Constructing Structures of Facial Identities on the View Sphere Using Kernel Discriminant Analysis

Yongmin Li, Shaogang Gong and Heather Liddell
Department of Computer Science
Queen Mary, University of London
London E1 4NS, UK
Email: {yongmin,sgg,heather}@dcs.qmw.ac.uk

Abstract

We present a novel approach to face recognition by constructing facial identity structures across views and over time, referred to as *identity surfaces*, in a Kernel Discriminant Analysis (KDA) feature space. This approach is aimed at addressing three challenging problems in face recognition: extracting the non-linear discriminating features, recognising faces across multiple views, and recognising moving faces over time. First, the KDA is developed to compute the most significant non-linear basis vectors with the intention of maximising the between-class variance and minimising the within-class variance. We applied KDA to the problem of multi-view face recognition, and a significant improvement has been achieved in robustness and accuracy. Second, *identity surfaces* are constructed to model the variance of facial appearance caused from rotation in depth. Recognition can then be conveniently performed by computing the pattern distances from the *identity surfaces*. Third, video-based online face recognition is performed by computing and matching object onto *identity surfaces* which encode the spatio-temporal dynamics of moving faces.

1 Introduction

Face recognition, as an important visual perceptual task, has been of great interest in recent years both theoretically and practically for applications including integrated surveillance, visually mediated interaction, human-machine interface, multimedia and teleconferencing.

Various approaches have been proposed to address the problem under different assumptions and conditions.

Template based methods were adopted in many previous studies. Early work by Baron [1] presented a Neural Network based approach to face recognition using raw images as system input. Recognition was performed based on the correlation of the resulting sequence of patterns with all model patterns. Brunelli and Poggio [5] presented and compared a geometrical feature based algorithm and a template based algorithm. They claimed that the results obtained for the testing sets show about 90% correct recognition using geometrical features and perfect recognition using template matching. In their subsequent research, Sung and Poggio [27] generated 6 face prototypes and 6 near-face-nonface prototypes as templates to match a new image pattern. A well-tuned Neural Network was employed to synthesise these matching results. Another approach using Support Vector Machines (SVMs) was presented by Osuna and Poggio, by which the most representative examples, known as Support Vectors, are extracted automatically [21, 22].

The high dimensionality of raw images is a problem in computation. To address this, Principal Component Analysis (PCA) has been widely adopted to reduce dimensionality and extract abstract features of faces. Sirovich and Kirby [26] first used PCA, also known as the Karhunen-Loeve transform, for face representation. Turk and Pentland [29] proposed the eigenface method which used a similar method to code face images and capture face features. PCA has also been used extensively in other face models such as the the Active Shape Model (ASM) and Active Appearance Model (AAM) [9, 8].

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 [12], proved to be a more suitable technique for class separation. Computationally, LDA can be solved as an eigen-decomposition problem similar to PCA. Swets and Weng [28] applied a subsequent LDA projection followed by PCA to derive the *Most Discriminating Features*. Zhao *et al.* [34] used LDA as a representation for frontal-view face recognition. Edwards *et al.* [11] adopted LDA to select *discriminant parameters* based on Active Appearance Models. They argued that these parameters can be used to decouple identity variance from pose, lighting and expression variance.

Though LDA can provide a significant discriminant improvement to the task of face recognition, it is still a linear technique in nature. When severe non-linearity is involved, this method is intrinsically poor. Another shortcoming of LDA lies in the fact that the number of basis vectors are limited by the number of face classes, therefore it would be less representative when small set of subjects are concerned. To extract the non-linear principal components, Kernel PCA (KPCA) was developed for pattern recognition [25]. Romdhani *et al.* [23] adopted KPCA to construct a nonlinear models aiming at corresponding dynamic appearances of both shape and texture across views. However, as with PCA, KPCA captures the *overall* variance of all patterns which are inadequate for discriminating purposes.

It is important to point out that most of the previous work in face recognition is mainly concerned with frontal-view. However, recognising faces across views is more challenging than that at a fixed view, e.g. frontal view, because of the severe non-linearity caused by rotation in depth, self-occlusion, self-shading and illumination change. The eigenface method has been extended to view-based and modular eigenspaces with the intention of recognising faces under varying views by Moghaddam and Pentland [20]. Li *et al.* [16] presented a view-based piece-wise SVM model of the face space. Cootes *et al.* [10] proposed the view-based Active Appearance Models which employ three models for profile, half-profile and frontal views. But the division of the face space in these methods is rather arbitrary, ad hoc and often coarse.

Another limitation of the previous work is that the methods proposed for recognition are 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). Knight and Johnston [14] showed that recognition of famous faces in photographic negatives can be significantly enhanced when the faces were shown moving rather than static. Bruce *et al.* [3, 4] extended this result to other conditions where recognition is made difficult, e.g. by thresholding the images or showing them in blurred or pixellated formats. Though some preliminary results obtained from techniques such as the temporal signature method [13] , the subspace method [33] and the identity trajectory method [15], have been reported, the issue of recognising the dynamics of faces under a spatio-temporal context remains largely unresolved.

In this paper, we present a novel and comprehensive approach to modelling facial identities across views and over time. To remedy the linearity limitation of LDA, Kernel Discriminant Analysis (KDA) is adopted, which employs the kernel technique to maximise the between-class variance and minimise the within-class variance. To address the multi-view face recognition problem, a spatio-temporal *identity surface* of each face class is constructed in a kernel discriminating feature space. A video-based approach using pattern distances and trajectory distances to the *identity surfaces* is presented to perform online face recognition dynamically. The rest part of this paper is arranged as follows: The KDA method is introduced in Section 2, then the issue of modelling faces using KDA is discussed in Section 3. *Identity surface* construction and performance evaluation are presented in Section 4, while Section 5 describes the approach to video-based face recognition. Experimental results are given in Section 6 and conclusions in Section 7.

2 Kernel Discriminant Analysis

As stated in the previous section, 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, the Kernel Discriminant Analysis, a nonlinear discriminating approach based on the kernel technique [25, 31, 32, 19, 2], is developed for extracting the nonlinear discriminating features.

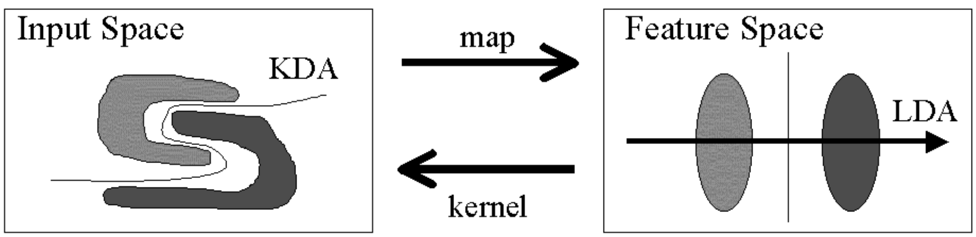


Figure 1: Kernel Discriminant Analysis.

The principle of KDA can be illustrated in Figure 1. It is difficult to directly compute the discriminating features between the two classes of patterns because of the severe non-linearity. By defining a non-linear map from the input space to a high-dimensional feature space, one obtains a linearly separable distribution in the feature space. Then LDA, the linear technique, can be performed in the feature space to extract the most significant discriminating features. However, the computation may be problematic or even impossible due to the high dimension. By introducing a kernel function, all the computation can be carried out in the input space conveniently.

2.1 Centred Data

For a set of training patterns $\{\boldsymbol{x}\}$ which are categorised into C classes, $\boldsymbol{\phi}$ is defined as a non-linear map from the input space to a high-dimensional feature space. By this map one assumes that an original nonlinear problem in the input space can be transformed to a linear problem in the high-dimensional feature space and solved using the regular linear techniques. If the map $\boldsymbol{\phi}$ satisfies the Mercer's condition [31, 32], then the inner product of two vectors in the feature space can be calculated through a kernel function

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

which can be conveniently computed in the input space.

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

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

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

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

$$\boldsymbol{S}_w = \frac{1}{C} \sum_{c=1}^{C} \frac{1}{N_c} \sum_{i=1}^{N_c} \boldsymbol{\phi}_i \boldsymbol{\phi}_i^{\mathrm{T}} \tag{4}$$

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

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

Assuming $\boldsymbol{S}_w$ is not singular, one 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 matrix

$$\boldsymbol{S} = \boldsymbol{S}_w^{-1} \boldsymbol{S}_b \tag{6}$$

Assuming $\boldsymbol{v}$ is one of the eigenvectors of matrix $\boldsymbol{S}$, and λ is its corresponding eigenvalue, i.e.

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

Combining (6) and (7),

$$\boldsymbol{S}_b \boldsymbol{v} = \lambda \boldsymbol{S}_w \boldsymbol{v} \tag{8}$$

Then taking inner product with vector $\boldsymbol{\phi}_m$ on both sides of equation (8) yields

$$(\boldsymbol{S}_b \boldsymbol{v} \cdot \boldsymbol{\phi}_m) = \lambda (\boldsymbol{S}_w \boldsymbol{v} \cdot \boldsymbol{\phi}_m), m = 1, 2, ..., N \tag{9}$$

A coefficient vector exists

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

that satisfies

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

Substituting (3), (4) and (11) in (9) yields

$$\sum_{n=1}^{N} \alpha_n \sum_{c=1}^{C} \frac{1}{N_c^2} \sum_{i=1}^{Nc} \sum_{j=1}^{Nc} (\boldsymbol{\phi}_{ci} \cdot \boldsymbol{\phi}_m)(\boldsymbol{\phi}_{cj} \cdot \boldsymbol{\phi}_n) = \lambda \sum_{n=1}^{N} \alpha_n \sum_{c=1}^{C} \frac{1}{N_c} \sum_{i=1}^{N_c} (\boldsymbol{\phi}_{ci} \cdot \boldsymbol{\phi}_m)(\boldsymbol{\phi}_{ci} \cdot \boldsymbol{\phi}_n) \tag{12}$$

Defining a $N \times N_c$ matrix $\boldsymbol{K}_c$ as

$$(\boldsymbol{K}_c)_{ij} := (\boldsymbol{\phi}_i \cdot \boldsymbol{\phi}_j) = k_{ij} \tag{13}$$

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

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

one obtains

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

Defining $N \times N$ matrix as

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

one derives

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

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

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

where

$$\boldsymbol{k}_x = (k(\boldsymbol{x}, \boldsymbol{x}_1), k(\boldsymbol{x}, \boldsymbol{x}_2), ..., k(\boldsymbol{x}, \boldsymbol{x}_N))^{\mathrm{T}} \tag{19}$$

Constructing the eigen matrix

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

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

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

2.2 Non-centred Data

In the general case, $\{\boldsymbol{\phi}(\boldsymbol{x}_i)\}, i = 1, 2, ..., N$, are not centred. A similar method to [24] is adapted here. By defining

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

one can use the method stated above since $\{\boldsymbol{\phi}_i\}, i = 1, 2, ..., N$ are now centred. The kernel matrix $\tilde{\boldsymbol{K}}_c$ can then be expressed by its non-centred counterpart $\boldsymbol{K}$ as follows:

$$\begin{aligned}
(\tilde{\boldsymbol{K}}_c)_{ij} &= (\tilde{\boldsymbol{\phi}}_i \cdot \tilde{\boldsymbol{\phi}}_j) \\
&= \left(\boldsymbol{\phi}_i - \frac{1}{N}\sum_{m=1}^{N}\boldsymbol{\phi}_m\right) \cdot \left(\boldsymbol{\phi}_j - \frac{1}{N}\sum_{n=1}^{N}\boldsymbol{\phi}_n\right) \\
&= (\boldsymbol{\phi}_i \cdot \boldsymbol{\phi}_j) - \frac{1}{N}\sum_{m=1}^{N}(\boldsymbol{\phi}_m \cdot \boldsymbol{\phi}_j) - \frac{1}{N}\sum_{n=1}^{N}(\boldsymbol{\phi}_i \cdot \boldsymbol{\phi}_n) \\
&\quad + \frac{1}{N^2}\sum_{m=1}^{N}\sum_{n=1}^{N}(\boldsymbol{\phi}_m \cdot \boldsymbol{\phi}_n) \\
&= k_{ij} - \frac{1}{N}\sum_{m=1}^{N}k_{mj} - \frac{1}{N}\sum_{n=1}^{N}k_{in} + \frac{1}{N^2}\sum_{m=1}^{N}\sum_{n=1}^{N}k_{mn}
\end{aligned} \tag{23}$$

Using $N \times N$ matrix $(\boldsymbol{K})_{ij} := k_{ij}$ and $\mathbf{1}_N$, one obtains

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

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

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

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

where

$$\begin{aligned}
(\tilde{\boldsymbol{k}}_x)_i &= \left(\boldsymbol{\phi}(\boldsymbol{x}) - \frac{1}{N}\sum_{m=1}^{N}\boldsymbol{\phi}(\boldsymbol{x}_m)\right)\left(\boldsymbol{\phi}(\boldsymbol{x}_i) - \frac{1}{N}\sum_{n=1}^{N}\boldsymbol{\phi}(\boldsymbol{x}_n)\right) \\
&= k(\boldsymbol{x}, \boldsymbol{x}_i) - \frac{1}{N}\sum_{m=1}^{N}k(\boldsymbol{x}_i, \boldsymbol{x}_m) - \frac{1}{N}\sum_{n=1}^{N}k(\boldsymbol{x}, \boldsymbol{x}_n) \\
&\quad + \frac{1}{N^2}\sum_{m=1}^{N}\sum_{n=1}^{N}k(\boldsymbol{x}_m, \boldsymbol{x}_n)
\end{aligned} \tag{26}$$

Defining an $N \times 1$ vector $\mathbf{1}'$ with all entries equal to 1, one obtains

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

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

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

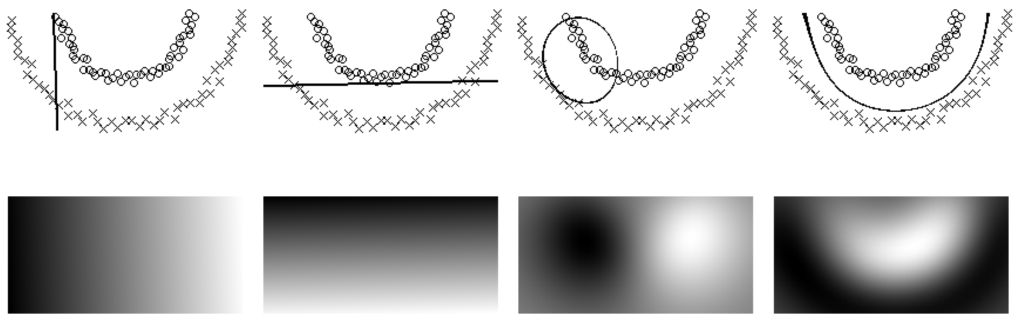


Figure 2: Solving a nonlinear classification problem with, from left to right, PCA, LDA, KPCA and KDA.

We use a toy problem to illustrate the characteristics of KDA as shown in Figure 2. Two classes of patterns denoted with circles and crosses respectively have a significant non-linear 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 value of the one-dimensional feature 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 variation though 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.

3 Modelling Multi-view Faces Using KDA

Due to the severe non-linearity 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 view, e.g. frontal view. Another difficulty of multi-view face recognition is that the appearances of different people from the same view are more similar than those of the same person from different views.

3.1 Constructing ***Shape-and-Pose-Free*** Textures

A multi-view dynamic face model, which consists of a sparse 3D Point Distribution Model (PDM) [9], a *shape-and-pose-free* texture model, and an affine geometrical model, is adopted to extract the *shape-and-pose-free* texture patterns for multi-view face recognition.

The 3D shape vector of a face is estimated from a set of 2D face images labelled with pose and 2D positions of a set of salient landmarks. To decouple the covariance between shape and texture, 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. 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. We applied PCA to the 3D shape patterns and *shape-and-pose-free* texture patterns respectively to obtain a low dimensional statistical model.

Figure 3 shows the sample face images used to construct the model, the 3D shape meshes overlaid on the images, and the extracted *shape-and-pose-free* texture patterns. [1]

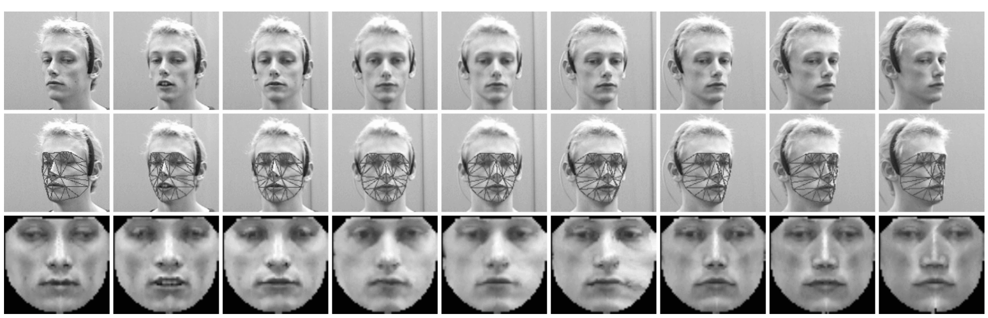


Figure 3: Extract the *shape-and-pose-free* texture patterns of multi-view face images using a multi-view dynamic face model.

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.

[1] When one side of a face becomes partially invisible, the texture pattern is constructed from the visible side using the bilateral symmetry of 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 $\mathbf{c} = (\mathbf{s}, \mathbf{t}, \alpha, \beta, dx, dy, r)^{\mathrm{T}}$ where $\mathbf{s}$ is the shape parameter, $\mathbf{t}$ is the texture parameter, (α, β) is pose in tilt and yaw, (dx, dy) is the translation of the centroid of the face, and r is its scale. More details of model construction and fitting are described in [17].

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.

3.2 Variation from Face Classes vs. Variation from Pose

Although the *shape-and-pose-free* facial texture patterns from different views may be more similar than their original forms, the underlying discriminating features for different face classes have not been represented explicitly. Therefore such a representation in itself would not be efficient for recognition. To illustrate this problem, we plot the *shape-and-pose-free* face texture patterns in the PCA space in Figure 4. For the sake of conciseness, only the patterns of four face classes are shown here. Figure 4(a) illustrates the variation of the first PCA dimension with respect to the pose change. The horizontal axis gives the index number of images in the order of $-20^\circ \sim +20^\circ$ in tilt and $-40^\circ \sim +40^\circ$ in yaw. The orders are identical for all face classes. The patterns belonging to a same face class are linked together. Figure 4(b) shows the distribution of the texture patterns in the first two PCA dimension. It is noted that the variation from different face classes is not efficiently separated from that for pose change, or more precisely, the former is even overwhelmed by the latter.

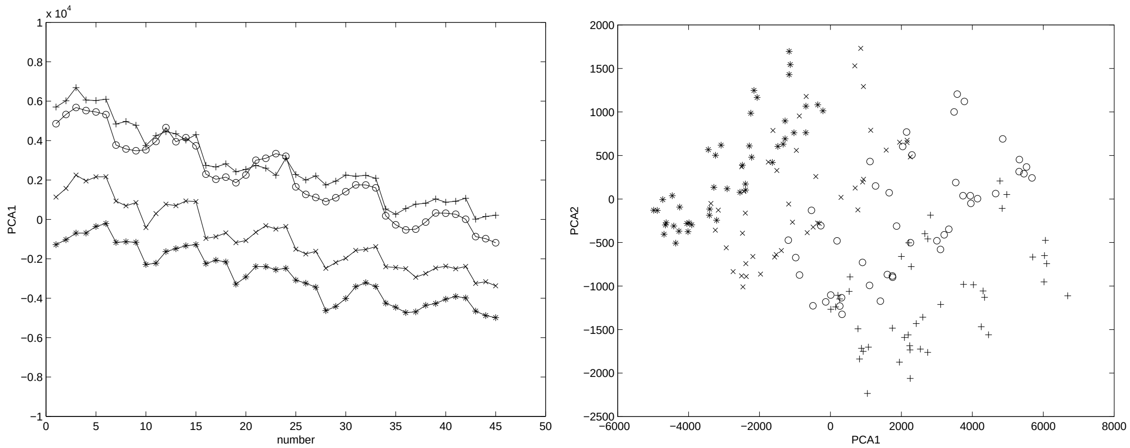
3.3 Extracting the KDA Features of Faces

We apply KDA to the *shape-and-pose-free* face patterns of the same face classes as shown in Figure 4. The *shape-and-pose-free* patterns are normalised to unit vectors before KDA. The Gaussian kernel is adopted,

$$k(\boldsymbol{x}, \boldsymbol{y}) = exp(-\frac{(\boldsymbol{x} - \boldsymbol{y})^2}{2\sigma^2}) \tag{29}$$

where $2\sigma^2 = 1$. [2]

[2] It is indicated in our experiments, including SVM based pose estimation [16], multi-view face detection [16, 18], and KDA based multi-view face recognition presented in this paper, that a satisfactory result is



(a) variation of the 1st PCA dimension with respect to pose change.

(b) pattern distribution in the first two PCA dimensions

Figure 4: Face class separability under multiple views: Variation from different face classes vs. variation from pose change. The horizontal axis in (a) gives the index number of pose changing between $[-20°, +20°]$ in tilt and $[-40°, +40°]$ in yaw.

The variation and distribution of the patterns are shown in Figure 5(a) and 5(b) respectively. Compared to the results of the PCA patterns in Figure 4, the improvement on class separability is significant. It is worth pointing out that such separability is achieved by using only two KDA dimensions.

4 Recognising Faces Using Identity Surfaces

One of the most commonly used techniques for recognition is to compute the probabilities of a set of known patterns or the similarities among templates of different classes before selecting the optimal value using a simple metric. For example, the Euclidean distance or the Mahalanobis distance can be adopted if the pattern distribution of each class is compact enough and separable from others. However, usually this simplistic method cannot provide satisfactory solutions to the problem of multi-view face recognition. The reasons are twofold: First, the representation adopted, e.g. the KDA, may not generate a *perfectly* compact distribution of each face class while separates one from another. Second, the

usually achieved when patterns are normalised to unit vectors and the kernel parameter is chosen as $2\sigma^2 = 1$.

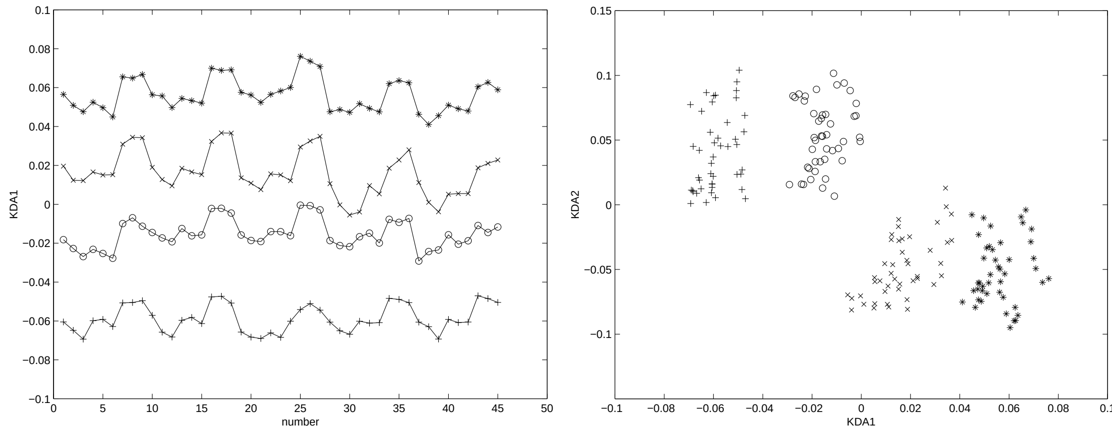


Figure 5: Distribution of the KDA patterns obtained from the same face images as in Figure 4.

distributions of each class cannot be guaranteed to be homogeneous.

When the distribution is irregular, the traditional statistical method for dealing with this problem is to estimate a multi-modal density function for each class. But a very large number of training examples are needed either for parametric or non-parametric modelling. In this work, we do not constrain ourselves to such a strict condition. Instead, we present a novel approach to construct an *identity surface* of each face class from a sparse sample of multi-view face patterns.

4.1 Identity Surfaces

As stated above, one of the key problems of multi-view face recognition is how to separate two kinds of variations: variation from different subjects and variation from pose. Observing the results presented in Figure 4(a) and Figure 5(a), we find that the features from different face classes share a similar variation tendency with respect to pose change. It suggests that a significant improvement to face identity modelling can be expected if the pose information is exploited explicitly. Based on this idea, we developed a method of multi-view face recognition using *identity surfaces*.

Assuming that only the appearance variation caused by rotation in depth is concerned, i.e. the variation from expression, illumination and facial make-up is excluded, each face class can be represented by a unique hyper surface based on the 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 features of faces, e.g. the KDA vectors. For each pair of tilt and yaw values, there is one unique "point" for a face class. The distribution of all the "points" of the same face class with regard to pose form a hyper surface in the space spanned by the discriminating features and pose. We call this surface an *identity surface*. Then face recognition can be performed by computing and comparing the distances between a given pattern and a set of *identity surfaces*.

4.2 Constructing Identity Surfaces of Faces

If sufficient patterns of a face class in different views are available, the *identity surface* of this face class can be constructed precisely. However, we do not require such a strict condition. In this work, we develop a method to synthesise the *identity surface* of a face class 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, $\boldsymbol{z}$ is the discriminating feature vector of a face pattern, e.g. a KDA vector. A list $(x_{01}, y_{01}), (x_{02}, y_{02}), ..., (x_{0N_v}, y_{0N_v})$ gives predefined views which discretise the view sphere into N_p grids. On each grid, the *identity surface* of a face class is approximated by a plane

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

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

$$\begin{aligned}
\mathcal{Q} &= \sum_i^{N_p}\sum_m^{M_i} \|\boldsymbol{a}_i x_{im} + \boldsymbol{b}_i y_{im} + \boldsymbol{c}_i - \boldsymbol{z}_{im}\|^2 && (31)\\
\text{subject to} &: \boldsymbol{a}_i x_{0k} + \boldsymbol{b}_i y_{0k} + \boldsymbol{c}_i = \boldsymbol{a}_j x_{0k} + \boldsymbol{b}_j y_{0k} + \boldsymbol{c}_j \\
& \quad k = 0, 1, ..., N_v, \\
& \quad \text{plane } i, j \text{ intersect at } (x_{0k}, y_{0k}). && (32)
\end{aligned}$$

This is a quadratic optimisation problem which can be solved using the interior point method [30].

Figure 6 shows a real identity surface of a face class using 45 example views ($-20° \sim +20°$ in tilt and $-40° \sim +40°$ in yaw with an interval of $10°$) and the synthesised identity surface using only 15 example views (same ranges but with an interval of $20°$).

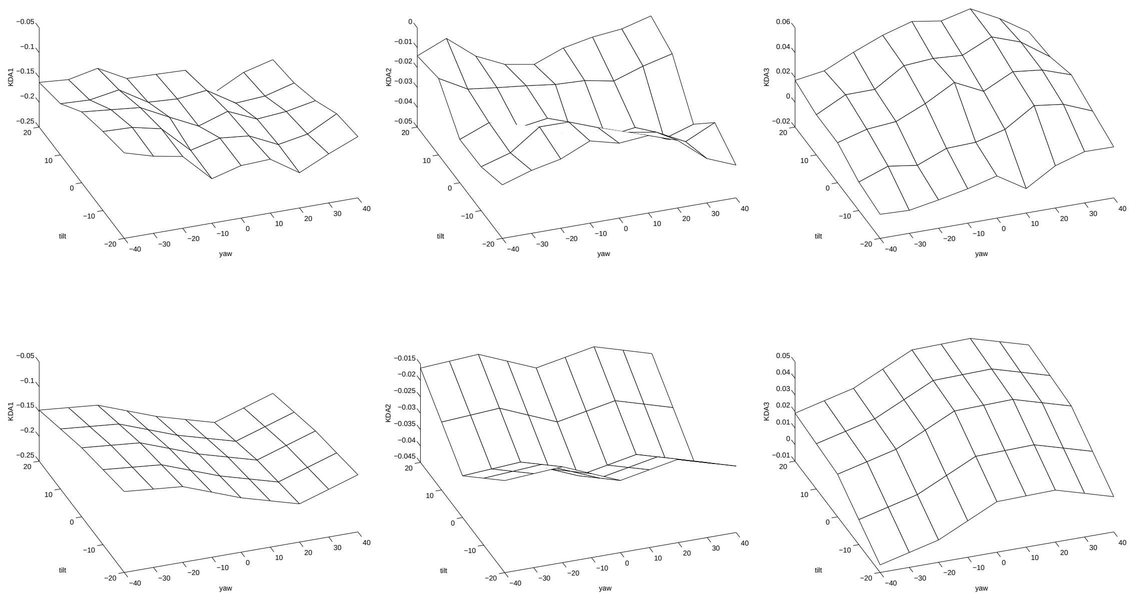


Figure 6: The identity surface constructed from all 45 views (first row) and that synthesised from 15 prototype patterns (second row). Only the first three KDA components are shown here.

4.3 Recognition by Pattern Distances to the *Identity Surfaces*

For an unknown face image, one first fits the multi-view dynamical face model [17] onto the image and projects onto the KDA feature space to yield a face pattern $(x, y, \boldsymbol{z}_0)$ where $\boldsymbol{z}_0$ is the KDA vector and x, y are the pose in tilt and yaw, then the pattern distance to one of the *identity surfaces* can be computed as the Euclidean distance between $\boldsymbol{z}_0$ and the corresponding point $\boldsymbol{z}$ on the *identity surface*

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

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

It is important to note that the Euclidean distance may be more appropriate for LDA or KDA while the Mahalanobis distance is more efficient when PCA or KPCA is adopted. This is because that the discriminating feature is crucial in the former case while the general variation of all patterns is concerned in the latter.

We constructed the *identity surfaces* of 12 subjects (one of them is shown in Figure 3) in the KDA feature space from 15 views of each subjects, and then performed a test on all face patterns (45 of each subject). The distances of the 45 patterns of the first subject from the 12 *identity surfaces* are shown in Figure 7. In this experiment, the dimension of KDA patterns are chosen as 10. The distances to the ground-truth *identity surface* are highlighted with circles and solid line. It can be seen clearly that all the 45 patterns have the shortest distances to the ground-truth *identity surface*, therefore are recognised correctly.

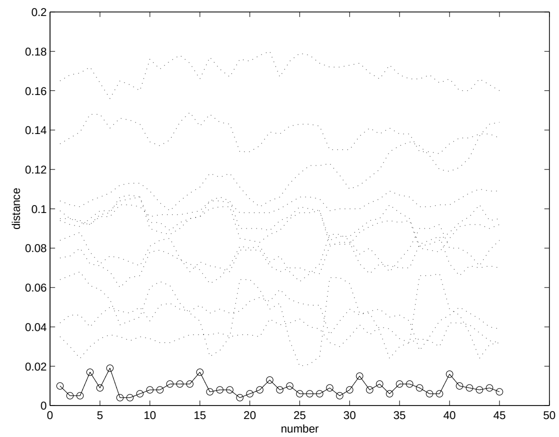


Figure 7: Recognising multi-view faces using distances to the *identity surfaces*. The solid line denotes the results of the ground-truth subject.

4.4 Robustness and Efficiency Analysis

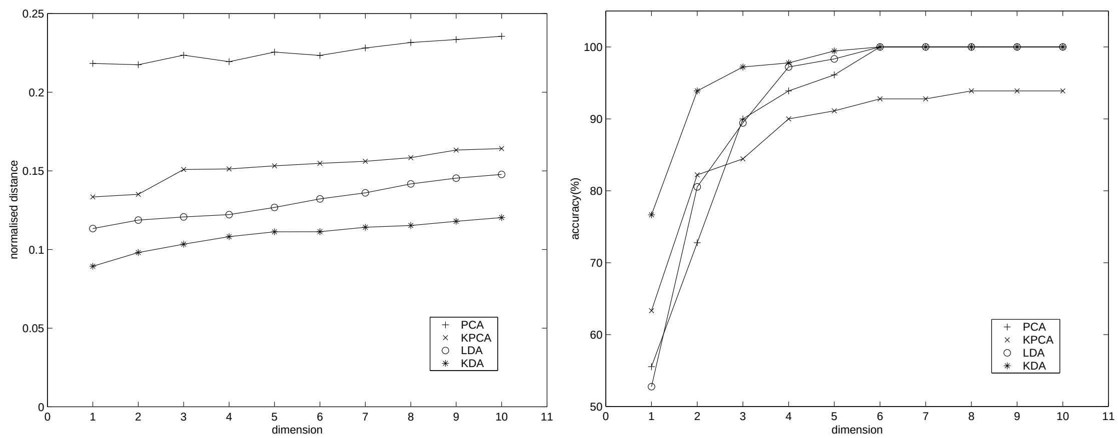
To compare with KDA, we constructed the *identity surfaces* in the PCA, KPCA, and LDA feature spaces using the same set of face patterns. To make the results of different representations comparable, we define the following criterion

$$d' = \sum_{i=1}^{M} \frac{d_{i0}}{\sum_{j=1}^{C} d_{ij}} \tag{34}$$

where C is the number of face classes, M is the total number of test face patterns, d_{ij} is the pattern distance between the ith test pattern and the jth *identity surface*, and d_{i0} is the pattern distance between the ith test pattern and the *identity surface* of the ground-truth face class. d_{ij} and d_{i0} are computed using (33).

Criterion d' can be interpreted as a summation of normalised pattern distances to their ground-truth *identity surfaces* over all test patterns. The smaller the d', the more robust the classification performance. Figure 8(a) 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 robust classification performance.

The recognition accuracies with respect to the dimension of feature spaces are shown in Figure 8(b). 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.



(a) Recognition robustness

(b) Recognition accuracy

Figure 8: Robust analysis.

5 Recognising Faces Dynamically in Video Sequences

Even for the human vision system, the performance of face recognition or facial analysis is not very reliable on *static images*. However, the situation can be improved dramatically when video input containing the faces concerned is available. Recall the description in Section 1, psychological and physiological research suggests that modelling and recognising moving faces dynamically have the potential for achieving a superior performance over that on static images.

5.1 Video-Based Online Face Recognition

We argue that the performance of face recognition for a computer based vision system can be dramatically enhanced if the facial dynamics is modelled in the following aspects:

1. Instead of the method of exhaustive scan, which is notoriously slow, *focus of attention* can be performed efficiently using enriched dynamical information such as motion, colour, and background.

2. Information from individual frames of a video input may be ambiguous, or even controversial. However, the accumulated evidence from all frames can provide a more reliable performance. This is the so-called *identity constancy* principle.

3. It is interesting to note that the human vision system works in an *interactive* rather than an *open-looped* manner. For example, when observing a moving face, we predict the next likely position, pose, and appearance of the face as well as collecting the information at the time being. Then the coincidence or difference between our prediction and observation adjusts the perception we receive about the face. Therefore a reinforced effect is achieved in this *interactive* manner. As for computer based vision systems, an improved performance can be achieved if the model parameters, or even the model itself, is adapted to the observations and measurements dynamically.

5.2 Recognising Faces Dynamically Using Identity Surfaces

For computer based vision systems, the issue of formulating and modelling the facial dynamics described in the previous sections is non-trivial and still largely under-developed. However, significant improvement in terms of recognition accuracy and robustness may still be achieved when the spatio-temporal information is modelled in a rather straightforward way, e.g. simply accumulating the discriminating evidences with the spatio-temporal order encoded in an input sequence. As a practical implementation, we formulate the following approach to video-based online face recognition using *identity surfaces*.

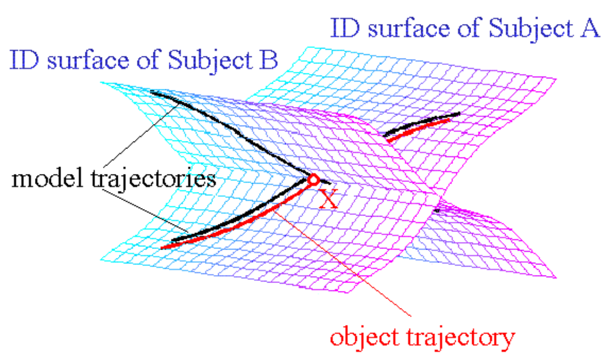


Figure 9: Identity surfaces for face recognition

As shown in Figure 9, when a face is detected and tracked in an input video sequence, one obtains the *object trajectory* of the face in the feature space. Also, its projection on each of the *identity surface* with the same poses and temporal order forms a *model trajectory* of the specific face class. It can be regarded as the ideal trajectory of this face class encoded by the same spatio-temporal information (pose information and temporal order from the video sequence) as the tracked face. Then face recognition can be carried out by matching the object trajectory with a set of model trajectories. Compared to face recognition on static images, this approach can be more robust and accurate. For example, it is difficult to decide whether the pattern X in Figure 9 belongs to subject A or B for a single pattern, however, if we know that X is tracked along the object trajectory, it is very clear that it is more likely to be subject A than B.

A complete process of this video-based face recognition includes:

- **Registration** Construct the *identity surface* for each face class from learning sequences;
- **Tracking** Fit the multi-view dynamic model on an input video sequence containing faces to be recognised, and extract the discriminating features;
- **Recognition** Compute the object and model trajectories and compare these trajectories.

5.3 Constructing Identity Surfaces from Learning Sequences

Before recognition is carried out, a face class should be registered with respect to the system by one or more learning sequences containing the faces of this face class. We record a small video clip of a subject while he/she rotates the head in front of a camera. After applying the multi-view dynamic face model [17] on the video sequence, we obtain a set of face patterns of this subject. Then these patterns are stored to construct the *identity surface* of this face class, and, if necessary, to train (or re-train) the KDA.

To simplify computation, normally we do not use all the patterns of each subject to train the KDA since the sizes of the kernel matrix $\boldsymbol{K}$ and $\boldsymbol{K}_c$ are directly related to the number of training examples. A pragmatic way to select the KDA training patterns is to factor-sample the patterns from the training sequences so that the result patterns uniformly cover the view sphere.

After KDA training, all face patterns can be projected onto the feature space spanned by the significant KDA base vectors. Then the method described in Section 4.2 is employed to construct the *identity surfaces*.

5.4 Trajectory Matching

In the recognition stage, we apply the same multi-view dynamic face model on a novel sequence containing faces to be recognised, then an object trajectory can be obtained by projecting the face patterns into the KDA feature space. On the other hand, the model trajectory can be built 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, recognition is performed by matching the object trajectory to a set of identity model trajectories.

A preliminary realisation of this approach is implemented by computing the trajectory distance

$$d_m = \sum_{i=1}^{t} w_i d_{mi} \tag{35}$$

where d_{mi} is the pattern distance to the *identity surface* of the mth face class in the ith frame computed using (33), and w_i is the weight on this distance. Recognition is performed by selecting the subject with minimum trajectory distance.

6 Experiments

We demonstrate the performance of this approach on a small scale multi-view face recognition problem. Twelve sequences, each from a set of 12 subjects, were used as training sequences to construct the *identity surfaces*. The number of frames contained in each sequence varies from 40 to 140. We randomly selected 180 images (15 images of each subject) to train the KDA. The first ten KDA basis vectors were used to construct the *identity surfaces*. Then recognition was performed on new test sequences of these subjects. Figure 10 shows the sample images fitted by our multi-view dynamic model and the warped *shape-and-pose-free* texture patterns from a test sequence. The object and model trajectories (in the first two KDA dimensions) are shown in Figure 11. The pattern distances from the *identity surfaces* in each individual frame are shown on the left side of Figure 12, while the trajectory distances shown on the right side. These results depict that a more robust performance is achieved when recognition is carried out using the trajectory distances which include the accumulated evidence over time, though the pattern distances to the *identity surfaces* in each individual frame already provides good recognition accuracy on a frame by frame basis.

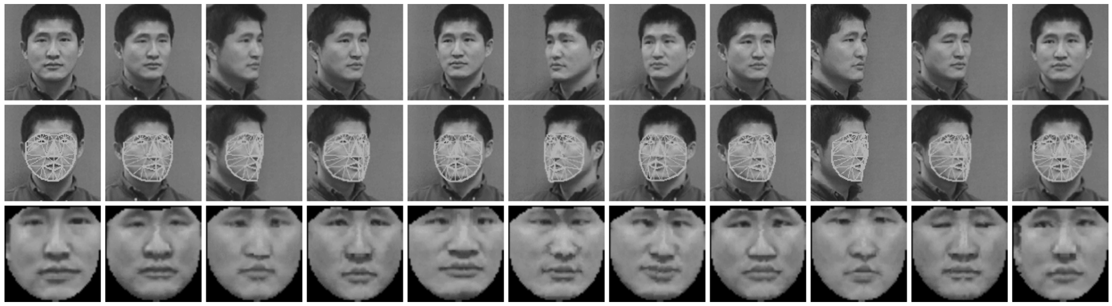


Figure 10: Video-based multi-view face recognition. From top to bottom, sample images from a test sequence with an interval of 10 frames, images fitted by the multi-view dynamic face model, and the *shape-and-pose-free* texture patterns.

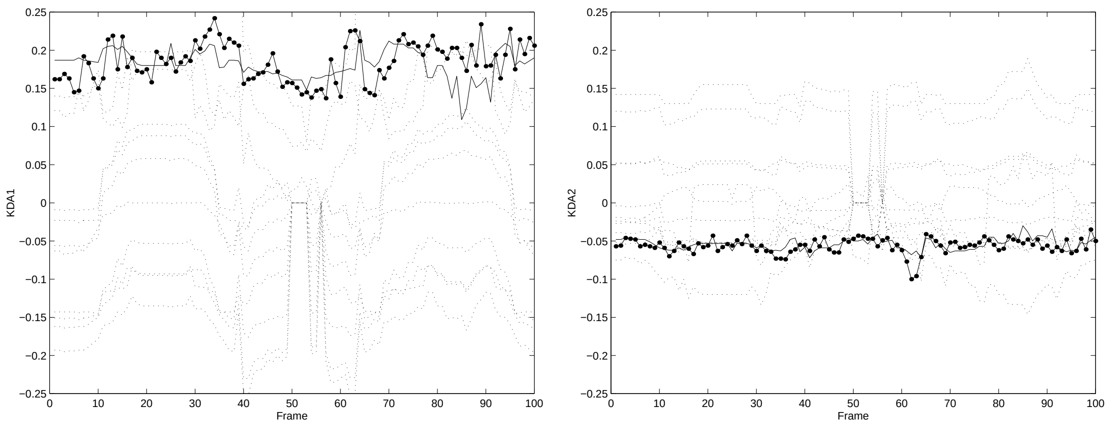


Figure 11: The object and model trajectories in the first two KDA dimensions. The object trajectories are the solid lines with dots denoting the face patterns in each frame. The others are model trajectories where the ones from the ground-truth subject highlighted with solid lines.

7 Conclusions

In this paper, we have presented a comprehensive approach to extract the non-linear discriminating features using KDA and to recognise faces across views and over time dynamically using *identity surfaces* in the KDA feature space. The key issues of this work can be summarised as follows:

1. 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 all patterns though it is non-linear. To

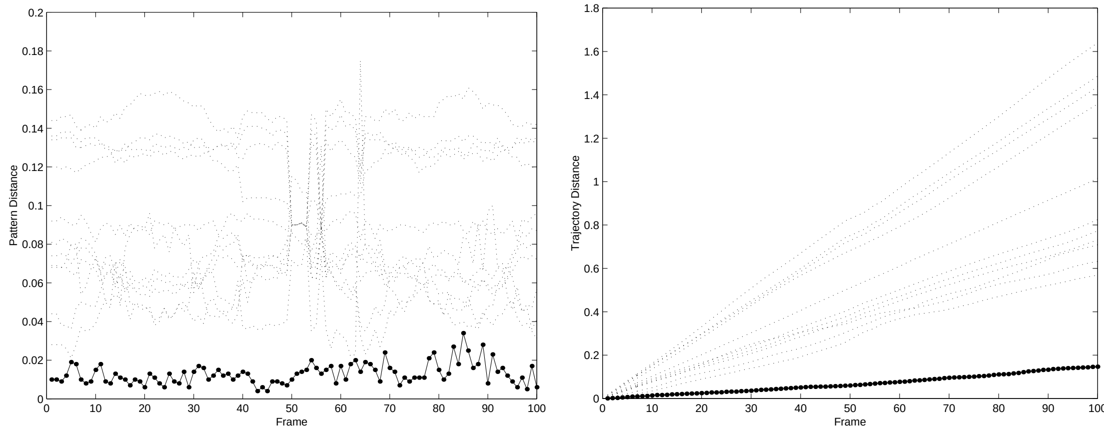


Figure 12: Pattern distances and trajectory distances. The ground-truth subject is highlighted with solid lines. By using KDA and *identity surfaces*, the pattern distances give good recognition for individual frames. However, the trajectory distances provide a more robust performance, especially its accumulated effects (i.e. discriminating ability) over time.

efficiently extract the discriminating features of multi-class patterns with severe non-linearity, the KDA is developed in this work. We applied this method to multi-view face recognition, and significant improvement has been achieved both in robustness and accuracy.

2. Recognising faces across views is more challenging than that from a fixed view because of the severe non-linearity caused by rotation in depth, self-occlusion, self-shading, and illumination change. To model the variance from rotation in depth, we propose the method of *identity surface* which can be constructed from a sparse sample of multi-view face images. Then recognition can be performed by computing the pattern distances or trajectory distances to a set of *identity surfaces*.

3. Psychological and physiological research suggests that modelling and recognising moving faces dynamically have the potential for achieving a superior performance over that on static images. Inspired by this idea, we present an approach to dynamic face recognition by computing and matching the object and model trajectories. A more reliable recognition is achieved since these trajectories encode the spatio-temporal information of a moving face and provide the accumulated evidence of identity.

One of the main drawbacks of this approach is the intensive computation involved in KDA. To obtain the KDA projection of an unknown pattern, one has to compute the kernel functions of this pattern with all training examples. Actually this is a common limitation of all kernel techniques such as KPCA and SVMs. Though some methods such as the reduced set technique [6, 7] can be adopted for computation reduction, an additional non-linear optimisation problem is usually introduced which is not guaranteed to provide a global optimal solution.

In addition, some of the implementation such as trajectory matching is still simplistic in its present form. The trajectory distance is computed as a weighted summation, therefore it does not make any difference to the results of recognition if the information of each frame comes either in a random order or in the temporal order as it being, though the temporal order is still very useful in the tracking process (recall the difference in human vision system described in Section 1). We believe it is an interesting issue for both psychological and artificial vision research to exploit the underlying mechanism of this spatio-temporal dynamics, and extensive further work needs to be conducted.

References

[1] R. Baron. Mechanisms of human facial recognition. *International Journal of Man Machine Studies*, 15:137–178, 1981.

[2] G. Baudat and F. Anouar. Generalized discriminant analysis using a kernel approach. *Neural Computation*, 12:2385–2404, 2000.

[3] V. Bruce, A. Burton, and P. Hancock. Comparisons between human and computer recognition of faces. In *IEEE International Conference on Automatic Face & Gesture Recognition*, pages 408–413, Nara, Japan, 1998.

[4] V. Bruce, P. Hancock, and A.Burton. Human face perception and identification. In Wechsler, Philips, Bruce, Fogelman-Soulie, and Huang, editors, *Face Recognition: From Theory to Applications*, pages 51–72. Springer-Verlag, 1998.

[5] R. Brunelli and T. Poggio. Face recognition: features vs. templates. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 15(10):1042–1062, 1993.

[6] C. J. C. Burges. Simplified support vector decision rules. In L. Saitta, editor, *Proceedings, 13th Intl. Conf. on Machine Learning*, pages 71–77, San Mateo, CA, 1996. Morgan Kaufmann.

[7] C. J. C. Burges and B. Scholkopf. Improving the accuracy and speed of support vector learning machines. In M. Mozer, M. Jordan, and T. Petsche, editors, *Advances in Neural Information Processing Systems 9*, pages 375–381. MIT Press, Cambridge, MA, 1997.

[8] T. Cootes, G. Edwards, and C. Taylor. Active appearance models. In *European Conference on Computer Vision*, volume 2, pages 484–498, Freiburg, Germany, 1998.

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

[10] T. Cootes, K. Walker, and C. Taylor. View-based active appearance models. In *IEEE International Conference on Automatic Face & Gesture Recognition*, pages 227–232, Grenoble, France, 2000.

[11] G. Edwards, A. Lanitis, C. Taylor, and T. Cootes. Statistical models of face images - improving specificity. In *British Machine Vision Conference*, volume 2, pages 765–774, Edinburgh, Scotland, 1996.

[12] K. Fukunaga. *Introduction to statistical pattern recognitiion*. Academic Press, 1972.

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

[14] B. Knight and A. Johnston. The role of movement in face recognition. *Visual Cognition*, 4:265–274, 1997.

[15] Y. Li, S. Gong, and H. Liddell. Recognising the dynamics of faces across multiple views. In *British Machine Vision Conference*, pages 242–251, Bristol, England, September 2000.

[16] Y. Li, S. Gong, and H. Liddell. Support vector regression and classification based multi-view face detection and recognition. In *IEEE International Conference on Automatic Face & Gesture Recognition*, pages 300–305, Grenoble, France, March 2000.

[17] Y. Li, S. Gong, and H. Liddell. Modelling faces dynamically across views and over time. In *IEEE International Conference on Computer Vision*, Vancouver, Canada, July 2001.

[18] Y. Li, S. Gong, J. Sherrah, and H. Liddell. Multi-view face detection using support vector machines and eigenspace modelling. In *The Fourth International Conference on Knowledge-Based Intelligent Engineering System & Allied Technologies*, pages 241–244, Brighton, UK, August 2000.

[19] S. Mika, G. Ratsch, J. Weston, B. Scholkopf, and K. Muller. Fisher discriminant analysis with kernels. In *IEEE Neural Networks for Signal Processing Workshop*, pages 41–48, 1999.

[20] B. Moghaddam and A. Pentland. Face recognition using view-based and modular eigenspaces. In *Automatic Systems for the Identification and Inspection of Humans, SPIE*, volume 2277, 1994.

[21] E. Osuna, R. Freund, and F. Girosi. Support vector machines: Training and applications. Technical report, Massachusetts Institute of Technology, 1997. AI Memo 1602.

[22] E. Osuna, R. Freund, and F. Girosi. Training support vector machines: An application to face detection. In *Proc. Computer Vision and Pattern Recognition'97*, pages 130–136, 1997.

[23] S. Romdhani, S. Gong, and A. Psarrou. On utilising template and feature-based correspondence in multi-view appearance models. In *European Conference on Computer Vision*, volume 1, pages 799–813, Dublin, Ireland, June 2000.

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

[25] B. Scholkopf, A. Smola, and K.-R. Muller. Kernel principal component analysis. In W. Gerstner, A. Germond, M. Hasler, and J.-D. Nicoud, editors, *Artificial Neural Networks – ICANN'97*, pages 583–588, Berlin, 1997. Springer Lecture Notes in Computer Science.

[26] L. Sirovich and M. Kirby. Low-dimensional procedure for the characterization of human faces. *Journal of Optical Society of America*, 4:519–524, 1987.

[27] K. Sung and T. Poggio. Example-based learning for view-based human face detection. Technical report, Massachusetts Institute of Technology, 1994. A.I. MEMO 1521.

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

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

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

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

[32] V. Vapnik. *Statistical Learning Theory.* John Wiley & Sons, 1998.

[33] O. Yamaguchi, K. Fukui, and K. Maeda. Face recognition using temporal image sequence. In *IEEE International Conference on Automatic Face & Gesture Recognition*, pages 318–323, Nara, Japan, 1998.

[34] W. Zhao, A. Krishnaswamy, R. Chellappa, D. Swets, and J. Weng. Discriminant analysis of principal components for face recognition. In Wechsler, Philips, Bruce, Fogelman-Soulie, and Huang, editors, *Face Recognition: From Theory to Applications*, pages 73–85. Springer-Verlag, 1998.