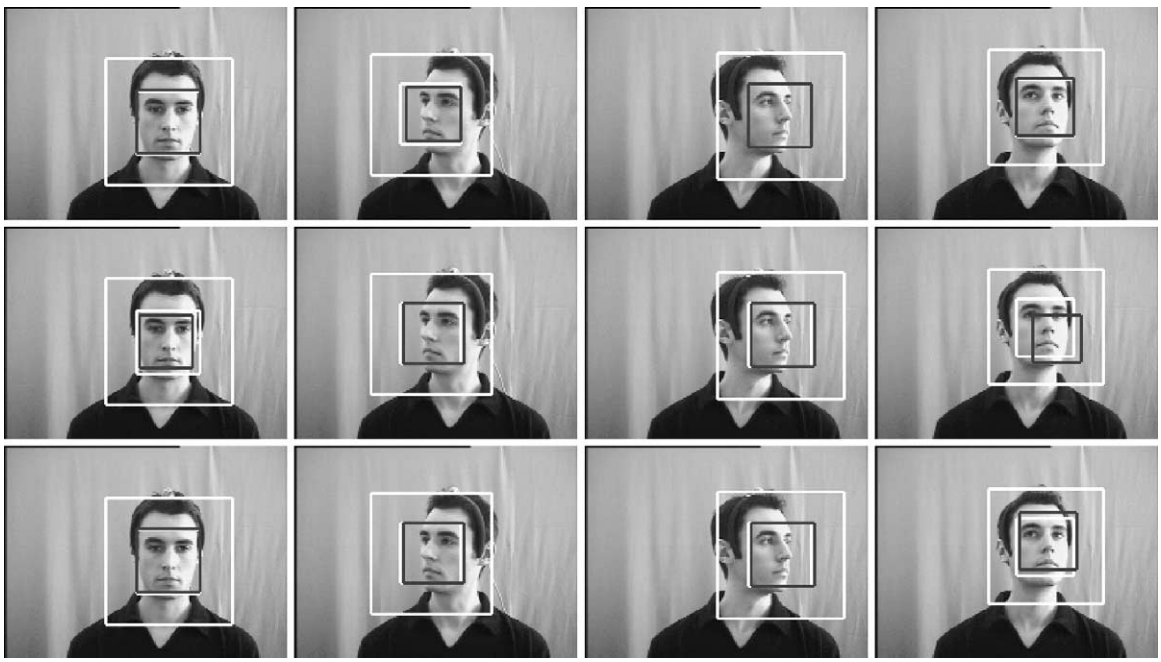


Fig. 8. Sample frames from a test sequence. From top to bottom are the face detection results of the SVM, Eigenface and hybrid methods. For each frame, detection is performed within the outer box. The small white box is the ground-truth position of the face, and the dark box is the detected face pattern.

face acquisition system described in Section 3.1. Face detection is performed within the outer box with a doubled size of the ground-truth box. The reason for using this bounding box is only to ensure that the computation on each frame (the number of image patches from the scan) is equal so that the results are comparable through the whole sequence. We will demonstrate in Section 3.4 that motion and skin-colour can be used effectively to determine the bounding boxes on which face detection is performed.

The experimental results indicate that (Fig. 9):

(1) The SVM method is the most accurate in terms of error in detection scale and location, but also the slowest;
(2) The Eigenface method is the fastest, but less accurate in certain frames;
(3) The hybrid method demonstrates the best balance between accuracy and speed; it is almost as accurate as the SVM method and not significantly slower than the Eigenface method in most frames.

3.4. Detecting faces dynamically from video

Once we construct both pose estimators and view-based face detectors, we can perform multi-view face detection from video input. In our implementation, we used the skin colour and motion detection to bootstrap regions of interest which may contain faces. The motion information normally sketches the *contour* and highly textured regions of a moving object. Skin colour, on the other hand, typically provides *regions* of pixels which are usually located on faces, hands, and arms. The motion and skin colour cues can be used in a complementary manner for selective attention.

We adopt the Gaussian mixture model to compute the probability of a pixel being skin colour [26,36].

$$p(\boldsymbol{\xi}) = \sum_{j=1}^{m} p(\boldsymbol{\xi}|j)P(j) \tag{11}$$

where $P(j)$ is the mixing parameter of component j, $\boldsymbol{\xi}$ is the pixel colour vector in HS format, and $p(\boldsymbol{\xi}|j)$, the density of component j, is constructed with mean $\boldsymbol{\mu}_j$ and covariance matrix $\boldsymbol{\Sigma}_j$:

$$p(\boldsymbol{\xi}|j) = \frac{1}{2\pi|\boldsymbol{\Sigma}_j|^{1/2}} \exp\left\{-\frac{1}{2}(\boldsymbol{\xi} - \boldsymbol{\mu}_j)^T \boldsymbol{\Sigma}_j^{-1}(\boldsymbol{\xi} - \boldsymbol{\mu}_j)\right\} \tag{12}$$

Once the model has been constructed, a look-up table can be created off-line for fast real-time performance [26,36].

Motion detection is performed simply by computing the temporal difference of two successive frames:

$$\frac{\partial I(x,y,t)}{\partial t} = I(x,y,t) - I(x,y,t-1) \tag{13}$$

where I is the image intensity, x, y are the pixel position in the images, and t is the time. For a colour image, I can be computed by averaging the three chromatic components: red, green and blue (RGB). If the value of Eq. (13) is above a preset threshold, the pixel (x, y) is regarded as being on a moving object.

Sample frames of a sequence with the results of motion-colour based selective attention (large boxes) and face detection (small boxes) are shown in Fig. 10.

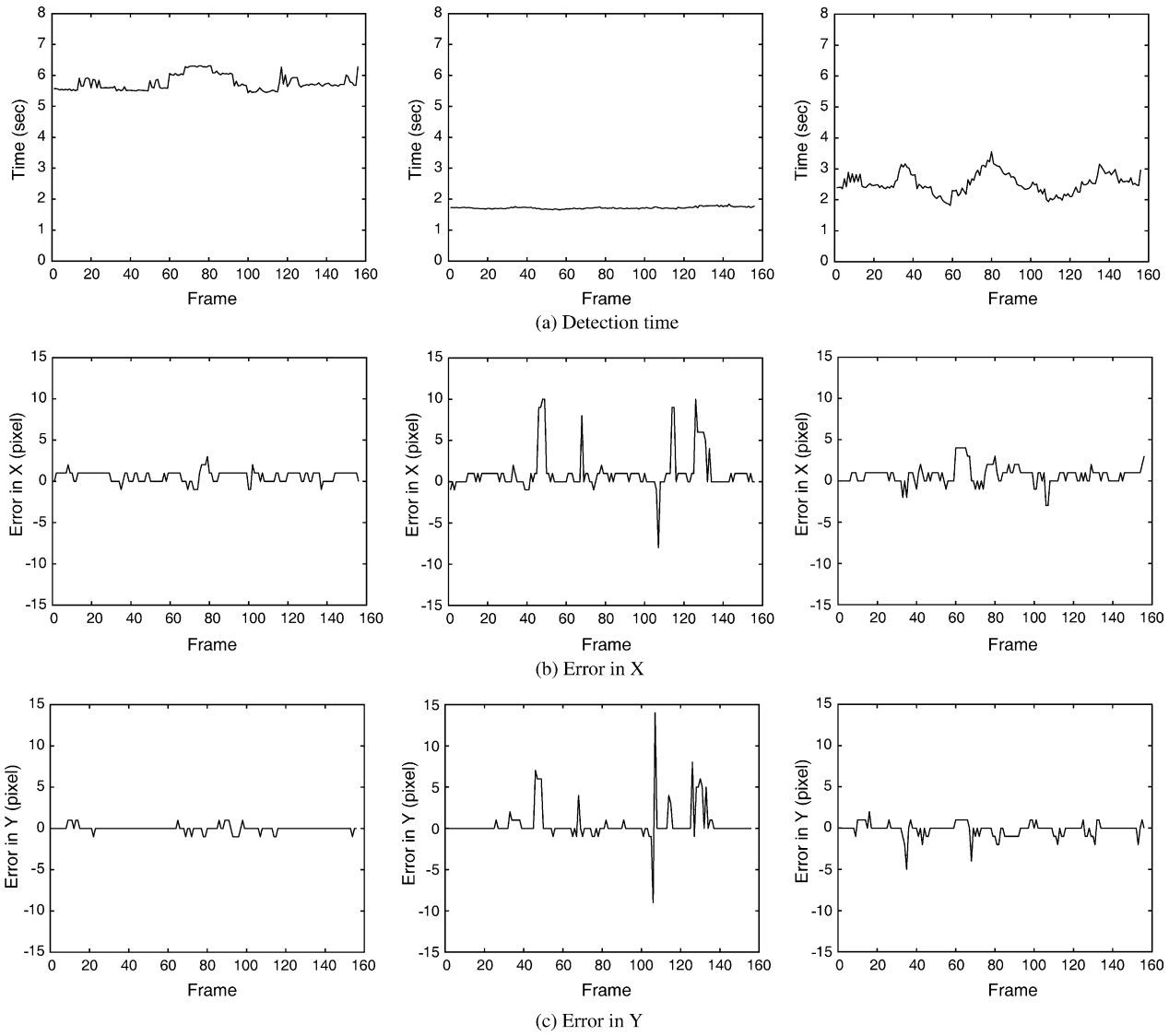


Fig. 9. Comparison results of, from left to right, the SVM, Eigenface and hybrid methods for multi-view face detection on a test sequence: (a) shows the detection time in seconds on each frame; (b) and (c) are the position errors in pixels from the ground-truth position in horizontal (X) and vertical (Y) direction, respectively.

Finally, there are several issues worth mentioning for implementing the face detection system.

(1) When face detection is performed from video input, it is not necessary to follow the procedure described in Section 2 strictly. Besides the selective attention issues discussed above, the pose information in the previous frame can be conveniently used for the detector selection in the current frame (the pose information can be updated after final detection). However, when the pose information is not available or not reliable, for example, in the first frame, or when detection failure occurs, the whole procedure is needed to recover detection.

(2) The pose information needed here is only for the purpose of choosing a detector, therefore a coarse pose estimator can be trained and used to reduce the computation. Nevertheless, when the final detection is obtained, a more precise pose estimation may be necessary for further processing such as face tracking and recognition.

(3) Selecting scales for the first frame of an image sequence (Step 2 of the detection process in Section 2) is arbitrarily determined in our experiment, for example, from 20 pixels to 1/4 of the image size. For the following frames however, the scales can be chosen according to the detected faces in the previous frames.

(4) As we have filtered the images with Sobel operators and normalised the final pattern vectors to unit vectors, the system can perform well with uniformly changing illuminations. We also tested it on cluttered background (Fig. 10).

Fig. 10. Face detection on a video sequence. The larger boxes are obtained by motion-colour based selective attention. Face detection is then performed on these bounding boxes only. The final detections are labelled with the smaller boxes inside the larger ones.

3.5. Frontal-view face recognition

As a case study, we performed *frontal-view* face recognition based on the results of detection. It is important to note that we have also tried multi-view face recognition but the results were unsatisfactory. In psychological vision research, it has been reported that a human's ability to memorise and recognise faces is better in 3/4 views than in frontal and profile views [2,15,20]. Unfortunately there have been few similar findings in machine based face recognition systems. We believe that, for *appearance based* face recognition which is the methodology adopted most widely in this community, correspondence is one of the most important issues for accurate recognition. Note that the correspondence should be in 3D which is often difficult from 2D input images. This is perhaps one of the key reasons why most face recognition systems can only perform well in frontal or near-frontal views—imagine that the appearance of a face is very different from different views, or in other words, misalignment in views may lead to a significant drop in performance.

The experiments were carried out on a small scale where the number of subjects is 10. The training set included 90 face images, nine of each subject. All the faces were collected in frontal view or near frontal view. We ran the multi-view face detectors on those images, then cropped the detected patches as the final training examples for face recognition. Ten one-against-all classifiers were trained using SVM on nine positive examples and 81 negative examples.

Table 3 lists some results from four test sequences. From left to right, the parameters are seq: sequence number, frame: number of frames in sequence, errT: average absolute error of tilt, errY: average absolute error of yaw, detected: number of frames where faces were detected, frontal: number of frames where faces are in frontal view (recognition is performed), recognised: number of frames where subjects were correctly recognised.

The results showed that the performance of face recognition on this small scale problem (10 subjects) is acceptable, therefore it may have potential applications such as access control in a small office environment. We must admit that the correspondence between faces is only implicitly implemented by the multi-view face detectors, therefore this method may lack scalability to a larger number of subjects.

Table 3
Test results on four sequences

Seq	Frame	err$T$	err$Y$	Detected	Frontal	Recognised
1	100	10.8	5.3	100 (100%)	36	32 (89%)
2	200	11.8	10.2	198 (99%)	64	61 (95%)
3	200	8.8	16.2	200 (100%)	56	53 (95%)
4	200	8.0	7.0	193 (97%)	101	93 (92%)
Total	700	9.7	10.3	691 (99%)	257	239 (93%)

4. Conclusions

We have presented in this paper an integrated approach to multi-view face detection. The contributions of this work include:

(1) A novel approach to multi-view face detection, where pose information is explicitly estimated first, and is used to select an appropriate face detector.
(2) An SVM regression based method to estimate the head pose.
(3) A hybrid algorithm combining SVM and Eigenface methods for face detection which provides improved performance in terms of accuracy and speed.

Face detection can be defined as a classification problem of separating face patterns from non-face patterns. Therefore, estimating a boundary which robustly separates the two classes of patterns is more promising than other methods such as probabilistic modelling of the face patterns or the density functions of both face and non-face patterns since only the face and non-face patterns located around the boundary are concerned. This is actually the underlying characteristic of SVMs. By iteratively collecting near-face negative patterns using a prototype face detector, we can gradually refine the detector, making it well fitted to the boundary between face and non-face patterns.

Unlike the frontal-view problem, detecting faces with large pose variation is more challenging since the severe nonlinearity caused by rotation in depth, self-shadowing and self-occlusion yields an extraordinarily irregular distribution of face patterns. The straightforward method, constructing a single universal detector, proved to be inefficient. Some researchers tried to build view-based piece-wise multiple models to solve this problem [17,27,31]. However, computation is intensified since a pattern needs to be evaluated on more models. In this work, we presented a novel approach to this problem by explicitly using the pose information. By determining firstly the possible pose of a pattern, only the classifier for this specific pose is needed for detection. Moreover, the computation on pose estimation, which may be intuitively regarded as an extra burden, does not impose a significant influence on the real-time performance of a system since a 'cheap' pose estimator, which provides a coarse estimation, is sufficient.

We have presented an appearance based approach to pose estimation in this paper. PCA is adopted to represent multi-view face patterns in a low-dimensional orthogonal feature space, and SVM regression is employed to construct the pose estimators. The advantage of the SVM pose estimator is that it can be trained directly from the data with little requirement of the prior knowledge about the data and it is guaranteed to converge.

To improve the overall performance of face detection in terms of both speed and accuracy, three methods for multi-view face detection were implemented and compared with

each other. Experimental results show that the Eigenface method is faster but less accurate as there is a relatively large overlap between the confidence distributions of face and non-face classes (see Fig. 4), while the SVM method is more accurate but slower since the number of SVs cannot be efficiently controlled at a low level. By combining the two methods together, a novel method is proposed which keeps the advantages and suppresses the disadvantages of both methods. The properties of the hybrid method include:

(1) Most 'obvious' patterns are determined by the Eigenface method which is fast;
(2) The ambiguous patterns are classified by the SVM method which is accurate;
(3) The SVM classifier is trained only on a small set of ambiguous patterns, thus it is more accurate and faster.

This hybrid method can also be applied to other classification problems.

Another interesting issue is dynamic face detection from video input. In this situation, motion, skin colour and background information provide enriched information for detection. Although robust motion estimation and colour constancy over time can be problematic, a relatively simple method, which adopts temporal differencing for motion estimation, mixtures of Gaussians for skin colour modelling, and grouping motion and skin colour for selective attention, has proved to be sufficient to improve the real-time performance.

As a case study, we have also implemented a frontal-view face recognition system based on the face detection results. Satisfactory results have been obtained on a small number of subjects.

Acknowledgements

The authors wish to thank Alex Smola for providing the code of the LOQO algorithm.

References

[1] D. Beymer, A. Shashua, T. Poggio. Example based image analysis and synthesis, Technical report, Massachusetts Institute of Technology, A.I. Memo 1431, 1993.
[2] V. Bruce, T. Valentine, The basis of the 3/4 view advantage in face recognition, Applied Cognitive Psychology 1 (1987) 109–120.
[3] M. Burl, P. Perona, Recognition of planar object classes, IEEE Conference on Computer Vision and Pattern Recognition (1996) 223–230.
[4] D. Comaniciu, P. Meer, Mean shift analysis and applications, IEEE International Conference on Computer Vision, vol. 2, Corfu, Greece, 1999, pp. 1197–1203.
[5] E. Elagin, J. Steffens, H. Neven, Automatic pose estimation system for human faces based on bunch graph matching technology, IEEE International Conference on Automatic Face and Gesture Recognition, Nara, Japan, 1998, pp. 136–141.
[6] A. Gee, R. Cipolla, Determining the gaze of faces in images, Image and Vision Computing 12 (10) (1994) 639–647.
[7] S. Gong, S. McKenna, J. Collins, An investigation into face pose distributions, Proceedings of the IEEE International Conference on Automatic Face and Gesture Recognition, Vermont, US, 1996, pp. 265–270.
[8] S. Gong, S. McKenna, A. Psarrou, Dynamic Vision: From Images to Face Recognition, World Scientific Publishing and Imperial College Press, 2000.
[9] S. Gong, E. Ong, P. Loft, Appearance-based face recognition under large head rotations in depth, Asian Conference on Computer Vision, Hong Kong, vol. 2, 1998, pp. 679–686.
[10] S. Gong, E.-J. Ong, S. McKenna, Learning to associate faces across views in vector space of similarities to prototypes, In British Machine Vision Conference, 54-64, Southampton, England, 1998.
[11] T. Horprasert, Y. Yacoob, L. Davis, Computing 3D head orientation from a monocular image sequence, IEEE International Conference on Automatic Face and Gesture Recognition, Vermont, USA, 1996, pp. 242–247.
[12] T.E. Hutchinson, K.P. White Jr., W.N. Martin, K.C. Reichert, L.A. Frey, Human-computer interaction using eye-gaze input, IEEE Transactions on Systems, Man, and Cybernetics 19 (6) (1989) 1527–1534.
[13] P.V.M. Jones, Rapid object detection using a boosted cascade of simple features, IEEE Conference on Computer Vision and Pattern Recognition, Hawaii, USA, 2001.
[14] R. Kjeldsen, J. Kender, Finding skin in color images, IEEE International Conference on Automatic Face and Gesture Recognition, Vermont, USA, 1996, pp. 312–317.
[15] F.L. Krouse, Effects of pose, pose change and delay on face recognition performance, Journal of Applied Psychology 66 (1981) 651–654.
[16] N. Kruger, M. Potzsch, C. von der Malsburg, Determination of face position and pose with a learned representation based on labelled graphs, Image and Vision Computing 15 (8) (1997) 665–673.
[17] S. Li, L. Zhu, Z. Zhang, H. Zhang, Statistical learning of multi-view face detection, European Conference on Computer Vision, Copenhagen, Denmark, 2002.
[18] Y. Li, S. Gong, H. Liddell, Support vector regression and classification based multi-view face detection and recognition, IEEE International Conference on Automatic Face and Gesture Recognition, Grenoble, France, 2000, pp. 300–305.
[19] Y. Li, S. Gong, J. Sherrah, H. Liddell, Multi-view face detection using Support Vector Machines and Eigenspace modeling, The Fourth International Conference on Knowledge-Based Intelligent Engineering System and Allied Technologies, Brighton, UK, 2000, pp. 241–244.
[20] R.H. Logie, A.D. Baddeley, M.M. Woodhead, Face recognition, pose and ecological validity, Applied Cognitive Psychology 1 (1987) 53–69.
[21] Y. Matsumoto, A. Zelinsky, An algorithm for real-time stereo vision implementation of head pose and gaze direction measurement, IEEE International Conference on Automatic Face and Gesture Recognition, Grenoble, France, 2000, pp. 499–504.
[22] T. Maurer, C. von der Malsburg, Tracking and learning graphs and pose on image sequences of faces, IEEE International Conference on Automatic Face and Gesture Recognition, Vermont, USA, 1996, pp. 176–181.
[23] S. McKenna, S. Gong, Tracking faces, IEEE International Conference on Automatic Face and Gesture Recognition, Vermont, US, 1996, pp. 271–276.
[24] S. McKenna, S. Gong, J. Collins, Face tracking and pose representation, British Machine Vision conference, Edinburgh, Scotland, 1996, pp. 755–764.
[25] S. McKenna, S. Gong, Y. Raja, Face recognition in dynamic scenes, British Machine Vision Conference, Colchester, UK, 1997, pp. 140–151.

[26] S. McKenna, S. Gong, Y. Raja, Modelling facial colour and identity with gaussian mixtures, Pattern Recognition 31 (12) (1998) 1883–1892.
[27] B. Moghaddam, A. Pentland, Face recognition using view-based and modular eigenspaces, Automatic Systems for the Identification and Inspection of Humans, SPIE, vol. 2277, 1994.
[28] B. Moghaddam, A. Pentland, Probabilistic visual learning for object representation, IEEE Transactions on Pattern Analysis and Machine Intelligence 19 (7) (1997) 137–143.
[29] B. Moghaddam, W. Wahid, A. Pentland, Beyond Eigenfaces: probabilistic matching for face recognition, IEEE International Conference on Automatic Face and Gesture Recognition, Nara, Japan, 1998, pp. 30–35.
[30] H. Murase, S.K. Nayar, Visual learning and recognition of 3-D objects from appearance, International Journal of Computer Vision 14 (1995) 5–24.
[31] J. Ng, S. Gong, Multi-view face detection and pose estimation using a composite support vector machine across the view sphere, IEEE International Workshop on Recognition, Analysis, and Tracking of Faces and Gestures in Real-Time Systems, Corfu, Greece, 1999, pp. 14–21.
[32] E. Osuna, R. Freund, F. Girosi. Support vector machines: training and applications, Training and applications, Technical report, Massachusetts Institute of Technology, AI Memo 1602, 1997.
[33] E. Osuna, R. Freund, F. Girosi, Training support vector machines: an application to face detection, Proceedings of the Computer Vision and Pattern Recognition' 97 (1997) 130–136.
[34] A. Pentland, B. Moghaddam, T. Starner, View-based and modular eigenspaces for face recognition, IEEE Conference on Computer Vision and Pattern Recognition, Seatle, 1994, pp. 84–94.
[35] Y. Raja, S. McKenna, S. Gong, Colour model selection and adaptation in dynamic scenes, European Conference on Computer Vision, Freiburg, Germany, 1998.
[36] Y. Raja, S. McKenna, S. Gong, Segmentation and tracking using colour mixture models, Asian Conference on Computer Vision, Hong Kong, 1998.
[37] H. Rowley, S. Baluja, T. Kanade, Neural network-based face detection, IEEE Conference on Computer Vision and Pattern Recognition, San Francisco, CA, USA, 1996, pp. 203–207.
[38] H. Rowley, S. Baluja, T. Kanade, Rotation invariant neural network-based face detection, Technical report, School of Computer Science, Carnegie Mellon University, CMU-CS-97-201, 1997.
[39] H. Rowley, S. Baluja, T. Kanade, Neural network-based face detection, IEEE Transactions on Pattern Analysis and Machine Inteligence 20 (1) (1998).
[40] H. Rowley, S. Baluja, T. Kanade, Rotation invariant neural network-based face detection, IEEE Conference on Computer Vision and Pattern Recognition, 1998.
[41] B. Scholkopf, P. Bartlett, A. Smola, R. Williamson, in: L. Niklasson, M. Bod'en, T. Ziemke (Eds.), Support vector regression with automatic accuracy control, Proceedings of the 8th International Conference on Artificial Neural Networks, Perspectives in Neural Computing, Springer, Berlin, 1998, pp. 111-116.
[42] B. Scholkopf, A. Smola, R. Williamson, P.L. Bartlett, New support vector algorithms, Neural Computation 12 (5) (2000) 1207–1245.
[43] A. Smola, B. Scholkopf, G. Rotsch, Linear programs for automatic accuracy control in regression, The Ninth International Conference on Artificial Neural Networks, London, 1999, pp. 575–580.
[44] F. Soulie, F. Viennet, B. Lamy, Multi-modular neural network architectures: applications in optical character and human face recognition, International Journal of Pattern Recognition and Artificial Intelligence 7 (4) (1993) 721–755.
[45] C. Stauffer, W. Grimson, Adaptive background mixture models for real-time tracking, IEEE Conference on Computer Vision and Pattern Recognition, Fort Collins, CO, USA, vol. 2, 1999, pp. 246–252.
[46] K. Sung, T. Poggio, Example-based learning for view-based human face detection, Technical report, Massachusetts Institute of Technology, AI MEMO 1521, 1994.
[47] K. Toyama, J. Krumm, B. Brumitt, B. Meyers, Wallflower: principles and practice of background maintenance, IEEE International Conference on Computer Vision, Kerkyra, Greece, vol. 1, 1999, pp. 255–261.
[48] R. Vanderbei, LOQO: an interior point code for quadratic programming, Technical report, Princeton University, Technical Report SOR 94-15, 1994.
[49] V. Vapnik, The Nature of Statistical Learning Theory, Springer, New York, 1995.
[50] M. Xu, T. Adatsuka, Detecting head pose from stereo image sequence for active face recognition, IEEE International, Conference on Automatic Face and Gesture Recognition, Nara, Japan, 1998, pp. 82–87.
[51] K. Yow, R. Cipolla, Detection of human faces under scale, orientation and viewpoint variations, IEEE International, Conference on Automatic Face and Gesture Recognition, Vermont, USA, 1996, pp. 295–300.
[52] K. Yow, R. Cipolla, A probalilistic framework for perceptual grouping of features for human face detection, IEEE International, Conference on Automatic Face and Gesture Recognition, Vermont, USA, 1996, pp. 16–21.
[53] Z. Zhang, L. Zhu, S. Li, H. Zhang, Real-time multi-view face detection, IEEE International Conference on Automatic Face and Gesture Recognition, Washington, DC, 2002.