Available at
www.ElsevierComputerScience.com
POWERED BY SCIENCE DIRECT
Image and Vision Computing 21 (2003) 1077–1086
www.elsevier.com/locate/imavis

Recognising trajectories of facial identities using kernel discriminant analysis

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

[a]Content and Coding Lab, BT Exact, Adastral Park, Ipswich IP5 3RE, UK
[b]Department of Computer Science, Queen Mary, University of London, London E1 4NS, UK

Accepted 13 August 2003

Abstract

We present a comprehensive approach to address three challenging problems in face recognition: modelling faces across multi-views, extracting the nonlinear discriminating features, and recognising moving faces dynamically in image sequences. A multi-view dynamic face model is designed to extract the *shape-and-pose-free* facial texture patterns. Kernel discriminant analysis, which employs the kernel technique to perform linear discriminant analysis in a high-dimensional feature space, is developed to extract the significant nonlinear features which maximise the between-class variance and minimise the within-class variance. Finally, an identity surface based face recognition is performed dynamically from video input by matching object and model trajectories.

Keywords: Face recognition; Kernel discriminant analysis; Identity surfaces; Multi-view face models

1. Introduction

Face recognition has been of considerable interest over the past decade. Various approaches such as Eigenfaces [20], Elastic Graph model [12], Linear Object Classes [24], Active Shape Models (ASMs) [6] and Active Appearance Models (AAMs) [5] have been proposed to address this problem. It is important to point out that most of the previous work in face recognition is mainly concerned with frontal view or near frontal views. Due to the severe nonlinearity caused by rotation in depth, self-occlusion, self-shading and illumination change, recognising faces with large pose variation is more challenging than that at a fixed view, e.g. frontal view.

Extracting the discriminating features, which maximise the between-class variance and minimise the within-class variance, is crucial to face recognition, especially when faces are undergoing large pose variation. Principal component analysis (PCA), also known as eigenface method, has been widely adopted in this area [18,20]. However, it is worth noting that the features extracted by PCA are actually 'global' features for all face classes, thus they are not necessarily representative for discriminating one face class from others. Linear discriminant analysis (LDA), which seeks to find a linear transformation by maximising the between-class variance and minimising the within-class variance, proved to be a more suitable technique for classification [8,19]. Although LDA can provide a significant discriminating improvement to the task of face recognition, it is still a linear technique in nature. When severe nonlinearity is involved, this method is intrinsically poor. Another shortcoming of LDA lies in the fact that the number of basis vectors is limited by the number of face classes, therefore it would be less representative when a small set of subjects is concerned. To extract the nonlinear principal components, Kernel PCA (KPCA) was developed [16,17]. However, as with PCA, KPCA captures the *overall* variance of all patterns which are inadequate for discriminating purposes.

Another limitation of the previous studies is that the methodology adopted for recognition is largely based on matching static face images. Psychology and physiology research depicts that the human vision system's ability to recognise animated faces is better than that on randomly ordered still face images (i.e. the same set of images, but displayed in random order without the temporal context of moving faces) [2,11]. For computer vision systems,

* Corresponding author.
E-mail address: yongmin.li@bt.com (Y. Li).

0262-8856/$ - see front matter © 2003 Elsevier B.V. All rights reserved.
doi:10.1016/j.imavis.2003.08.010

although some work has been reported [7,9,10], the problem of recognising the dynamics of human faces in a spatio-temporal context remains largely unresolved.

In this work, we present a comprehensive approach to address the three challenging problems in face recognition stated above. A multi-view dynamic face model is designed to extract the shape-and-pose-free facial texture patterns for accurate across-view registration. Kernel discriminant analysis (KDA), a kernel based technique, is developed to compute the nonlinear discriminating basis vectors. Finally face recognition is performed dynamically by matching an object trajectory tracked from a video input with model trajectories synthesised on *identity surfaces*.

2. Kernel discriminant analysis

As stated in Section 1, both PCA and LDA are limited to linear problems, and KPCA is designed to deal with the overall rather than the *discriminating* variance. In this work, KDA, a nonlinear discriminating approach based on the kernel technique [22] is developed for extracting the nonlinear discriminating features.

The underlying principle of KDA is described as follows. Assume we have a set of training patterns $\{\mathbf{x}\}$ which are categorised into C classes. Owing to nonlinearity, these patterns cannot be linearly separated in the space of $\{\mathbf{x}\}$. With an appropriately selected nonlinear mapping $\boldsymbol{\phi}$, we project the patterns onto a high-dimensional feature space where the images of the patterns are linearly separable. Then by performing LDA in the feature space, one can obtain a nonlinear representation in the original input space. However, computing $\boldsymbol{\phi}$ explicitly may be problematic or even impossible. In the rest of this section, we will discuss in details how to apply the kernel technique to solve the problem. Following the idea of Ref. [16], we first derive the algorithm in the case of centred data, and then extend it to the non-centred case.

It is important to point out that we have noticed similar algorithms have been published by Roth and Steinhage [15], Mika et al. [14] and Baudat and Anouar [1]. Basically these algorithms are equivalent although algorithm formulation and procedure are slightly different. For example, we use one matrix inversion and one eigen-decomposition, while two steps of eigen-decomposition are performed in Ref. [1].

2.1. Centred data

If the nonlinear mapping $\boldsymbol{\phi}$ satisfies Mercer's condition [22,23], then the inner product of two vectors in the feature space can be computed through a kernel function

$$k(\mathbf{x},\mathbf{y}) = (\boldsymbol{\phi}(\mathbf{x})\cdot\boldsymbol{\phi}(\mathbf{y})) \tag{1}$$

which can be conveniently performed in the input space.

Let us first consider the centred data set in the feature space, i.e.

$$\sum_{i=1}^{N} \boldsymbol{\phi}_i = 0 \tag{2}$$

where N is the total number of training patterns. Define a between-class scatter matrix $\mathbf{S}_\mathrm{b}$ and a within-class scatter matrix $\mathbf{S}_\mathrm{w}$ in the feature space as

$$\mathbf{S}_\mathrm{b} = \frac{1}{C}\sum_{c=1}^{C} \boldsymbol{\mu}_c\boldsymbol{\mu}_c^\mathrm{T} \tag{3}$$

$$\mathbf{S}_\mathrm{w} = \frac{1}{C}\sum_{c=1}^{C}\frac{1}{N_c}\sum_{i=1}^{N_c} \boldsymbol{\phi}_{ci}\boldsymbol{\phi}_{ci}^\mathrm{T} \tag{4}$$

where N_c is the number of patterns in the cth class, and $\boldsymbol{\mu}_c$ is the mean vector of class c

$$\boldsymbol{\mu}_c = \frac{1}{N_c}\sum_{i=1}^{N_c} \boldsymbol{\phi}_{ci} \tag{5}$$

Since the data are centred, we do not need to subtract the global mean vector from $\boldsymbol{\mu}_c$ and $\boldsymbol{\phi}_i$. Thus the scatter matrices $\mathbf{S}_\mathrm{b}$ and $\mathbf{S}_\mathrm{w}$ can be expressed in a simplistic form. We will return to the general case where the data are not centred in Section 2.2.

If it is assumed that $\mathbf{S}_\mathrm{w}$ is not singular, we can maximise the between-class variance and minimise the within-class variance of vectors $\boldsymbol{\phi}_i$ in the feature space by performing eigen-decomposition on the matrix

$$\mathbf{S} = \mathbf{S}_\mathrm{w}^{-1}\mathbf{S}_\mathrm{b} \tag{6}$$

If we assume $\mathbf{v}$ is one of the eigenvectors of matrix $\mathbf{S}$, and λ is its corresponding eigenvalue, we get

$$\mathbf{S}\mathbf{v} = \lambda\mathbf{v} \tag{7}$$

If we combine Eqs. (6) and (7), we obtain

$$\mathbf{S}_\mathrm{b}\mathbf{v} = \lambda\mathbf{S}_\mathrm{w}\mathbf{v} \tag{8}$$

If we then take the inner product with vector $\boldsymbol{\phi}_m$ on both sides of Eq. (8), we obtain

$$(\mathbf{S}_\mathrm{b}\mathbf{v}\cdot\boldsymbol{\phi}_m) = \lambda(\mathbf{S}_\mathrm{w}\mathbf{v}\cdot\boldsymbol{\phi}_m), \qquad m = 1,2,\dots,N \tag{9}$$

A coefficient vector exists

$$\boldsymbol{\alpha} = (\alpha_1,\alpha_2,\dots,\alpha_N)^\mathrm{T} \tag{10}$$

that satisfies

$$\mathbf{v} = \sum_{n=1}^{N} \alpha_n\boldsymbol{\phi}_n \tag{11}$$

If we substitute Eqs. (3)–(5) and (11) in Eq. (9), we obtain

$$\sum_{n=1}^{N}\alpha_n\sum_{c=1}^{C}\frac{1}{N_c^2}\sum_{i=1}^{N_c}\sum_{j=1}^{N_c}(\boldsymbol{\phi}_{ci}\cdot\boldsymbol{\phi}_m)(\boldsymbol{\phi}_{cj}\cdot\boldsymbol{\phi}_n)$$

$$=\lambda\sum_{n=1}^{N}\alpha_n\sum_{c=1}^{C}\frac{1}{N_c}\sum_{i=1}^{N_c}(\boldsymbol{\phi}_{ci}\cdot\boldsymbol{\phi}_m)(\boldsymbol{\phi}_{ci}\cdot\boldsymbol{\phi}_n) \tag{12}$$

For each class c, if we define an $N\times N_c$ matrix $\mathbf{K}_c$ as

$$(\mathbf{K}_c)_{ij} := (\boldsymbol{\phi}_i\cdot\boldsymbol{\phi}_j) = k(\mathbf{x}_i,\mathbf{x}_{cj}) \tag{13}$$

and a $N_c\times N_c$ matrix $\mathbf{1}_{N_c}$ as

$$(\mathbf{1}_{N_c})_{ij} := 1 \tag{14}$$

we obtain

$$\left(\sum_{c=1}^{C}\frac{1}{N_c^2}\mathbf{K}_c\mathbf{1}_{N_c}\mathbf{K}_c^{\mathrm{T}}\right)\boldsymbol{\alpha}=\lambda\left(\sum_{c=1}^{C}\frac{1}{N_c}\mathbf{K}_c\mathbf{K}_c^{\mathrm{T}}\right)\boldsymbol{\alpha} \tag{15}$$

If we define $N\times N$ matrix as

$$\mathbf{A}=\left(\sum_{c=1}^{C}\frac{1}{N_c}\mathbf{K}_c\mathbf{K}_c^{\mathrm{T}}\right)^{-1}\left(\sum_{c=1}^{C}\frac{1}{N_c^2}\mathbf{K}_c\mathbf{1}_{N_c}\mathbf{K}_c^{\mathrm{T}}\right) \tag{16}$$

we derive

$$\mathbf{A}\boldsymbol{\alpha}=\lambda\boldsymbol{\alpha} \tag{17}$$

By eigen-decomposing matrix $\mathbf{A}$, we obtain the coefficient vector $\boldsymbol{\alpha}$. Therefore, for a new pattern $\mathbf{x}$ in the original input space, we can calculate its projection onto $\mathbf{v}$ in the high-dimensional feature space by

$$(\boldsymbol{\phi}(x)\cdot\mathbf{v})=\sum_{i=1}^{N}\alpha_i(\boldsymbol{\phi}_i\cdot\boldsymbol{\phi}(x))=\sum_{i=1}^{N}\alpha_i k(\mathbf{x},\mathbf{x}_i)=\boldsymbol{\alpha}^{\mathrm{T}}\mathbf{k}_x \tag{18}$$

where

$$\mathbf{k}_x=\left(k(\mathbf{x},\mathbf{x}_1),k(\mathbf{x},\mathbf{x}_2),\ldots,k(\mathbf{x},\mathbf{x}_N)\right)^{\mathrm{T}} \tag{19}$$

If we construct the eigenmatrix

$$\mathbf{U}=[\boldsymbol{\alpha}_1,\boldsymbol{\alpha}_2,\ldots,\boldsymbol{\alpha}_M] \tag{20}$$

from the first M significant eigenvectors of $\mathbf{A}$, the projection of $\mathbf{x}$ in the M-dimensional KDA space is given by

$$\mathbf{y}=\mathbf{U}^{\mathrm{T}}\mathbf{k}_x \tag{21}$$

2.2. Non-centred data

In the general case, $\{\boldsymbol{\phi}(\mathbf{x}_i)\}$, $i=1,2,\ldots,N$, are not centred in the feature space. A similar method to Ref. [16] is adopted here. By defining

$$\tilde{\boldsymbol{\phi}}_i := \boldsymbol{\phi}_i-\frac{1}{N}\sum_{n=1}^{N}\boldsymbol{\phi}_n \tag{22}$$

we can use the method stated above since $\{\tilde{\boldsymbol{\phi}}_i\}$, $i=1,2,\ldots,N$ are now centred. The $N\times N$ kernel matrix $\tilde{\mathbf{K}}$ can then be expressed as:

$$(\tilde{\mathbf{K}})_{ij}=(\tilde{\boldsymbol{\phi}}_i\cdot\tilde{\boldsymbol{\phi}}_j)=\left(\boldsymbol{\phi}_i-\frac{1}{N}\sum_{m=1}^{N}\boldsymbol{\phi}_m\right)\cdot\left(\boldsymbol{\phi}_j-\frac{1}{N}\sum_{n=1}^{N}\boldsymbol{\phi}_n\right)$$

$$=(\boldsymbol{\phi}_i\cdot\boldsymbol{\phi}_j)-\frac{1}{N}\sum_{m=1}^{N}(\boldsymbol{\phi}_m\cdot\boldsymbol{\phi}_j)-\frac{1}{N}\sum_{n=1}^{N}(\boldsymbol{\phi}_i\cdot\boldsymbol{\phi}_n)$$

$$+\frac{1}{N^2}\sum_{m=1}^{N}\sum_{n=1}^{N}(\boldsymbol{\phi}_m\cdot\boldsymbol{\phi}_n)$$

$$=k_{ij}-\frac{1}{N}\sum_{m=1}^{N}k_{mj}-\frac{1}{N}\sum_{n=1}^{N}k_{in}+\frac{1}{N^2}\sum_{m=1}^{N}\sum_{n=1}^{N}k_{mn} \tag{23}$$

If we use $N\times N$ matrix $(\mathbf{K})_{ij} := k(\boldsymbol{\phi}_i\cdot\boldsymbol{\phi}_j)$ and $\mathbf{1}_N$, we obtain

$$\tilde{\mathbf{K}}=\mathbf{K}-\frac{1}{N}\mathbf{1}_N\mathbf{K}-\mathbf{K}\frac{1}{N}\mathbf{1}_N+\frac{1}{N^2}\mathbf{1}_N\mathbf{K}\mathbf{1}_N \tag{24}$$

Therefore $\tilde{\mathbf{K}}_c$ can be obtained as a sub-matrix of $\tilde{\mathbf{K}}$. Then by substituting $\mathbf{K}_c$ with $\tilde{\mathbf{K}}_c$ in Eq. (16) and eigen-decomposing $\mathbf{A}$, we obtain the matrix $\mathbf{U}$ in Eq. (20).

Similar to the centred case given in Eq. (18), the projection of a new pattern $\mathbf{x}$ onto an eigenvector $\tilde{\mathbf{v}}$ in the feature space is given by

$$(\tilde{\boldsymbol{\phi}}(\mathbf{x})\cdot\tilde{\mathbf{v}})=\sum_{i=1}^{N}\alpha_i(\tilde{\boldsymbol{\phi}}(\mathbf{x})\cdot\tilde{\boldsymbol{\phi}}(\mathbf{x}_i))=\tilde{\mathbf{k}}_x\boldsymbol{\alpha} \tag{25}$$

where

$$(\tilde{\mathbf{k}}_x)_i=\left(\boldsymbol{\phi}(\mathbf{x})-\frac{1}{N}\sum_{m=1}^{N}\boldsymbol{\phi}(\mathbf{x}_m)\right)$$

$$\times\left(\boldsymbol{\phi}(\mathbf{x}_i)-\frac{1}{N}\sum_{n=1}^{N}\boldsymbol{\phi}(\mathbf{x}_n)\right)$$

$$=k(\mathbf{x},\mathbf{x}_i)-\frac{1}{N}\sum_{m=1}^{N}k(\mathbf{x}_i,\mathbf{x}_m)-\frac{1}{N}\sum_{n=1}^{N}k(\mathbf{x},\mathbf{x}_n)$$

$$+\frac{1}{N^2}\sum_{m=1}^{N}\sum_{n=1}^{N}k(\mathbf{x}_m,\mathbf{x}_n) \tag{26}$$

If we define an $N\times 1$ vector $\mathbf{1}'$ with all entries equal to 1, we obtain

$$\tilde{\mathbf{k}}_x=\mathbf{k}_x-\frac{1}{N}\mathbf{K}\mathbf{1}'-\frac{1}{N}\mathbf{k}_x\mathbf{1}_N+\frac{1}{N^2}\mathbf{1}'\mathbf{K}\mathbf{1}_N \tag{27}$$

Finally, the projection of $\mathbf{x}$ in the M-dimensional KDA space is given by

$$\mathbf{y}=\mathbf{U}^{\mathrm{T}}\tilde{\mathbf{k}}_x \tag{28}$$

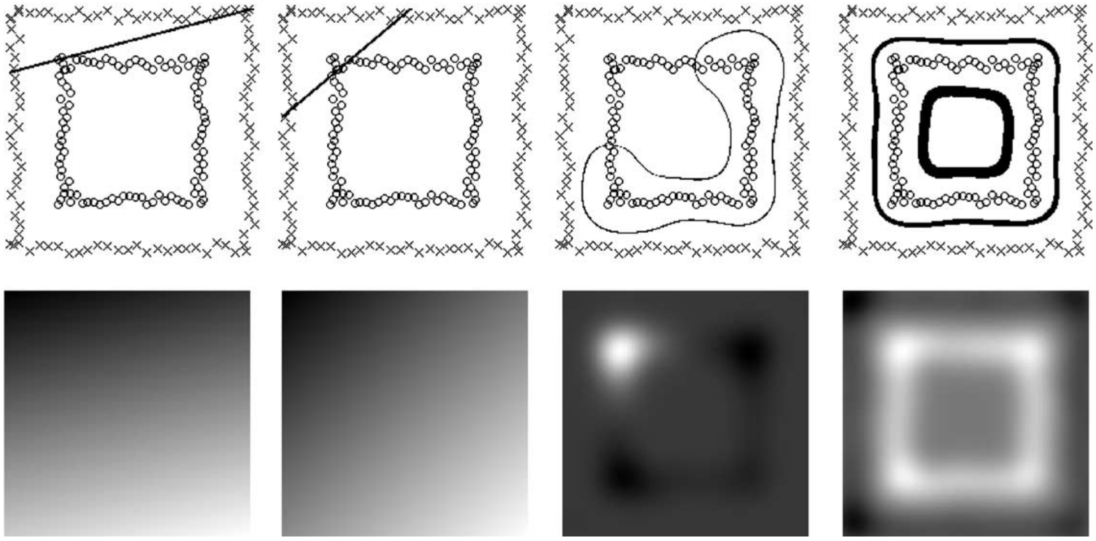


Fig. 1. Solving a nonlinear classification problem with, from left to right, PCA, LDA, KPCA and KDA.

2.3. A toy problem

We use a 'toy' problem to illustrate the characteristics of KDA as shown in Fig. 1. Two classes of patterns, denoted by circles and crosses, respectively, have a significant nonlinear distribution. We try to separate them with a *one-dimensional* decision boundary of PCA, LDA, KPCA or KDA. The upper row shows the patterns and the discriminating curves computed by the four different methods. The lower row illustrates the intensity values of the one-dimensional features computed from PCA, LDA, KPCA and KDA. It can be seen clearly that PCA and LDA are incapable of providing correct classification because of their linear nature. Neither does KPCA do so since it is designed to extract the overall rather than the discriminating variance although it is nonlinear in principle. KDA gives the correct classification boundary: the discriminating curve accurately separates the two classes of patterns, and the feature intensity correctly reflects the actual pattern distribution.

2.4. Remarks on computations

In the training stage, two steps of computation are intensive: the inversion of an $N \times N$ matrix in Eq. (16) and the eigen-decomposition of an $N \times N$ matrix in Eq. (17), both with complexity of $O(N^3)$. This implies that the method as presented above has limitations when applying to problems with large training examples. A simple strategy of selecting a small number of examples for training is usually helpful in many applications. In the experiments presented later in this paper, we adopted this strategy for KDA training.

Another limitation of the algorithm is computing the projection of a new vector in Eq. (27), where all the N training vectors are involved. Note that this is also a limitation of KPCA. Some method such as the reduced set technique [3,4] may be used to relieve the computation burden.

3. Multi-view dynamic face model

Due to the severe nonlinearity caused by rotation in depth, self-occlusion, self-shading and illumination change, modelling the appearance of faces across multiple views is much more challenging than that from a fixed, e.g. frontal view. Another significant difficulty for multi-view face recognition comes from the fact that the appearances of different people from the same view are often more similar than those of the same person from different views.

A multi-view dynamic face model, which consists of a sparse 3D Point Distribution Model (PDM) [6], a shape-and-pose-free texture model, and an affine geometrical model, has been developed in this work [13]. This model is extended from AAMs [5], but is distinguished from AAMs by: (1) a 3D shape model is constructed from 2D images, and (2) the texture model is built from shape-and-pose-free texture patterns.

The 3D shape vector of a face is estimated from a set of 2D face images in different views, i.e. given a set of 2D face images with known pose and 2D positions of the landmarks, the 3D shape vector can be estimated using linear regression.

To establish the correspondence between faces with different shapes and across multiple views, a face image fitted by the shape model is warped to the mean shape at frontal view (with 0° in both tilt and yaw), obtaining a shape-and-pose-free texture pattern. This is implemented by forming a triangulation from the landmarks and employing a piece-wise affine transformation between each triangle pair.

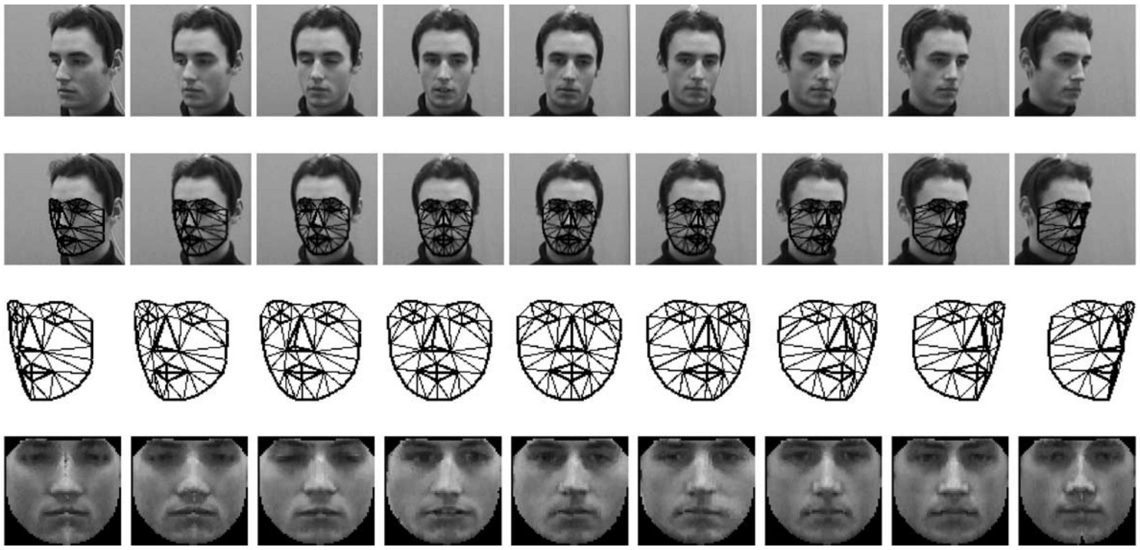


Fig. 2. Multi-view dynamic face model. From top to bottom are sample training face images, the landmarks labelled on the images, the estimated 3D shape rotating from -40 to $+40°$ in yaw and with tilt fixed on 0°, and the extracted *shape-and-pose-free* texture patterns.

By warping to the mean shape, one obtains the shape-free texture of the given face image. Furthermore, by warping to the frontal view, a pose-free texture representation is achieved.[1]

We applied PCA to the 3D shape patterns and shape-and-pose-free texture patterns, respectively, to obtain a low dimensional statistical model. Fig. 2 shows the sample face images used to construct the model, the landmarks labelled on each image, the 3D shape estimated from these labelled face images, and the extracted shape-and-pose-free texture patterns.

Based on the analysis above, a face pattern can be represented in the following way. First, the 3D shape model is fitted to a given image or video sequence containing faces. Then the face texture is warped onto the mean shape of the 3D PDM model in frontal view. Finally, by adding parameters controlling pose, shift and scale, the complete parameter set of the dynamic model for a given face pattern is $\mathbf{c} = (\mathbf{s}, \mathbf{t}, \alpha, \beta, \mathrm{d}x, \mathrm{d}y, r)^{\mathrm{T}}$ where $\mathbf{s}$ is the shape parameter, $\mathbf{t}$ is the texture parameter, (α, β) is pose in tilt and yaw, $(\mathrm{d}x, \mathrm{d}y)$ is the translation of the centroid of the face, and r is its scale.

The shape-and-pose-free texture patterns obtained from model fitting are adopted for face recognition. In our experiments, we also tried to use the shape patterns for recognition, however, the performance was not as good as that of using textures.

[1] There may still be some kinds of residual variation from other sources which are pose-dependent, for example, the self-shading and illumination change while a face is rotating out of the image plane. At this stage, we ignore these variations and focus on the variation from the geometrical rotation of faces.

4. Extracting the nonlinear discriminating features of multi-view face patterns

For face recognition, one needs to deal with two kinds of variation: variation from identities (between-class variance) and variation from other sources (within-class variance). An effective representation of face patterns should provide a mechanism to emphasise the former and suppress the latter.

Although the within-class variation of the shape-and-pose-free facial texture patterns has been reduced from their original form, the underlying discriminating features for different face classes have not been represented explicitly. Therefore such a representation in itself may not be efficient for recognition. This situation is shown in Fig. 4a where the shape-and-pose-free texture patterns of 12 face classes are represented by PCA, the widely used method in face recognition. For the sake of clarity, only patterns of the first four subjects are displayed. It is noted that the variance from different face classes is not efficiently separated from that for pose change, or more precisely, the former is even overshadowed by the latter. The original face images and warped texture patterns from one of these subjects are shown in Fig. 3.

We also applied LDA, KPCA and KDA on the same set of face patterns. The distributions of the face patterns of the first four subjects are shown in the first two significant dimensions as in Fig. 4. It is noted that

(1) the pattern distributions using PCA and KPCA are not satisfactorily separable since these two techniques are not designed for discriminating purpose;
(2) LDA performs better than PCA and KPCA, but not as good as KDA;

(a)The original images of a faceclass

(a)The warped facial texture patterns

Fig. 3. The original face images and the warped facial texture patterns of one of the 12 subjects.

(3) KDA provides the best separation performance among the four methods.

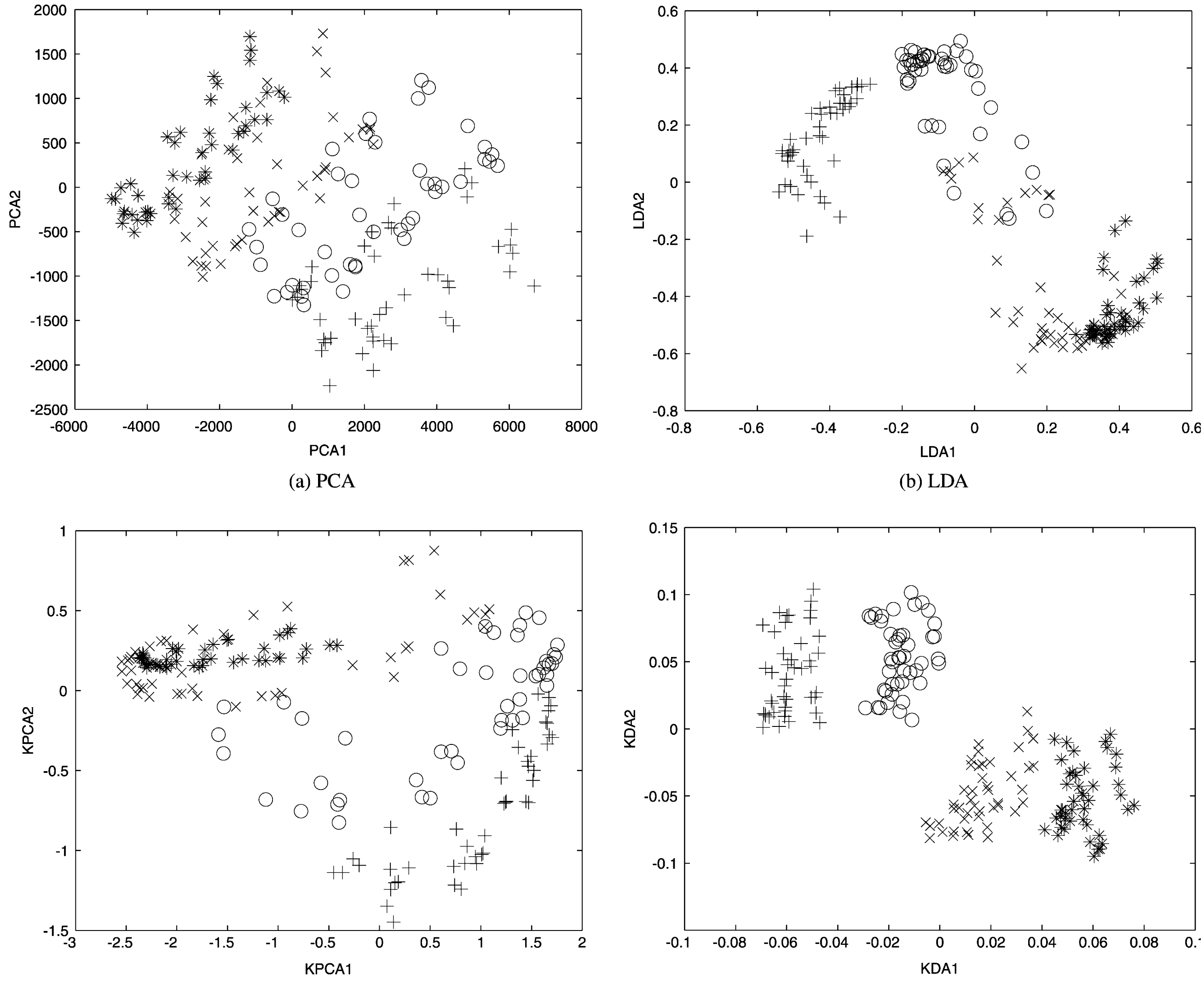
In this experiment, the Gaussian kernel is adopted for KPCA and KDA,

$$k(\mathbf{x}, \mathbf{y}) = \exp\left(-\frac{\|\mathbf{x} - \mathbf{y}\|^2}{2\sigma^2}\right) \tag{29}$$

where $2\sigma^2 = 1$.

5. Recognising multi-view faces using identity surfaces

The traditional techniques for face recognition include computing the Euclidean or Mahalanobis distance to a face template and estimating the density of patterns using multi-modal models. However, the problem of *multi-view* face recognition can be solved more efficiently if the pose information is available. Based on this idea, we propose an approach to multi-view face recognition by constructing identity surfaces in a discriminating feature space.



(a) PCA (b) LDA (c) KPCA (d) KDA

Fig. 4. Distribution of multi-view face patterns in the first two significant PCA, LDA, KPCA and KDA dimensions.

As shown in Fig. 5, each subject to be recognised is represented by a unique hyper surface based on pose information. In other words, the two basis coordinates stand for the head pose: tilt and yaw, and the other coordinates are used to represent the discriminating feature patterns of faces. For each pair of tilt and yaw, there is one unique 'point' for a face class. The distribution of all these points of a same face class forms a hyper surface in this feature space. We call this surface an identity surface.

5.1. Synthesising identity surfaces

We propose to synthesise the identity surface of a subject from a small sample of face patterns which sparsely cover the view sphere. The basic idea is to approximate the identity surface using a set of N_p planes separated by a number of N_v predefined views. The problem can be formally defined as follows.

Suppose x, y are tilt and yaw, respectively, $\mathbf{z}$ is the discriminating feature vector of a face pattern, i.e. the KDA

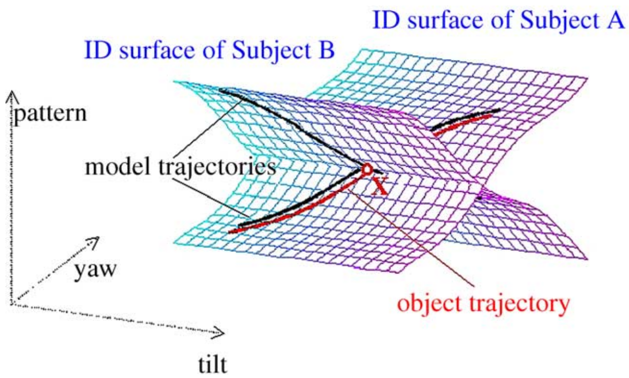


Fig. 5. Identity surfaces for dynamic face recognition.

vector. $(x_{01}, y_{01}), (x_{02}, y_{02}), \ldots, (x_{0N_v}, y_{0N_v})$ are predefined views which separate the view plane into N_p pieces. On each of these N_p pieces, the identity surface is approximated by a plane

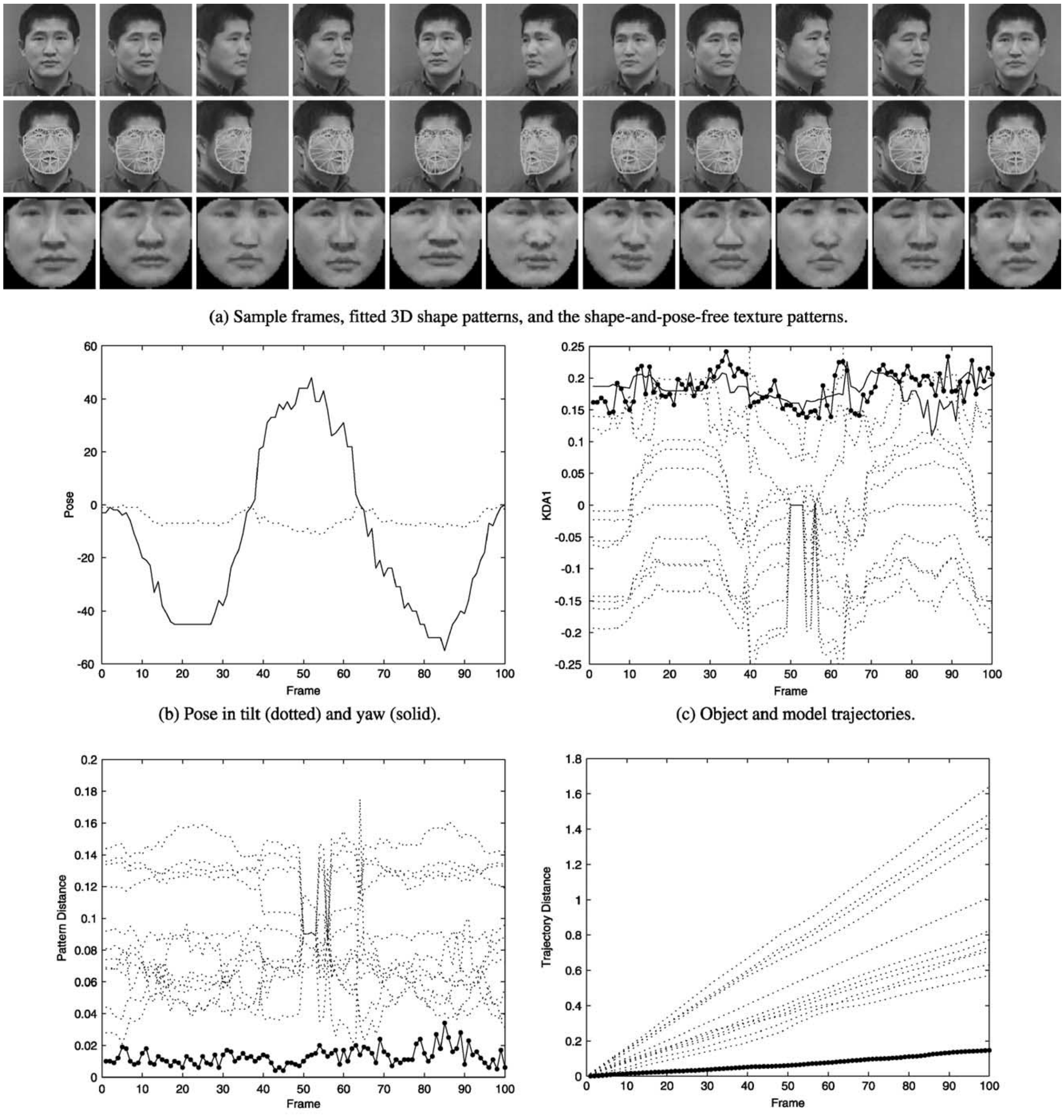
$$\mathbf{z} = \mathbf{a}x + \mathbf{b}y + \mathbf{c} \tag{30}$$

Suppose the M_i sample patterns covered by the ith plane are $(x_{i1}, y_{i1}, \mathbf{z}_{i1}), (x_{i2}, y_{i2}, \mathbf{z}_{i2}), \ldots, (x_{iM_i}, y_{iM_i}, \mathbf{z}_{iM_i})$, then one minimises

$$Q = \sum_{i}^{N_p} \sum_{m}^{M_i} \|\mathbf{a}_i x_{im} + \mathbf{b}_i y_{im} + \mathbf{c}_i - \mathbf{z}_{im}\|^2 \tag{31}$$

subject to

$$\begin{aligned} &\mathbf{a}_i x_{0k} + \mathbf{b}_i y_{0k} + \mathbf{c}_i = \mathbf{a}_j x_{0k} + \mathbf{b}_j y_{0k} + \mathbf{c}_j, \\ &k = 0, 1, \ldots, N_v, \text{ planes } i, j \text{ intersect at } (x_{0k}, y_{0k}) \end{aligned} \tag{32}$$



(a) Sample frames, fitted 3D shape patterns, and the shape-and-pose-free texture patterns.

(b) Pose in tilt (dotted) and yaw (solid).

(c) Object and model trajectories.

(d) Pattern distances.

(e) Trajectory distances.

Fig. 6. Video-based multi-view face recognition. (c) shows the object trajectory (solid line with dots) and model trajectories in the first KDA dimension where the model trajectory from the ground-truth subject is highlighted with solid line. It is noted from (d) and (e) that the pattern distances can give an accurate recognition result; however, the trajectory distances provide a more robust performance, especially its accumulated effects (i.e. discriminating ability) over time.

This is a quadratic optimisation problem which can be solved using the interior point method [21].

5.2. Dynamic face recognition by trajectory matching

For an unknown face pattern $(x, y, \mathbf{z}_0)$ where $\mathbf{z}_0$ is the KDA vector and x, y are the pose in tilt and yaw, one can classify this pattern into one of the known face classes by computing the distance to each of the identity surfaces as the Euclidean distance[2] between $\mathbf{z}_0$ and the corresponding point on the identity surface $\mathbf{z}$

$$d = \|\mathbf{z}_0 - \mathbf{z}\| \tag{33}$$

where $\mathbf{z}$ is given by Eq. (30).

As shown in Fig. 5, when a face is tracked continuously in a video sequence using the multi-view dynamic face model described in Section 3, an object trajectory is obtained by projecting the face patterns into the KDA feature space. On the other hand, according to the pose information of the face patterns, one can build the model trajectory on the identity surface of each subject using the same pose information and temporal order of the object trajectory. Those two kinds of trajectories, i.e. object and model trajectories, encode the spatio-temporal information of the tracked face. And finally, the recognition problem can be solved by matching the object trajectory to a set of model trajectories. A preliminary realisation of trajectory matching is implemented by computing the trajectory distances up to time slice t

$$d_m = \sum_{i=1}^{t} w_i d_{mi} \tag{34}$$

where d_{mi}, the pattern distance between the face pattern captured in the ith frame and the identity surface of the mth subject, is computed from Eq. (33), and w_i is the weight on this distance. Finally, the optimal m with minimum d_m is chosen as the recognition result.

6. Experiments

We demonstrate the performance of this approach on a small scale multi-view face recognition problem. Twelve sequences, one of each subject, were used as training sequences. The sequence length varies from 40 to 140 frames. We randomly selected 180 images (15 images of each subject) to train KDA. Then recognition was performed on new test sequences of these subjects.

Fig. 6 shows the results on one of the test sequences. It is noted that a more robust performance is achieved when recognition is carried out using the trajectory distances which include the accumulated evidence over time, although the pattern distances in each individual frame already provides good recognition accuracy on a frame by frame basis.

To compare with KDA, we applied the PCA, KPCA, and LDA techniques using the same set of face patterns. To make the results of different representations comparable, we define the following criterion

$$d' = \frac{1}{N} \sum_{i=1}^{N} \frac{C d_{i0}}{\sum_{j=1}^{C} d_{ij}} \tag{35}$$

where C is the number of face classes, N is the total number of test face patterns, d_{ij} is the pattern distance between the ith test pattern and the jth face class, and d_{i0} is the pattern distance between the ith test pattern and the ground-truth face class.

Criterion d' can be interpreted as a summation of normalised pattern distances to their ground-truth face class. The smaller the d', the more reliable the classification. Fig. 7 shows the values of d' for different representations, PCA, KPCA, LDA and KDA, with respect to the dimension of the feature spaces. The results indicate that KDA gives the most reliable classification performance.

The recognition accuracies with respect to the dimension of feature spaces are shown in Fig. 8. It is interesting to note that the KDA features are very efficient. A 93.9% recognition accuracy was achieved when the dimension of the KDA vector was only 2. It is also observed that, for the small scale problem (12 subjects), all the methods except for KPCA achieved a 100% recognition accuracy when the dimension of features is not less than 6. We will investigate how these techniques perform on large scale problems in future work.

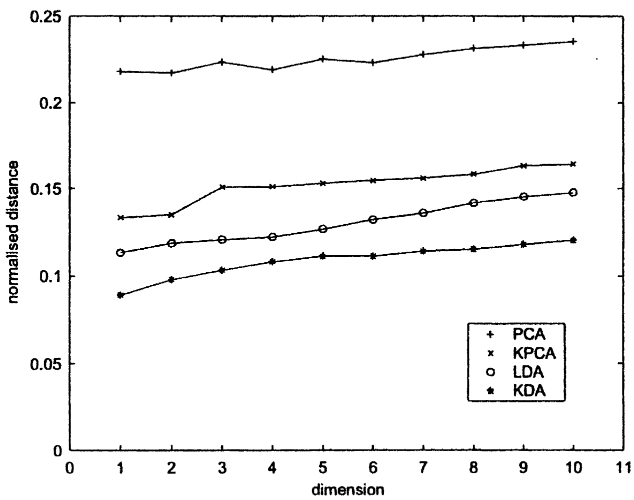


Fig. 7. Recognition reliability.

[2] It is important to note that Euclidean distance is more appropriate for KDA and LDA while Mahalanobis distance is more efficient for PCA and KPCA since the discriminating features are extracted in the former case and the general variation is concerned in the latter.

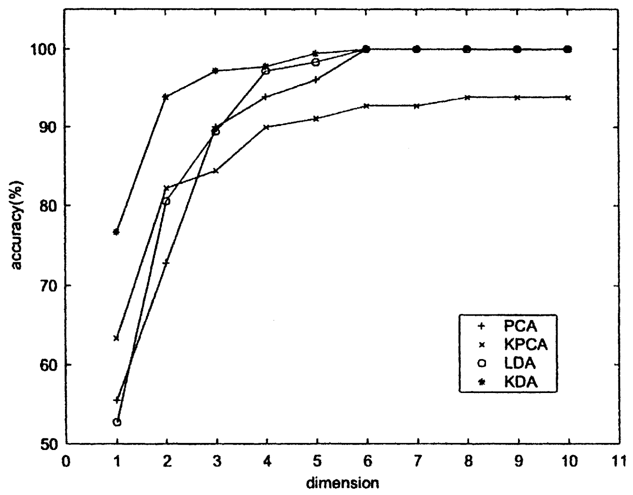


Fig. 8. Recognition accuracy.

7. Conclusions

In this paper, we have presented a comprehensive approach to multi-view dynamic face recognition. This approach is designed to address three challenging problems: modelling faces across multi-views, extracting nonlinear discriminating features, and recognising moving faces dynamically in image sequences.

To model faces with large pose variation, we developed a dynamic face model, which includes a 3D PDM, a shape-and-pose-free texture model, and an affine geometrical model. By representing faces with the shape-and-pose-free texture patterns, the variance from pose change is suppressed.

PCA, LDA and KPCA have been widely used in face recognition. But PCA and LDA are limited to the linear applications while KPCA intends to capture the overall rather than the discriminating variance of patterns even though it is nonlinear. To efficiently extract the discriminating features of multi-class patterns with severe nonlinearity, KDA, which implicitly performs LDA in a nonlinear feature space through a kernel function, is developed in this work. When applying KDA to the shape-and-pose-free texture patterns, the variance from different identities is emphasised and the variance from other sources is further reduced.

Instead of matching templates or estimating multi-modal density, the identity surfaces of face classes are constructed in a discriminating feature space. Recognition is then performed dynamically by matching an object trajectory tracked from a video input with a set of model trajectories synthesised on the identity surfaces. Experimental results showed that this approach provided robust and accurate recognition.

References

[1] G. Baudat, F. Anouar, Generalized discriminant analysis using a kernel approach, Neural Computation 12 (2000) 2385–2404.

[2] V. Bruce, A. Burton, P. Hancock, Comparisons between human and computer recognition of faces, in: IEEE International Conference on Automatic Face and Gesture Recognition, Nara, Japan, 1998, pp. 408–413.

[3] C.J.C. Burges, Simplified support vector decision rules, in: L. Saitta (Ed.), Proceedings, 13th International Conference on Machine Learning, Morgan Kaufmann, San Mateo, CA, 1996, pp. 71–77.

[4] C.J.C. Burges, B. Scholkopf, Improving the accuracy and speed of support vector learning machines, in: M. Mozer, M. Jordan, T. Petsche (Eds.), Advances in Neural Information Processing Systems 9, MIT Press, Cambridge, MA, 1997, pp. 375–381.

[5] T. Cootes, G. Edwards, C. Taylor, Active appearance models, in: European Conference on Computer Vision, Freiburg, Germany, vol. 2, 1998, pp. 484–498.

[6] T. Cootes, C. Taylor, D. Cooper, J. Graham, Active shape models-their training and application, Computer Vision and Image Understanding 61 (1) (1995) 38–59.

[7] G. Edwards, C. Taylor, T. Cootes, Learning to identify and track faces in sequences, in: IEEE International Conference on Automatic Face and Gesture Recognition, Nara, Japan, 1998, pp. 260–267.

[8] K. Fukunaga, Introduction to Statistical Pattern Recognition, Academic Press, New York, 1972.

[9] S. Gong, S. McKenna, A. Psarrou, Dynamic Vision: From Images to Face Recognition, World Scientific Publishing/Imperial College Press, 2000.

[10] S. Gong, A. Psarrou, I. Katsouli, P. Palavouzis, Tracking and recognition of face sequences, in: European Workshop on Combined Real and Synthetic Image Processing for Broadcast and Video Production, Hamburg, Germany, 1994, pp. 96–112.

[11] B. Knight, A. Johnston, The role of movement in face recognition, Visual Cognition 4 (1997) 265–274.

[12] M. Lades, J. Vorbruggen, J. Buhmann, J. Lange, C. Malsburg, R. Wurtz, W. Konen, Distortion invariant object recognition in the dynamic link architecture, IEEE Transactions on Computers 42 (3) (1993) 300–311.

[13] Y. Li, S. Gong, H. Liddell, Modelling faces dynamically across views and over time, in: IEEE International Conference on Computer Vision, Vancouver, Canada, 2001, pp. 554–559.

[14] S. Mika, G. Ratsch, J. Weston, B. Scholkopf, K. Muller, Fisher discriminant analysis with kernels, in: IEEE Neural Networks for Signal Processing Workshop (1999) 41–48.

[15] V. Roth, V. Steinhage, Nonlinear discriminant analysis using kernel functions, in: S. Solla, T. Leen, K.-R. Müller (Eds.), Advances in Neural Information Processing Systems 12, MIT Press, Cambridge, MA, 1999, pp. 568–574.

[16] B. Scholkopf, Support Vector Learning, R. Oldenbourg Verlag, Munich, 1997.

[17] B. Scholkopf, A. Smola, K.-R. Muller, Kernel principal component analysis, in: W. Gerstner, A. Germond, M. Hasler, J.-D. Nicoud (Eds.), Artificial Neural Networks—ICANN'97, Lecture Notes in Computer Science, Springer, Berlin, 1997, pp. 583–588.

[18] L. Sirovich, M. Kirby, Low-dimensional procedure for the characterization of human faces, Journal of Optical Society of America 4 (1987) 519–524.

[19] D. Swets, J. Weng, Using discriminant eigenfeatures for image retrieval, IEEE Transactions on Pattern Analysis and Machine Intelligence 18 (8) (1996) 831–836.

[20] M. Turk, A. Pentland, Eigenfaces for recognition, Journal of Cognitive Neuroscience 3 (1) (1991) 71–86.

[21] R. Vanderbei, Loqo: an interior point code for quadratic programming, Technical report, Princeton University, Technical Report SOR 94-15, 1994.

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

[23] V. Vapnik, Statistical Learning Theory, Wiley, New York, 1998.

[24] T. Vetter, T. Poggio, Linear object classes and image synthesis from a single example image, IEEE Transactions on Patternx Analysis and Machine Intelligence 19 (7) (1997) 733–742.