# A STUDY OF APPEARANCE-BASED FEATURE EXTRACTION FOR FACE RECOGNITION

WANG JINGHUA

Ph.D

The Hong Kong Polytechnic University

2013

**The Hong Kong Polytechnic University**

**Department of Computing**

# A Study of Appearance-based Feature Extraction for Face Recognition

WANG Jinghua

A Thesis

Submitted in partial fulfillment of the requirements

for the degree of

Doctor of Philosophy

October 2012

## CERTIFICATE OF ORIGINALITY

I hereby declare that this thesis is my own work and that, to the best of my knowledge and belief, it reproduces no material previously published or written, nor material that has been accepted for the award of any other degree or diploma, except where due acknowledgement has been made in the text.

(Signed)

WANG Jinghua (Name of Student)

# Abstract

Compared with the other biometrics techniques, face recognition is non-intrusive, natural, and easy to use. In a face recognition system, the feature extraction procedure aims to improve the recognition accuracy and robustness. The most popular feature extraction methods are appearance-based methods, which regard the face images as points in the image space and learn the feature extraction scheme based on the relationship between these points.

The work in this thesis is in three parts. The first part proposes fast kernel Fisher discriminant analysis (FKFDA) to accelerate the nonlinear feature extraction. The second part proposes a method to extract pose-invariant feature. The third part extracts features for face verification.

*Fast kernel Fisher discriminant analysis (FKFDA)*

Kernel Fisher discriminant analysis (KFDA) extracts a nonlinear feature from a sample by calculating as many kernel functions as the training samples. Thus, its computational efficiency is inversely proportional to the size of the training sample set. In section 3, we propose FKFDA for fast feature extraction. This FKFDA consists of two procedures. First, we select a portion of training samples based on two criteria produced by approximating the kernel principal component analysis (AKPCA). Then, referring to the selected training samples as nodes, we formulate FKFDA to improve the efficiency. In FKFDA, the discriminant vectors are expressed as linear combinations of nodes in the kernel feature space, and the extraction of a feature from a sample only requires calculating as many kernel functions as the nodes. Therefore, the proposed FKFDA has a much faster feature extraction procedure compared with the naive

kernel-based methods. Experimental results suggest that the proposed FKFDA can generate well classified features.

*Pose-invariant feature extraction*

Recognizing face images across pose is one of the challenging tasks for reliable face recognition. Section 4 presents a new method to tackle this challenge based on orthogonal discriminant vector (ODV). The result of our theoretical analysis shows that an individual's probe image captured with a new pose can be represented by a linear combination of his/her gallery images. Based on this observation, in contrast to the conventional methods which model face images of different individuals on a single manifold, we propose to model face images of different individuals on different linear manifolds. The contribution of our approach includes: 1) to prove that the orthogonality to ODVs is a pose-invariant feature.; 2) to categorize each person with a set of ODVs, where his/her face images posses zero projections while other persons' images are characterized by maximum projections; 3) to define a metric to measure the distance between a face image and an ODV, and classify the face images based on this metric. Our experimental results validate the feasibility of modelling the face images of different individuals on different linear manifolds.

*Feature extraction for verification*

In face verification, while the positive samples are the images of one person, the negative samples can be anything else. These two classes can be quite different in both size and distribution. This imbalance degrades the performance of many feature extraction methods and classifiers. Section 5 proposes a method for extracting minimum positive and maximum negative features (in terms of absolute value) for face verification. We develop two models to yield the feature

extractors. Model 1 first generates a set of candidate extractors that can minimize the positive features, and then chooses the ones among these candidates that can maximize the negative features. Model 2 first generates a set of candidate extractors that can maximize the negative features, and then chooses the ones that can minimize the positive features. Compared with the traditional feature extraction methods and classifiers, the proposed models are less likely affected by the imbalance.

# Publications arising from the thesis

## Referred Journal Article

[1] Jinghua Wang, Yong Xu, David Zhang, Jane You, “an efficient method for computing orthogonal discriminant vectors”, *Neurocomputing*, vol. 73, no. 10-12, 2168–2176, 2010.

[2] Jinghua Wang, Qin Li, Jane You, Qijun Zhao, “fast kernel Fisher discriminant analysis via approximating the kernel principal component analysis”, *Neurocomputing*, vol. 74, no. 17, 3313–3322, 2011.

[3] Jinghua Wang, Jane You, Qin Li, Yong Xu, “extract minimum positive and maximum negative features for imbalanced binary classification”, *Pattern Recognition*, vol. 45, no. 3, 1136–1145, 2012.

[4] Jinghua Wang, Jane You, Qin Li, Yong Xu, “orthogonal discriminant vectors for face recognition across poses”, *Pattern Recognition*, vol. 45, no. 12, 4069-4079, 2012.

[5] Jinghua Wang, Peng Wang, Qin Li, Jane You, “Improvement of the kernel minimum squared error model for fast feature extraction”, Neural Computing & Applications, DOI 10.1007/s00521-012-0813-9, 2012.

## Working Journal Article

[6] Jinghua Wang, Jane You, Yong Xu, Qin Li, “enlarge the training set based on interpersonal relationship for face recognition from one image per person”, *to be submitted*

[7] Jinghua Wang, Jane You, Yong Xu, Qin Li, “class-specific representation with spatially smooth residue for robust face recognition”, *to be Submitted*

# Other Publications

## Referred Journal Article

[8] Yong Xu, Qi Zhu, Jinghua Wang, “Breast cancer diagnosis based on a kernel orthogonal transform”, *Neural Computing and Applications*, DOI 10.1007/s00521-011-0547-0.

## Referred Conference Article

[9] Jinghua Wang, Yong Xu, Jane You, “Sparse residue for occluded face image reconstruction and classification”, *21st International Conference on Pattern Recognition*. (Accepted)

[10] Qin Li, Jane You, Jinghua Wang, Allan Wong, “A fully automated system for retinal vessel tortuosity diagnosis using scale dependent vessel tracing and grading”, *IEEE 23rd International Symposium on Computer-Based Medical Systems (CBMS)*, Perth, Australia, 12-15 October, 2010, pp. 221 – 225.

[11] Qin Li, Jinghua Wang, Jane You, Bob Zhang, Fakhri Karray, “Refractive Error Detection via Group Sparse Representation”, *2010 International Conference on Autonomous and Intelligent Systems (AIS)*, Povoa de Varzim, Portugal, 21-23, June, 2010, pp. 1-5.

# Acknowledgements

I would like to take this opportunity to express my sincere thanks to everyone who generously gave me their care and support.

First of all, I would like to express my extreme gratitude to my chief supervisor, Prof. Jane You, for her enduring support, guidance and patience throughout my studies. She is always energetic and passionate about work. I benefited a lot from “Jane’s Style”.

Secondly, I would like to show my utmost gratefulness to Prof. David ZHANG, Prof. Yong Xu, Dr. Lei Zhang, and Dr. Qin Li for their constructive comments on my research work. I have learnt much from frequent discussion with them.

Thirdly, I would like to thank my colleagues in the BRC lab, Dr. Guangming Lu, Dr. Qinghua Hu, Dr. Zhenhua Guo, Dr. Qijun Zhao, Dr. King-Hong CHEUNG, Dr. Lin Zhang, Dr. Dongmin Guo, Xingzheng Wang, Xiaofeng Qu, Feng Liu, Yahui Liu, Bo Shen, Peng Wang, Wei Li, Meng Yang, Bo Peng, Jin Xie, Zhizhao Feng, Kaihua Zhang, and and Dr. Yinghui Chen, for their great help in my study. And also I would like to thank my friends in Shenzhen, Dr. Jinxing Liu, Dr. Jianxun Mi, Dr. Zhihui Lai, Qi Zhu, Zizhu Fan, Jinrong cui, Jiajun Wen and others for the helpful discussions between us.

Finally, I would like to thank my parents and grandparents for their love, support, and all their self-sacrifice on my behalf throughout the past years. All that I have done would be impossible without the support, help, and patience of my wife Linlin Chen, who has been at my side in the difficult times and shared the happy times with me.

# Table of contents

Abstract ..... i
Publications arising from the thesis ..... iv
Referred Journal Article ..... iv
Working Journal Article ..... iv
Other Publications ..... v
Referred Journal Article ..... v
Referred Conference Article ..... v
Acknowledgements ..... vi
Table of contents ..... vii
List of figures ..... ix
List of tables ..... xi
List of tables ..... xi
**Chapter 1 Introduction** ..... 1
1.1 Face recognition ..... 1
1.2 Challenges ..... 4
1.3 Feature extraction ..... 6
**Chapter 2 Literature review** ..... 8
2.1 1D method ..... 8
2.2 2D method ..... 13
**Chapter 3 Fast KFDA via approximating KPCA for nonlinear feature extraction** ..... 17
3.1 Introduction ..... 17
3.2 The KFDA procedure ..... 19
3.3 FKFDA Development ..... 22
3.3.1 Criteria for node selection ..... 23
3.3.2 Node selection scheme ..... 27
3.3.3 FKFDA formulation using nodes ..... 28
3.4 Discussion ..... 30
3.4.1 Computational complexity ..... 31
3.4.2 Singular generalized eigenequation problem ..... 33
3.5 Experimental results ..... 34
3.5.1 Face databases ..... 35
3.5.2 UCI datasets ..... 41
3.5.3 Discussion on the parameter ..... 43
3.6 Conclusion ..... 46
**Chapter 4 Pose-invariant feature extraction** ..... 48
4.1. Introduction ..... 48
4.2. Intrapersonal relationship among face images across pose ..... 50
4.2.1 From 3-D face to 2-D image ..... 51
4.2.2 From 2-D image to 3-D face ..... 52
4.2.3 2-D image prediction ..... 54
4.2.4 Discussion ..... 54
4.3. Orthogonal discriminant vectors ..... 56
4.3.1 Basic ideas ..... 57
4.3.2 The existence of the ODV ..... 58
4.3.3 The calculation of the ODV ..... 62

4.3.4 ODV-based face classification ..... 65
4.3.5 Computational complexity ..... 67
4.4. Experiments ..... 69
4.4.1 Residue investigation ..... 70
4.4.2 face verification ..... 72
4.4.3 Face recognition ..... 74
4.5. Conclusion and future work ..... 79
**Chapter 5 Feature extraction for face verification** ..... 82
5.1. Introduction ..... 82
5.2. Background and Motivation ..... 85
5.3. Proposed method ..... 88
5.3.1 Basic idea ..... 89
5.3.2 Model 1 ..... 91
5.3.3 Model 2 ..... 93
5.3.4 Classification and discussion ..... 95
5.4. Experiments ..... 97
5.4.1 Synthetic Data classification ..... 97
5.4.2 Face verification ..... 101
5.5. Conclusion ..... 103
**Chapter 6 Conclusion and Future work** ..... 105
6.1 Conclusion ..... 105
6.2 Future work ..... 107
**Bibliography** ..... 109

# List of figures

Figure 1.1 Taxonomy of biometrics ..... 1
Figure 1.2 Flowchart of face recognition ..... 3
Figure 1.3 Two modes of face recognition: identification and verification ..... 4
Figure 1.4 Intrapersonal variations in illumination, pose, occlusion, and expression. (The images are from the AR database [11]) ..... 4
Figure 1.5 Similarity of frontal faces between ..... 5
Figure 1.6 the possible relationship between a) face and non-face images; b) face images of two different persons ..... 5
Figure 1.7 illustration of the geometry-based features [13] ..... 6
Figure 2.1 Ten eigenfaces of the ORL database ..... 10
Figure 2.2 image reconstruction results using 20 eigenfaces. The first line shows the original images and the second line shows the reconstruction result. ..... 10
Figure 2.3 Comparison between PCA and LDA ..... 11
Figure 3.1 An example. ..... 26
Figure 3.2 The relationship between FDA, KFDA, and the proposed FKFDA ..... 31
Figure 3.3 Sample images from the AR database ..... 35
Figure 3.4 Sample images from the yaleB database ..... 36
Figure 3.5 Accuracy versus number of features on AR database: ..... 38
Figure 3.6 Accuracy versus number of features on YaleB database: ..... 39
Figure 3.7 Node selection time of different methods on face database: ..... 40
Figure 3.8 Classification accuracy vs. node percentage on AR database: ..... 45
Figure 4.1 Procedure for generating the intrapersonal relationship among 2-D face images. (a) 2-D face images generation; (b) 3-D face reconstruction; (c) 2-D face images generation from reconstructed 3-D face; (d) relationship generation among 2-D face images ..... 53
Figure 4.2 Face images of the same individual have the similar configuration ..... 54
Figure 4.3 Linear expression of a novel face image using gallery face images ..... 55
Figure 4.4 The distance of the ODV $v$ to the images from two classes ..... 65
Figure 4.5 Inter- and intra-class residue distribution on (a) AR face database; (b) YaleB face database; (c) FERET face database; (d) PIE database ..... 71
Figure 4.6 The distribution of the face images in CMU PIE database ..... 73
Figure 4.7 The distribution of the face images in YaleB database ..... 73
Figure 4.8 The ROC curves. (a) The subset of CMU PIE database; (b) The YaleB database ..... 74
Figure 4.9 Classification accuracy comparisons of different methods on different databases: (a) AR face database; (b) CMU PIE face database; (c) YaleB face database ..... 77
Figure 5.1 The distribution of the face images of three different individuals ..... 87
Figure 5.2 The classification of an outlier. The outlier is misclassified into the positive class by the solid line ..... 88
Figure 5.3 separate class 1 (positive) from class 2 (negative) using two

parallel hyperplanes ..... 90
Figure 5.4 The projections of samples onto feature extractors: (a) one feature extractor; (b) two feature extractors ..... 96
Figure 5.5 The distribution of the first synthetic dataset ..... 98
Figure 5.6 the distribution of the second synthetic dataset ..... 99
Figure 5.7 Examples in the CMU PIE database ..... 101

# List of tables

Table 1.1 Typical applications of face recognition [4] ..... 2
Table 3.1 Training computational complexity of different methods ..... 32
Table 3.2 The number of nodes captured under different conditions ..... 37
Table 3.3 Training time of different methods ($\times 10^5$ seconds) ..... 39
Table 3.4 Number of training samples and of nodes and their ratios on UCI datasets ..... 41
Table 3.5 Classification accuracy (%) with polynomial kernel function on UCI datasets..... 42
Table 4.1 Classification accuracies (%) of methods on FERET face database ..... 76
Table 4.2 Training time (seconds) of different methods in three face databases..... 79
Table 5.1 the performance (TPR and TNR) of different methods on synthetic datasets (%)..... 100
Table 5.4 Experimental results of face verification on CMU PIE subset TPR and TNR (%); and D (number of feature extractors) ..... 102

# Chapter 1 Introduction

## 1.1 Face recognition

Body characteristics including face, voice, and odor et al. are what we use to recognize each other for thousands of years. In mid-19$^{th}$ century, Alphonse Bertillon proposed to identify criminals using a number of body measurements [1]. Since then, this idea gains popularity and leads to the studies of biometric techniques, or biometrics. Biometrics techniques recognize a person based on whom a person really is, and can be used in any applications requiring access or security control. Compared with the traditional technologies which are based on what a person know or has, biometrics technology is safer from attacking. The biometrics techniques fall into two groups: behavioral and physical, as shown in Figure 1.1.

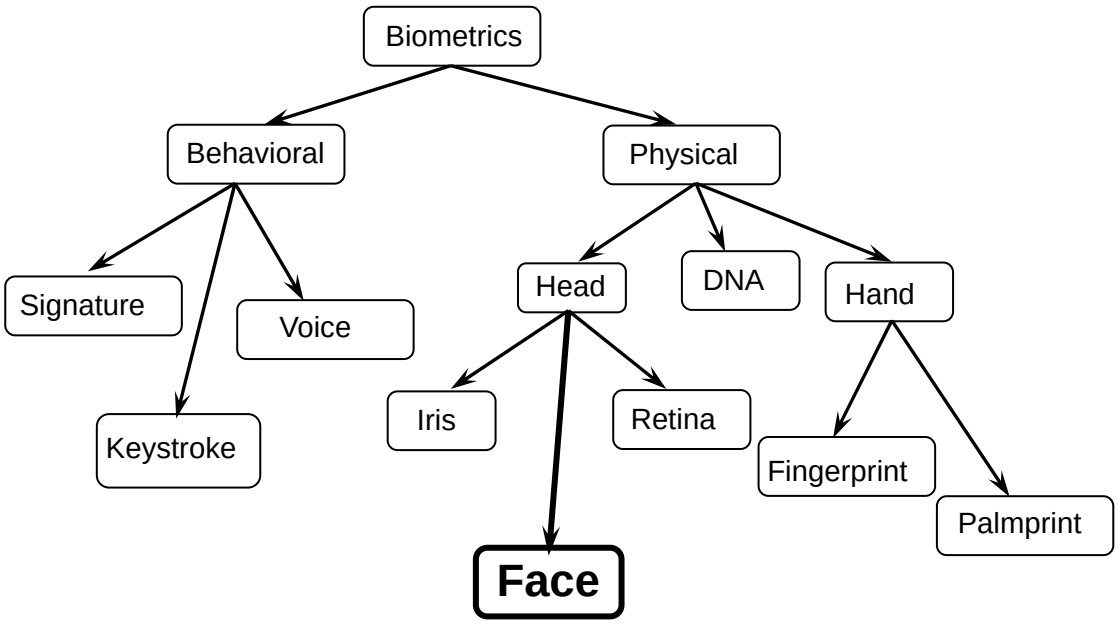



Figure 1.1 Taxonomy of biometrics

Among all the biometrics, face recognition is probably the most commonly used

in our daily life. Compared with other biometrics, face recognition is non-intrusive, natural, and easy to use [2]. Face recognition has numerous practical applications such as access control, bankcard identification, security monitoring, Mug shots searching, and surveillance system [3]. Table 1.1 summarizes the applications of face recognition [4].

Table 1.1 Typical applications of face recognition [4]

| Areas | Specific application |
|---|---|
| Law enforcement and surveillance | Portal control, post-event analysis |
| | Shoplifting, suspect tracking and investigation |
| | CCTV control |
| Smart cards | Welfare fraud |
| | Immigration, national ID, voter registration |
| | Drivers' licences, entitlement programs |
| Information security | Intranet and internet security, medical records |
| | Secure trading terminals |
| | Application security, database security |
| | TV parental control, desktop logon |
| Entertainment | Human-computer-interaction |
| | Video game, virtual reality, training programs |

Studies of face recognition go back at least to 1950s in psychology [5]. Some other earliest works in face recognition study the model method [6] and facial expression of emotions [7]. In 1970s, researchers began to study automatic recognition of face images [8-9], which laid the foundation for the next 40 years.

Nowadays, face recognition can be defined to be the automatic recognition of individuals based on their face images. A face recognition system generally consists of

four modules [2, 10] as shown in Figure 1.2. Face detection segments the face image from the background based on the coarse estimates of location and size. Face alignment normalizes the detected face images. Feature extraction aims to extract information that is useful for discriminant images of different persons. The final procedure of matching calculates the similarity between the extracted feature and those of the enrolled face stored in the database. This thesis focuses on the feature extraction methods.

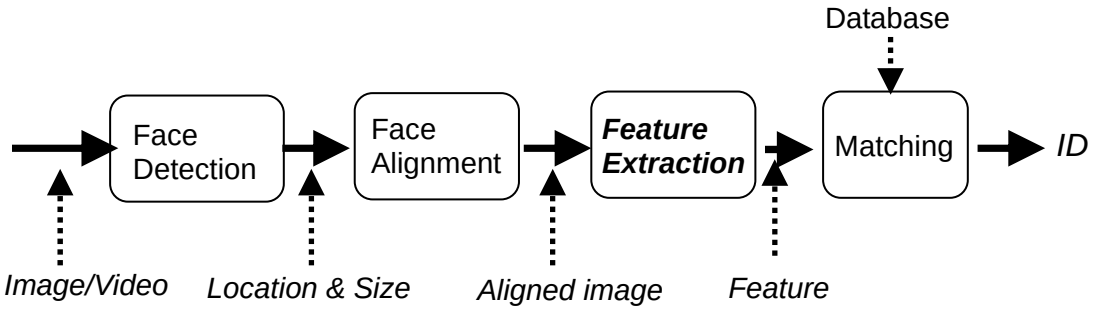



Figure 1.2 Flowchart of face recognition

There are two modes of face recognition: face verification and face identification, as shown in Figure 1.3. A face verification system investigates whether the identity of the query face image is same to the one it claimed. An individual presents both his image and identity to a verification system, and then the system matches his image with the particularly claimed template in the database. This one-to-one matching task tries to answer the question “am I the identity I claim to be?” The outcomes can be “genuine” or “imposter”. A face recognition system matches a provided image with every template in the database to determine the identity of the provided image. This is a one-to-many problem to answer the question “who am I?” The outcome is expected to be the identity of the face image. If the distinction of verification and identification is not important, this thesis will use the generic term recognition.

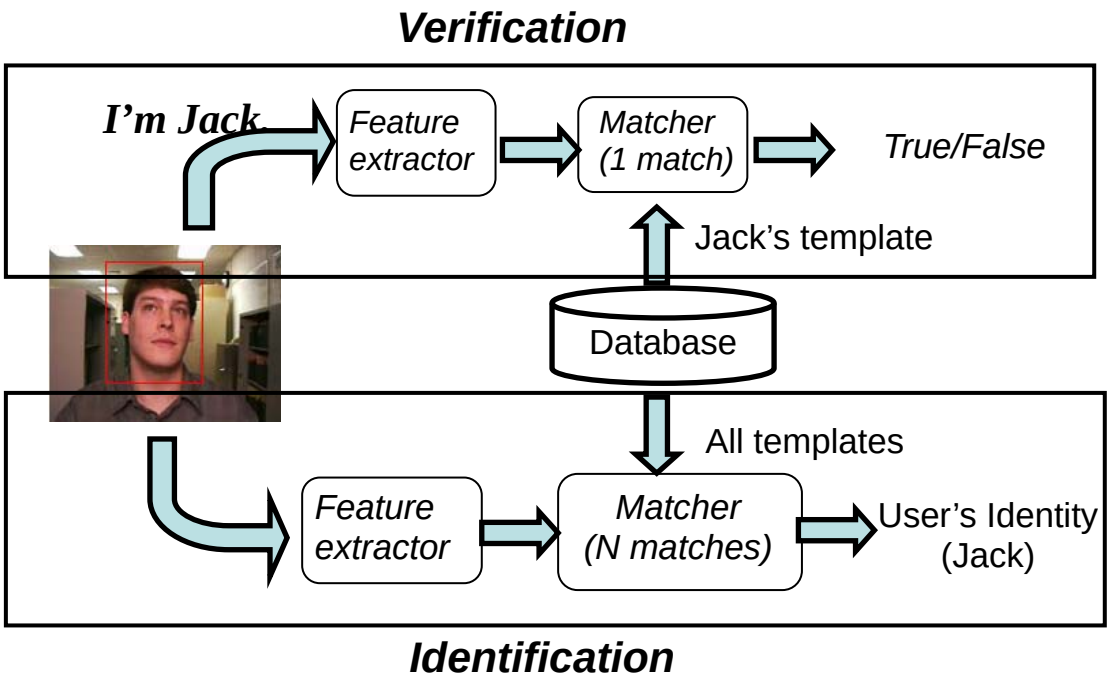



Figure 1.3 Two modes of face recognition: identification and verification

## 1.2 Challenges

Figure 1.4 Intrapersonal variations in illumination, pose, occlusion, and expression. (The images are from the AR database [11])

Two face images from the same person are not exactly the same due to the imperfect imaging conditions, illumination changes, pose variation, and expression variation. Figure 1.4 shows the face images of the same person captured under different conditions. The study of how these conditions effect the face images are still ongoing. Thus, two images output by the second module of face alignment in Figure 1.2 can be quite different even though they are from the same person. Studies show

that the intrapersonal differences induced by the variations of viewing direction and illumination are almost always larger than interpersonal difference [12]. In addition, small interpersonal difference shown makes the face recognition more difficult, as shown in Figure 1.5. The distribution of the face images in the image space is complicated and requires further studies. Figure 1.6 a) illustrates a possible relationship between face images and non-face images, and b) illustrates the possible face image distributions of two different persons.


Figure 1.5 Similarity of frontal faces between
(a) Twins (downloaded from www.marykateandashley.com); and
(b) A father and his son (downloaded from BBC news, news.bbc.co.uk).

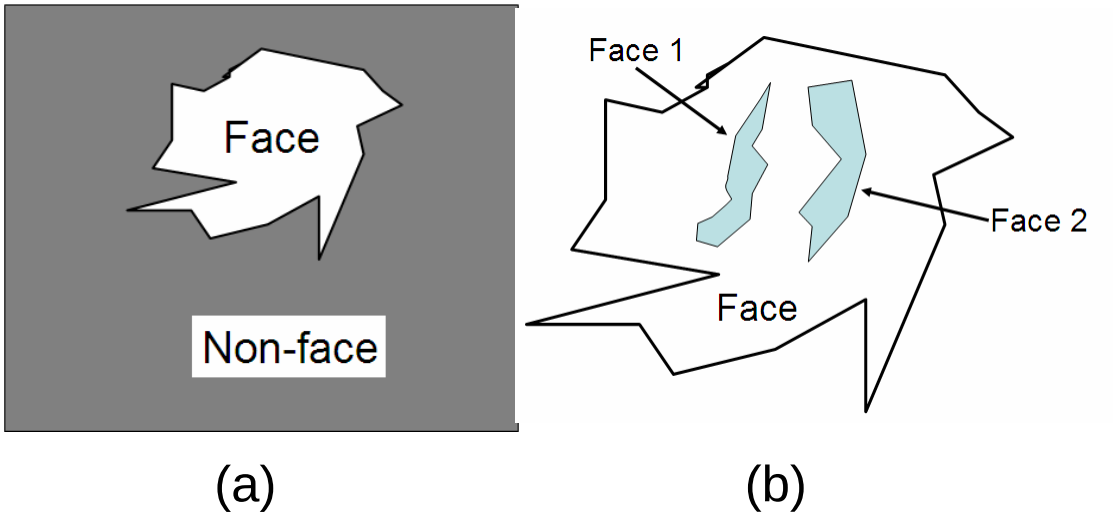



Figure 1.6 the possible relationship between a) face and non-face images; b) face images of two different persons

## 1.3 Feature extraction

Feature extraction procedure improves the recognition accuracy. If we perform the matching procedure directly based on the face images, the recognition accuracy will be low due to the large intrapersonal and some small interpersonal distances. The feature extraction procedure aims to depress the intrapersonal distance and increase the interpersonal distance, and thus improve the accuracy.

Feature extraction procedure improves the robustness of the classifier, and alleviates the small sample size (SSS) problem. The face images commonly reside in a Euclidean space with dimensionality larger than ten thousand. Nevertheless, the number of training images from each person is typically smaller than one hundred, and even just one in some situations. This induces the SSS problem. The classifier trained on so few sample for the high dimensional samples may not generalize well and lack of robustness. Feature extraction procedure reduces the dimensionality of the face images significantly and alleviates these problems.

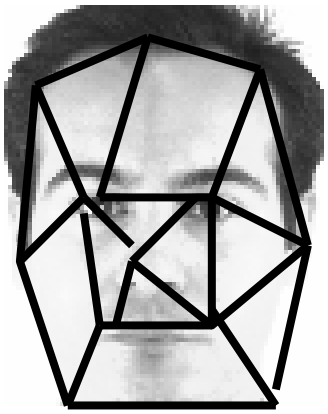


Figure 1.7 illustration of the geometry-based features [13]

Two different groups of feature extraction methods for face recognition are:

geometry-based methods and appearance-based methods. Geometry-based feature extraction methods [13-18] study the relations (e.g. angles and distances) between facial features such as chin, mouth, nose, and eyes, as shown in Figure 1.7. Though they are known as insensitive to small variations in viewpoint and illumination, these geometry-based methods are quite sensitive to measurement process and not reliable enough [19]. Geometric features are also argued to be ineffective in finding local image features and inferring identity by the geometric relations [20].

More popular feature extraction methods are appearance-based methods [21]. The appearance-based methods consider the $w$ by $h$ face images as a point in $w \times h$ dimensional Euclidean space, and learn the feature extraction scheme based on the relationship between these points. Compared with the geometry-based methods, it can extract many more discriminative features and achieve higher classification accuracy. In addition, the appearance-based method is more computationally efficient. Section 2 reviews the representative appearance-based methods. All of the methods proposed in this thesis are appearance-based method.

# Chapter 2 Literature review

This section reviews the appearance-based feature extraction methods for face recognition. The appearance-based methods roughly fall into two groups based on their input: one-dimensional (1D) method and two-dimensional (2D) method. While the 1D method takes the vectorized 2D image as the input, 2D method works directly on the 2D image. Section 2.1 and 2.2 respectively review the representative 1D and 2D appearance-based feature extraction methods.

Let the gallery set consists of $n = \sum_{i=1}^{c}$ face images from $c$ persons, $n_i$ images are from the $i$ th person. Denote the $w$ by $h$ matrix $A_j^k \left(1 \le j \le n_i; 1 \le k \le c\right)$ the $j$ th image of the $k$ th person, where $w$ and $h$ are number of rows and columns of the face images. The one dimensional vector $x_j^k \in R^{w \times h}$ is the vectorized version of the two dimensional image $A_j^k$. Denote $m_i$ and $m$ respectively the $i$ th class mean and total sample mean of the one dimensional vectors.

## 2.1 1D method

In 1987, Kirby and Sirovich pioneered the research of appearance-based method by introducing principal component analysis (PCA) [22] to face recognition. In 1991, Turk and Pentland proposed Eigenface [23], which is one of the most popular face recognition methods. Study shows that there exist significant statistical redundancies in the normalized face images [24]. PCA is an unsupervised method that seeks the most representative low dimensional representations for the face images. PCA calculates a set of eigenfaces which are factually the eigenvectors of the covariance

matrix

$$S_t \alpha = \lambda \alpha \tag{2.1}$$

where $S_t = \sum_{i=1}^{c} \sum_{j=1}^{n_i} \left( x_j^i - m \right) \left( x_j^i - m \right)^T$ is the covariance matrix, $\lambda$ and $\alpha$ are respectively the eigenvalue and eigenvector. Each face image is represented as a linear combination of these eigenfaces and the weight is the projection of the face image onto the eigenface. This vector of weights is the feature extraction result. Figure 2.1 shows 10 eigenfaces from the ORL database [25]. We use twenty eigenfaces to reconstruct the original face images of size 112 by 92. Figure 2.2 shows the original images in the first row and the reconstruction results in the second row. The face images of size 112 by 92 are well reconstructed using only 20 eigenfaces. Later, PCA is further developed by researchers [26-27].

Fisher discriminant analysis [28] (LDA or FDA) is another successful feature extraction method for face recognition. LDA aims to extract features that can maximize the between-class scatter matrix $S_b$ and simultaneously minimize the within-class scatter matrix $s_w$

$$S_w = \sum_{i=1}^{c} \sum_{j=1}^{n_i} \left( x_j^i - m_i \right) \left( x_j^i - m_i \right)^T \tag{2.2}$$

$$S_b = \sum_{i=1}^{c} n_i \left( m_i - m \right) \left( m_i - m \right)^T \tag{2.3}$$

To yield the feature extractor, LDA maximizes the Fisher criterion, which is the ratio between the determinants of between and within-class scatter matrix of the projected samples, as follows

$$J(\alpha) = \frac{\left| \alpha^T S_b \alpha \right|}{\left| \alpha^T S_w \alpha \right|} \tag{2.4}$$

Figure 2.1 Ten eigenfaces of the ORL database

Figure 2.2 image reconstruction results using 20 eigenfaces. The first line shows the original images and the second line shows the reconstruction result.

The feature extractors that maximize the equation (2.4) are proved to be the eigenvectors of a generalized eigenequation

$$S_b\alpha = \lambda S_w\alpha \tag{2.5}$$

Solving the above generalized eigenequation problem is time consuming and sensitive to some noise. If the matrix $S_w$ is nonsingular, the discriminant vectors in (2.5) are the eigenvectors of $S_w^{-1}S_b$. This converts the generalized eigenequation problem into a traditional eigenequation problem, and saves computational burden significantly. However, this approach can not be applied directly in face recognition, as the

dimensionality of the face images are high and the matrix $S_w$ is commonly singular. To solve this problem, researchers proposed pseudo-inverse method [29], perturbation method [30], rank decomposition method [31], PCA plus LDA method [32], and so on [33-34]. Some methods can increase the discriminant information of LDA, such as the null space method [35], PCA plus null space method [31-32], and direct-LDA [36].

While PCA extracts representative features, LDA extracts the most discriminative features in terms of Fisher criterion, as shown in Figure 2.3. In Figure 2.3, the LDA features of class 1 and class 2 are separable, but the PCA features are mixed together and non-separable. Due to this, researchers tend to prefer LDA to PCA when comparing them [37]. This is true when the data of each class can be represented by a Gaussian distribution and share the comment covariance matrix. However, this assumption does not always hold on the task of face recognition. In 2001, Martinez and Kaka present their experimental results and show PCA might outperform LDA on face recognition [38]. It is claimed that LDA is inferior to PCA when the training data do not reveal the underlying distribution or the number of samples per class is small [39-41].

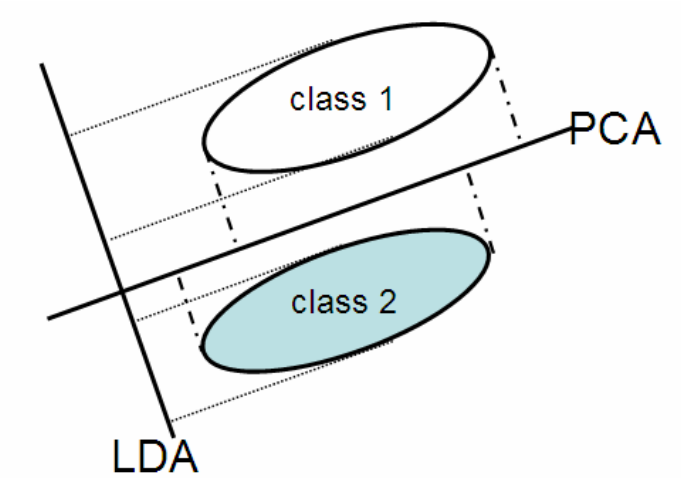



Figure 2.3 Comparison between PCA and LDA

Independent component analysis (ICA) [42] is an extension of PCA which

extracts the discriminative information by minimizing dependencies in order higher than two. Different from the feature extractors in PCA, these in ICA are not necessary to be orthogonal. Notable algorithms for ICA are FastICA [43] and Jade [44]. There are two different architectures of ICA. The first architecture outputs spatially local image to represent the face image. The images are variables and pixels are outcomes in this architecture. Regarding the pixels as variables and images as outcomes, the second architecture codes the images factorially.

Another popular appearance-based method is locality preserving projections (LPP) [45], which aims to preserve the local structure of the face image space. LPP consists of two steps. The first step constructs an adjacency graph of all the face images and assigns a weight to each edge between two nodes. The second step is to find the feature extractors that preserve most local information. To alleviate the computational burden, PCA is performed before LPP as a preprocessing procedure. Orthogonal locality preserving projections [46] is an extension of LPP that yields mutually orthogonal feature extractors.

Manifold learning methods take the face images as a set of related points and consider them span a low dimensional manifold [47]. Study shows that geometric nature and local structures of the data space possess much discriminant information [48-49], particularly in the face recognition scenario [2]. A number of manifold learning algorithms aim at discovering the geometric nature of the face space and keep it in the feature space, such as locally linear embedding [50], Isomap [51], Laplacian Eigenmaps [52], and Semidefinite Embedding [53]. Since the feature extraction strategy only defined for the training samples, these methods are not applicable to the testing face images [50-53]. When the previous manifold methods learn a single manifold for all of the training images, they eliminate much

discriminant information in the feature extraction procedure. In Section 4 of this thesis we propose a method to model face images of different individuals in different manifolds. This study is based on our theoretical analysis of the intrapersonal relationship across poses.

Kernel-based techniques are known for its effectiveness in extracting nonlinear features [54]. The basic idea of kernel-based techniques is to implicitly map the face images into a high (and some times infinite) dimensional kernel space. The nonlinear problems in the image space are expected to be transformed into a linear problem in the kernel space. Kernel principal component analysis (KPCA) [55-60] is the PCA procedure in the high dimensional space, and kernel Fisher Linear discriminant analysis (KFDA) [61-64] is the nonlinear counterpart of LDA. The kernel methods express the feature extractors as liner combinations of the training samples. Due to this, the feature extraction efficiency is inversely proportional to the size of the training set. In section 3 of this thesis, we propose a fast feature extraction using KFDA. In the proposed method, we express the feature extractor as a linear combination of a portion of the training samples, which are selected by approximating the KPCA procedure.

## 2.2 2D method

One-dimensional (1D) methods need to transform the two dimensional (2D) $r$ by $h$ face image into a one dimensional vector in the space $R^{r\times h}$. Normally, this transformation leads to a $r\times h$ by $r\times h$ covariance matrix, which is very important in many appearance-based feature method. This large covariance matrix is normally singular and difficult to estimate accurately [65]. Also, the transformation from the 2D image to the 1D vector ignores the underlying structure of the face image [66]. To

save as much discriminant information as possible and reduce the computational complexity, researchers proposed 2D appearance-based feature extraction methods.

In the 2D face image space, the total scatter matrix $R_t$, within-class scatter matrix $R_w$, and between-class scatter matrix $R_b$ are defined as follows

$$R_t = \sum_{i=1}^{c} \sum_{j=1}^{n_i} \left( A_j^i - M \right)^T \left( A_j^i - M \right) \tag{2.6}$$

$$R_w = \sum_{i=1}^{c} \sum_{j=1}^{n_i} \left( A_j^i - M_i \right)^T \left( A_j^i - M_i \right) \tag{2.7}$$

$$R_t = \sum_{i=1}^{c} n_i \left( M_i - M \right)^T \left( M_i - M \right) \tag{2.8}$$

where $M_i$ and $M$ are the mean 2D image of the $i$ th class and all the face images.

Two dimensional principal component analysis (2DPCA) maximizes the projected total scatter matrix [65, 67]

$$W_{2DPCA} = \arg\max_{W \in R^{h \times h'}, W^T W = I} \left| W^T R_t W \right| \tag{2.9}$$

Working directly on the 2D image, 2DPCA is more computational effect than PCA. For example, if the images are of size 128 by 128, the size of the covariance matrix in PCA is 16384 by 16384 and that in 2DPCA is still 128 by 128. Different from the one directional feature extraction approaches [65, 67], two directional method [68] extracts features from both of the two direction: vertical and horizontal. In the vertical 2DPCA, the total scatter matrix is defined as follows

$$C_t = \sum_{i=1}^{c} \sum_{j=1}^{n_i} \left( A_j^i - M \right) \left( A_j^i - M \right)^T = R_t^T \tag{2.10}$$

In [68], Yang proved the properties of 2DPCA: The horizontal (vertical) 2DPCA is invariant to vertical (horizontal) image translations and vertical mirror imaging. Because 2DPCA is less sensitive to face cropping and imprecise eye detection, it can

improve the traditional discriminant analysis methods for face verification [68].

Similar to LDA, two dimensional linear discriminant analysis (2DLDA) [69-70] is defined based on the following Fisher criterion

$$W_{2DLDA} = \arg\max_{W \in R^{h \times h'}, W^T W = I} \left| \frac{W^T R_b W}{W^T R_w W} \right| \tag{2.11}$$

Compared to LDA, 2DLDA is more computationally efficient to extract discriminant features and is able to preserve the underlying structure. In addition, the within-class scatter matrix is usually nonsingular in 2DLDA and the small sample problem (SSS) does not occur [69]. 2DLDA also can be performed from two directional, and in the vertical 2DLDA the between-class scatter matrix $C_b$ and within-class scatter matrix $C_w$ are defined as follows

$$\begin{cases} C_b = R_b^T \\ C_w = R_w^T \end{cases} \tag{2.12}$$

The Fisher criterions in two directional 2DLDA are as follows [71-72]

$$\begin{cases} W_{2DLDA\_R} = \arg\max_{W \in R^{h \times h'}, W^T W = I} \left| \frac{W^T R_b W}{W^T R_w W} \right| \\ W_{2DLDA\_C} = \arg\max_{W \in R^{r \times r'}, W^T W = I} \left| \frac{W^T C_b W}{W^T C_w W} \right| \end{cases} \tag{2.13}$$

The 2-D Fisherface [73] selects the principal components and discriminant vectors automatically. Though 2-D Fisherface [73] is more computationally efficient than the naïve 2D methods in feature extraction [74], it may lose some important discriminant information.

As both 2DLDA and 2DPCA extract features based on the columns or rows of the face image, the extracted features may still contain some redundant information. To further remove the redundancy, researchers propose to perform an additional LDA after 2DPCA [75] and 2DLDA [76]. DiaPCA [77] and DiaLDA [78] are another two

representative 2D extensions. Their advantages lie on their ability in preserving the correlation between variations of columns and rows of the face images [77].

Two dimensional locality preserving projections (2DLPP) [79-80] is an extension of LPP. 2DLPP directly solves the singular problem of LPP and preserves as much information as possible. The 2DLPP is claimed to be superior to 2DLDA in face recognition [81]. However, it is very difficult to fix the parameter optimally in 2DLPP.

# Chapter 3 Fast KFDA via approximating KPCA for nonlinear feature extraction

## 3.1 Introduction

Kernel-based learning machines [54] such as support vector machine (SVM) [82], kernel principal component analysis (KPCA) [55-56], kernel Fisher discriminant analysis (KFDA) [61, 64], and kernel minimum squared error (KMSE) [83] have attracted much attention in the field of pattern recognition. While KPCA [55-56] is formulated as a nonlinear form of principal component analysis (PCA) [84], KFDA proposed in [64] and [61] are nonlinear schemes of Fisher discriminant analysis (FDA) for, respectively, two-class and multi-class problems. KFDA is very effective in extracting nonlinear features and has been used in many real-world applications [85-86], such as face recognition [87] and handwritten digit classification [57]. However, KFDA-based feature extraction is not efficient.

The feature extraction efficiency of a naive kernel-based method is inversely proportional to the size of the training sample set. This is because, in order to extract a feature from a sample, the kernel-based methods must calculate as many kernel functions as the training samples. Thus, the kernel-based feature extraction methods are computationally inefficient or even unfeasible when the training sample set is large. Some algorithms have been proposed to accelerate the feature extraction of kernel-based methods [88-92]. The methods [88-89, 93] simplify computation by replacing the kernel matrix with its submatrix; however, they do not consider how representative the related samples are and tend to produce overfitting. Some other improved methods [90-92] have been proposed based on the assumption that the

discriminant vectors are linear combinations of “nodes”, i.e. a portion of the training samples. However, the node selection procedures in [90-92] are very time consuming.

In this chapter, we propose a fast kernel Fisher discriminant analysis (FKFDA) technique based on the idea of “node” to improve the efficiency of nonlinear feature extraction. This work is an extension of our previous work [94] which accelerates the feature extraction procedure of KPCA. The nodes are most representative training samples in this thesis. They replace the total training samples to express the discriminant vectors in the kernel feature space. The proposed FKFDA consists of two parts: node selection and FKFDA formulation. In the node selection part, we define a pseudo-eigenvalue for every training sample and use it as a criterion to assess how representative this sample is. In addition, we bound the similarity between the nodes to reduce the size of node set. In the FKFDA formulation part, we use the nodes to express discriminant vectors. We simplify this formulation procedure by using the novely presented derivation of the scatter matrices.

Our FKFDA is more computationally efficient than naive KFDA [61, 64]. In the training procedure, while the scatter matrices in KFDA scale with the number of all the training samples, they scale with the number of nodes in the designed FKFDA. In the testing procedure, to extract a feature from a sample, the FKFDA needs to calculate only as many kernel functions as the nodes (instead of as many as all the training samples in naive KFDA). As there are many fewer nodes than training samples, FKFDA is much more computationally efficient than naive KFDA in both training and testing procedure.

The rest of the section is organized as follows: subsection 3.2 briefly reviews the KFDA procedure and presents a novel derivation for the scatter matrices. Subsection 3.3 describes our FKFDA. Subsection 3.4 discusses the computational complexity and

singular generalized eigenequation problem of our FKFDA. Subsection 3.5 performs experiments to evaluate the proposed FKFDA. Subsection 3.6 offers our conclusion and outlines future work.

## 3.2 The KFDA procedure

Besides briefly introducing the KFDA procedure, this subsection presents novel derivations for the scatter matrices. Suppose there are $n = \sum_{i=1}^{c} n_i$ training samples from $c$ classes, where $n_i \left(1 \le i \le c\right)$ denotes the number of samples in the $i$ th class. The $d$ dimensional vector $x_j^i \left(1 \le j \le n_i\right)$ denotes the $j$ th sample of the $i$ th class. Let $\Phi$ be an implicit nonlinear map which maps the sample space $R^d$ into kernel feature space $\Phi : R^d \to F,\ x \to \varphi\left(x\right)$, and the inner product of any two mapping samples $\varphi\left(x_i\right)$ and $\varphi\left(x_j\right)$ in $F$ can be computed using the kernel function

$$k\left(x_i, x_j\right) = \varphi^T\left(x_i\right)\varphi\left(x_j\right) \tag{3.1}$$

Note that the dimensionality of the kernel feature space $F$ could be arbitrarily large or even infinite.

The KFDA is a procedure of Fisher discriminant analysis (FDA) in a kernel feature space induced by the kernel function. The Fisher criterion in the kernel feature space is defined as follows:

$$J^{\varphi}\left(\phi\right) = \frac{\phi^T S_b^{\varphi} \phi}{\phi^T S_w^{\varphi} \phi} \tag{3.2}$$

where $\phi$ is the discriminant vector; $S_w^{\varphi}$ and $S_b^{\varphi}$ defined in equations (3.3) and (3.4) are the within-class scatter matrix and between-class scatter matrix, respectively:

$$S_w^{\varphi} = \sum_{i=1}^{c} \sum_{j=1}^{n_i} \left(\varphi\left(x_j^i\right) - m_i^{\varphi}\right)\left(\varphi\left(x_j^i\right) - m_i^{\varphi}\right)^T \tag{3.3}$$

$$S_b^{\varphi} = \sum_{i=1}^{c} n_i \left( m_i^{\varphi} - m_0^{\varphi} \right) \left( m_i^{\varphi} - m_0^{\varphi} \right)^T \tag{3.4}$$

where $m_i^{\varphi} = \frac{1}{n_i} \sum_{j=1}^{n_i} \varphi\left( x_i^j \right)$ is the mean vector of the mapping training samples of the $i$ th class; and $m_0^{\varphi} = \frac{1}{n} \sum_{i=1}^{c} n_i m_i^{\varphi}$ is the mean vector across all the mapping samples.

Before presenting the discriminant vector calculation, we present novel derivations for the within-class and between-class scatter matrix in the kernel feature space. To the best of our knowledge, this procedure of derivation has appeared nowhere else.

Let $X_i = \left[ \varphi\left( x_1^i \right) \quad \varphi\left( x_2^i \right) \quad \cdots \quad \varphi\left( x_{n_i}^i \right) \right]$ denote the sample matrix of the $i$ th class and $X = \left[ X_1 \quad X_2 \quad \cdots \quad X_c \right]$ denote that of all the classes. Then, we have the following two equations

$$m_i^{\varphi} = \frac{1}{n_i} \sum_{j=1}^{n_i} \varphi\left( x_j^i \right) = \frac{1}{n_i} X_i 1_{n_i} \tag{3.5}$$

and

$$\varphi\left( x_j^i \right) - m_i^{\varphi} = X_i \left( e_j^i - \frac{1}{n_i} 1_{n_i} \right) \tag{3.6}$$

where $1_{n_i} \in R^{n_i \times 1}$ and $e_j^i \in R^{n_i \times 1}$ are $n_i$ dimensional vectors. All the elements of $1_{n_i} \in R^{n_i \times 1}$ are 1s. All the elements of $e_j^i$ are 0, except the $j$ th element is 1.Using (3.5) and (3.6), we can rewrite (3.3) as follows

$$\begin{aligned} S_w^{\varphi} &= \sum_{i=1}^{c} \sum_{j=1}^{n_i} \left( \varphi\left( x_j^i \right) - m_i^{\varphi} \right) \left( \varphi\left( x_j^i \right) - m_i^{\varphi} \right)^T \\ &= \sum_{i=1}^{c} \sum_{j=1}^{n_i} X_i \left( e_j^i - \frac{1}{n_i} 1_{n_i} \right) \left( e_j^i - \frac{1}{n_i} 1_{n_i} \right)^T X_i^T \\ &= \sum_{i=1}^{c} X_i P_i X_i^T = X P X^T \end{aligned} \tag{3.7}$$

where

$$P_i = \sum_{j=1}^{n_i} \left( e_j^i - \frac{1}{n_i} 1_{n_i} \right) \left( e_j^i - \frac{1}{n_i} 1_{n_i} \right)^T \in R^{n_i \times n_i} \tag{3.8}$$

In addition, the total scatter matrix can be expressed as follows

$$\begin{aligned} S_t^{\varphi} &= \sum_{i=1}^{c} \sum_{j=1}^{n_i} \left( \varphi\left(x_j^i\right) - m_0^{\varphi} \right) \left( \varphi\left(x_j^i\right) - m_0^{\varphi} \right)^T \\ &= \sum_{i=1}^{c} \sum_{j=1}^{n_i} \left( \varphi\left(x_j^i\right) - m_0^{\varphi} \right) \left( \varphi\left(x_j^i\right) - m_0^{\varphi} \right)^T \\ &= \sum_{i=1}^{c} \sum_{j=1}^{n_i} \varphi\left(x_j^i\right) \varphi\left(x_j^i\right)^T - n m_0^{\varphi} \left( m_0^{\varphi} \right)^T \\ &= X X^T - \frac{1}{n} X 1_n 1_n^T X^T \\ &= X \left( I - \frac{1}{n} 1_n 1_n^T \right) X^T \end{aligned} \tag{3.9}$$

As the total scatter matrix equals to the summation of the within-class and between-class scatter matrix, we have the following equation for the between-class scatter matrix

$$S_b^{\varphi} = S_t^{\varphi} - S_w^{\varphi} = X Q X^T \tag{3.10}$$

where

$$Q = I - \frac{1}{n} 1_n 1_n^T - P \tag{3.11}$$

Different from the formulas presented in [54, 57, 64, 88, 90, 95], both (3.7) and (3.10) take the sample matrix $X$ as a separated multiplier. The reasoning behind doing so is to simplify the formulation of the FKFDA in subsection 3.3.

According to Mercer's theory, the discriminant vector can be expressed as a linear combination of all the mapping samples

$$\phi = \sum_{i=1}^{c} \sum_{j=1}^{n_i} \alpha_j^i \varphi\left(x_j^i\right) = X \alpha \tag{3.12}$$

Substituting equations (3.7), (3.10), and (3.12) into equation (3.2), we rewrite the

Fisher criterion as

$$J^{\varphi}(\alpha)=\frac{\alpha^T X^T XPX^T X\alpha}{\alpha^T X^T XQX^T X\alpha}=\frac{\alpha^T KPK\alpha}{\alpha^T KQK\alpha}=\frac{\alpha^T M\alpha}{\alpha^T N\alpha} \tag{3.13}$$

where

$$\begin{aligned} &K=X^T X;\ \ K_{ij}=\varphi(x_i)^T\varphi(x_j)\\ &M=KPK\in R^{n\times n}\\ &N=KQK\in R^{n\times n}\end{aligned} \tag{3.14}$$

The discriminant vectors that maximize the Fisher criterion are the generalized eigenvectors corresponding to the largest eigenvalues in the following eigenequation [96]:

$$M\alpha_i=\lambda N\alpha_i \tag{3.15}$$

If the matrix $N$ is nonsingular, we can simplify the computation by converting this generalized eigenequation into a conventional eigenequation as follows

$$N^{-1}M\alpha_i=\lambda\alpha_i \tag{3.16}$$

The feature extraction result for a sample is composed of many components; each of them is the projection of the sample onto a discriminant vector. We use the following formula to extract the $k$ th feature component $y_k$ from the sample $x$

$$y_k=\alpha_k^T K_x=\sum_{i=1}^{c}\sum_{j=1}^{n_i}(\alpha_k)_j^i\,k(x_j^i,x) \tag{3.17}$$

where $K_x=\left[k(x_1^1,x)\ \cdots\ k(x_{n_1}^1,x)\ \cdots\ k(x_1^c,x)\ \cdots\ k(x_{n_c}^c,x_j^i)\right]^T\in R^{n\times 1}$ is a column vector, and $\alpha_k=\left[(\alpha_k)_1^1\ \cdots\ (\alpha_k)_{n_1}^1\ \cdots\ (\alpha_k)_1^c\ \cdots\ (\alpha_k)_{n_c}^c\right]$ is the $k$ th discriminant vector.

## 3.3 FKFDA Development

As shown in equation (3.17), the generation of a single feature from a sample

demands the calculation of as many kernel functions as the training samples. Thus, the efficiency of KFDA-based feature extraction procedure is inversely proportional to the size of the training sample set. Hence, large training sample set poses a serious problem for the KFDA-based feature extraction. One has to find means of fast nonlinear feature extraction. This subsection formulates a FKFDA (fast Kernel Fisher discriminant analysis) procedure to speed up feature extraction.

Different from KFDA that expresses discriminant vectors as linear combinations of all the training samples, FKFDA expresses them as linear combinations a portion of training samples, which are referred to as nodes in this thesis. The proposed FKFDA is developed based on the following idea. When the training samples are linearly combined to express the discriminant vectors in KFDA, some ones make substantially greater contributions than others. So, if we can identify the nodes that made the greatest contributions, it is possible to express the discriminant well by those nodes.

In 3.1, we define two criteria for node selection. In 3.2, we design a scheme to select nodes based on these two criteria. In 3.3, we formulate the FKFDA using nodes.

### 3.3.1 Criteria for node selection

Two criteria are defined here for node selection: (1) pseudo-eigenvalue; and (2) similarity between nodes. We can investigate how representative a sample is in the unknown kernel feature space through these two criteria. These two criteria are extracted by approximating the kernel principal component analysis (AKPCA).

*Criterion (1): pseudo-eigenvalue.*

Different from the Mercer's theorem [97] that expresses the discriminant vectors using all the training samples , our FKFDA expresses the discriminant vectors as linear combinations of nodes. Because the nodes are only a portion of the training

samples, a vector that can be linearly expressed by all the training samples is not necessarily linearly expressible using the nodes. However, if we can reconstruct all the training samples using the nodes, we can express the discriminant vectors (the linear combinations of the training samples) using the nodes. Thus, in order to approximate the discriminant vectors using the nodes as well as possible, the nodes should be the most representative training samples. This idea is similar to that of of KPCA, which seeks best representative directions in a least-square sense. If the nodes can represent all the training samples well, these nodes can replace the training samples to express the discriminant vectors.

Pseudo-eigenvalue is a scalar defined to assess how representative the related training sample is, and acts like the eigenvalue which is used to assess the corresponding feature extractor in KPCA. When selecting nodes, a training sample corresponding to a larger pseudo-eigenvalue is preferred. The pseudo-eigenvalue is defined in the following paragraphs.

In KPCA, the optimal vectors for projection are the eigenvectors of the total scatter matrix corresponding to leading eigenvalues. The total scatter matrix is defined as follows

$$S_t^{\varphi} = \sum_{i=1}^{c} \sum_{j=1}^{n_i} \varphi\left(x_j^i\right) \varphi^T\left(x_j^i\right) = XX^T \tag{3.18}$$

where we assume that the mapping samples are centered in $F$ (details of how to center the mapping samples can be found in [56]). We assess a given feature extractor in the KPCA by its eigenvalue, and the larger the eigenvalue is the more representative its corresponding eigenvector is. The eigenvalue to assess the feature extractor $\beta$ can be worked out using

$$\lambda = \frac{\beta^T S_t^{\varphi} \beta}{\beta^T \beta} \tag{3.19}$$

Based on this observation, we assess a mapping sample $\varphi(x)$ using the following Rayleigh quotient:

$$l(x)=\frac{\varphi^T(x)S_t^{\varphi}\varphi(x)}{\varphi^T(x)\varphi(x)} \tag{3.20}$$

This scalar is referred to pseudo-eigenvalue in this section. Regarding a matrix $T\in R^{n\times n}$, there are only $n$ eigenvalues defined for $n$ eigenvectors, but the pseudo-eigenvalue is defined for every vector in the $n$ dimensional vector space. If an $n$ dimensional vector $v$ is the eigenvector of $T\in R^{n\times n}$, then the pseudo-eigenvalue defined for $v$ is identical to the eigenvalue corresponding to it. In our node selection case, if one training sample $\varphi(x)$ happens to be a feature extractor of the KPCA, i.e. the eigenvector of $S_t^{\varphi}$, the criterion defined in equation (3.20) is just its eigenvalue. A larger $\lambda$ means a smaller reconstruction error in the KPCA. Similarly, a larger pseudo-eigenvalue means a more representative node.

As the nonlinear map $\varphi$ is implicit, the calculation of the scalar $l$ is not trivial. We can reformulate it as follows

$$l=\frac{\varphi^T(x)S_t^{\varphi}\varphi(x)}{\varphi^T(x)\varphi(x)}=\frac{\varphi^T(x)XX^T\varphi(x)}{\varphi^T(x)\varphi(x)}=\frac{\sum_{i=1}^{c}\sum_{j=1}^{n_i}k\left(x,x_j^i\right)^2}{k(x,x)} \tag{3.21}$$

Using equation (3.21), we can calculate the pseudo-eigenvalue for every training sample easily.

*Criterion (2): similarity between nodes.* Apart from the pseudo-eigenvalue, we must also consider the similarity between the nodes. Or else, we may select many redundant nodes. To reduce the size of the node set, we try to avoid using two similar nodes. An example in Figure 3.1 can explain this clearly. In this figure, $x_1$, $x_2$, and

$x_3$ are three of the training samples. The ellipse denotes the area where all the training samples scatter, and $x_1 = a * x_2 (a > 0)$ and $x_3$ are the eigenvectors of the total scatter matrix. The relationship between the pseudo-eigenvalues is $l(x_1) = l(x_2) > l(x_3)$. Now we want to select nodes from these three samples $x_1$, $x_2$, and $x_3$. If only the pseudo-eigenvalue is considered, $x_1$ and $x_2$ should be the first two nodes. However, they are in the same direction and only one of them is sufficient to represent the information in this direction. Though $x_3$ corresponds to a smaller eigenvalue than $x_1$ and $x_2$, along with one of $x_1$ and $x_2$, it can represent the training sample set well and express any discriminant vector exactly. Thus, the node set $\{x_1, x_3\}$ (or $\{x_2, x_3\}$) is preferred to $\{x_1, x_2\}$.

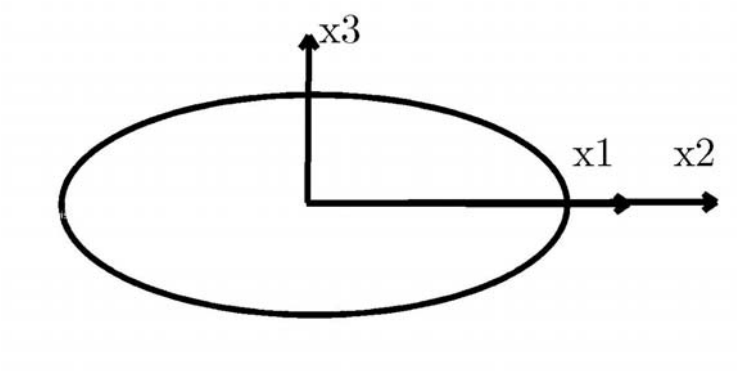



Figure 3.1 An example. $x_1$, $x_2$, and $x_3$ are three samples and the ellipse denotes the area where all the training samples scatter. The node set $\{x_1, x_3\}$ (or $\{x_2, x_3\}$) is preferred to $\{x_1, x_2\}$

We define the similarity between two mapping samples as the absolute cosine value of the angle formed by them, as follows

$$s(x_i, x_j) = \left|\cos(x_i, x_j)\right| = \frac{\left|\varphi^T(x_i)\varphi(x_j)\right|}{\sqrt{\varphi^T(x_i)\varphi(x_i)}\sqrt{\varphi^T(x_j)\varphi(x_j)}} \tag{3.22}$$

From equation (3.22), we know that the similarity between samples ranges from 0 to 1.When $s\left(x_i, x_j\right)$ is close to 1, the two mapping samples $\varphi\left(x_i\right)$ and $\varphi\left(x_j\right)$ point in almost the same direction and should not be taken as nodes simultaneously. By permitting only low degrees of similarity between nodes, we significantly reduce the number of nodes.

### 3.3.2 Node selection scheme

Using the criteria defined above, we design a scheme to select nodes. Among all the training samples, we take the one that corresponds to the largest pseudo-eigenvalue as the first node $x_1^*$. Then, we pick out the nodes one by one. The $i$ th node $x_i^*$ is the training sample that corresponds to the maximum pseudo-eigenvalue under the following constraints

$$s\left(x_i^*, x_j^*\right) < \xi \tag{3.23}$$

where $j = 1, 2, \cdots, i-1$. The selection of the $i$ th node is modeled as follows

$$\max\left|l\left(x_i^*\right)\right| \quad s.t. \quad \mathrm{s}\left(x_i^*, x_j^*\right) < \xi \quad where \quad j = 1, 2, \cdots, i-1 \tag{3.24}$$

where $0 \le \xi \le 1$ is a parameter to bound the similarity between nodes. The node selection scheme is implemented as follows.

Initially, we set the node set to be null $NS = null$, and the candidate set for nodes to be the training samples $CS = \left\{\varphi\left(x_i\right) | 1 \le i \le n\right\}$. Also, we calculate the pseudo-eigenvalue for each of the training samples. Then, we select nodes by executing the following two steps iteratively until the candidate set is null.

*Step (1).* Find the sample $\varphi\left(x_j\right)$ that corresponds to the largest pseudo-eigenvalue among the members of the candidate set $CS$; take $\varphi\left(x_j\right)$ as a

node and add it to the node set, i.e. $NS = NS \cup \{\varphi(x_j)\}$.

*Step (2).* Visit all the samples in $CS$ and delete the sample $\varphi(x_k)$ from $CS$, if $s(x_k, x_j) > \xi$.

Step 1 takes the most representative member of the candidate set as a node. Step 2 reduces the size of the candidate set by deleting the training samples that have large similarity to the newly selected node.

To summary, the above procedures generate a node set (NS) that contains the most representative nodes $\varphi(x_i)$ and does not have any pair of nodes $\varphi(x_i)$ and $\varphi(x_j)$ satisfying $s(x_i, x_j) > \xi$, where $0 \le \xi \le 1$ is a parameter.

### 3.3.3 FKFDA formulation using nodes

In subsection 3.3.2, we pick out a set of nodes that can represent the training sample set well. In FKFDA, these nodes replace the training samples to linearly express the discriminant vector $\phi$ in the kernel feature space, as follows

$$\phi = \sum_{t=1}^{h} \alpha_t^* \varphi(x_t^*) = X^* \alpha \tag{3.25}$$

where the number of nodes $h$ is smaller than $n$ and $x_t^* \in \{x_j^i \mid 1 \le i \le c, 1 \le j \le n_i\}$ is a node selected from the training sample set. The matrix $X^* = [\varphi(x_1^*) \quad \varphi(x_2^*) \quad \cdots \quad \varphi(x_h^*)]$ is the node matrix. Substituting equations (3.7), (3.10), and (3.25) into equation (3.2), the Fisher criterion is converted to

$$J^{\varphi}(\alpha) = \frac{\alpha^T X^{*T} XPX^T X^* \alpha}{\alpha^T X^{*T} XQX^T X^* \alpha} = \frac{\alpha^T K^* PK^{*T} \alpha}{\alpha^T K^* QK^{*T} \alpha} = \frac{\alpha^T M^* \alpha}{\alpha^T N^* \alpha} \tag{3.26}$$

where

$$K^* = X^{*T} X = \left[ K_1^* \quad K_2^* \quad \cdots \quad K_c^* \right] \in R^{h \times n}$$
$$\left( K_i^* \right)_{gf} = \varphi^T \left( x_g^* \right) \varphi \left( x_f^i \right) = k \left( x_g^*, x_f^i \right) \quad 1 \le i \le c \quad 1 \le g \le h \quad 1 \le f \le n_i \tag{3.27}$$
$$M^* = K^* P K^{*T} \in R^{h \times h}, N^* = K^* Q K^{*T} \in R^{h \times h}$$

We can see the advantage of expressing the within-class and between-class scatter matrix as equations (3.7) and (3.10). Separating the sample matrix $X$ as a multiplier makes the formulation of FKFDA easier. If the scatter matrices are expressed as the formulas presented in [54, 57, 64, 88, 90, 95], this FKFDA can be constructed but only in a much more complex way. This is because the scatter matrices are computed through a series of computations and the sample matrix $X$ is not separated as a multiplier explicitly. In addition, this formulation procedure of FKFDA also makes the mathematical relationship between the Fisher criterions defined in equations (3.13) and (3.26) clearly; only the sample matrix $X$ is replaced by the node matrix $X^*$ in some places.

The vectors $\alpha$ that maximize the Fisher criterion defined by equation (3.26) are the discriminant vectors. They are the eigenvectors corresponding to the maximum eigenvalues of the following generalized eigenequation.

$$M^* \alpha_i = \lambda N^* \alpha_i \tag{3.28}$$

If $N^*$ is a nonsingular matrix, the discriminant vectors will be the eigenvectors corresponding to the maximum eigenvalues of the following conventional eigenequation:

$$\left( N^* \right)^{-1} M^* \alpha_i = \lambda \alpha_i \tag{3.29}$$

The $k$th feature of sample $x$ corresponding to the discriminant vector $\alpha_k$ should be

$$y_k = \alpha_k^T K_x^* = \sum_{i=1}^{h} \left( \alpha_k \right)_i k \left( x_i^*, x \right) \tag{3.30}$$

where $K_x^* = \left[ k\left(x_1^*, x\right) \quad \cdots \quad k\left(x_h^*, x\right) \right]$ and $\alpha_k = \left[ \left(\alpha_k\right)_1 \quad \left(\alpha_k\right)_2 \quad \cdots \quad \left(\alpha_k\right)_h \right]^T$.

## 3.4 Discussion

In this subsection, we discuss the computational complexity and singular generalized eigenequation problem of our FKFDA. Figure 3.2 shows the relationship among FDA, KFDA, and FKFDA. Both KFDA and FKFDA, formulated in the kernel feature space induced by the kernel function, are the nonlinear generalizations of the FDA. While the discriminant vectors are linearly expressed by all the training samples in the KFDA, they are linearly expressed only by the most representative ones in our FKFDA.

The node selection procedure of our method is quite different from the previous methods. The methods [89, 92-93] associate every sample with a column of the matrix $K$, where $K_{ij} = k\left(x_i, x_j\right)$ and focus on the approximation of this matrix using its submatrix. Then, the samples corresponding to the submatrix are taken as the nodes. The method in [89] minimizes the difference between $K$ and its approximation in terms of Kronecker metric. Taking into consideration both discrimination and generalization, the method in [92] minimizes a new criterion to generate the approximation for $K$. The method in [93] selects the submatrix randomly and approximates $K$ based on Nystrom theorem [98]. Instead of focusing on the approximation of discriminant vectors, all of these methods aim to generate a well approximation for the matrix $K$. However, it is not necessary that the selected nodes can replace all the training samples to express discriminant vectors.

Based on the derivation in Figure 3.2, this subsection first calculates the computational complexities of the proposed FKFDA, and then concludes through

analysis that the singular generalized eigenequation problem does not occur in the FKFDA under some conditions.

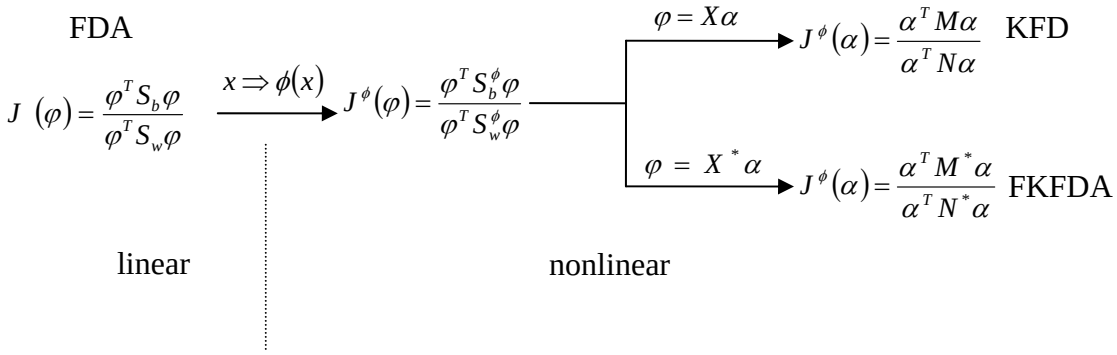



Figure 3.2 The relationship between FDA, KFDA, and the proposed FKFDA

### 3.4.1 Computational complexity

The training procedure consists of two phases: node selection and FKFDA formulation. We will analyze both. Table 3.1 presents the training computational complexity of different methods.

Firstly, we calculate the computational complexity of node selection scheme described in section 3.3.2. To initialize, we calculate the scalar $l$ for all samples with computational complexity of $\mathrm{O}(n^2)$, where $n$ is the number of training samples. In $h$ iterations, we pick out $h$ nodes. In each iteration, we visit no more than $n$ training samples with computational complexity no more than $\mathrm{O}(n)$. Thus, the node selection algorithm has the computational complexity of $\mathrm{O}(nh+n^2)$.

Then, we use equation (15) to formulate the FKFDA and (17) to compute the discriminant vectors. The computational complexity of computing the scatter matrices and discriminant vectors are respectively $\mathrm{O}(nh)$ and $\mathrm{O}(h^3)$.

In all, the computational complexity of the training procedure of the FKFDA is

$\mathrm{O}\left(nh+h^3+n^2\right)$. In our experiments, the number of nodes $h$ is much smaller than the number of samples $n$ ($h$ is even smaller than 2% of $n$ in some cases). Therefore the computational complexity of the training procedure of the FKFDA approximates to $\mathrm{O}\left(n^2\right)$. As the training procedure of naive KFDA has computational complexities of $\mathrm{O}\left(n^3\right)$, our FKFDA is faster.

Table 3.1 Training computational complexity of different methods

| Procedure | Method in [88] | Method in [89] | Method in [90-92] | Proposed method |
|---|---|---|---|---|
| Calculating the scalar $l$ | Null | Null | Null | $\mathrm{O}\left(n^2\right)$ |
| Selecting nodes | $\mathrm{O}\left(n^3\right)$ | $O(n^2h^2)$ | $O\left(hn^3\right)$ | $\mathrm{O}\left(nh\right)$ |
| Generating Fisher criterion | $\mathrm{O}\left(nh\right)$ | $\mathrm{O}\left(nh\right)$ | $\mathrm{O}\left(nh\right)$ | $\mathrm{O}\left(nh\right)$ |
| Producing the discriminant vectors | $\mathrm{O}\left(h^3\right)$ | $\mathrm{O}\left(h^3\right)$ | $\mathrm{O}\left(h^3\right)$ | $\mathrm{O}\left(h^3\right)$ |
| Total | $\mathrm{O}\left(h^3+n^3+nh\right)$ | $\mathrm{O}\left(h^3+n^2h^2+nh\right)$ | $\mathrm{O}\left(h^3+n^3h+nh\right)$ | $\mathrm{O}\left(nh+h^3+n^2\right)$ |

Further, the following analysis shows that the state-of-the-art node selection methods [88-92] are much slower than that in FKFDA. The node selection method in [88] involves a matrix decomposition with computational complexity of $\mathrm{O}\left(n^3\right)$. After picking out $i$ nodes, the method in [89] chooses a node only from a subset of the rest $n-i$ training samples and applies the greedy algorithm. If the subset consists of $j$ samples, the computational complexity of the greedy algorithm is $O(nh^2j)$. To obtain a representative node set, the size of the subset should be proportional to the size of the training sample set, i.e. $j=\gamma n\left(0<\gamma<1\right)$. Thus, the

computational complexity of the method can be considered as $O(n^2h^2)$. To select the $i$th node, the methods [90-92] have to perform matrix inversion at least once with computational complexity $O\left(i^3\right)$ and a series of matrix multiplications with computational complexity $\mathrm{O}\left(n^3\right)$. In all, to select $h$ nodes, the computational complexity of the node selection procedure in [90-92] is $O\left(hn^2\right)+\sum_{i=1}^{h}O\left(n^3\right)=O\left(hn^3\right)$.

Based on these observations, we can say that the proposed FKFDA is indeed faster than the naive KFDA and the other algorithms proposed in [88-92].

### 3.4.2 Singular generalized eigenequation problem

KFDA often suffers from the SSS (small sample size) problem in real-world applications [99-100]. In this subsection, we analyze the singular generalized eigenequation problem in our FKFDA. From the definition of the within-class scatter matrix equation (3.3), we know that $rank\left(S_w^{\varphi}\right)\le n-c$, where $n$ is the number of samples and $c$ is the number of classes. Theoretically, the matrix $N=X^TS_w^{\varphi}X\in R^{n\times n}$ in KFDA cannot be nonsingular, as $rank\left(N\right)\le rank\left(S_w^{\varphi}\right)\le n-c$. In other words, the singular generalized eigenequation problem is sure to occur in KFDA.

The singular generalized eigenequation problem does not occur in our FKFDA under some conditions. The matrix $N^*$ is computed using the formula $N^*=X^{*T}S_w^{\varphi}X^*\in R^{h\times h}$ in the FKFDA. The inequality $rank\left(N^*\right)\le rank\left(S_w^{\varphi}\right)=n-c$ is rarely useful here, as the number of nodes $h$ (the maximum value of $rank\left(N^*\right)$)

is always much smaller than $n-c$ in our experiments. Though the rank of the $N^*$ can theoretically vary from 0 to $h$, this matrix $N^*$ is indeed nonsingular when the nodes satisfy the following two conditions: (1) the nodes are linearly independent in the kernel feature space; (2) the linear combinations of the nodes do not lie in the null space of $S_w^\varphi$. As the similarity between the nodes is considered in the node selection procedure, the nodes can be regarded as linearly independent. Whether a node lies in the null space of $S_w^\varphi$ can be checked out easily. If $\varphi(x)^T S_w^\varphi \varphi(x) = 0$ holds, the sample $\varphi(x)$ lies in the null space of $S_w^\varphi$; otherwise, the sample $\varphi(x)$ does not lie in the null space of $S_w^\varphi$. Mathematically, we cannot prove that all the linear combinations of the nodes do not lie in the null space of $S_w^\varphi$. In our experiments, however, the matrix $N^*$ is tested to be nonsingular. Even if it is singular in other applications, we can take this into consideration when selecting nodes: if the inclusion of one node makes $S_w^\varphi$ singular, we can delete it from the candidate set and select a new node. The matrix $N^*$ is nonsingular and the generalized eigenequation (17) can be converted into the conventional eigenequation (18), which means that the proposed FKFDA scheme avoids the singular generalized eigenequation problem.

## 3.5 Experimental results

To test the proposed method, this subsection not only performs experiments on two face databases, but also on seven UCI datasets. We compare the performances of our FKFDA and five other methods, specifically, two classic nonlinear feature extraction methods (KFDA [64] and KPCA [55]) and three accelerated nonlinear feature extraction methods proposed in [89], [92], and [93]. Subsection 3.5.1 describes the experiments on AR and YaleB Face databases. Subsection 3.5.2

describes the experiments on the UCI datasets. This experiment validates the feasibility of FKFDA as a general nonlinear feature extraction method, not only for face images. Subsection 5.3 discusses parameter settings for FKFDA.

### 3.5.1 Face databases

Figure 3.3 Sample images from the AR database

The AR face database [11] contains 3 120 images corresponding to the faces of 120 people. The images include frontal view faces with the following 13 different conditions: 1. neutral expression, 2. smile, 3. anger, 4. scream, 5. wearing sun glasses, 6. wearing scarf, 7. left light on, 8. wearing sun glasses and left light on, 9. wearing scarf and left light on, 10. right light on, 11. wearing sun glasses and right light on, 12. wearing scarf and right light on, 13. all sides lights on; second sessions repeated same conditions. Each person participated in two sessions, separated by intervals of two weeks. The same pictures were taken in both sessions. All the images of this database are used. Figure 3.3 depicts ten face images of the AR database.

The Yale Face Database B [101] contains 5 850 images, 585 images for each of 10 individuals. The images of one individual are captured under 9 different poses and illumination conditions. Also, an image with ambient illumination was captured under all of the 9 different poses for each individual. All the images of this database are used. Figure 3.4 depicts ten face images of the YaleB database.

Figure 3.4 Sample images from the yaleB database

In both AR and YaleB face databases, we separate the face images of each person in two halves, one half for training and one for testing. Thus, the size of the training set for these two databases are respectively 1 560 and 2 925. Two popular kernel functions are used in our experiments. One is the polynomial kernel $k(x,y)=(x^T y+1)^2$ and the other is the Gaussian kernel $k(x,y)=\exp(-\|x-y\|/\sigma^2)$. We first calculate the variances of all the data components in the training sample set and set $\sigma^2$ to be the sum of these variances. This method can set the parameter of the kernel function automatically, and it was used in [91-92]. We trained all of the five methods 20 times with random training sets and calculated the classification accuracy of each training set and testing set pair.

The similarity between each pair of training samples can be worked out easily using (3.22), the parameter $\xi$ in the FKFDA (in step 2 of the node selection procedure) is set to be the mean value of all these similarities. With this setting, the node selection procedure generates node sets of size 371 and 394 for the AR face database when the polynomial and Gaussian kernel functions are respectively used. These nodes account for 23.78% and 25.26% of all the training samples. For the YaleB database, corresponding to the polynomial and Gaussian kernel functions, the node sets (with size of 316 and 304) respectively accounts for 10.80% and 10.39% of all the training samples. To extract a feature from a sample, the naïve kernel methods calculate as many kernel functions as the training samples and our method only

calculates as many as nodes. Hence, our method is much more efficient in nonlinear feature extraction.

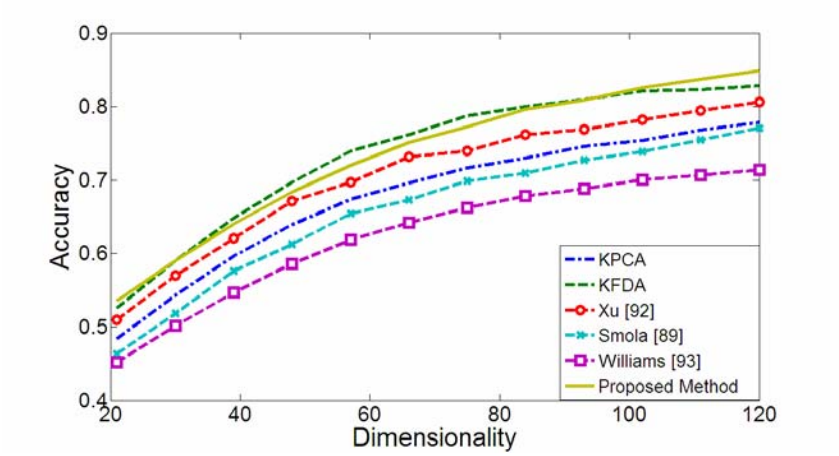
Table 3.2 The number of nodes captured under different conditions

| Different conditions | 1. neural expression | 2. smile | 3. anger | 4. scream | 5. sun glasses | 6. scarf |
|---|---|---|---|---|---|---|
| Polynomial | 29 | 27 | 30 | 28 | 30 | 31 |
| Gaussian | 30 | 27 | 32 | 29 | 33 | 32 |

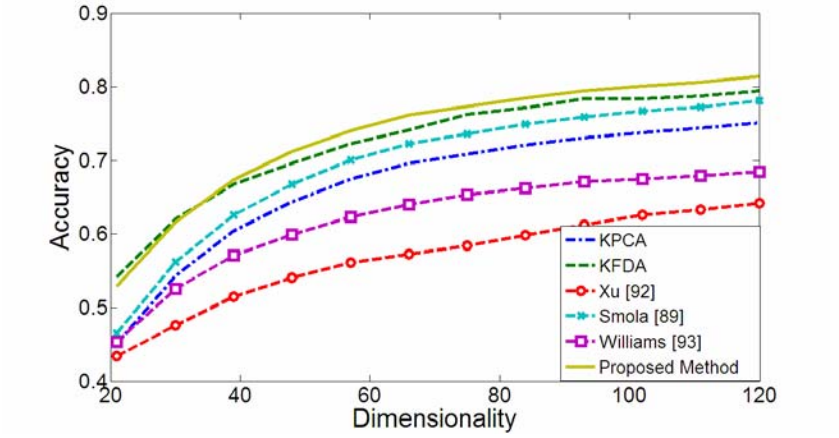
| Different conditions | Left light on | | | Right light on | | | 13. all sides lights on |
|---|---|---|---|---|---|---|---|
| | 7. | 8. sun glasses | 9. scarf | 10. | 11. sun glasses | 12. scarf | |
| Polynomial | 32 | 27 | 26 | 31 | 25 | 27 | 28 |
| Gaussian | 31 | 28 | 27 | 30 | 31 | 30 | 34 |

Table 3.2 shows the number nodes captured under different conditions in the AR database. We can see that the numbers of nodes under the 13 conditions are almost the same. They are no larger than 34 and no less than 25. This indicates that the images captured under different conditions are equally important in expressing the discriminant vectors. We also find that all the nodes corresponding to large combination weights for at least one naïve KPCA's basis vector in terms of absolute value, though they do not necessarily correspond to the largest weights. There are many samples corresponding to large combination weights are not selected as nodes, mainly because they are similar to some previously selected nodes.

The feature extraction results are classified using the nearest neighbor classifier and in Figure 3.5 and Figure 3.6 the classification accuracy is plotted against the number of features, i.e. the dimensionality of the feature for each sample. We plot the time of node selection procedure against the number of nodes in Figure 3.7. We also list the training time of different methods in.




(a)



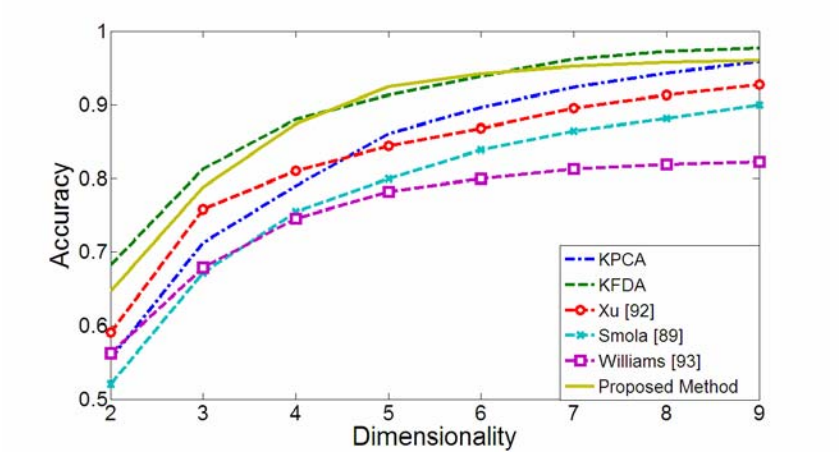

(b)

Figure 3.5 Accuracy versus number of features on AR database:
(a) polynomial kernel, (b) Gaussian kernel

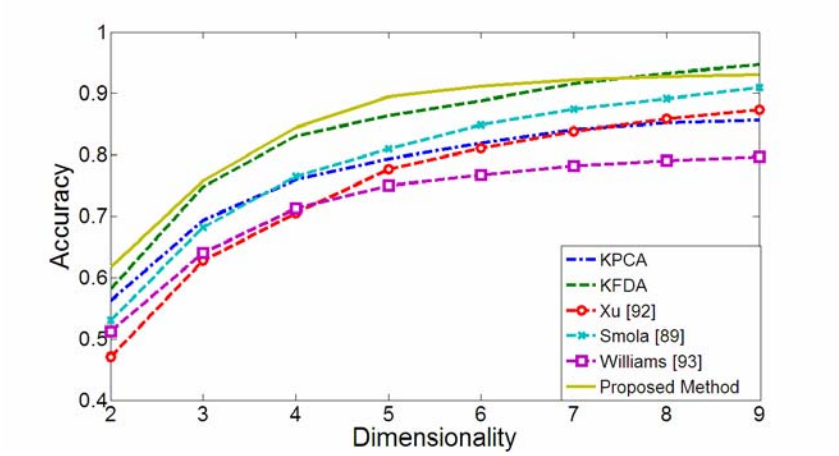
In Figure 3.5 and Figure 3.6, the proposed method can achieve comparable classification accuracy with these of the naïve KFDA. This indicates that the selected nodes can replace all the training samples to express the discriminant vectors well. However, when tested on these two face databases, the methods in [89], [92], and [93], which only consider the kernel matrix when selecting nodes, cannot generate satisfying discriminant vectors and their classification accuracies are usually smaller.

Table 3.3 Training time of different methods ($\times 10^5$ seconds)

| | KPCA | KFDA | Smola [89] | Xu [92] | Williams [93] | Proposed method |
|---|---|---|---|---|---|---|
| AR | 0.027 | 0.065 | 0.506 | 2.369 | 0.147 | 0.034 |
| YaleB | 0.324 | 0.380 | 2.780 | 8.133 | 1.925 | 0.460 |




(a)

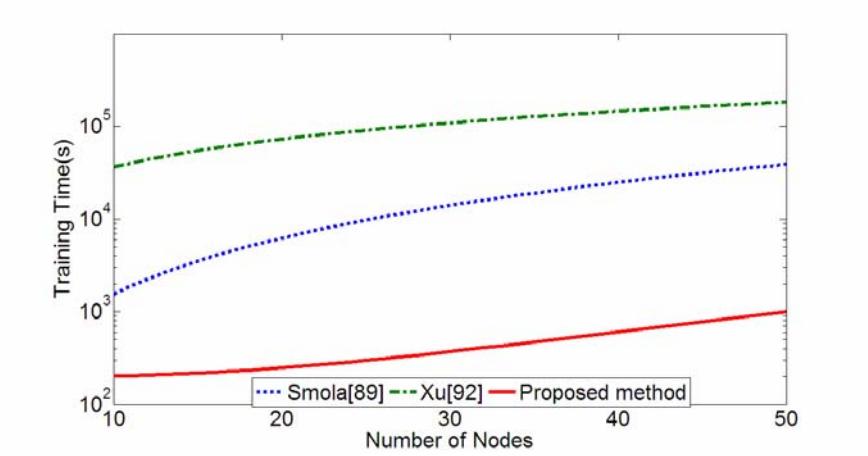



(b)

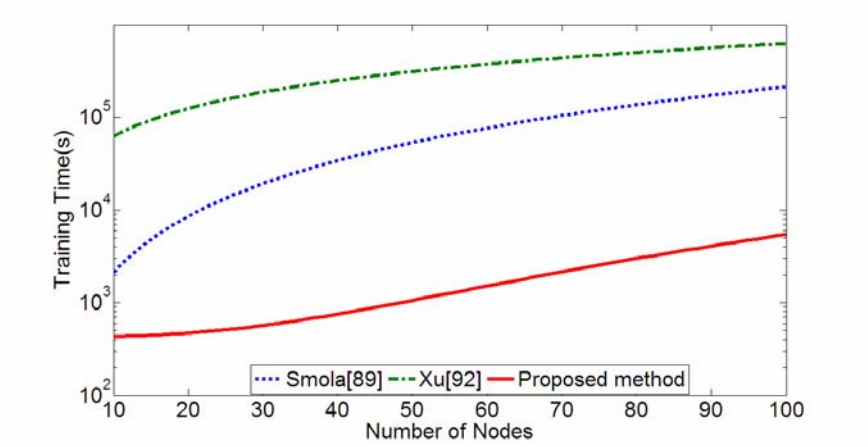
Figure 3.6 Accuracy versus number of features on YaleB database:
(a) polynomial kernel, (b) Gaussian kernel

Figure 3.7 shows that the proposed node selection procedure is much faster than the procedures in [89] and [92]. To select 50 nodes from the AR database and 100 nodes from the YaleB database, our method is 10 times faster than these two methods. As [93] selects nodes randomly, we don't compare its node selection procedure with the proposed method.

Table 3.3 shows that the proposed method has a faster training procedure than [89, 92-93]. Though the proposed method is slower than naïve KFDA and KPCA in training procedure in some cases, its feature extraction efficiency is more than 4 times faster than them.




(a)




(b)

Figure 3.7 Node selection time of different methods on face database: (a) AR; (b) YaleB

In our experiments, the proposed FKFDA has a nonsingular matrix $N^*$ and avoids the singular generalized eigenequation problem. This enables us to simplify the computation significantly by converting the generalized eigenequation into a conventional eigenequation. To overcome the singular generalized eigenequation

problem in KFDA, the regularization method in [64] is adopted.

Table 3.4 Number of training samples and of nodes and their ratios on UCI datasets
( $N$ : Number of training samples; $n_1$ : Number of nodes with ploynomial kernel;
$n_2$ : Number of nodes with kernel function)

| dataset | Car | dermatology | ecoli | Glass | iris | Wine | Zoo |
|---|---|---|---|---|---|---|---|
| $N$ | 864 | 183 | 168 | 107 | 75 | 178 | 55 |
| $n_1$ | 56 | 11 | 14 | `2 | 3 | 11 | 5 |
| Ratio( $n_1 / N$ ) | 6.48% | 6.01% | 8.33% | 1.87% | 4.00% | 6.18% | 9.09% |
| $n_2$ | 41 | 13 | 16 | 2 | 3 | 14 | 5 |
| Ratio( $n_2 / N$ ) | 4.85% | 7.10% | 9.52% | 1.87% | 4.00% | 7.87% | 9.09% |

### 3.5.2 UCI datasets

We also conduct experiments on seven UCI datasets (http://archive.ics.uci.edu/ml/). We randomly divide each dataset into two halves and then test the methods with the first half as the training set and the second half as the testing set. We repeat this training and testing procedure 20 times for each dataset. Because every test subset corresponds to one classification accuracy, we can figure out the average classification accuracy and take that as the classification accuracy of the whole dataset. In these experiments, we use the polynomial kernel $k(x, y) = (x^T y + 1)^2$ and Gaussian kernel $k(x, y) = \exp(-\|x - y\| / \sigma^2)$ . The parameters are fixed in the same way as in the face recognition experiments.

Table 3.5 Classification accuracy (%) with polynomial kernel function on UCI datasets

| | | KFDA | KPCA | Smola [89] | Xu [92] | Williams [93] | Proposed |
|---|---|---|---|---|---|---|---|
| car | Polynomial | 96.99 | 96.32 | 96.54 | 94.35 | 93.26 | 97.02 |
| | Gaussian | 96.51 | 96.32 | 96.24 | 94.35 | 93.27 | 96.53 |
| dermatology | Polynomial | 91.28 | 82.07 | 89.16 | 90.74 | 87.42 | 92.93 |
| | Gaussian | 90.32 | 81.52 | 88.82 | 87.74 | 86.55 | 91.65 |
| ecoli | Polynomial | 85.88 | 86.47 | 88.69 | 83.53 | 82.15 | 89.88 |
| | Gaussian | 83.53 | 78.24 | 85.43 | 84.71 | 82.36 | 87.06 |
| glass | Polynomial | 93.58 | 97.12 | 96.23 | 89.91 | 87.58 | 97.17 |
| | Gaussian | 93.58 | 96.25 | 94.73 | 87.16 | 90.65 | 97.33 |
| iris | Polynomial | 96.00 | 97.33 | 95.87 | 94.67 | 93.07 | 97.33 |
| | Gaussian | 97.33 | 97.33 | 97.33 | 96.00 | 94.36 | 97.33 |
| wine | Polynomial | 72.61 | 68.89 | 83.15 | 83.55 | 80.01 | 84.78 |
| | Gaussian | 94.44 | 68.89 | 88.40 | 73.33 | 86.50 | 94.57 |
| zoo | Polynomial | 96.18 | 94.23 | 96.73 | 90.38 | 92.16 | 98.84 |
| | Gaussian | 96.15 | 86.54 | 93.53 | 92.31 | 90.12 | 97.69 |

Table 3.4 lists the ratios of nodes to training samples. We can see that the smallest ratio is only 1.87%, and the largest is 9.52%. The lower the ratio is, the more efficient the feature extraction procedure of the proposed method is. As the ratio is always less than 10%, the feature extraction procedure of the proposed method is more than 10 times faster than that of the conventional kernel methods.

Table 3.5 shows the average recognition rates when using respectively the polynomial kernel function and the Gaussian kernel function. The classification accuracy of the proposed method is larger than, or at least comparable with, the other methods.

### 3.5.3 Discussion on the parameter

In the node selection algorithm, if one sample $\varphi(x_j)$ is selected as a node, all the samples $\varphi(x_k)$ that satisfy $\left|\cos(x_k, x_j)\right| > \xi$ will be deleted from the node candidate set and have no further opportunity to be taken as a node. Thus, the parameter $\xi$ is the maximum similarity that two nodes could have. In the previous experiments, we set it to be the mean value of all the similarities among pairs of training samples. With this setting, the designed node selection algorithm can generate an appropriate node set for the FKFDA construction. In this subsection, we will discussion how the parameter $\xi$ affects the performance of the designed scheme.

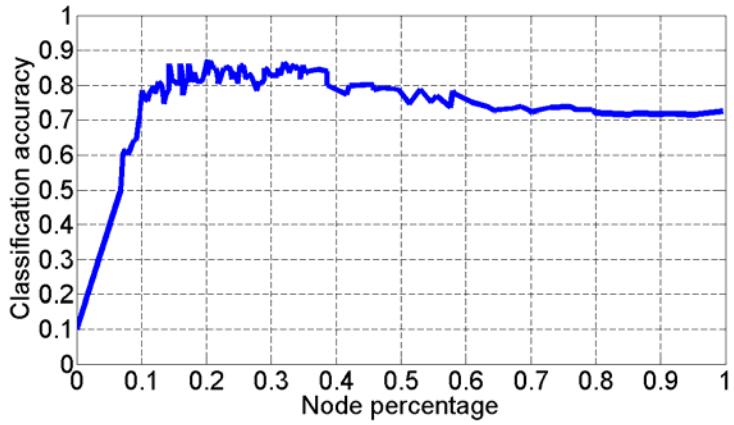
If the parameter $\xi$ is set to its minimum value, i.e. zero, the generated node set would consist of only one sample. When we select the first node, we will delete all the other samples from the node candidate set at the same time. Oppositely, if we set the parameter $\xi$ to its maximum value, i.e. one, the generated node set would consist of all the training samples and no samples at all would be deleted from the node candidate set. The node selection algorithm does nothing but sorts the samples in descending order based on the pseudo-eigenvalue. In this case, the FKFDA degenerates into the naive KFDA procedure that uses all the training samples in its construction. Generally, the node set generated by a larger $\xi$ will contain no fewer nodes than the node set that is generated by a smaller $\xi$.

Along with interpreting the second node selection criterion (similarity between

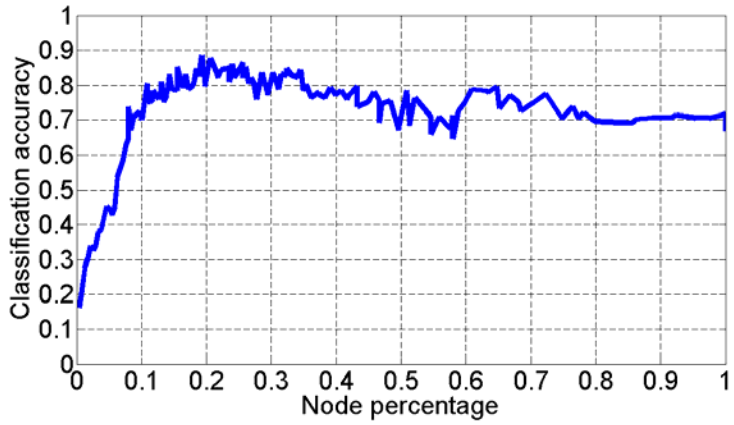
nodes) from the KPCA point of view, we can interpret it intuitively as follows. The class labels of two samples $\varphi(x_j)$ and $\varphi(x_k)$ that satisfy $\left|\cos(x_k, x_j)\right| > \xi$ may or may not be the same. If they are from the same class, then one of them will suffice to represent the information of this class in this direction. If $\varphi(x_j)$ and $\varphi(x_k)$ are from different classes, neither $\varphi(x_j)$ nor $\varphi(x_k)$ are promising discriminant vectors as the samples from these two classes may mix together after projection on their directions. However, to decrease the reconstruction error, we take one of them as a node. In both of these two cases, we take the one that corresponds to the larger pseudo-eigenvalue as node. In a word, it does not matter whether $\varphi(x_j)$ and $\varphi(x_k)$ are from the same class or not, it is still wise to take only one of them as a node if $\left|\cos(x_k, x_j)\right| > \xi$.

In our experiments, we vary $\xi$ to generate different numbers of nodes and formulate the proposed FKFDA using these nodes. Normally, how the parameter $\xi$ affects the number of nodes is application dependent. Here, we limit the focus on how the number of nodes affects the classification accuracy, other than on how $\xi$ affects the number of nodes. By manually varying $\xi$, we can generate a node set as small as including only few samples and a node set as large as including almost all the training samples. For the AR face databases, we take half of the face images of each person as training samples and the other half as testing samples. Both polynomial and Gaussian Kernel functions are used. In Figure 3.8, we plot the average classification accuracy of ten times running on the AR face databases versus $p$, the percentage of nodes accounting for all the training samples. As can be seen from these figures, the classification accuracy increases sharply with the increase of $p$. However, the

classification accuracy decreases for the larger $p$. This indicates that discriminant vectors expressed by too many training samples do not necessarily generate well separated nonlinear features. Usually, the classification accuracy achieves its maximum when $p \in (15\%, 30\%)$. This can explain why the proposed method can outperform the naive KFDA in some cases. The curves are not smooth, which means that the accuracy can change markedly for an additional node. Thus, it is necessary to design node selection methods.




(a)




(b)

Figure 3.8 Classification accuracy vs. node percentage on AR database:
(a) polynomial kernel, (b) Gaussian kernel

## 3.6 Conclusion

When KFDA extracts a feature from a sample, it should calculate as many kernel functions as there are training samples. Therefore, KFDA is not applicable to cases where the training sample set is large. In this section, we proposed a FKFDA for fast nonlinear feature extraction. Our FKFDA consists of two phases: node selection and FKFDA formulation. In the node selection phase, we presented two criteria: pseudo-eigenvalue and similarity between nodes generated by approximating kernel principal component analysis. The nodes selected by our scheme correspond to large pseudo-eigenvalues and the similarities between them are small. As large pseudo-eigenvalues mean small reconstruction errors, these nodes can represent the training samples well. FKFDA expresses the discriminant vectors as the linear combinations of these nodes, instead of expressing them as the linear combinations of all the training samples as in KFDA. In the FKFDA formulation phase, a novel derivation for the scatter matrices simplifies the formulation procedure and clarifies the mathematical relationship between KFDA and FKFDA. The scatter matrices in FKFDA are much smaller than those in naive KFDA. These smaller scatter matrices not only reduce the training computational complexity but also avoid the singular generalized eigenequation problem in our experiments. Additionally, to extract a feature from a sample, our FKFDA only needs to calculate as many kernel functions as there are nodes, instead of as many as all the training samples in the KFDA. In all, our FKFDA has a lower computational complexity than KFDA both in the training and the testing procedure and avoid the singular generalized eigenequation problem that occurs in KFDA.

In future, we would like to investigate the relationship between the nodes and the eigenfaces generated by PCA. Also, we wish to consider the parameter $\xi$, the

maximum allowable similarity between nodes, which plays an important role in the node selection scheme. In the experiments, we set it to be the mean value of all the similarity between pairs of training samples according to our experience. However, the optimal value of this parameter $\xi$ is yet unidentified. The optimal parameter may vary in different cases. We will analysis how this parameter affects the number of nodes and in turn the classification accuracy.

# Chapter 4 Pose-invariant feature extraction

## 4.1. Introduction

Face recognition is motivated by the potential applications as well as the fundamental challenges in the past decades [102-105]. One of the most popular methods for face recognition is the appearance-based method [28, 45, 57, 106-108]. Appearance based method represents a $w$-by-$h$ face image by a vector in the $wh$-dimensional image space. It is concluded that the dimensionality of the face space is too high to allow robust face recognition [109], due to the variations of pose [110], occlusion [104], and illumination [111].

A number of algorithms are proposed to recognize face images from different poses [112-117]. Multidimensional scaling (MDS)-based approach performs well to recognize low resolution probe face images using high resolution gallery images [112]. Random Regression Forests [113] can estimate the head pose of different face images. To overcome large pose variation challenges, the ensemble based approach [114] boosts linear tied factor analysis (TFA) models and achieve high accuracy. The method [117] stabilizes regressor against the pose difference and uses it to recognize face images under different poses. Both active appearance models-based landmarkings [116] and domain frequency based holistic features [115] are also proven to be helpful to solve the pose problem.

Though face images are high dimensional vectors, they are proven to reside on a low dimensional submanifold [45, 50, 118-120]. To understand the submanifold, many manifold learning techniques have been developed, which include marginal Fisher analysis (MFA) [121], neighborhood preserving embedding (NPE) [122], local

discriminant embedding (LDE) [123], etc. Given face images in a high-dimensional space, these methods can extract the geometric properties of the images, such as intrinsic dimensionality, connected components, Euclidean embedding, etc [46].

Manifold learning methods [45-46, 50, 118-123] model face images of different individuals on a single manifold, which can efficiently identify the holistic structure of the original face images. However, these methods also eliminate very important discriminative information when learning the submanifold. Considering face images as points in a high dimensional space, we can regard the face images of each individual to span a linear manifold. The difference among these manifolds is a kind of discriminative information that is eliminated when modeling face images on a single manifold.

This section presents a new appearance-based method for face recognition across pose by modeling face images of different individuals on different linear manifolds. Based on the idea that a linear manifold can be characterized by its norm vector, an orthogonal discriminant vector (ODV) for each manifold is defined and used to discriminate face images associated with this manifold from other images. (This new definition of ODV is different from what is described in [124-126], where orthogonal discriminant vectors are a set of mutually orthogonal vectors that maximize the Fisher criterion.) An ODV associated with one manifold is orthogonal to the face images on this manifold and not orthogonal to the rest face images. Association with different set of ODVs indicates the difference among linear manifolds.

The following lists the major contributions of this section:

1. The introduction of a scheme to evaluate the intrapersonal relationship among face images with different poses by comparing an individual face image under a new pose with his/her gallery face images. The result of

theoretical analysis shows that the face image under a new pose lies on the linear manifold spanned by his/her gallery face images despite of the change of poses.

2. The proposal of an identity-dependent and pose-invariant feature for face recognition across pose based on the intrapersonal relationship among face images. The new feature is the orthogonality to ODVs.
3. The development of a comprehensive procedure to examine the existence of ODV in face recognition and implementation of an effective two-step algorithm to calculate ODVs.

The remaining of this section is organized as follows. Subsection 4.2 highlights the fundamental issues of the theoretical analysis to support the proposed algorithm while Subsection 4.3 describes the new ODV-based face recognition method. The experiments and performance evaluation are reported in Subsection 4.4. Finally, the conclusion and further discussion are presented in Subsection 4.5.

## 4.2. Intrapersonal relationship among face images across pose

We assume that, for each individual, only a few gallery 2-D face images under different poses are known and the probe face images are captured under novel poses. By predicting the face image under a novel pose using gallery face images, this subsection investigates the intrapersonal relationship among face images across pose. The prediction task takes the latent 3-D face object where the intrapersonal relationship among face images originates from as a medium. If the face images are well aligned, this subsection draws the conclusion that one's probe face image under a novel pose can be linearly expressed using his/her gallery face images.

Subsection 4.2.1 and 4.2.2 present two procedures in reverse directions: 2-D image generation under a certain pose from the 3-D face and 3-D face reconstruction

using the gallery face images. These two procedures are respectively indicated by a) and b) in Figure 4.1. With the help of these two procedures, subsection 4.2.3 predicts a 2-D image under a novel pose, indicated by c) in Figure 4.1. Subsection 4.2.4 analyzes the prediction result and reveals the intrapersonal relationship among face images.

### 4.2.1 From 3-D face to 2-D image

Regarding the 3-D surface of a specific face as Lambertian [127], the intensity of the surface point $(x, y, z)$ under a given lighting source $\vec{s}$ can be computed as follows

$$F(x, y, z) = \rho(x, y, z)\cos\alpha \tag{4.1}$$

where $\alpha$ is the angle formed by lighting direction $\vec{s}(x, y, z)$ and normal $\vec{n}(x, y, z)$, $\rho(x, y, z)$ is the albedo of given point. After ordering the intensities of surface points lexicographically, we represent a 3-D face by a vector $F \in R^{N_3 \times 1}$, where the scalar $N_3$ denotes the number of pixels. For a fixed pose, we can derive the $N_2$ dimensional 2-D face image $f_0 \in R^{N_2 \times 1}$ from the 3-D face $F$ by selecting the visible points (Figure 4.1 (a)). It is proved to be a linear orthogonal projection procedure from $F$ to $f_0$ [127-128], and can be expressed as follows

$$f_0 = V_0 F \tag{4.2}$$

where $V_0 \in R^{N_2 \times N_3}$ is the pose-dependent projection operator. To achieve the goal of dropping all invisible points from $F$ and only keeping all the visible ones in the face image $f_0$, the elements of $V_0$ are set as follows: if the $j$ th pixel in $F$ is invisible, $(V_0)_{ij} = 0$ for $1 \le i \le N_2$; or else, if it is visible and projected as the $i$ th pixel of $f_0$ ,

$(V_0)_{ij} = 1$ and $(V_0)_{kj} = 0 (1 \le k \le N_2, i \ne k)$.

### 4.2.2 From 2-D image to 3-D face

Assume that $n$ face images $f_1, f_2, \cdots, f_n$ under different poses $V_1, V_2, \cdots, V_n$ are generated from the same 3-D face $F$

$$f_i = V_i F \quad (i = 1, 2, \cdots, n) \tag{4.3}$$

If there is one point in $F$ which is invisible in any of these images, the reconstruction of $F$ from $f_1, f_2, \cdots, f_n$ will be theoretically ill-posed. However, based on the observation that

$$f = VF \tag{4.4}$$

where the matrix $f$ consists of all the face images $f = \left[ f_1^T \quad f_2^T \quad \cdots \quad f_n^T \right]^T \in R^{n*N_2 \times 1}$ and the matrix $V$ consists of all the pose-dependent projections $V = \left[ V_1^T \quad V_2^T \quad \cdots \quad V_n^T \right]^T \in R^{n*N_2 \times N_3}$, the 3-D face can be estimated as

$$\begin{aligned} \overline{F} &= \overline{V}^+ \left( V^T f \right) \\ &= \overline{V}^+ \left( V^T V F \right) = \overline{V}^+ \left\{ \left( \sum_{i=1}^{n} V_i^T V_i \right) F \right\} = \overline{V}^+ \overline{V} F \end{aligned} \tag{4.5}$$

where both $\overline{V} = \sum_{i=1}^{n} V_i^T V_i \in R^{N_3 \times N_3}$ and $\overline{V}^+ \in R^{N_3 \times N_3}$ are diagonal matrices. The diagonal elements of the matrix $\overline{V}^+$ are set as follows: $\overline{V}_{jj}^+ = 1 / \overline{V_{jj}}$, if the $j$ th $(1 \le j \le N_3)$ diagonal element of $\overline{V}$ is nonzero; and $\overline{V}_{jj}^+ = 1$, if $\overline{V_{jj}} = 0$.

As mentioned in subsection 4.2.1, if and only if the $k$ th pixel of $f_i$ is the projection of the $j$ th pixel of $F$, $(V_i)_{kj}$ equals one and, in turn, the $j$ th diagonal

element of $\overline{V_i} = V_i^T V_i$ equals one, i.e. $\left(\overline{V_i}\right)_{jj} = 1$. Thus, the $j$ th diagonal element of $\overline{V} = \sum_{i=1}^{n} V_i^T V_i$ counts the number of face images in which the $j$ th pixel of the 3-D face $F$ is visible. If every pixel is visible in only one of the gallery face images, the task of 3-D face reconstruction has the unique solution $F = V^T f$. However, this case rarely occurs. Though recovering the pixels that are invisible in any of the face images is difficult, the unfavorable effects that some pixels are visible in multiple face images can be eliminated by the multiplication of the diagonal matrix $\overline{V}^+$. If a pixel is visible in $m$ different images, its value in $V^T f$ is $m$ times larger than that in the real face $F$. After the left multiplication of matrix $\overline{V}^+$, the reconstructed face $\overline{F} = \overline{V}^+ V^T f$ is different from $F$ only in the pixels that are visible in none of gallery face images.

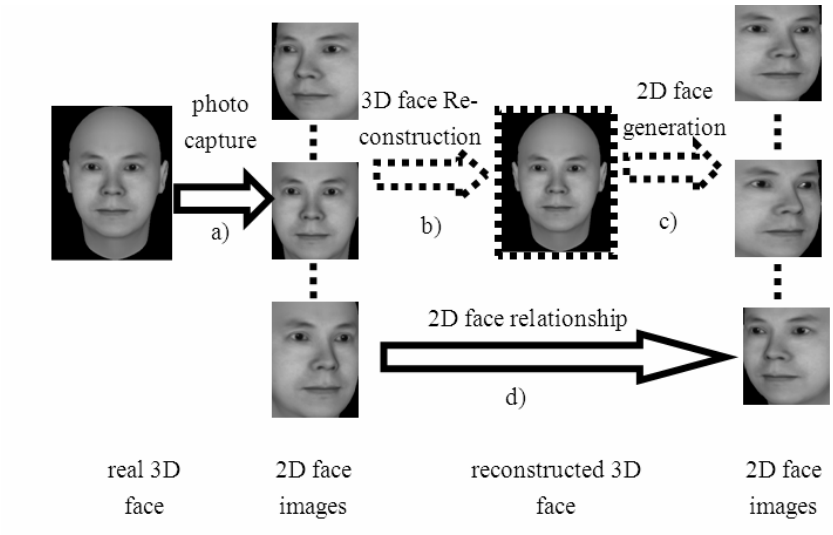



Figure 4.1 Procedure for generating the intrapersonal relationship among 2-D face images. (a) 2-D face images generation; (b) 3-D face reconstruction; (c) 2-D face images generation from reconstructed 3-D face; (d) relationship generation among 2-D face images

### 4.2.3 2-D image prediction

With the reconstructed 3-D face using equation (4.5), we can predict face images under any pose. Replacing the unknown real 3-D face with the reconstructed 3-D face $\overline{F}$, the equation (4.2) for calculating the face image under a novel pose $V_0$ can be rewritten as

$$\begin{aligned} f_0 &= V_0 \overline{V}^{+} V^T f = Wf \\ &= \begin{bmatrix} W_1 & W_2 & \cdots & W_n \end{bmatrix} \begin{bmatrix} f_1 \\ f_2 \\ \vdots \\ f_n \end{bmatrix} = \sum_{i=1}^{n} W_i f_i \end{aligned} \tag{4.6}$$

where $W = V_0 \overline{V}^{+} V^T = \begin{bmatrix} W_1 & W_2 & \cdots & W_n \end{bmatrix}$. Equation (4.6) states that one's face image $f_0$ under a novel pose can be expressed by his/her gallery face images. The underlying assumption of this prediction procedure is that the pixels of the probe face image are visible in at least one gallery face image. Note that, *the probe face image needs not to have a similar pose to one of the gallery face images.*

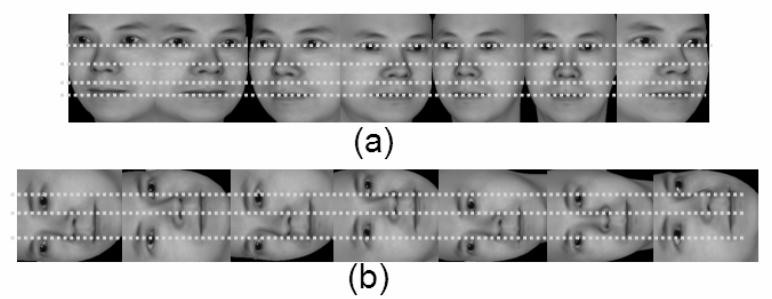



Figure 4.2 Face images of the same individual have the similar configuration

### 4.2.4 Discussion

As can be seen from Figure 4.2, the well aligned face images of the same individual but under different poses are similar in terms of the main organs' spatial

configuration and shape. Thus, if a novel face image $f_0$ can be expressed by a set of face images $f_1, f_2, \cdots, f_n$, this globally linear expression should also hold in locally areas, such as different organs in the face. In other words, if the equation $f_0 = \sum_{i=1}^{n} \lambda_i f_i$ holds, we can linearly express the "eye" $e_0$ of the face $f_0$ by the "eyes" $e_1, e_2, \cdots, e_n$ of the gallery faces $f_1, f_2, \cdots, f_n$, i.e. $e_0 = \sum_{i=1}^{n} \lambda_i e_i$. Intuitively, it does not make sense to express the "eye" of a face image by the combination of all the organs, such as eye, mouth, nose et al.

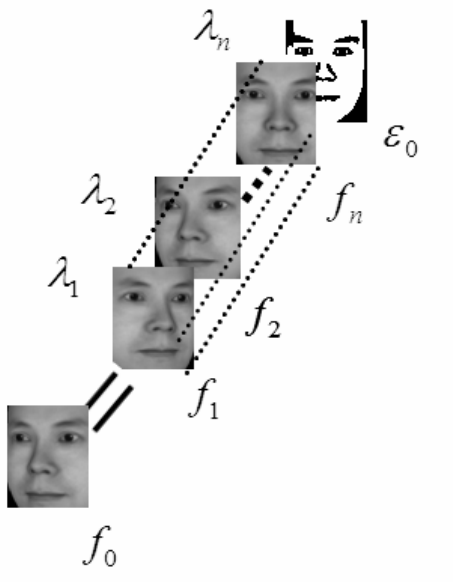



Figure 4.3 Linear expression of a novel face image using gallery face images

Under the assumption that the matrix $W_i$ is a scalar matrix in equation (4.6), i.e. $W_i = \lambda_i I$ ($I$ is the identity matrix), the unreasonable situation that kinds of organs are combined together to form a specific organ is avoided and all parts of the face images are linearly expressed only by their counterparts in other face images, as shown in Figure 4.3. Then, we have the following theorem

**Theorem 1** *If the gallery and probe face images are well aligned, one's probe face image captured under a novel pose can be approximately expressed as a*

*linear combination of his/her gallery face images, i.e.*

$$f_0 = \sum_{i=1}^{n} \lambda_i f_i \tag{4.7}$$

Theorem 1 shows that the probe face image under a novel pose lies on the linear manifold spanned by gallery face images. Thus, we can associate an individual with a linear manifold and model all of his/her face images on it.

In practice, both the gallery and probe face images may be polluted by noise and a novel face image may be only linearly expressible by the gallery face images with a nonzero residue $\varepsilon_0$, as follows

$$f_0 = \sum_{i=1}^{n} \lambda_i f_i + \varepsilon_0 \tag{4.8}$$

There are many ways to measure the residue in equation (4.8). For most of the measurements, it is time consuming to seek the optimal coefficient that minimizes the residue. This thesis, however, just takes advantage of this intrapersonal relationship among face images and avoids seeking the coefficients for the linear expression in equation (4.8).

## 4.3. Orthogonal discriminant vectors

This subsection proposes a method for face recognition across pose based on orthogonal discriminant vector (ODV). Subsection 4.3.1 presents the basic idea and defines the ODV. Subsection 4.3.2 investigates the existence of ODV in face recognition. Subsection 4.3.3 develops an algorithm to calculate ODVs. Subsection 4.3.4 investigates the classification of face images using ODV. Subsection 4.3.5 analyzes the computational complexity of the proposed method.

### 4.3.1 Basic ideas

Geometrically, theorem 1 indicates that one's novel face image lies on the linear manifold spanned by his/her gallery face images. Because of this, a vector $v$ is orthogonal to the probe face image, if $v$ is orthogonal to all the gallery face images. This is derived from the geometrical theorem that if a vector $v$ is orthogonal to a set of vectors $x_i\left(i=1,2,\cdots n\right)$, i.e. $v \perp x_i\left(i=1,2,\cdots n\right)$, it is orthogonal to any linear combination of them, i.e. $v \perp \sum_{i=1}^{n}\alpha_i x_i$, where $\alpha_i\left(i=1,2,\cdots n\right)$ are coefficients. Thus, we have the following corollary 1 from Theorem 1:

**Corollary 1** *Being orthogonal to a certain vector is a pose-invariant feature for the face images of an individual*.

Assume the face images of two individuals span linear manifolds $S_1$ and $S_2$. The methods [28, 45, 57, 106-108] overlook the difference among these two linear manifolds when modeling all the face images on the union manifold $S = S_1 \cup S_2$. Typically, the difference between face linear manifolds $S_1$ and $S_2$ is not null, i.e. $S_1 - S_2 \neq \phi$ and $S_2 - S_1 \neq \phi$. We can decompose $S_1$ into two orthogonal linear manifolds $S_1 = S_{11} + S_{12}$ and $S_{11} \cap S_{12} = \phi$, where $S_{11} \perp S_2$ and $S_{12} \subset S_2$. While the face images of the second individual have nonzero projections onto the vectors in $S_{11}$, the face images of the first individual have zero projections. So, being orthogonal to any vector in $S_{11}$ is a pose-invariant feature for the first individual.

We define the orthogonal discriminate vector (ODV) associating with the $i$th individual as follows:

**Definition 1** *If a vector $v$ is orthogonal to all the gallery face images of the $i$th individual and not orthogonal to any gallery images of the other individuals, this*

*vector $v$ is an ODV associating with the $i$ th individual.*

As revealed by theorem 1, one's probe face image captured under a novel pose can be linearly expressed by his/her gallery face images. So, the ODVs that are orthogonal to the gallery face images are also orthogonal to the face images under novel poses. Thus, being orthogonal to ODVs is a pose-invariant feature, which can be used in face recognition across pose.

### 4.3.2 The existence of the ODV

In this subsection, we study the existence of ODV in the task of face recognition. Let the $N$ dimensional vectors $x_i \in R^{N\times 1}\left(1 \le i \le n_1\right)$ be $n_1$ face images of one individual, and $y_j \in R^{N\times 1}\left(1 \le j \le n_2\right)$ be $n_2$ face images of the others. We consider the face recognition as a two-class classification problem. The face image matrix of the first class is denoted as $X = \left[\begin{array}{cccc} x_1 & x_2 & \cdots & x_{n_1} \end{array}\right] \in R^{N\times n_1}$ and the one of the second class is denoted as $Y = \left[\begin{array}{cccc} y_1 & y_2 & \cdots & y_{n_2} \end{array}\right] \in R^{N\times n_2}$.

Let $Z = \left\{ z_i \in R^{N\times 1}, 1 \le i \le N \right\}$ be a set of mutually orthogonal unit vectors that span the $N$ dimensional vector space. Then, the face images from these two classes can be linearly expressed by the bases, as follows

$$\begin{cases} x_i = Zd_i^1, 1 \le i \le n_1 \\ y_j = Zd_j^2, 1 \le j \le n_2 \end{cases} \tag{4.9}$$

and

$$\begin{cases} X = ZD_1 \\ Y = ZD_2 \end{cases} \tag{4.10}$$

where $Z = \left[\begin{array}{cccc} z_1 & z_2 & \cdots & z_N \end{array}\right]$ is the base matrix, $d_i^1, d_j^2 \in R^N$ are coefficient vectors associating with $x_i$ and $y_j$. The matrices $D_1 = \left[\begin{array}{cccc} d_1^1 & d_2^1 & \cdots & d_{n_1}^1 \end{array}\right] \in R^{N\times n_1}$ and

$D_2 = \begin{bmatrix} d_1^2 & d_2^2 & \cdots & d_{n_1}^2 \end{bmatrix} \in R^{N \times n_2}$ are two coefficient matrices.

The theorem of generalized singular value decomposition (GSVD) states as follows: *for the given two matrices* $D_1 \in R^{N \times n_1}$, $D_2 \in R^{N \times n_2}$, $D = \begin{pmatrix} D_1^T \\ D_2^T \end{pmatrix}$ *and* $t = rank(D)$, *there exist orthogonal matrices* $L_1 \in R^{n_1 \times n_1}$, $L_2 \in R^{n_2 \times n_2}$, $W \in R^{t \times t}$, *and* $Q \in R^{N \times N}$ *such that*

$$L_1^T D_1^T Q = \Sigma_1 \left( \underbrace{W^T K}_{t} \quad \underbrace{0}_{N-t} \right) \tag{4.11}$$

*and*

$$L_2^T D_2^T Q = \Sigma_2 \left( \underbrace{W^T K}_{t} \quad \underbrace{0}_{N-t} \right) \tag{4.12}$$

*where*

$$\sum_{1 \atop n_1 \times t} = \begin{pmatrix} I_1 & & \\ & J_1 & \\ & & O_1 \end{pmatrix} \quad \textit{and} \quad \sum_{2 \atop n_2 \times t} = \begin{pmatrix} O_2 & & \\ & J_2 & \\ & & I_2 \end{pmatrix} \tag{4.13}$$

*and* $K \in R^{t \times t}$ *is nonsingular with its singular values equal to the nonzero singular values of* $D$. *The matrices* $I_1 \in R^{r \times r}$ *and* $I_2 \in R^{(t-r-s) \times (t-r-s)}$ *are identity matrices, where*

$$r = rank \begin{pmatrix} D_1^T \\ D_2^T \end{pmatrix} - rank\left(D_2^T\right) \quad \textit{and} \quad s = rank\left(D_1^T\right) + rank\left(D_2^T\right) - rank \begin{pmatrix} D_1^T \\ D_2^T \end{pmatrix} \tag{4.14}$$

*The matrices* $O_1 \in R^{(n_1-r-s) \times (t-r-s)}$ *and* $O_2 \in R^{(n_2-t-s) \times r}$ *are zero matrices with possible no rows or no columns. The matrices* $J_1 = diag\left(a_{r+1}, \cdots, a_{r+s}\right)$ *and* $J_2 = diag\left(b_{r+1}, \cdots, b_{r+s}\right)$ *satisfy*

$$1 > a_{r+1} \geq \cdots \geq a_{r+s} > 0 \textit{ and } 0 < b_{r+1} \leq \cdots \leq b_{r+s} < 1 \tag{4.15}$$

$a_i^2 + b_i^2 = 1 \quad for \quad i = r+1, \cdots, r+s$.

Based on (4.11) and (4.12), we have

$$\begin{cases} D_1 = Q \begin{bmatrix} K^T W \\ 0 \end{bmatrix} \Sigma_1 L_1^T \\ \\ D_2 = Q \begin{bmatrix} K^T W \\ 0 \end{bmatrix} \Sigma_2 L_2^T \end{cases} \tag{4.16}$$

Using (4.16), we can rewrite (4.10) as follows

$$X = ZD_1 = ZQ \begin{bmatrix} K^T W \\ 0 \end{bmatrix} \Sigma_1 L_1^T = ZQ \begin{bmatrix} K^T W \\ 0 \end{bmatrix} \begin{bmatrix} I_1 & & \\ & J_1 & \\ & & 0 \end{bmatrix} L_1^T \tag{4.17}$$

and

$$Y = ZD_2 = ZQ \begin{bmatrix} K^T W \\ 0 \end{bmatrix} \begin{bmatrix} 0 & & \\ & J_2 & \\ & & I_2 \end{bmatrix} L_2^T \tag{4.18}$$

Using the above two equations, we can prove theorem 2

**Theorem 2** *The column vectors of* $V_\perp$ *are ODVs of the second class, where*

$$V_\perp = ZQ \begin{bmatrix} K^T W \\ 0 \end{bmatrix} \begin{bmatrix} I_1 \\ 0 \\ 0 \end{bmatrix} \tag{4.19}$$

**Proof.** We only need to prove the columns of $V_\perp$ are orthogonal to the images of the second class and are not orthogonal to the images of the first class.

(a)

$$V_\perp^T Y = \begin{bmatrix} I_1 & 0 & 0 \end{bmatrix} \begin{bmatrix} W^T K & 0 \end{bmatrix} Q^T Z^T ZQ \begin{bmatrix} K^T W \\ 0 \end{bmatrix} \begin{bmatrix} 0 & & \\ & J_2 & \\ & & I_2 \end{bmatrix} L_2^T \tag{4.20}$$

Because both $Z$ and $Q$ are orthogonal matrices, we have

$$V_{\perp}^{T}Y=\begin{bmatrix}I_1 & 0 & 0\end{bmatrix}W^{T}KIK^{T}W\begin{bmatrix}0 & & \\ & J_2 & \\ & & I_2\end{bmatrix}L_2^{T}=0 \tag{4.21}$$

Thus, the column vectors of $V_{\perp}$ are orthogonal to all the images of the second class.

(b)

$$\begin{aligned}V_{\perp}^{T}X&=\begin{bmatrix}I_1 & 0 & 0\end{bmatrix}\begin{bmatrix}W^{T}K & 0\end{bmatrix}Q^{T}Z^{T}ZQ\begin{bmatrix}K^{T}W \\ 0\end{bmatrix}\begin{bmatrix}I_1 & & \\ & J_1 & \\ & & 0\end{bmatrix}L_1^{T}\\ &=\begin{bmatrix}I_1 & 0 & 0\end{bmatrix}W^{T}KK^{T}W\begin{bmatrix}I_1 & & \\ & J_1 & \\ & & 0\end{bmatrix}L_1^{T}=W^{T}KK^{T}WL_1^{T}\end{aligned} \tag{4.22}$$

Because $L_1$ and $W$ are orthogonal matrices, and $K$ is a nonsingular matrix, we have

$$V_{\perp}^{T}X\neq 0 \tag{4.23}$$

Thus, the column vectors of $V_{\perp}$ are not orthogonal to all the images of the first class. In summary, the column vectors of $V_{\perp}$ are orthogonal to all the images in the second class and not orthogonal to all the images in the first class. Thus, the columns of $V_{\perp}$ are ODVs of the second class.

In a similar way, we can prove that the columns of the matrix $V_{\perp}^{'}$ in equation (4.24) are ODVs of the first class

$$V_{\perp}^{'}=ZQ\begin{bmatrix}K^{T}W \\ 0\end{bmatrix}\begin{bmatrix}0 \\ 0 \\ I_2\end{bmatrix} \tag{4.24}$$

As $I_1\in R^{r\times r}$ and $r=rank(D)-rank(D_2^{T})$, we know that the ODV of the second class does not exist when $rank(D)=rank(D_2^{T})$. Similarly, the ODV of the first class does not exist when $rank(D)=rank(D_1^{T})$. Thus, when

$rank(D) = rank(D_1^T) = rank(D_2^T)$, there is no ODV in this two-class classification problem. However, the equation $rank(D) = rank(D_1^T) = rank(D_2^T)$ holds only when all the samples from one class can be linearly expressed by the combinations of the samples from the other class. For high dimensional face images, such a situation hardly happens. This is because one individual must have a character that differentiates his/her face images from those of others. Our experimental results verify the existence of the ODV.

### 4.3.3 The calculation of the ODV

For simplicity in description, we first consider a two-class classification problem. We regard the face images $X = \begin{bmatrix} x_1 & x_2 & \cdots & x_{n_1} \end{bmatrix} \in R^{N \times n_1}$ of one individual as the first class and the face images $Y = \begin{bmatrix} y_1 & y_2 & \cdots & y_{n_2} \end{bmatrix} \in R^{N \times n_2}$ of the others as the second class. Also, we assume the first class has fewer images than the second class, i.e. $n_1 < n_2$.

We propose the following model to calculate the ODVs for the first class

$$\max \left\| Y^T v \right\|_2 \quad s.t. \;\; X^T v = 0 \tag{4.25}$$

By minimizing the projections of the first class to be zero and maximizing the projections of the second class, this model aims to generate the most discriminative ODVs. We propose a two-step procedure to solve the model in (4.25). Step 1 generates a set of candidate vectors for the ODV. From candidate set, step 2 chooses the most discriminative ODVs onto which the images of the second class have the maximum projections.

Step 1: generate candidate vectors for ODVs

To generate the candidate vectors, step 1 only needs to solve the following linear

equation system

$$X^T Y \mu = 0 \tag{4.26}$$

In fact, we have a theorem as follows

**Theorem 3** *The nonzero vector* $v = Y\mu \neq 0$ *is an ODV of the first class, if* $\mu$ *is a nonzero solution vector of equation (4.26).*

**Proof.** We only need to prove that the following two formulas regarding the vector $v$ hold

$$\begin{cases} v^T X = (Y\mu)^T X = \mu^T Y^T X = \left(X^T Y \mu\right)^T = 0 \\ v^T Y = (Y\mu)^T Y = \mu^T Y^T Y \neq 0 \end{cases} \tag{4.27}$$

While the first formula in (4.27) is certain to be true and does not need further proof, the second formula can be proved by contradiction as follows.

Suppose $\mu^T Y^T Y = 0$, then

$$\mu^T Y^T Y = 0 \Rightarrow \mu^T Y^T Y \mu = 0 \Leftrightarrow (Y\mu)^T Y\mu = 0 \tag{4.28}$$

As we know $v^T v = (Y\mu)^T Y\mu = \sum_{i=1}^{N} v_i^2 \geq 0$, where $N$ is the dimensionality of the face images. If and only if $v_i = 0$ for $1 \leq i \leq N$, i.e. $v = 0$, the equation $v^T v = 0$ holds. But it is impossible for $v$ to be zero vector, which contradicts $v = Y\mu \neq 0$. This completes the proof.

Suppose $U = \left[\mu_1 \quad \mu_2 \quad \cdots \quad \mu_l\right] \in R^{n_2 \times l}$ is a set of linearly independent solutions of (4.26), we can denote the candidate ODV set as follows

$$S = \left\{ v = YU\alpha \mid \alpha \in R^{l \times 1} \right\} \tag{4.29}$$

where $\alpha$ is a coefficient vector. In fact, solving (4.26) (obtaining the matrix $U$) implies finding the null space of the coefficient matrix $X^T Y$. This set includes all the candidate ODVs as $U\alpha$ is the general form for the solutions of (4.26). Among the

infinite candidate vectors in $S$, step 2 picks out the most discriminative ones and takes them as ODVs.

Step 2: choose the most discriminative ODVs

Among all the vectors in $S$, step 2 chooses the ODVs corresponding to the longest second class projections for the further classification. We formulate it as follows

$$\max_{v \in S} \left\| Y^T v \right\|_2 \tag{4.30}$$

Substituting $v$ by $YU\alpha$ in (4.30), we have

$$\max \left\| Y^T v \right\|_2^2 = \max \left\| v^T Y Y^T v \right\|_2 = \max \left\| \alpha^T U^T Y^T Y Y^T Y U \alpha \right\|_2 \tag{4.31}$$

It can be easily proved that the coefficient vectors $\alpha$ should be the eigenvectors of the matrix $M = U^T Y^T Y Y^T Y U \in R^{l \times l}$ associating with the leading eigenvalues. If the eigenvectors of the matrix $M$ that associate with the $k$ largest eigenvalues are $\alpha_i \left( i = 1, 2, \cdots, k \right)$, the ODVs for the first class are

$$v_i = YU\alpha_i \left( i = 1, 2, \cdots, k \right) \tag{4.32}$$

Equation (4.32) shows that the ODVs for the first class are in fact linear combinations of the face images from the second class.

To sum up, the following two-step algorithm can generate the most discriminative ODVs:

Firstly, solve the equation $X^T Y \mu = 0$ and output a set of independent solutions $\mu_1, \mu_2, \cdots, \mu_l$;

Secondly, perform eigendecompositon of matrix $M = U^T Y^T Y Y^T Y U$ to solve the maximization problem (4.31) and generate the leading eigenvectors $\alpha_i$; output the ODVs $v_i = YU\alpha_i \left( i = 1, 2, \cdots, k \right)$;

### 4.3.4 ODV-based face classification

Based on the cosine metric, we define the distance between a face image $x$ and an ODV $v$ as follows

$$d(x,v) = 1 - \left|\cos\langle x, v\rangle\right| = 1 - \left|\frac{x^T v}{\|x\|\|v\|}\right| \tag{4.33}$$

Such defined distance achieves its maximum value, i.e. one, if $x \perp v$; and achieves its minimum value, i.e. zero, if $x // v$.

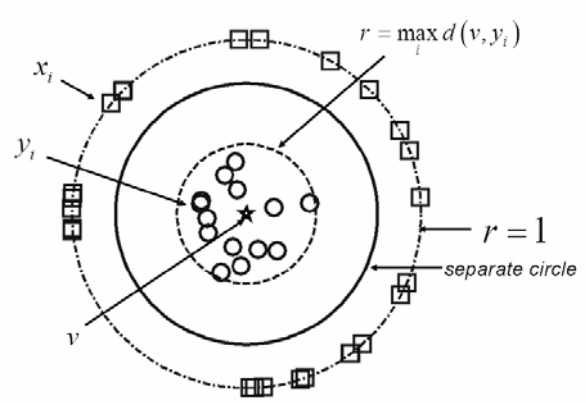



Figure 4.4 The distance of the ODV $v$ to the images from two classes

**4.3.4.1 Two-class classification problem**

In a two-class classification problem, the ODV $v$ for the first class is orthogonal to the first class image $x_i$ and not orthogonal to the second class image $y_j$, i.e. $v^T x_i = 0$ and $v^T y_j \neq 0$. So, the distances between $v$ and the first class face images equal one, and those between $v$ and the second class face images are smaller than one. If we use the common center of the circles $O$ to represent the ODV $v$ and the metric defined in (4.33) to measure the distances between $v$ and the face images, the first class images (represented by "□") scatter on the unit circle centered at $O$ for a 2-D case, whereas the second class images (represented by "○") scatter inside of the unit circle, as shown in Figure 4.4. Theoretically, a circle

can correctly separate the images from these two classes, if its radius $r$ satisfies $\max_{1\le i\le n_2} d\left(v, y_i\right) < r < 1$.

#### 4.3.4.2 Multi-class classification problem

If the face images belong to more than two individuals, the face recognition task becomes a multi-class classification problem. We take the one-to-many strategy and calculate a set of ODVs $v_j^i \left(1 \le i \le c; 1 \le j \le k_i\right)$ for every class, where $c$ $(c > 2)$ is the number of individuals and $k_i$ is the number of ODVs associating with the $i$th class. For a gallery face image $x$ belonging to the $c(x)$ class, its distance to the ODV $v_j^i$ satisfies

$$\begin{cases} d\left(x, v_j^i\right) = 1 & c(x) = i \\ d\left(x, v_j^i\right) < 1 & c(x) \ne i \end{cases} \tag{4.34}$$

Normally, the larger the distance between a face image and the ODV $v_j^i$, the more likely this face image belongs to the $i$th class. Here, we classify a probe face image $x$ into the $c(x)$th class, if

$$\frac{1}{k_{c(x)}} \sum_{j=1}^{k_{c(x)}} d\left(x, v_j^{c(x)}\right) = \max_{1\le i\le c} \frac{1}{k_i} \sum_{j=1}^{k_i} d\left(x, v_j^i\right) \tag{4.35}$$

The complete face recognition procedure is summarized in the following algorithm.

Algorithm 4.1 ODV-based face recognition

Training stage: For $1 \le i \le c$, the following procedure is performed to calculate a set of ODVs for the $i$th class

Step 1. Take the face images in $i$th class as columns of $X$ and those of other classes as columns of $Y$;

Step 2. Solve the linear equation system $X^T Y\mu = 0$, and obtain a set of linearly independent solutions $U = \left[\mu_1 \quad \mu_2 \quad \cdots \quad \mu_l\right] \in R^{n_2 \times l}$, where $l$ is the number of linearly independent solutions;

Step 3. Perform eigendecomposition of the matrix $M = U^T Y^T Y Y^T Y U$, and denote the eigenvectors associating with nonzero eigenvalues as $\alpha_j^i \left(j = 1, 2, \cdots k_i\right)$;

Step 4. Calculate the ODVs using $v_j^i = YU\alpha_j^i \left(j = 1, 2, \cdots k_i\right)$;

Testing stage:

Step 1. Calculate the distances (defined in (4.33)) between probe face image $x$ and the ODVs of each class;

Step 2. The probe face image $x$ is classified into the $c(x)$th class according to (4.35).

### 4.3.5 Computational complexity

Now, let us analyze the computational complexity of Algorithm 4.1 in subsection 4.3.4.

In the training stage, step 1 only picks out the face images associating with the $i$th individual with computational complexity of $O(n)$, where $n$ is the total number of images. As $X \in R^{N \times n_1}$ and $Y \in R^{N \times n_2}$, the construction of the linear equation system $X^T Y\mu = 0$ has computational complexity of $O(n_1 n_2 N)$ in step 2, where $N$ is the dimensionality of the face images and $n_1, n_2$ are number of images in the two classes. Step 2 solves the linear equation system with computational complexity of $O(n^3)$.

Step 3 involves a series of matrix multiplication and an eigendecomposition

procedure. To calculate the matrix $M$ efficiently, we reformulate it as follows

$$M = \left(U^T \left(Y^T Y\right)\right)\left(\left(Y^T Y\right)U\right) = \left(\left(Y^T Y\right)U\right)^T \left(\left(Y^T Y\right)U\right) = M_1^T M_1 \tag{4.36}$$

Calculation of the matrix $M_1 = \left(Y^T Y\right)U \in R^{n_2 \times l}$ has the computational complexity of $\mathrm{O}\left(n_2^2 N + n_2^2 l\right)$, and calculation of $M = M_1^T M_1$ has the computational complexity of $\mathrm{O}\left(n_2 l^2\right)$, where $l\left(l < n\right)$ is the number of solutions generated in step 2. The eigendecomposition procedure of the matrix $M \in R^{l \times l}$ needs computational complexity of $\mathrm{O}\left(l^3\right)$. Thus, the step 3 has computational complexity of

$$\mathrm{O}\left(n_2^2 N + n_2^2 l\right) + \mathrm{O}\left(n_2 l^2\right) + \mathrm{O}\left(l^3\right) < O\left(n^3 + n^2 l + n^2 N\right) \tag{4.37}$$

Step 4 performs a series of matrix multiplication to generate ODV $v_j^i = YU\alpha_j^i \left(j = 1, 2, \cdots k_i\right)$ with computational complexity of $O\left(Nnl\right)$.

Thus, the calculation of ODVs of one class has computational complexity of

$$O\left(n_1 n_2 N\right) + O\left(n^3\right) + O\left(n^3 + n^2 l + n^2 N\right) + O\left(Nnl\right) < O\left(Nn^2\right) \tag{4.38}$$

These four steps are performed $c$ times, where $c$ is the number of classes. Totally, the computational complexity of the training stage is $O\left(cNn^2\right)$. This indicates that the computational complexity of the training stage is mainly determined by the number of face images other than the dimensionality of them which is usually larger.

In the testing stage, when classifying a probe face image, we only need to calculate its distances to $\sum_{i=1}^{c} k_i$ ODVs (where $k_i$ is the number of ODVs associating with the $i$ th class) with the computational complexity of $O\left(N\left(\sum_{i=1}^{c} k_i\right)\right)$. From the calculation procedure of the ODV, we know that the number of ODVs associating with one class is strictly smaller than the number of gallery images, i.e.

$k_i < n$. Thus, the computational complexity of classifying a probe face image satisfies

$$O\left(N \times \sum_{i=1}^{c} k_i\right) < O\left(Ncn\right) \tag{4.39}$$

Note that the computational complexity of both the training and testing procedure is in direct proportion to the dimensionality of the face images.

## 4.4. Experiments

This subsection presents our experiments on the popular face databases. Subsection 4.4.1 validates the theorem 1 by investigating the residues in the linear expressions of probe face images using gallery face images. Subsection 4.4.2 and 4.4.3 respectively show the experimental results of face verification and recognition.

One standard for evaluating the face recognition technologies is the Facial Recognition Technology (FERET) database [129], which was sponsored by US Department of Defense through the DARPA Program. A subset of the FERET database including 1 386 images of 198 individuals (7 images for each individual) is used in our experiments. An image is included in this subset, if its name is marked with any of the following two character strings: “ba”, “bd”, “be”, “bf”, “bg”, “bj”, and “bk” [57]. The images in this subset have pose variations of $\pm 15^o$, $\pm 25^o$, and also the variations of illumination and the expression. The original images are cropped and resized to $80 \times 80$ pixels.

Another standard database for evaluating the face recognition across pose is the Carnegie Mellon University Pose, Illumination and Expression database (CMU PIE database) [130]. The CMU PIE database totally consists of more than 40 000 facial images of 68 people. In the construction of this database, the images of each individual are captured under 43 different illumination conditions, across 13 different poses, and with 4 different expressions. We use a subset contains five poses (C05, C07,

C09, C27, C29) and all different illuminations and expressions. There are 170 images for each of the 50 individuals.

The Yale Face Database B [101] contains 5 850 images, 585 images for each of 10 individuals. The images of one individual are captured under 9 different poses and illumination conditions. Also, an image with ambient illumination is captured under all of the 9 different poses for each individual.

The AR face database [11] contains 3 120 images corresponding to the faces of 120 people. The images include frontal view faces with different facial expressions, conditions of illumination, and occlusions (sun glasses and scarf). Each person participates in two sessions, separated by intervals of two weeks. The same pictures are taken in both sessions.

### 4.4.1 Residue investigation

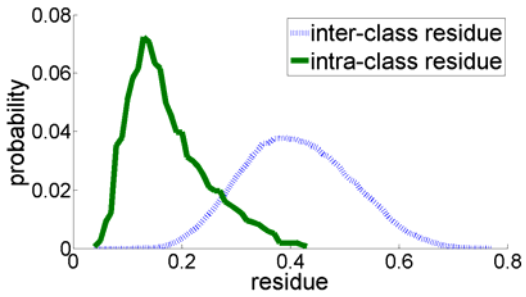
This subsection verifies Theorem 1 through investigating the residues in the linear expressions of probe face images using gallery images. Assume that a novel face image $z$ can be linearly expressed by the gallery face images $x_1, x_2, \cdots, x_n$ of one individual with a residue $\varepsilon$ as follows

$$z = \lambda_1 x_1 + \lambda_2 x_2 + \cdots + \lambda_n x_n + \varepsilon \tag{4.40}$$

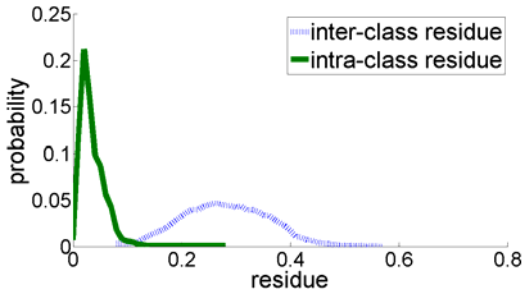
We calculate the minimum residue in terms of $l_2$-norm using the following method. Firstly, we calculate the covariance matrix of $x_1, x_2, \cdots, x_n$ and its eigenvectors $g_1, g_2, \cdots g_n$. Then, we can obtain the minimum residue as follows

$$\varepsilon = z - \sum_{i=1}^{n} x^T g_i \tag{4.41}$$

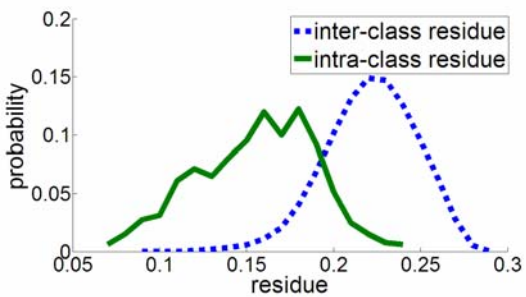
The reasoning behind doing so is that the two spaces respectively spanned $x_1, x_2, \cdots, x_n$ and $g_1, g_2, \cdots g_n$ are the same.



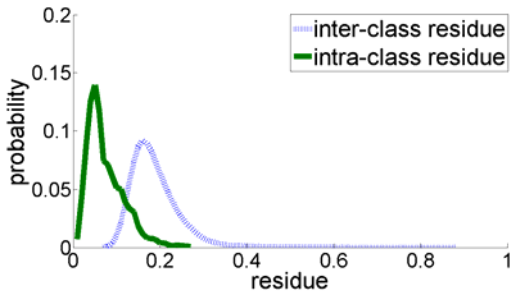

(a)




(b)




(c)




(d)

Figure 4.5 Inter- and intra-class residue distribution on (a) AR face database; (b) YaleB face database; (c) FERET face database; (d) PIE database

In this experiment, we divide the face images of each individual into two halves.

The first half is the gallery set and the second half is the probe set. The residue in (4.41) is an intra-class residue if $z$ and $x_1, x_2, \cdots, x_n$ associate with the same individual. The residue is an inter-class residue if they associate with different individuals.

Figure 4.5 shows the probability distribution of the norm of intra- and inter-class residues. It can be seen that the intra-class residue is usually much smaller than the inter-class residue. One's face image is more likely to be well expressed by his/her face images other than the images of others. The large inter-class residue indicates that one's probe face image is far away from the linear manifold spanned by the face images of another person. So, the face images of different individuals span different linear manifolds, and the ODV that is orthogonal to only one of the linear manifolds exists.

### 4.4.2 Face verification

The goal of face verification is to determine whether a probe image belongs to a particular person. In the proposed ODV-based method, we work out a distance between the probe image and an ODV, and compare it with a defined threshold. The probe image is classified into the class which the ODV associating with, only if the distance is larger than the threshold. We use the subset of CMU PIE containing 8500 face images of 50 different individuals and the whole YaleB databases. Both of them are randomly divided into two halves; one half is taken as gallery and the other half is taken as probe.

The distances between ODVs and face images can be classified into two groups:

1. Intra-class distance: the face image and ODV associate with the same class. The distance in this group is expected to be as large as possible.

2. Inter-class distance: the face image and ODV associate with other class. The distance in this group is expected to be as small as possible.

In Figure 4.6 and Figure 4.7, the original represents ODV and the dots represent the face images. If the distance between a face image and the ODV is $d_1$, this face image is represented by a random dot on the circle with radius equals to $d_1$. Here, the distance is regularized using the following equation

$$d_{new} = \frac{d - \min(d)}{\max(d) - \min(d)} \tag{4.42}$$

In both Figure 4.6 and Figure 4.7, the circles in figures (a) and (b) have the same radius which is smaller than 95% intra-class distances, i.e. only 5% of the dots scatter inside of the circle in Fig. (a). As can be seen from Fig. (b), only a small portion of the inter-class distances are larger than the radius of the circle.

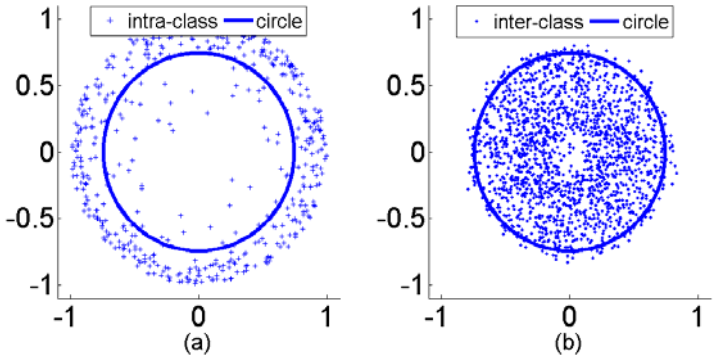



Figure 4.6 The distribution of the face images in CMU PIE database

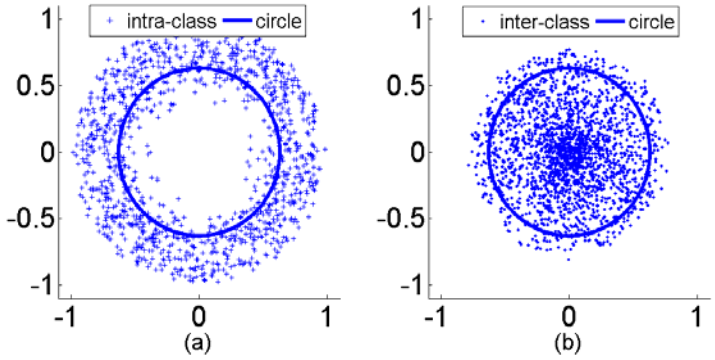
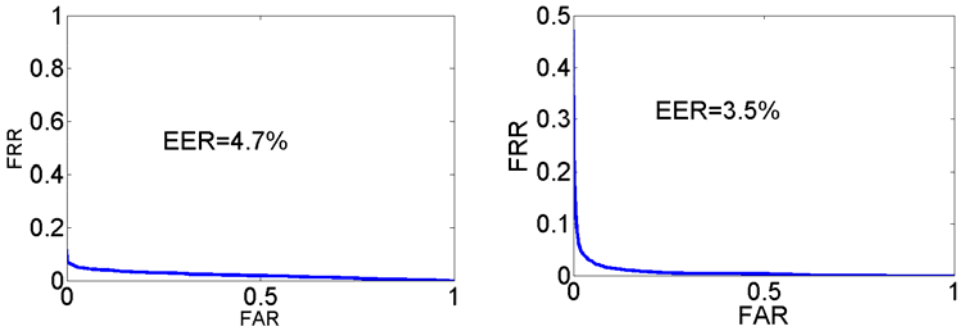



Figure 4.7 The distribution of the face images in YaleB database




(a) (b)

Figure 4.8 The ROC curves. (a) The subset of CMU PIE database; (b) The YaleB database

There are two kinds of misclassification in face verification: false acceptance (FA), where the system incorrectly classifies a face image into a class which it does not belong to; and false rejection (FR), where the system fails to classify a face image into the class which it belongs to. For each threshold, there will be one pair of false acceptance rate (FAR) and false rejection rate (FRR). By tuning the threshold, we can have a series of FARs and the corresponding FRRs. Then, we can plot FRR verse FAR and obtain a receiver operator curve (ROC) for each database. Figure 4.8 (a) and (b) respectively show the ROC for the subset of CMU PIE and the Yale B database. The Equal Error Rate (EER) is defined to be the FRR when it equals to FAR. The EER for the CMU PIE face database is 4.7%, which is lower than those of the biohashing algorithm (11.93%) [131], original Fisherface method (18.18%) [132] and its improvements (larger than 5%) in [132]. The EER for the Yaleb database is 3.5%, a little lower than the results reported in [133].

### 4.4.3 Face recognition

To test the proposed method, this subsection presents the face recognition experiments on the AR, FERET, CMU PIE, and YaleB face databases. We compare the performance of the proposed method with other six appearance-based methods:

principal component analysis (PCA) [134], Fisher discriminant analysis (FDA) [28], (Maximum a posterior discriminant analysis) MLDA [115], Kernel principal component analysis (KPCA) [56, 135], Kernel Fisher discriminant analysis (KFDA) [57], and local preserving projection (LPP) [45]. All of the above five methods and our method classify a face image using the nearest neighbor classifier. Both KFDA and KPCA adopt the Gaussian Kernel.

Seven-fold cross validation is used on the FERET face database. The subset of FERET are divided into seven portions based on the names of images, which are marked with “ba”, “bd”, “be”, “bf”, “bg”, “bj”, and “bk”. These seven portions are respectively captured under different circumstances. Six portions are used for training and the rest portion is used for testing in our experiments. Thus, the sizes of the training and testing subset are respectively 1 188 and 198. The proposed method generates 198 ODVs, and other methods generate 198 projection vectors. The classification results are listed in Table 4.1.

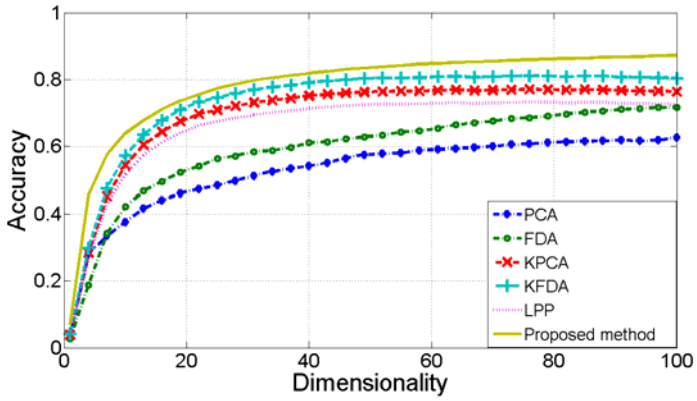
Table 4.1 shows that the proposed method outperforms the other five methods in most cases. When the “bg” portion is used for testing, the proposed method achieves the classification accuracy of 95.5%, more than 5% higher than the other methods. Though KFDA achieves a little higher accuracy (0.5%) on “bd” and “bf”, KFDA is very time consuming in both training and testing procedure. Moreover, it is very difficult to fix the parameter for kernel function in both KFDA and KPCA. All the methods (PCA, FDA, and LPP) make an implicit assumption that the face images of each individual cluster together. However, as it is widely recognized, the variations induced by pose change can be larger than those induced by identity change. Different from these three methods, our method models face images of different individuals on different linear manifolds and does not require the images of one individual cluster

together. Because of this, our method achieves accuracies 2%~10% higher than PCA, LDA, and LPP.

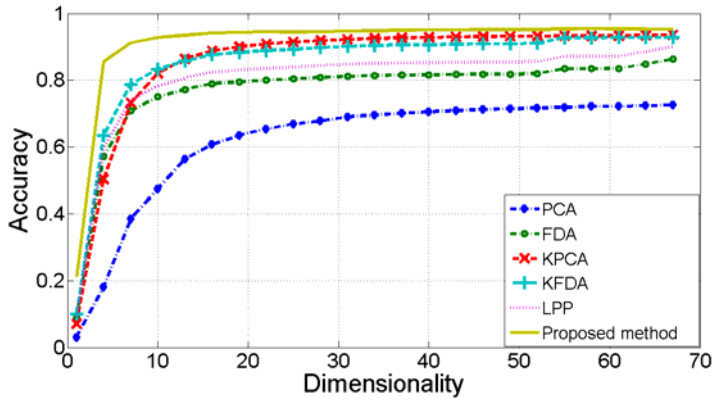
Table 4.1 Classification accuracies (%) of methods on FERET face database

| Portion for test | ba | bd | be | bf | bg | bj | bk |
|---|---|---|---|---|---|---|---|
| PCA | 81.5 | 67.0 | 69.5 | 52.5 | 77.5 | 68.0 | 78.5 |
| FDA | 89.5 | 68.5 | 90.5 | 47.5 | 84.0 | 65.5 | 86.5 |
| KPCA | 86.0 | 76.0 | 75.5 | 58.0 | 89.5 | 68.5 | 82.5 |
| KFDA | 90.5 | 76.5 | 82.0 | 60.5 | 88.0 | 67.0 | 91.0 |
| LPP | 84.0 | 71.5 | 90.0 | 51.0 | 89.5 | 66.5 | 84.5 |
| MLDA | 87.5 | 72.5 | 89.0 | 52.0 | 81.5 | 66.0 | 78.0 |
| Proposed method | 92.0 | 76.0 | 92.5 | 60.0 | 95.5 | 69.0 | 94.0 |

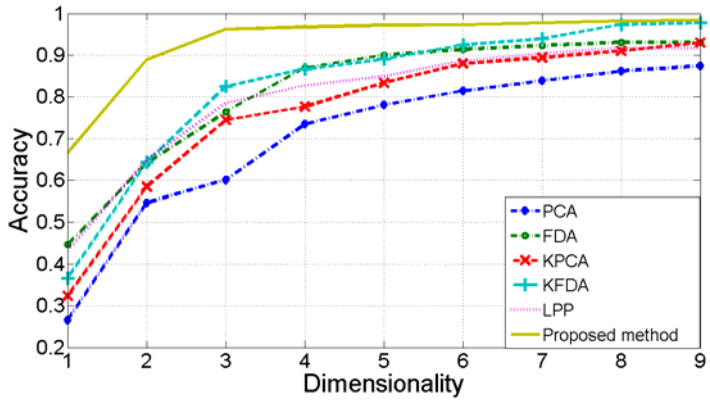
The other three databases are randomly divided into two disjoint subsets, one for training (gallery images) and one for testing (probe images). For the YaleB database, the training subset contains 10 images for each individual (account for about 1.8%) and the testing subset contains 575 images for each individual (account for about 98.2%). The training subset and the testing subset in the CMU PIE face database respectively have size of 3 400 and 8 160, and respectively contain 50 and 120 images for each individual. The AR database is randomly divided into two halves, one for training and one for testing. These randomly divisions are repeated 20 times. The average classification accuracies are plotted versus the number of projection vectors (ODVs in the proposed method) in Figure 4.9.




(a) AR face database




(b) CMU PIE face database




(c) YaleB face database

Figure 4.9 Classification accuracy comparisons of different methods on different databases: (a) AR face database; (b) CMU PIE face database; (c) YaleB face database

In these figures, the dimensionality means the number of ODVs in our method and number of projectors in other methods. As can be seen, the proposed method

performs better than other methods. Note that, the proposed method can achieve much higher classification accuracy than other methods when the dimensionality is low, especially on the CMU PIE and YaleB face databases. On the CMU PIE face database, the proposed method can achieve classification accuracy of 85.55% using two ODVs, and 91.16% using three ODVs. On the contrast, the other methods cannot achieve classification accuracy higher than 65% using two projection vectors and 80% using three projection vectors. On the YaleB face database, the classification accuracies of the proposed method are 20% higher than other methods when dimensionality is less than three. None of appearance-based face recognition methods [28, 45-46, 56, 104, 124, 135] has achieved accuracy as high as ours with such low dimensionalities.

Table 4.2 lists the training time of different methods on the three face databases. In the AR and YaleB face databases, the dimensionality of the face images is much larger than the number of training images. While the computational complexity of our method largely depends on the number of training images, the ones of [28, 45, 134] largely depend on the dimensionality. This explains why our method is much faster than PCA, FDA and LPP. The nonlinear methods [56-57, 135] are time consuming because it (1) has an additional procedure to fix the kernel parameter and (2) must perform an implicit nonlinear transformation instead of working with the original images. Our method is slower than PCA and FDA when tested on CMU PIE database mainly because the number of training images is larger than the dimensionality of the face images in this database. However, this is not the general case, because our training face images are obtained by windowing and scaling the original 640 by 486 images. It is widely accepted that in the task of face recognition, the dimensionality of the images is normally larger than the number of images available for training. Though the computational complexity of our method grows proportionally as the

number of classes grows, this number is much smaller compared with dimensionality of the face images. With computational complexity grows proportional to the dimensionality of images, our method is faster than the other methods whose computational complexity grows quadratically as the dimensionality grows.

Table 4.2 Training time (seconds) of different methods in three face databases

| Database | Size | PCA | FDA | LPP | KPCA | KFDA | Proposed method |
|---|---|---|---|---|---|---|---|
| AR | 80×100 | 349 | 405 | 864 | 3447 | 4558 | 325 |
| CMU PIE | 32×32 | 703 | 891 | 1093 | 2420 | 1779 | 981 |
| YaleB | 160×120 | 54 | 69 | 206 | 1432 | 1030 | 48 |

## 4.5. Conclusion and future work

We conclude that our new approach to face recognition across pose is effective in comparison with the existing methods. Unlike the traditional methods that investigate the face images on a single manifold, our algorithm explores the differences among linear manifolds spanned by face images of different individuals. Based on the comprehensive theoretical analysis of intrapersonal relationship among face images across pose, it is found that a person's face image with a new pose can be linearly expressed by his/her gallery face images. The experimental results reported in Subsection 4.4.1 validate our observation presented in subsection 4.2. As a result, images of one individual can be characterized by the orthogonality to certain vectors. By introducing the concept of orthogonal discriminant vector (ODV) which is the vector orthogonal to the images of the same person, we can discriminate images of one person from others. Our experimental results show that the intra-class residue is much smaller than the inter-class residue. This means that the probe images can be

linearly approximated by the gallery images, which confirms the feasibility to represent the intrapersonal relationship among face images as derived in Subsection 4.2 .

The existence of ODV for face recognition is further proven theoretically by introducing a two-step algorithm to calculate ODVs via solving a linear equation system. The distance between a face image and an ODV is measured by a novel distance metric to categorize face images for classification. Our experimental results demonstrate that the new measurement is more effective than the existing methods and achieves lower EERs for face verification and higher accuracy for face recognition.

It is noted that all of face images are generated from the 3D face. Therefore, both the probe and gallery face images contain pixels of the 3D face. If a probe face image is formed by pixels which are contained in the gallery face images, it can be linearly expressed by the gallery images. This is tested by the experiment in subsection 4.1, where the intra-class residue is much smaller than the inter-class residue. Consequently, the intrapersonal relationship among face images is validated and our method performs well. Furthermore, as stated in Subsection 2.3, our method is easier to implement because it does not require the pose of the probe face image to approximate that of any gallery images.

However, if the probe face image has many pixels that are not contained in gallery face images, the intrapersonal relationship among face images across pose becomes inaccurate. The probe face image cannot be correctly represented by the linear combination of the gallery images. Thus, the distances between the probe image and ODVs do not necessarily approximate one, even if the ODVs are orthogonal to the gallery images.

As part of our future work, quantitative estimation of the reliability of an individual’s pose manifold under various conditions including scare pose situations will be studies. Occlusion is a very difficult issue in face recognition. The theoretical analysis of our proposed algorithm for intrapersonal relationship among face images under different poses without occlusion would laid the foundation for further development of a powerful approach to occluded face recognition. In addition, a general model of 3-D face will be developed based on the gallery images, where the gallery images can be used for face modeling with different parameters to achieve robustness with flexibility.

# Chapter 5 Feature extraction for face verification

## 5.1. Introduction

With a probe face image and a claimed identity, a face verification system tries to determine whether the image belongs to the claimed subject. It only compares the similarity between the probe image and the template of the claimed identity. The output can be genuine or imposter. From the point view of pattern recognition, face verification is a binary classification problem. In this task, the positive class consists of the representations of one object and negative class consists of anything else. It is an imbalanced problem because (1) the positive class has fewer samples than the negative class; (2) the positive samples (representations of one object) form a cluster while the negative samples (which can be anything that different from the positive samples) do not. In this section, we treat face verification as an imbalanced binary classification problem and propose a feature extraction and classification method.

As one of the fundamental problems in machine learning, learning from imbalanced datasets has attracted much attention in recent years [136-137]. One of the latest surveys is [156]. In this section, we limit our study on the imbalanced binary classification problem (or face verification). The imbalance has at least two forms. One form of imbalance is the number of samples, where one class has much more samples than the other class. Another form of imbalance is that the distributions of different classes are quite different.

Imbalanced data degrade the performances of many dimension reduction or feature extraction methods. When presented with imbalanced datasets, some methods tend to yield feature extractors that favor the majority class, such as principal

component analysis (PCA) [134]. The unsupervised PCA seeks the feature extractors that maximize the total scatter. Its feature extractor will be largely determined by the majority class if one class has much more samples than the other class. Some feature extraction methods cannot perform well on imbalanced datasets because they are essentially developed only for the balanced datasets, such as Fisher discriminant analysis (FDA) [138-139]. The supervised FDA aims to maximize the between class scatter and minimize the within class scatter. It is developed based on the assumption that samples from two classes are subjected to Gaussian distributions.

Many standard classifiers tend to favor the majority class on imbalanced data. Support vector machine (SVM) refers to the samples that near boundaries as support vectors and seeks the separating hyperplane that maximizes the separation margin between the hypothesized concept boundary and the support vectors [136]. The SVMs are inherently biased toward the majority class because they aim to minimize the total error. Multilayer perceptron (MLP) is proved to have difficulty in learning from imbalanced datasets [140]. Because of their ability of avoiding the so-called overfitting, the simple and robust linear classifiers are attractive, such as linear discriminant analysis (LDA), minimum square error (MSE), support vector machine (SVM) [141]. These classifiers make an implicit assumption that the positive and negative classes can be roughly separated by a hyperplane [96]. However, this assumption is violated in many imbalanced datasets (including face verification) where only the positive samples form a cluster, as detailed in section 5. 2. This explains why the performances of these linear classifiers are significantly degraded by the imbalanced datasets.

Different from the discriminative methods (LDA, MSE, and SVM), Gaussian mixture model (GMM) [142] is a generative method. In GMM, the distribution of the

samples is modeled by a linear combination of two or more Gaussian distributions [143-145]. GMM has been used in many fields [143-145], and can deal with the imbalanced problem if the parameters of the Gaussian distributions are well fixed. The main difficulty in GMM is to estimate the number of Gaussians to use [146].

This section proposes an imbalanced binary classification method for face verification. The proposed method seeks feature extractors that can generate minimum positive and maximum negative features in terms of absolute value. In other words, the positive features extracted by a feature extractor are expected to be in an interval $[-\xi,\xi]$, and the negative features fall into $(-\infty,-\xi)\cup(\xi,+\infty)$, where $\xi$ is a positive scalar. This agrees with the situation in a verification task where positive samples cluster together and the negative samples may not. To obtain the feature extractors, this section proposes two models and designs algorithms to solve these models. While model 1 first minimizes the positive features then maximizes the negative features, model 2 first maximizes the negative features then minimizes the positive features. After projecting the samples onto feature extractors, the proposed method classifies the features based on their weighted distances to the origin.

The advantages of the proposed method are mainly summarized as follows:

Firstly, the proposed method is less likely affected by the imbalanced distributions of the positive and negative classes in two aspects. Different from the traditional feature extraction methods that assume the positive and negative samples respectively cluster together, the proposed method only requires the positive samples cluster together (the negative samples can either cluster together or not). Different from the traditional linear classifiers that require the samples can be roughly separated by a single hyperplane, the proposed method can perform well if two parallel hyperplanes can separate the positive samples from the negative ones.

Secondly, the proposed method is less likely affected by the imbalanced sizes of the positive and negative classes. The positive and negative samples are independently input to two steps in the proposed algorithms. Thus, the two classes have equal power in determining the final feature extractors even though one class may consist of much more samples than the other class.

Thirdly, the proposed method significantly reduces the misclassification of the outliers into positive class. Different from traditional methods that assign two symmetric half-spaces to positive class and negative class, our method assigns two asymmetrical areas to these two classes. As the area for the positive class is much smaller than that of the negative class, the outliers are not likely to be misclassified into the positive class.

The rest of this section is organized as follows. Subsection 5.2 describes the background and motivation. Subsection 5.3 presents the proposed method. Subsection 5.4 presents the experiments and subsection 5.5 draws a conclusion.

## 5.2. Background and Motivation

We consider a binary classification problem, where the $d$ dimensional column vectors $x_1, x_2, \cdots, x_{n_1}$ are samples (i.e. face images) from the positive class with class label $y_i = 1\left(1 \le i \le n_1\right)$ and $x_{n_1+1}, x_{n_1+2}, \cdots, x_{n_1+n_2}$ from the negative class with class label $y_i = -1\left(n_1 + 1 \le i \le n\right)$. The total number of samples is $n$, where $n = n_1 + n_2$. We denote the matrix consists of all the training samples as $X = \left[x_1 \quad x_2 \quad \cdots \quad x_n\right]$, and the vector consists of all the class labels as $Y = \left[y_1 \quad y_2 \quad \cdots \quad y_n\right]^T$.

Imbalanced datasets degrade the performance of many feature extraction methods. Due to “curse of dimensionality” [147-148], a feature extraction procedure is

necessary in some tasks [149-151]. The subspace-based feature extraction methods [23-24, 42, 57, 152-153] perform well on balanced data. However, they tend to generate feature extractors that favor the majority class if one class dominates the other.

Imbalanced datasets degrade the performances of many classifiers. After feature extraction, a classifier maps the input feature vector space to the output class label space. In our binary classification problem, the class labels are +1 and -1. Most classifiers try to estimate the separate surface of these two classes in some way [154]. Three popular classifiers are K-nearest neighbor (KNN), multilayer perceptron (MLP), and support vector machine (SVM). Though these classifiers perform well on balanced datasets, they are proved to have difficulty in classifying imbalanced datasets [136, 140-141].

Because of their ability of avoiding the so-called overfitting, the simple and robust linear classifiers (LDA, MSE, SVM) are attractive [141]. However, their performances are significantly degraded by the imbalance of the dataset if: (1) the sizes of the positive and negative classes are imbalanced; (2) the samples of one class form a cluster while those of the other class do not. Another problem of these linear classifiers is that they tend to misclassify outliers into positive class. The rest of this subsection shows two problems of the linear classifiers (on imbalanced datasets), which classify a sample $x$ based on the sign of the value

$$f(x) = x^T w + w_0 \tag{5.1}$$

where $w$ is the coefficient vector and $w_0$ is the threshold.

Firstly, the linear classifiers fail to work if their common implicit assumption does not hold. The goal of a linear classifier is to seek a hyperplane for classification. This hyperplane divides the sample space into two half-spaces, and in them

respectively fall the samples of two classes. This goal is achievable only under an implicit assumption that the positive and negative classes can be roughly classified by a single hyperplane. Figure 5.1 shows the distribution of face images belonging to three different persons. All of these images are from the Yale face database [101]. In the verification of face 3 in Figure 5.1, the negative class consists of two distant subclusters (face 1 and 2). This violates the above assumption and thus incapacitates the linear classifiers. As the positive class represents a particular object while the negative class represents the whole "rest of the world" in a verification problem [152], it is common that the negative class and positive class are not linearly separable. So, the linear classifiers are usually not applicable in this imbalanced binary classification problem.

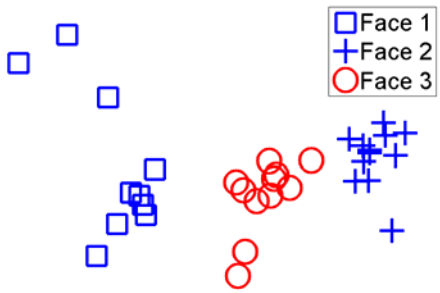



Figure 5.1 The distribution of the face images of three different individuals

Secondly, linear classifiers tend to misclassify outliers. Considering $x^T w + w_0$ as the feature of sample $x$, linear classifiers classify $x$ only based on the sign of this feature. They classify a sample into positive class if and only if it associates with a positive feature. This feature can infinitely approach zero or be infinitely large. However, it is problematic in some situations to classify the testing sample $x$ into the positive class if its feature is too large. Figure 5.2 shows an example. In this figure, the crosses ("+") denote the positive training samples and the squares ("□") denote

the negative training samples. The solid line (separating hyperplane) can separate the positive samples from the negative samples. Traditional methods consider a novel sample is positive if it lies right to the solid line. Based on this, the circle ("o") representing a testing sample will be classified into the positive class. However, this circle is far away from all positive samples and should be considered as an outlier in the negative class. Such outliers are unavoidable in verification problems, because it is hardly possible to collect a representative training set for the negative class.

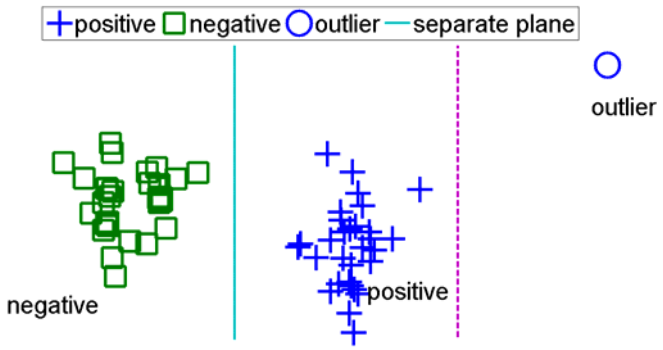



Figure 5.2 The classification of an outlier. The outlier is misclassified into the positive class by the solid line

Traditional classifiers misclassify the outlier into the positive class mainly because they classify a sample only based on the sign of the feature $x^T w + w_0$ and do not take the absolute value of this feature into consideration. One reasonable way to avoid such misclassification is bounding the positive feature from both below and above using two values, instead of only bounding it from below using zero.

## 5.3. Proposed method

In this subsection, we propose a new method for imbalanced binary classification. For simplicity, we consider the samples include an extra dimension with fix value 1 and the threshold $w_0$ (in equation (5.1)) turns to be an additional dimension of the coefficient vector. Also, as only the direction of the coefficient vector $w$ is important

for the classification, we restrict it to have a unit norm. This coefficient vector is also referred to as the feature extractor.

Subsection 5.3.1 introduces the basic idea of the proposed method. Subsection 5.3.2 and 5.3.3 propose two models and algorithms to solve these models. Subsection 5.3.4 presents the classification procedure of the proposed method and discussion.

### 5.3.1 Basic idea

The principal of the proposed method is to seek a pair of parallel hyperplanes $h^{\pm}(x): w^T x = \pm\xi$ for classification, as shown in Figure 5.3. The positive samples are expected to be clustered in the belt area $A$ defined as follows

$$A: -\xi \le w^T x \le +\xi \tag{5.2}$$

The negative samples are expected to be in the area $\overline{A}$ defined as follows

$$\overline{A}: w^T x > +\xi \cup w^T x < -\xi \tag{5.3}$$

Compared with the negative samples, the positive samples are nearer to the hyperplane $h^0(x): w^T x = 0$. Different from traditional linear classifiers that assign two symmetric half-spaces to positive class and negative class, our method assigns two asymmetrical areas to these two classes. To reflect the imbalance of the positive class and negative class, our method assigns a “larger” area for the negative class.

Alternatively, we can regard the scalar $w^T x$ as the feature of sample $x$ after projecting onto the feature extractor $w$. From equations (5.2) and (5.3), we know that the positive features fall into the interval $[-\xi, \xi]$, and the negative features fall into $(-\infty, -\xi) \cup (\xi, +\infty)$. To enlarge the separable, this method seeks the minimum positive and maximum negative features in terms of absolute value for classification.

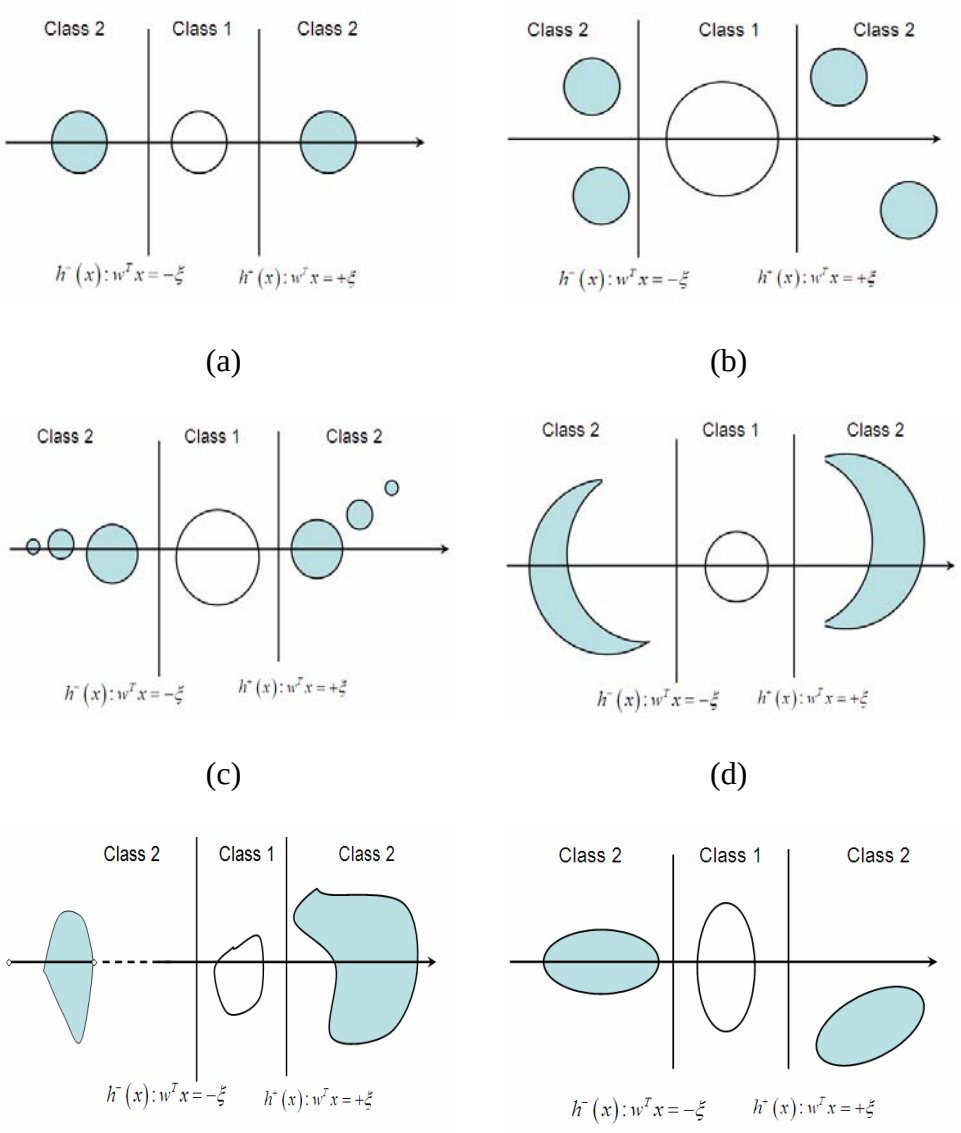



Figure 5.3 separate class 1 (positive) from class 2 (negative) using two parallel hyperplanes

Ideally, we can obtain the feature extractor $w$ by solving the following $n$ inequalities

$$\begin{cases} \left|w^T x_i\right| \le \xi & i = 1, 2, \cdots, n_1 \\ \left|w^T x_i\right| > \xi & i = n_1 + 1, n_1 + 2, \cdots, n \end{cases} \tag{5.4}$$

However, there are three problems in solving these inequalities. Firstly, there is no solution for the inequalities (5.4) in some cases. The existence of a solution for

inequalities (5.4) means we can classify the training samples using the hyperplanes $h^{\pm}(x): w^T x = \pm\xi$ with one hundred percent. This is not the case for many real classification problems. Secondly, when inequalities (5.4) are solvable and have infinite solutions, we have no straightforward criterion to assess the solutions and choose the best ones. Thirdly, solving a set of inequalities as many as the training samples is time consuming.

In the following, we modify the model (5.4) and generate two new models. By solving the new models, we work out the feature extractors $w$ efficiently.

### 5.3.2 Model 1

Model 1 first minimizes the positive features, and then maximizes the negative features in terms of absolute value. We can formulate it as follows

$$\max_{\min\left\|X_1^T v\right\|_2} \left\|X_2^T v\right\|_2 \tag{5.5}$$

where the matrices $X_1$ and $X_2$ respectively consists of all the positive and negative samples. We can use a similar procedure to be introduced in 5.3.3 to solve this model by solveing two eigendecomposition problems.

However, we can modify the model in (5.4) by setting the parameter $\xi$ to be zero, and obtain a special case of the model in (5.4), as follows

$$\max_w \left\|X_2^T w\right\|_2 \quad s.t. \quad \left\|X_1^T w\right\|_2 = 0 \tag{5.6}$$

Note that, model 1 maximizes the norm of the negative feature vector, instead of maximizing the smallest negative feature. If it is necessary to focus on the classification of the boundary samples, we can revise this model as follows

$$\max_w \underline{f} \quad s.t. \quad \left\|X_1^T w\right\|_2 = 0 \tag{5.7}$$

where $\underline{f} = \inf\left\{x_i^T w \mid n_1 + 1 \le i \le n\right\}$. Because solving (5.6) is much faster than solving (5.7), we adopt the model in equation (5.6) in this subsection

. Model in equation (5.6) has open solutions which are detailed in the following paragraphs.

To solve model in equation (5.6) efficiently, we design a two-step procedure. The first step generates a set of candidate feature extractors onto which the positive samples have zero projections. From this set, the second step takes the vectors onto which the negative samples have the maximum projections as the feature extractors.

To generate a set of vectors onto which the positive samples have zero projections, the first step solves the following linear equation system

$$X_1^T X_2 \mu = M \mu = 0 \tag{5.8}$$

In verification problems, positive samples are normally fewer than negative samples, i.e. $n_1 < n_2$. The matrix $M = X_1^T X_2 \in R^{n_1 \times n_2}$ has fewer rows than columns. Thus, the linear equation system (5.8) has a set of nonzero solutions. It can be easily proved that, corresponding to each nonzero solution $\mu$ of (5.8), the nonzero vector $X_2 \mu$ is orthogonal to all the positive samples. If $U = \left[\mu_1 \quad \mu_2 \quad \cdots \quad \mu_k\right]$ are a set of linearly independent solutions of (5.8), we can easily prove that the positive samples have zero projections on the vectors in the following set

$$S_1 = \left\{v \mid v = X_2 U \alpha, \alpha \in R^{k \times 1}\right\} \tag{5.9}$$

where $\alpha$ is a coefficient vector.

From the set $S_1$, the second step chooses vectors that can generate maximum negative projections. The projections of the negative samples onto $v = X_2 U \alpha$ form a vector $X_2^T X_2 U \alpha \in R^{n_2 \times 1}$. We can maximize this projection vector as follows

$$\begin{aligned}\max\left\|X_2^T X_2 U\alpha\right\|_2 &= \max\left\|X_2^T X_2 U\alpha\right\|_2^2 \\ &= \max \alpha^T U^T X_2^T X_2 X_2^T X_2 U\alpha \\ &= \max \alpha^T N\alpha\end{aligned} \tag{5.10}$$

The vector $\alpha$ should be the eigenvector of the matrix $N = U^T X_2^T X_2 X_2^T X_2 U \in R^{k\times k}$ corresponding to the leading eigenvalues. As can be seen from (5.9), there is a one-to-one correspondence between the $\alpha$ and $v$. Thus, we can work out the feature extractor $v$ once obtaining the vector $\alpha$.

In summary, we perform the following algorithm to solve model 1

Algorithm 5.1

Step 1: solve the linear equation system (5.8) and generate a set of linear independent solutions $\mu_1 \quad \mu_2 \quad \cdots \quad \mu_k$;

Step 2: solve the maximization problem (5.10) by performing an eigendecomposition procedure; work out the feature extractor $v_i = X_2 U\alpha_i$ where $\alpha_i$ is an eigenvector of the matrix $N$.

If the dimensionality of the training data is high, step 1 can generate many linearly independent vectors that orthogonal to the positive samples. Then, step 2 takes the most discriminative vectors as the feature extractors.

### 5.3.3 Model 2

We propose the second model as follows

$$\min_{\max\left\|X_2^T v\right\|_2} \left\|X_1^T v\right\|_2 \tag{5.11}$$

Among all the vectors $v$ onto which the negative samples have maximum projections, this model picks out the ones onto which the positive samples have

minimum projections and takes them as the feature extractors.

We design a two-step procedure to solve this model. The first step generates a set of vectors onto which the negative samples have projections as large as possible. From this set, the second step picks out the vectors onto which the positive samples have minimum projections.

The first step generates a set of vectors onto which the negative samples have maximum projections by solving the following maximization problem

$$\max_w \left\| X_2^T v \right\|_2 = \max_w \left\| X_2^T v \right\|_2^2 = \max_w v^T X_2 X_2{}^T v = \max_w v^T P v \tag{5.12}$$

where $P = X_2 X_2{}^T \in R^{d \times d}$ and $v \in R^{d \times 1}$ is a coefficient vector. The eigenvectors $e_1, e_2, \cdots, e_j$ of the matrix $P$ corresponding to the nonzero eigenvalues are the solution of the maximization problem in (5.12). Thus, $v$ should be a in the subspace spanned by these eigenvectors, and in the following set

$$S_2 = \left\{ v \mid v = E\beta, \beta \in R^{j \times 1} \right\} \tag{5.13}$$

where $E = \begin{bmatrix} e_1 & e_2 & \cdots & e_j \end{bmatrix} \in R^{d \times j}$ and $\beta$ is the coefficient vector.

The second step picks out vectors from $S_2$ onto which the positive samples have projections as small as possible by solving the following minimization problem

$$\min \left\| X_1{}^T E\beta \right\| = \min \beta^T E^T X_1 X_1{}^T E\beta = \min \beta^T Q \beta \tag{5.14}$$

where $Q = E^T X_1 X_1{}^T E \in R^{j \times j}$. The solutions of the minimization problem in equation (5.14) are the eigenvectors of the matrix $Q$ corresponding to the minimum eigenvalues. As $Q$ is a semi-positive definite matrix, its eigenvalues are larger than or equal to zero. Denoting the eigenvectors corresponding to the minimum eigenvalues as $\beta_1, \beta_2, \cdots, \beta_h$, we work out the coefficient vectors using $v_i = E\beta_i$.

In summary, we perform the following algorithm to solve model 2

Algorithm 5.2

Step 1: solve the maximization problem (5.14) by eigendecomposing the matrix $P = X_2 X_2^T$ and generate a set of eigenvectors $E = \begin{bmatrix} e_1 & e_2 & \cdots & e_j \end{bmatrix}$ corresponding to the maximum eigenvalues;

Step 2: solve the minimization problem (5.14) by eigendecomposing the matrix $Q = E^T X_1 X_1^T E$ and generate a set of eigenvectors $\beta_1, \beta_2, \cdots, \beta_h$ corresponding to the minimum eigenvalues; calculate the feature extractors using $v_i = E\beta_i$.

### 5.3.4 Classification and discussion

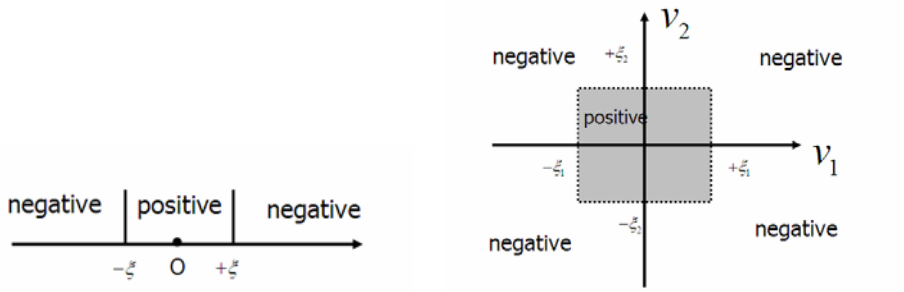
Projecting the samples onto the feature extractors $v_i \left(i = 1, 2, ..., n\right)$ output by algorithm 5.1 and 5.2, we can obtain minimum positive features and maximum negative features. Ideally, the positive features are within the interval $\left[-\xi_i, -\xi_i\right]$ and the negative features within $\left(-\infty, -\xi_i\right) \cup \left(\xi_i, +\infty\right)$. If the proposed method generates only one feature extractor, the positive features are near to the origin and negative features are far away from the origin, as shown in Figure 5.4 (a). The features of these two classes can be separated by two points. If the proposed method generates two feature extractors, the positive features are in a rectangle and the negative features are out of the rectangle as shown in Figure 5.4 (b). The situation is similar when we have more feature extractors. In Figure 5.4, the positive and negative features are expected to lie in two asymmetrical areas of the feature space. While positive features cluster together, the negative features can scatter anywhere else. This agrees with the situation in verification problems and avoids the two problems mentioned in subsection 5.2. Firstly, as the proposed method does not require the negative samples cluster together, it can perform well when the negative class consists of a number of

distant subclusters. Secondly, as the proposed method confines the positive samples in a small area, it can correctly classify an outlier into the negative class.

The feature extraction results of a test sample $x$ form a vector $z=\begin{bmatrix} z_1 & z_2 & \cdots & z_h \end{bmatrix}^T$, where $z_i$ is the projection of $x$ onto the feature extractor $v_i$. We can classify the feature $z$ based on its distance to the origin. In this thesis, we adopt the weighted block distance as follows

$$d(x)=\sum_{i=1}^{h} w_i |z_i| = \sum_{i=1}^{h} w_i \left| v_i^T x \right| \tag{5.15}$$

where $w_i$ is the weight for the $i$th feature extractor. If this distance is larger than a threshold, $x$ is classified into the negative class; or else, it is classified into the positive class.




(a) (b)

Figure 5.4 The projections of samples onto feature extractors: (a) one feature extractor; (b) two feature extractors

The same to the other methods, the proposed method takes into account both the positive and negative samples in the training stage. However, different from the traditional methods that input the positive and negative samples concurrently, the proposed method inputs one class after the other. This keeps our method away from the influence of the imbalanced sizes of the positive and negative classes. In the proposed two-step algorithms for the two models, either the positive or negative class

is independently input to one step. Even if the training set is imbalanced, the majority class cannot dominate the minority class.

Taking the one-against-others strategy, we can extend the proposed method to deal with the $c$-class problem ($c$ is the number of classes), as follows:

Training procedure: (generate feature extractors for each class)

For each $1 \le l \le c$, take the samples in the $l$th class as the positive samples and the rest as the negative samples; perform algorithm 5.1 or 5.2 to generate the feature extractors $v_1^l, v_2^l, \cdots, v_k^l$ for the $l$th class.

Classification procedure (classify the test sample $x$)

For each $1 \le l \le c$, calculate $d_l(x) = \sum_{i=1}^{k_l} w_i \left| \left(v_i^l\right)^T x \right|$ and classify $x$ into the $j$th class if $d_j(x) = \min_{1 \le l \le c} d_l(x)$.

## 5.4. Experiments

In this subsection, we first compare our method with different classifiers (back propagation, GMM, and five different forms of SVM) on two synthetic datasets in subsection 5.4.1. Then, we compare our method with different feature extraction methods (FDA, PCA, and LPP) on face verification in subsection 5.4.2. The experimental results validate the feasibility of the proposed method not only for face verification but also for general imbalanced binary classification problem.

### 5.4.1 Synthetic Data classification

The first dataset is drawn from three 2-dimensional random vectors $\Omega_0$, $\Omega_1$ and $\Omega_2$, each of which has Gaussian distribution with covariance matrix of $diag\{1,3\}$. The mean of the first random vector $\Omega_0$ is $(0,0)$, and those of $\Omega_1$ and

$\Omega_2$ are respectively $(5,1)$ and $(-5,1)$. We draw 50 positive training samples from $\Omega_0$, and draw 500 negative training samples respectively from $\Omega_1$ and $\Omega_2$. Thus, the training set consists of 50 positive samples and 1000 negative samples. Figure 5.5 shows the distribution of the training samples. The testing set also consists of 50 positive samples drawn from $\Omega_0$, and 1000 negative samples drawn from $\Omega_1$ and $\Omega_2$.

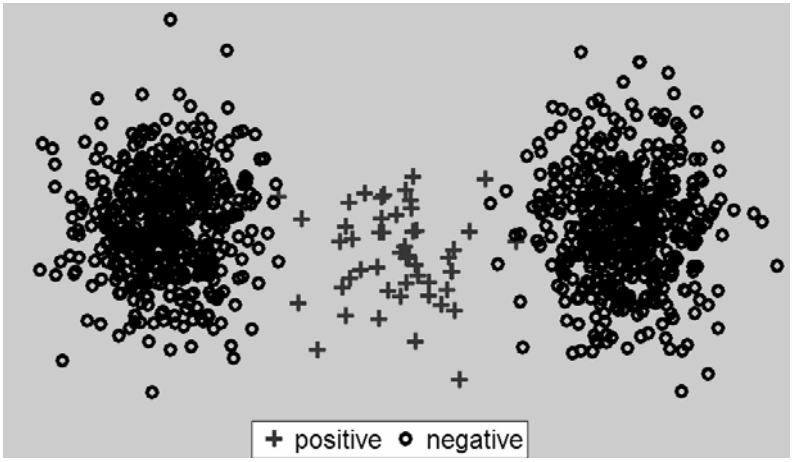



Figure 5.5 The distribution of the first synthetic dataset

The second dataset is drawn from five 2-dimensional random vectors $\Xi_0$, $\Xi_1$, $\Xi_2$, $\Xi_3$, and $\Xi_4$, each of which has Gaussian distribution with covariance matrix of $diag\{0.1,0.1\}$. The means of these five random vectors are respectively $(0,0)$, $(-1.5,-1.5)$, $(+1.5,-1.5)$, $(-1.5,+1.5)$, $(+1.5,+1.5)$. We draw 20 positive training samples from $\Xi_0$, and draw 40, 400, 40, 400 negative training samples respectively from $\Xi_1$, $\Xi_2$, $\Xi_3$, and $\Xi_4$. Thus, the training set consists of 20 positive samples and 880 negative samples. Figure 5.6 shows the distribution of the training samples. The testing set also consists of 20 positive samples from $\Xi_0$, and 40, 400, 40, 400 negative samples respectively drawn from $\Xi_1$, $\Xi_2$, $\Xi_3$, and $\Xi_4$.

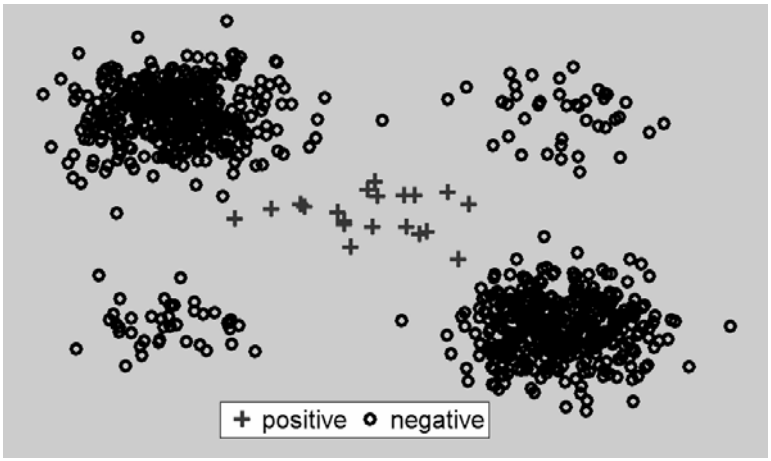



Figure 5.6 the distribution of the second synthetic dataset

Both of these two synthetic datasets consist of imbalanced positive and negative samples because of two reasons. Firstly, the positive samples are much fewer than the negative samples. The minority class only account for 4.76% in the first dataset and 2.22% in the second dataset of all the training samples. Secondly, the distributions of the positive and negative samples are quite different. While the positive samples are drawn from a single random vector, the negative samples are drawn from no less than two random vectors. While the positive samples form a single cluster, the negative samples form no less than two clusters.

In a binary classification problem, a true positive (TP) means a positive sample is correctly classified into the positive class and a true negative (TN) means a negative sample is correctly classified into the negative class. We use true positive rate (TPR) and true negative rate (TNR) to evaluate the performance of different methods. TPR is the ratio between the number of TP and that of all the positive samples and TNR is the ratio between the number of TN and that of the negative samples.

We compare our method with Multilayer perceptron (MLP) [140] which is a popular artificial neural network, and four different forms of SVM: linear support vector machine (LSVM) [96], Gaussian support vector machine (GSVM) [96], polynomial support vector machine (PSVM) [96], and one-class support vector

machine (OSVM) [155]. We also compare our method with GMM [142]. In GMM, we suppose the numbers of Gaussian distributions are known (3 for the first dataset and 5 for the second dataset). Table 5.1 lists the performances of these methods on the synthetic datasets.

Table 5.1 shows that each of the listed methods has a TNR higher than 95%. It indicates that these methods can correctly classify the majority negative samples. However, many minority samples are misclassified by LSVM, GSVM, PSVM, OSVM, and MLP. The TPRs of them are lower than 90% on both of these datasets. It indicates that these methods are biased toward the majority negative class on these imbalanced binary classification problems. Because the samples are drawn from random vectors that follow the Gaussian distributions and the number of these distributions are known, GMM achieves both high TPRs and TNRs.

Table 5.1 the performance (TPR and TNR) of different methods on synthetic datasets (%)

| methods | | LSVM | GSVM | PSVM | OSVM | MLP | GMM | Model 1 | Model 2 |
|---|---|---|---|---|---|---|---|---|---|
| The first dataset | TPR | 86.0 | 86.0 | 89.0 | 62.0 | 32.7 | 98.0 | 98.0 | 99.7 |
| | TNR | 99.8 | 99.8 | 99.3 | 99.9 | 97.4 | 99.6 | 95.2 | 99.5 |
| The second dataset | TPR | 13.6 | 85.0 | 77.5 | 68.0 | 60.8 | 95.0 | 98.7 | 92.6 |
| | TNR | 99.8 | 99.8 | 99.7 | 99.9 | 98.9 | 99.6 | 99.5 | 99.8 |

Our two models are robust to the imbalances in size and distribution and achieve high TPRs as well as high TNRs (larger than 92%). As can be seen from algorithm 5.1 and algorithm 5.2, the positive and negative samples are input independently in two steps. Thus, the feature extractors are not affected by the imbalanced class sizes. Also, as the proposed models only require the positive samples cluster together, they can achieve good performances on these datasets where the negative samples are drawn from several random vectors.

### 5.4.3 Face verification

One standard face database is the Carnegie Mellon University Pose, Illumination and Expression database (CMU PIE database) [130]. The CMU PIE database totally consists of more than 40 000 facial images of 68 people. In the construction of this database, the images of each individual are captured under 43 different illumination conditions, across 13 different poses, and with 4 different expressions.

We use a subset contains the face images of 10 individuals under five poses (C05, C07, C09, C27, C29) and all different illuminations and expressions. Figure 5. shows 40 face images of a person. There are 170 images for each of the 50 individual. The training set consists of 30 images of each individual, and the test set consists of the rest images.

Figure 5.7 Examples in the CMU PIE database

We verify each of the ten individuals 10 times with independent training sample set, and list the TPR, TNR and D (number of feature extractors) in Table 5.2. In the verification of one individual, positive samples are his/her face images and negative samples are images of the others. In this experiment, we compare our feature extraction methods with other methods, including PCA [23, 134], FDA [138-139] and LPP [45]. The features extracted by our method are classified based their weighted distances to the origin, as shown in equation (5.15). The features extracted by other methods are classified using the GSVM.

The feature extractors in PCA are eigenvectors of the total scatter matrix corresponding to all the nonzero eigenvalues. With 300 training samples, we obtain 299 feature extractors at the most. The number of feature extractors in FDA is $c-1$,

where $c$ is the number of classes. In a verification task, there are two classes and we

Table 5.2 Experimental results of face verification on CMU PIE subset TPR and TNR (%); and D (number of feature extractors)

| | Face 1 | | | Face 2 | | | Face 3 | | | Face 4 | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | TPR | TNR | D | TPR | TNR | D | TPR | TNR | D | TPR | TNR | D |
| PCA | 80.7 | **98.1** | 299 | 72.9 | 96.0 | 299 | 92.1 | 95.9 | 299 | 79.2 | **99.4** | 299 |
| FDA | 37.8 | 95.2 | 1 | 27.9 | 87.4 | 1 | 39.3 | 91.6 | 1 | 35.3 | 93.8 | 1 |
| LPP | 71.4 | 96.2 | 30 | 85.0 | 93.9 | 30 | 71.4 | 96.3 | 30 | 69.3 | **97.9** | 30 |
| Model 1 | **92.1** | 97.3 | 7 | **93.6** | **98.1** | 8 | **97.1** | **98.0** | 3 | **92.1** | 95.9 | 5 |
| Model 2 | **92.9** | **97.5** | 19 | **92.9** | **96.3** | 17 | **96.4** | **97.2** | 19 | **97.1** | 90.0 | 12 |

| | Face 5 | | | Face 6 | | | Face 7 | | | Face 8 | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | TPR | TNR | D | TPR | TNR | D | TPR | TNR | D | TPR | TNR | D |
| PCA | 81.4 | **99.0** | 299 | 85.0 | **98.2** | 299 | 76.4 | **94.8** | 299 | 84.3 | **96.2** | 299 |
| FDA | 22.9 | 89.7 | 1 | 42.4 | 92.2 | 1 | 22.9 | 88.3 | 1 | 29.3 | 93.9 | 1 |
| LPP | 79.3 | **99.1** | 30 | 82.8 | 97.2 | 30 | 88.6 | 93.9 | 30 | 84.3 | 95.7 | 30 |
| Model 1 | **94.3** | 90.1 | 6 | **99.3** | 96.1 | 3 | **92.1** | **95.2** | 8 | **91.4** | **95.8** | 3 |
| Model 2 | **92.9** | 91.3 | 18 | **96.4** | **98.9** | 14 | **92.9** | 92.7 | 9 | **93.5** | 95.7 | 19 |

| | Face 9 | | | Face 10 | | |
|---|---|---|---|---|---|---|
| | TPR | TNR | D | TPR | TNR | D |
| PCA | 70.8 | **96.3** | 299 | **97.8** | **98.9** | 299 |
| FDA | 23.7 | **93.8** | 1 | 58.6 | 96.5 | 1 |
| LPP | 82.1 | 92.9 | 30 | 91.4 | 97.5 | 30 |
| Model 1 | **91.4** | 93.3 | 5 | 96.4 | 98.0 | 3 |
| Model 2 | **93.9** | 92.7 | 12 | **97.9** | **98.4** | 4 |

have only one FDA-based feature extractor. The LPP-based feature extractors are obtained by solving a generalized eigenvalue problem. We keep all the PCA and LPP

feature extractors for dimension reduction. When solving the maximization problem (5.10) in model 1 and (5.14) in model 2, we keep the feature extractors that account for 95% of the spectrum.

Generally, the proposed models achieve higher TPR and TNR when verifying these 10 faces, as shown in Table 5.2. They achieve higher both TPR and TNR than the other methods on face 2 and face 3. In some cases, the other three feature extraction methods can achieve higher TNR than the proposed models. However, none of them can achieve higher TPR. In other words, compared with the proposed method, the other methods are more likely to misclassify the samples in the minority class. This indicates these methods are affected by the imbalance of the training dataset. The Wilcoxon signed-rank tests (with the same settings in 4.2) demonstrate that the differences of classification accuracies are statistically meaningful. Also, the proposed models have fewer feature extractors than PCA and LPP. The numbers of feature extractors in model 1 are no larger than 8 and those in model 2 are no larger than 19, while PCA and LPP respectively have 299 and 30 feature extractors. In summary, this table shows that the proposed method achieve higher classification accuracy using fewer feature extractors.

## 5.5. Conclusion

This section proposes a method for extracting minimum positive and maximum negative features in terms of absolute value for imbalanced binary classification. Corresponding to each feature extractor is a pair of parallel hyperplanes to separate the positive samples from the negative ones, as shown in Figure 5.3. This differentiates our method from the traditional linear classifiers that try to separate the samples using a single hyperplane. To obtain the feature extractors, this section

presents two models. Model 1 first generates a set of candidate feature extractors that can minimize the positive features to be zeros, and then chooses the ones among these candidates that can maximize the negative features. Model 2 first generates a set of candidate feature extractors that can maximize the negative features, and then chooses the ones among these candidates that can minimize the positive features.

In our experiments, while the positive samples are representations of one object, the negative samples are representations of more than two objects. Different from the positive samples that cluster together, the negative samples form no less than two subclusters. In other words, the distributions of the positive and negative classes are imbalanced. This degrades the performance of many traditional feature extraction methods and classifiers. However, as the proposed method only requires the positive samples cluster together, it can perform well and achieve high accuracy. Thus, the proposed method is less likely affected by the imbalanced distributions of the positive and negative samples.

In the training stage of many traditional feature extraction methods and classifiers, the positive and negative samples are input concurrently. In the proposed two models, however, the positive and negative samples are input independently in two steps. This alleviates the effect of the imbalanced sizes of the positive and negative classes.

In the classification stage, the proposed method assigns two asymmetrical areas to the imbalanced positive and negative classes. This restricts the positive features in a relatively small area. Thus, the outliers are less likely misclassified into the positive class.

# Chapter 6 Conclusion and Future work

## 6.1 Conclusion

The study of face recognition is motivated by its potential applications to the society ranging from law enforcement and surveillance to entertainment. In applications, face recognition is non-intrusive, natural, and easy to use. These advantages make face recognition becoming an important member of biometrics. How to extract features from the face images is a key problem in face recognition systems.

This thesis studies appearance-based feature extraction methods, which are the most popular in the field of face recognition in last two decades. The feature extraction procedure takes the cropped face image as input and outputs the low dimensional features. A good feature extraction method can make the follow-up classification procedure easier and improve the classification accuracy significantly. To achieve these two goals, the features should be abundant in discriminant information and be low in dimensional. This thesis studies and improves the feature extraction methods in three different situations.

In section 3, we propose fast kernel Fisher discriminant analysis (FKFDA) via approximating kernel principal component analysis (KPCA). To accelerate the nonlinear feature extraction procedure, the proposed FKFDA employs the idea of “node”. In stead of expressing the nonlinear feature extractors using all of the training samples, FKFDA expresses it using the selected nodes. We select nodes based on two criteria: pseudo-eigenvalue and similarity to the other nodes. We use the first criterion to assure that the nodes are most representative samples and qualified to replace the total samples to express discriminant vectors. We use the second criterion to control the number of nodes, and assure that no pair of nodes is highly similar to each other. A

higher threshold for allowable similarity means a larger node set. A different threshold means a different node set. The influence of this threshold is still unknown and needs further study.

In section 4, we study the intrapersonal relationship between face images and propose orthogonal discriminant vectors for face recognition. In this section, we consider the 2D images captured under different poses are projections of the 3D face object. We take the implicit 3D face object as a medium and use the training images (captured under different poses) to express the probe image in novel pose. Our theoretical analysis concludes that the face images of the same person under different poses lie on a manifold. Based on this conclusion we define a pose-invariant feature for recognition: orthogonality to the ODVs. We also study the existence of the ODV theoretically. After that, we propose an algorithm to calculate ODVs and use it perform recognition.

Different from section 3 and 4 which study identification as a multi-class problem, Section 5 studies verification as an imbalanced binary classification problem. We analyze not only the imbalanced character of verification, but also why this imbalance degrades the performance of feature extraction and classification methods. In this imbalanced binary classification task, the minority positive samples cluster together and the majority negative samples do not. Almost all of the feature extraction and classification methods favor the majority class. We propose the idea of separating these two classes using a pair or numerous pairs of parallel lines. Projecting the samples onto the perpendiculars of these lines, we obtain minimum positive and maximum negative features in terms of absolute value. We propose two algorithms to calculate the feature extractors. In these two algorithms, the minority class and majority class have equal power in determining the final feature extractor.

## 6.2 Future work

Though we human beings recognize each other by “face recognition” easily, it is still a difficult task to let computer has the equal capability. One of the unsolved problems is how to extract the optimal feature from the face images. Though the methods proposed in this thesis perform well in our experiments, they are far from satisfying applications. There are problems awaits us to consider.

1. *Illumination problem.* Ambient lighting changes from time to time. As the projection of the 3D face object, the 2D face images may have some shadows when the lighting source in some direction. The shadows can diminish some facial features. As tested by experiments, the intrapersonal distance is larger than the interpersonal distance in this situation [157]. Most feature extraction methods fail in this situation. One work is proposed based on the hypothesis that the face images (for a fixed pose) under different illuminations form a convex cone in the image space [101]. This work achieves higher classification accuracy than many other works. However, its computational complexity is very high. We would like to consider the face images as points in image space, and study the relationship between the face images captured under different illuminiations. This relationship may lead us to an approach for face recognition under different illumination conditions.
2. *Occlusion problem.* As almost all of the appearance-based methods can only extract global features, they cannot handle the partially occluded face images. One way to handle the occlusion problem is to partition the face images into several blocks and then use a voting strategy to make the final decision. The work [158] is superior to many other works by modeling each block using a mixture of Gaussians. However, the optimal partition of the face images is

still an unsolved problem. The sparse representation classification [104] expresses the test image as a linear combination of all training samples and makes decision based on their contribution. This global method can deal with the occlusion problem with a high computational burden. This tells us that appearance-based method may recognize face images under occlusion robustly. In the future, we would like to study the possibility of recognize face images robustly, as well as efficiently.

3. *Single sample problem.* Face recognition from one image per person (also referred to as single sample problem) is another important sub-area, which recently attracts increasing attention [159]. Single sample problem is particularly significant in some large scale identification problems, such as passport card identification, driver license identification, and law enforcement. LDA, LPP, and many other methods are not applicable to the one sample problem. We would like to stuy the reasons that make single sample problem more difficult than the many-sample problem. Also, we would like to propose a method to synthesize new face images. This turns the single sample problem into many-sample problem and makes the traditional feature extraction methods applicable.

# Bibliography


[1] A. K. Jain, A. Ross, and S. Prabhakar, "An introduction to biometric recognition," *IEEE Transactions on Circuits and Systems for Video Technology,* vol. 14, no. 1, pp. 4-20, 2004.

[2] S. Z. Li and A. K. Jain, *Handbook of face recognition*: Springer Science & Business, 2004.

[3] A. Tolba, A. El-Baz, and A. El-Harby, "Face recognition: a literature review," *International Journal of Signal Processing* vol. 2, no. 2, pp. 88-103, 2006.

[4] W. Zhao, R. Chellappa, A. Rosenfeld, and P. J. Phillips, "Face Recognition: A Literature Survey," *ACM Computing Surveys,* vol. 35, no. 4, pp. 399-458, 2003.

[5] I. S. Bruner and R. Tagiuri, *The perception of people. In Handbook of Social Psychology, Vol. 2, G. Lindzey, Ed., Addison-Wesley, Reading, MA, 634-654.*, 1954.

[6] W. W. Bledsoe, "The model method in facial recognition. ," Tech. rep. PRI:15, Panoramic research Inc., Palo Alto, CA.1964.

[7] C. Darwin, "The Expression of the Emotions in Man and Animals," *John Murray, London,U.K.*, 1972.

[8] M. D. Kelly, *Visual identification of people by computer*: Tech. rep. AI-130, Stanford AI Project, Stanford, CA., 1970.

[9] T. Kanade, *Computer recognition of human faces*. Basel, Switzerland, and Stuttgart, Germany, 1973.

[10] L. D. Introna and H. Nissenbaum, "Facial Recognition Technology. A Survey of Policy and Implementation Issues," Report of the Center for Catastrophe

Preparedness and Response, NYU, 2009.

[11] A. M. Martinez and R. Benavente, "The AR Face Database," CVC Technical Report #24, 1998.

[12] Y. Moses, Y. Adini, and S. Ullman, "Face recognition: The problem of compensating for changes in illumination direction," in *Proceedings of the European Conference on Computer Vision*, vol. A, pp. 286-296.

[13] L. Wiskott, J. M. Fellous, N. Kuiger, and C. von der Malsburg, "Face recognition by elastic bunch graph matching," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 19, no. 7, pp. 775-779, 1997.

[14] R. Brunelli and T. Poggio, "Face recognition: features versus templates," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 15, no. 10, pp. 1042-1052, 1993.

[15] L. D. Harmon, M. K. Khan, R. Lasch, and P. F. Ramig, "Machine identification of human faces," *Pattern Recognition*, vol. 13, no. 2, pp. 97-110, 1981.

[16] L. D. Harmon, S. C. Kuo, P. F. Ramig, and U. Raudkivi, "Identification of human face profiles by computer," *Pattern Recognition*, vol. 10, no. 5–6, pp. 301-312, 1978.

[17] G. J. Kaufman and K. J. Breeding, "The Automatic Recognition of Human Faces from Profile Silhouettes," *IEEE Transactions on Systems, Man and Cybernetics*, vol. SMC-6, no. 2, pp. 113-121, 1976.

[18] S. Z. Li and L. Juwei, "Face recognition using the nearest feature line method," *IEEE Transactions on Neural Networks*, vol. 10, no. 2, pp. 439-443, 1999.

[19] I. J. Cox, J. Ghosn, and P. N. Yianilos, "Feature-based face recognition using

mixture-distance," in Proceedings of the 1996 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR '96), pp. 209-216, 1996.

[20] P. N. Belhumeur, "Ongoing Challenges in Face Recognition," Reports on Leading-Edge Engineering from the 2005 Symposium, 2005.

[21] A. Rao and S. Noushath, "Subspace methods for face recognition," *Computer Science Review*, vol. 4, no. 1, pp. 1-17, 2010.

[22] L. Sirovich and M. Kirby, "Low-dimensional procedure for the characterization of human faces," *J. Opt. Soc. Am. A*, vol. 4, no. 3, pp. 519-524, 1987.

[23] M. Turk and A. Pentland, "Eigenfaces for Recognition," *Journal of Cognitive Neuroscience*, vol. 3, no. 1, pp. 71-86, 1991.

[24] P. S. Penev and J. J. Atick, "Local feature analysis: a general statistical theory for object representation," *Network: Computation in Neural Systems*, vol. 7, no. 3, pp. 477-500, 1996.

[25] C. AT&T Laboratories. The Database of Faces (available online http://www.cl.cam.ac.uk/research/dtg/attarchive/facedatabase.html) [Online].

[26] B. Moghaddam and A. Pentland, "Probabilistic visual learning for object representation," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 19, no. 7, pp. 696-710, 1997.

[27] J.-T. Chien and C.-C. Wu, "Discriminant waveletfaces and nearest feature classifiers for face recognition," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 24, no. 12, pp. 1644-1649, 2002.

[28] P. N. Belhumeur, J. P. Hespanha, and D. J. Kriegman, "Eigenfaces vs. Fisherfaces: recognition using class specific linear projection," *IEEE*

*Transactions on Pattern Analysis and Machine Intelligence*, vol. 19, no. 7, pp. 711-720, 1997.

[29] Q. Tian, M. Barbero, and Z. H. Gu, "Image classification by the Foley-Sammon transform," *Optical Engineering*, vol. 25, no. 7, pp. 834-840, 1986.

[30] W. Zhao, R. Chellappa, and A. Krishnaswamy, "Discriminant analysis of principal components for face recognition," in Proceedings of the Third IEEE International Conference on Automatic Face and Gesture Recognition, pp. 336-341, 1998.

[31] Y.-Q. Cheng, Y.-M. Zhuang, and J.-Y. Yang, "Optimal fisher discriminant analysis using the rank decomposition," *Pattern Recognition*, vol. 25, no. 1, pp. 101-111, 1992.

[32] D. L. Swets and J. J. Weng, "Using discriminant eigenfeatures for image retrieval," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 18, no. 8, pp. 831-836, 1996.

[33] R. Huang, Q. Liu, H. Lu, and S. Ma, "Solving the small sample size problem of LDA," in *Proceedings of the 16th International Conference on Pattern Recognition*, vol. 3, pp. 29-32, 2002.

[34] J. Yang, D. Zhang, and J. Y. Yang, "A generalised K-L expansion method which can deal with small sample size and high-dimensional problems," *Pattern Analysis & Applications*, vol. 6, no. 1, pp. 47-54, 2003.

[35] L.-F. Chen, H.-Y. M. Liao, M.-T. Ko, J.-C. Lin, and G.-J. Yu, "A new LDA-based face recognition system which can solve the small sample size problem," *Pattern Recognition*, vol. 33, no. 10, pp. 1713-1726, 2000.

[36] H. Yu and J. Yang, "A direct LDA algorithm for high-dimensional data — with

application to face recognition," *Pattern Recognition*, vol. 34, no. 10, pp. 2067-2070, 2001.

[37] J. Yu, Q. Tian, T. Rui, and T. S. Huang, "Integrating Discriminant and Descriptive Information for Dimension Reduction and Classification," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 17, no. 3, pp. 372-377, 2007.

[38] A. M. Martinez and A. C. Kak, "PCA versus LDA," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 23, no. 2, pp. 228-233, 2001.

[39] J. R. Beveridge, K. She, B. A. Draper, and G. H. Givens, "A nonparametric statistical comparison of principal component and linear discriminant subspaces for face recognition," in *Proceedings of the 2001 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR 2001)*, vol. 1, pp. 535-542, 2001.

[40] J. Yu, Q. Tian, T. Rui, and T. S. Huang, "Integrating Discriminant and Descriptive Information for Dimension Reduction and Classification," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 17, no. 3, pp. 372-377, 2007.

[41] L.-H. Chan, S.-H. Salleh, C.-M. Ting, and A. K. Ariff, "Face identification and verification using PCA and LDA," in *Proceedings of the International Symposium on Information Technology (ITSim 2008)*, vol. 2, pp. 1-6, 26-28 Aug. 2008.

[42] M. S. Bartlett, J. R. Movellan, and T. J. Sejnowski, "Face recognition by independent component analysis," *IEEE Transactions on Neural Networks*, vol. 13, no. 6, pp. 1450-1464, 2002.

[43] A. Hyvärinen and E. Oja, "Independent component analysis: algorithms and

applications," *Neural Networks*, vol. 13, no. 4–5, pp. 411-430, 2000.

[44] J.-F. Cardoso, "High-order contrasts for independent component analysis," *Neural Computation*, vol. 11, no. 1, pp. 157-192, 1999.

[45] X. He, S. Yan, Y. Hu, P. Niyogi, and H.-J. Zhang, "Face recognition using Laplacianfaces," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 27, no. 3, pp. 328-340, 2005.

[46] D. Cai, X. He, J. Han, and H.-J. Zhang, "Orthogonal Laplacianfaces for Face Recognition," *IEEE Transactions on Image Processing*, vol. 15, no. 11, pp. 3608-3614, 2006.

[47] Y. Fu, S. Yan, and T. S. Huang, "Correlation Metric for Generalized Feature Extraction," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 30, no. 12, pp. 2229-2235, 2008.

[48] A. Weingessel and K. Hornik, "Local PCA algorithms," *IEEE Transactions on Neural Networks*, vol. 11, no. 6, pp. 1242-1250, 2000.

[49] L. K. Saul and S. T. Roweis, "Think globally, fit locally: unsupervised learning of low dimensional manifolds," *J. Mach. Learn. Res.*, vol. 4, no. 1, pp. 119-155, 2003.

[50] S. Roweis and L. Saul, "nonlinear dimensionality reduction by locally linear embedding " *Science*, vol. 290, no. 5500, pp. 2323-2326, 2000.

[51] J. B. Tenenbaum, V. d. Silva, and J. C. Langford, "A Global Geometric Framework for Nonlinear Dimensionality Reduction," *Science*, vol. 290, no. 5500, pp. 2319-2323, December 22, 2000.

[52] M. Belkin and P. Niyogi, "Laplacian Eigenmaps for dimensionality reduction and data representation," *Neural Comput.*, vol. 15, no. 6, pp. 1373-1396, 2003.

[53] K. Q. Weinberger and L. K. Saul, "Unsupervised learning of image manifolds

by semidefinite programming," in *Proceedings of the 2004 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR 2004)*, vol. 2, pp. 988-995, 27 June-2 July 2004.

[54] K. R. Muller, S. Mika, G. Ratsch, K. Tsuda, and B. Scholkopf, "An introduction to kernel-based learning algorithms," *IEEE Transactions on Neural Networks*, vol. 12, no. 2, pp. 181-201, 2001.

[55] K. I. Kim, K. Jung, and H. J. Kim, "Face recognition using kernel principal component analysis," *IEEE Signal Processing Letters*, vol. 9, no. 2, pp. 40-42, 2002.

[56] B. Schölkopf, A. Smola, and K.-R. Müller, "Nonlinear Component Analysis as a Kernel Eigenvalue Problem," *Neural Computation*, vol. 10, no. 5, pp. 1299-1319, 1998.

[57] J. Yang, A. F. Frangi, J.-Y. Yang, D. Zhang, and Z. Jin, "KPCA plus LDA: a complete kernel Fisher discriminant framework for feature extraction and recognition," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 27, no. 2, pp. 230-244, 2005.

[58] L.-H. Zhao, X.-L. Zhang, and X.-H. Xu, "Face recognition base on KPCA with polynomial kernels," in *Proceedings of the International Conference on Wavelet Analysis and Pattern Recognition (ICWAPR '07)*, vol. 3, pp. 1213-1216, 2-4 Nov. 2007.

[59] J. Xie, "KPCA Based on LS-SVM for Face Recognition," in *Second International Symposium on Intelligent Information Technology Application (IITA '08)*, vol. 2, pp. 638-641, 20-22 Dec. 2008.

[60] X. Xie and K.-M. Lam, "Gabor-based kernel PCA with doubly nonlinear mapping for face recognition with a single face image," *IEEE Transactions on*

*Image Processing*, vol. 15, no. 9, pp. 2481-2492, 2006.

[61] G. Baudat and F. Anouar, "Generalized Discriminant Analysis Using a Kernel Approach," *Neural Computation*, vol. 12, no. 10, pp. 2385-2404, 2000.

[62] P. Howland and H. Park, "Generalizing discriminant analysis using the generalized singular value decomposition," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 26, no. 8, pp. 995-1006, 2004.

[63] L. Juwei, K. N. Plataniotis, and A. N. Venetsanopoulos, "Face recognition using kernel direct discriminant analysis algorithms," *IEEE Transactions on Neural Networks*, vol. 14, no. 1, pp. 117-126, 2003.

[64] S. Mika, G. Ratsch, J. Weston, B. Scholkopf, and K. R. Mullers, "Fisher discriminant analysis with kernels," in Proceedings of the 1999 IEEE Signal Processing Society Workshop Neural Networks for Signal Processing IX, pp. 41-48, 1999.

[65] J. Yang, D. Zhang, A. F. Frangi, and Y. Jing-yu, "Two-dimensional PCA: a new approach to appearance-based face representation and recognition," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 26, no. 1, pp. 131-137, 2004.

[66] W.-H. Yang and D.-Q. Dai, "Two-Dimensional Maximum Margin Feature Extraction for Face Recognition," *IEEE Transactions on Systems, Man, and Cybernetics, Part B: Cybernetics*, vol. 39, no. 4, pp. 1002-1012, 2009.

[67] J. Yang and J.-y. Yang, "From image vector to matrix: a straightforward image projection technique—IMPCA vs. PCA," *Pattern Recognition*, vol. 35, no. 9, pp. 1997-1999, 2002.

[68] J. Yang and C. Liu, "Horizontal and Vertical 2DPCA-Based Discriminant Analysis for Face Verification on a Large-Scale Database," *IEEE Transactions*

*on Information Forensics and Security*, vol. 2, no. 4, pp. 781-792, 2007.

[69] M. Li and B. Yuan, "2D-LDA: A statistical linear discriminant analysis for image matrix," *Pattern Recognition Letters*, vol. 26, no. 5, pp. 527-532, 2005.

[70] J. Yang, D. Zhang, X. Yong, and J.-y. Yang, "Two-dimensional discriminant transform for face recognition," *Pattern Recognition*, vol. 38, no. 7, pp. 1125-1129, 2005.

[71] S. Noushath, G. Hemantha Kumar, and P. Shivakumara, "(2D)2LDA: An efficient approach for face recognition," *Pattern Recognition*, vol. 39, no. 7, pp. 1396-1400, 2006.

[72] J. Ye, "Generalized Low Rank Approximations of Matrices," *Machine Learning*, vol. 61, no. 1, pp. 167-191, 2005.

[73] X.-Y. Jing, H.-S. Wong, and D. Zhang, "Face recognition based on 2D Fisherface approach," *Pattern Recognition*, vol. 39, no. 4, pp. 707-710, 2006.

[74] W.-H. Yang and D.-Q. Dai, "Two-Dimensional Maximum Margin Feature Extraction for Face Recognition," *IEEE Transactions on Systems, Man, and Cybernetics, Part B: Cybernetics*, vol. 39, no. 4, pp. 1002-1012, 2009.

[75] W. Zuo, D. Zhang, J. Yang, and K. Wang, "BDPCA plus LDA: a novel fast feature extraction technique for face recognition," *IEEE Transactions on Systems, Man, and Cybernetics, Part B: Cybernetics*, vol. 36, no. 4, pp. 946-953, 2006.

[76] J. Ye, R. Janardan, and Q. Li, "Two-dimensional linear discriminant analysis," in *Advances in Neural Information Processing Systems*. vol. 17, ed: Cambridge, MA: MIT Press, 2005, pp. 1669-1576.

[77] D. Zhang, Z.-H. Zhou, and C. Songcan, "Diagonal principal component analysis for face recognition," *Pattern Recognition*, vol. 39, no. 1, pp. 140-142,

2006.

[78] S. Noushath, G. Hemantha Kumar, and P. Shivakumara, "Diagonal Fisher linear discriminant analysis for efficient face recognition," *Neurocomputing*, vol. 69, no. 13–15, pp. 1711-1716, 2006.

[79] S. Chen, H. Zhao, M. Kong, and B. Luo, "2D-LPP: A two-dimensional extension of locality preserving projections," *Neurocomputing*, vol. 70, no. 4–6, pp. 912-921, 2007.

[80] D. Hu, G. Feng, and Z. Zhou, "Two-dimensional locality preserving projections (2DLPP) with its application to palmprint recognition," *Pattern Recognition*, vol. 40, no. 1, pp. 339-342, 2007.

[81] R. Zhi and Q. Ruan, "Facial expression recognition based on two-dimensional discriminant locality preserving projections," *Neurocomputing*, vol. 71, no. 7–9, pp. 1730-1734, 2008.

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

[83] J. H. Xu, X. G. Zhang, and Y. Li, "Kernel MSE algorithm: a unified framework for KFD, LS-SVM and KRR," in Proceedings of the International Joint Conference on Neural Networks(IJCNN-2001), Washington, pp. 1486-1491, 2001.

[84] K. Fukunaga, *Introduction to statistical pattern recognition*, second edition ed. New York: Academic Press, Inc., 1990.

[85] Y. Xu, D. Zhang, and J.-Y. Yang, "A feature extraction method for use with bimodal biometrics," *Pattern Recognition*, vol. 43, no. 3, pp. 1106-1115, 2010.

[86] D. Zhang, F. X. Song, Y. Xu, and Z. Z. Liang, *Advanced Pattern Recognition Technologies with Applications to Biometrics*. New York: IGI Global, 2009.

[87] M.-H. Yang, "Kernel Eigenfaces vs. Kernel Fisherfaces: Face recognition using kernel methods," in Proceedings of the Fifth IEEE International Conference on Automatic Face and Gesture Recognition, pp. 215-220, 2002.

[88] S. A. Billings and K. L. Lee, "Nonlinear Fisher discriminant analysis using a minimum squared error cost function and the orthogonal least squares algorithm," *Neural Networks*, vol. 15, no. 2, pp. 263-270, 2002.

[89] A. J. Smola and B. Scholkopf, "Sparse greedy matrix approximation for machine learning," in Proceedings of the 17th International Conf. on Machine Learning, San Francisco, pp. 911-918, 2000.

[90] Y. Xu, J.-y. Yang, J. Lu, and D.-j. Yu, "An efficient renovation on kernel Fisher discriminant analysis and face recognition experiments," *Pattern Recognition*, vol. 37, no. 10, pp. 2091-2094, 2004.

[91] Q. Zhu, "Reformative nonlinear feature extraction using kernel MSE," *Neurocomputing*, vol. 73, no. 16–18, pp. 3334-3337, 2010.

[92] Y. Xu, D. Zhang, Z. Jin, M. Li, and J.-Y. Yang, "A fast kernel-based nonlinear discriminant analysis for multi-class problems," *Pattern Recognition*, vol. 39, no. 6, pp. 1026-1033, 2006.

[93] C. K. I. Williams and M. Seeger, "Using the Nystrom method to speed up kernel machines," in *T. Leen, T. Dietterich, V. Tresp, editors, Advances in Neural Information Processing Systems 13*, ed Cambridge, MA: MIT Press, 2001.

[94] J. Wang, B. Xie, J. Xu, and H. Chen, "A fast KPCA-based nonlinear feature extraction method," in *Proceedings of the Asia-Pacific Conference on Computational Intelligence and Industrial Applications (PACIIA 2009)*, vol. 2, pp. 232-235, 28-29 Nov. 2009.

[95] J. Yang, Z. Jin, J.-y. Yang, D. Zhang, and A. F. Frangi, "Essence of kernel Fisher discriminant: KPCA plus LDA," *Pattern Recognition*, vol. 37, no. 10, pp. 2097-2100, 2004.

[96] O. D. Richard, E. H. Peter, and G. S. David, *Pattern Classification*. Beijing: China Machine Press, 2004.

[97] J. Mercer, "Functions of positive and negative type and their connection with the theory of integral equations," in Philos. Trans. Roy. Soc. London, pp. 415-446., 1909.

[98] M. Gibbs and D. J. C. Mackay, "Efficient implementation of Gaussian process," Technical report, Cavendish Laboratory, Cambridge, UK1997.

[99] W.-S. Chen, P. C. Yuen, J. Huang, and D.-Q. Dai, "Kernel machine-based one-parameter regularized Fisher discriminant method for face recognition," *IEEE Transactions on Systems, Man, and Cybernetics, Part B: Cybernetics*, vol. 35, no. 4, pp. 659-669, 2005.

[100] S. Mika, G. Ratsch, J. Weston, B. Scholkopf, A. Smola, and K. R. Muller, "Constructing descriptive and discriminative nonlinear features: Rayleigh coefficients in kernel feature spaces," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 25, no. 5, pp. 623-628, 2003.

[101] A. S. Georghiades, P. N. Belhumeur, and D. J. Kriegman, "From few to many: illumination cone models for face recognition under variable lighting and pose," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 23, no. 6, pp. 643-660, 2001.

[102] C. Kotropoulos, A. Tefas, and I. Pitas, "Frontal face authentication using morphological elastic graph matching," *IEEE Transactions on Image Processing*, vol. 9, no. 4, pp. 555-560, 2000.

[103] L. Chen, H. Man, and A. V. Nefian, "Face recognition based on multi-class mapping of Fisher scores," *Pattern Recognition*, vol. 38, no. 6, pp. 799-811, 2005.

[104] J. Wright, A. Y. Yang, A. Ganesh, S. S. Sastry, and M. Yi, "Robust Face Recognition via Sparse Representation," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 31, no. 2, pp. 210-227, 2009.

[105] S.-W. Lee, J. Park, and S.-W. Lee, "Low resolution face recognition based on support vector data description," *Pattern Recognition*, vol. 39, no. 9, pp. 1809-1812, 2006.

[106] R. Gross, I. Matthews, and S. Baker, "Appearance-based face recognition and light-fields," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 26, no. 4, pp. 449-465, 2004.

[107] F. S. Samaria and A. C. Harter, "Parameterisation of a stochastic model for human face identification," in Proceedings of the Second IEEE Workshop on Applications of Computer Vision, pp. 138-142, 1994.

[108] Y. Xu, F. Song, G. Feng, and Y. Zhao, "A novel local preserving projection scheme for use with face recognition," *Expert Systems with Applications*, vol. 37, no. 9, pp. 6718-6721, 2010.

[109] D. Cai, X. He, Y. Hu, J. Han, and T. Huang, "Learning a Spatially Smooth Subspace for Face Recognition," in Proceedings of the 2007 IEEE Conference on Computer Vision and Pattern Recognition (CVPR '07), pp. 1-7, 2007.

[110] X. Zhang and Y. Gao, "Face recognition across pose: A review," *Pattern Recognition*, vol. 42, no. 11, pp. 2876-2896, 2009.

[111] S.-W. Lee, S.-H. Moon, and S.-W. Lee, "Face recognition under arbitrary illumination using illuminated exemplars," *Pattern Recognition*, vol. 40, no. 5,

pp. 1605-1620, 2007.

[112] S. Biswas, G. Aggarwal, and P. J. Flynn, "Pose-robust recognition of low-resolution face images," in Proceedings of the 2011 IEEE Conference on Computer Vision and Pattern Recognition (CVPR 2011), pp. 601-608, 2011.

[113] G. Fanelli, J. Gall, and L. Van Gool, "Real time head pose estimation with random regression forests," in Proceedings of the 2011 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), pp. 617-624, 2011.

[114] S. Khaleghian, H. R. Rabiee, and M. H. Rohban, "Face recognition across large pose variations via Boosted Tied Factor Analysis," in 2011 IEEE Workshop on Applications of Computer Vision (WACV), pp. 190-195, 2011.

[115] I. Wijaya, "Pose Invariant Face Recognition Using Dominant Frequency Based Holistic Features and Statistical Classifier," Doctor, Computer Science and Electrical Engineering, kumamoto University, 2010(available via google).

[116] L. Teijeiro-Mosquera, J. L. Alba-Castro, and D. Gonzalez-Jimenez, "Face Recognition Across Pose with Automatic Estimation of Pose Parameters through AAM-Based Landmarking," in Proceedings of the 20th International Conference onPattern Recognition (ICPR 2010), pp. 1339-1342, 2010.

[117] A. Li, S. Shan, and W. Gao, "Coupled Biased-Variance Tradeoff for Cross-Pose Face Recognition," *IEEE Transactions on Image Processing*, vol. 21, no. 1, pp. 305-315, 2012.

[118] Y. Chang, C. Hu, and M. Turk, "Manifold of facial expression," in Proceedings of the IEEE International Workshop on Analysis and Modeling of Faces and Gestures (AMFG 2003), pp. 28-35, 2003.

[119] K.-C. Lee, J. Ho, M.-H. Yang, and D. Kriegman, "Video-based face recognition using probabilistic appearance manifolds," in *Proceedings of the*

*2003 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (2003)*, vol. 1, pp. 313-320, 18-20 June 2003.

[120] A. Shashua, A. Levin, and S. Avidan, "Manifold pursuit: a new approach to appearance based recognition," in *Proceedings of the 16th International Conference onPattern Recognition*, vol. 3, pp. 590-594, 2002.

[121] S. Yan, D. Xu, B. Zhang, H.-J. Zhang, Q. Yang, and S. Lin, "Graph Embedding and Extensions: A General Framework for Dimensionality Reduction," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 29, no. 1, pp. 40-51, 2007.

[122] X. He, D. Cai, S. Yan, and H.-J. Zhang, "Neighborhood preserving embedding," in *Proceedings of the Tenth IEEE International Conference on Computer Vision (ICCV 2005)*, vol. 2, pp. 1208-1213, 17-21 Oct. 2005.

[123] H.-T. Chen, H.-W. Chang, and T.-L. Liu, "Local discriminant embedding and its variants," in *Proceedings of the 2005 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR 2005)*, vol. 2, pp. 846-853, 20-25 June 2005.

[124] J. Wang, Y. Xu, D. Zhang, and J. You, "An efficient method for computing orthogonal discriminant vectors," *Neurocomputing*, vol. 73, no. 10–12, pp. 2168-2176, 2010.

[125] T. Xiong, J. Ye, and V. Cherkassky, "Kernel Uncorrelated and Orthogonal Discriminant Analysis: A Unified Approach," in *Proceedings of the 2006 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR 2006)*, vol. 1, pp. 125-131, 17-22 June 2006.

[126] Y. Hamamoto, A. Ohama, T. Kanaoka, and S. Tomita, "Orthogonal discriminant analysis based on a modified Fisher criterion [feature

extraction]," in *Proceedings of the 11th IAPR International Conference on Pattern Recognition, Conference B: Pattern Recognition Methodology and Systems*, vol. II, pp. 363-366, 30 Aug-3 Sep 1992.

[127] X. Chai, S. Shan, X. Chen, and W. Gao, "Locally Linear Regression for Pose-Invariant Face Recognition," *IEEE Transactions on Image Processing*, vol. 16, no. 7, pp. 1716-1725, 2007.

[128] D. Beymer and T. Poggio, "Face recognition from one example view," in Proceedings of the Fifth International Conference on Computer Vision, pp. 500-507, 1995.

[129] P. J. Phillips, M. Hyeonjoon, S. A. Rizvi, and P. J. Rauss, "The FERET evaluation methodology for face-recognition algorithms," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 22, no. 10, pp. 1090-1104, 2000.

[130] T. Sim, S. Baker, and M. Bsat, "The CMU Pose, Illumination, and Expression (PIE) database," in Proceedings of the Fifth IEEE International Conference on Automatic Face and Gesture Recognition, pp. 46-51, 2002.

[131] A. B. J. Teoh, A. Goh, and D. C. L. Ngo, "Random Multispace Quantization as an Analytic Mechanism for BioHashing of Biometric and Random Identity Inputs," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 28, no. 12, pp. 1892-1901, 2006.

[132] Y. C. Feng, P. C. Yuen, and A. K. Jain, "A Hybrid Approach for Generating Secure and Discriminating Face Template," *IEEE Transactions on Information Forensics and Security*, vol. 5, no. 1, pp. 103-117, 2010.

[133] Y. Z. Goh, A. B. J. Teoh, and K. O. M. Goh, "Wavelet-based illumination invariant preprocessing in face recognition," *Journal of Electronic Imaging*,

vol. 18, no. 2, pp. 1-12, 2009.

[134] M. Kirby and L. Sirovich, "Application of the Karhunen-Loeve procedure for the characterization of human faces," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 12, no. 1, pp. 103-108, 1990.

[135] Y. Xu, D. Zhang, F. Song, J.-Y. Yang, Z. Jing, and M. Li, "A method for speeding up feature extraction based on KPCA," *Neurocomputing*, vol. 70, no. 4–6, pp. 1056-1061, 2007.

[136] H. He and E. A. Garcia, "Learning from Imbalanced Data," *IEEE Transactions on Knowledge and Data Engineering*, vol. 21, no. 9, pp. 1263-1284, 2009.

[137] T. M. Khoshgoftaar, J. Van Hulse, and A. Napolitano, "Supervised Neural Network Modeling: An Empirical Investigation Into Learning From Imbalanced Data With Labeling Errors," *IEEE Transactions on Neural Networks*, vol. 21, no. 5, pp. 813-830, 2010.

[138] Y. Xu, J.-Y. Yang, and Z. Jin, "A novel method for Fisher discriminant analysis," *Pattern Recognition*, vol. 37, no. 2, pp. 381-384, 2004.

[139] J. Yang, J.-y. Yang, and D. Zhang, "What's wrong with Fisher criterion?," *Pattern Recognition*, vol. 35, no. 11, pp. 2665-2668, 2002.

[140] Y. L. Murphey, H. Guo, and L. A. Feldkamp, "Neural Learning from Unbalanced Data," *Applied Intelligence*, vol. 21, no. 2, pp. 117-128, 2004.

[141] W. Chen, C. E. Metz, M. L. Giger, and K. Drukker, "A Novel Hybrid Linear/Nonlinear Classifier for Two-Class Classification: Theory, Algorithm, and Applications," *IEEE Transactions on Medical Imaging*, vol. 29, no. 2, pp. 428-441, 2010.

[142] B. Scherrer, "Gaussian Mixture Model Classifiers," no. 2007. (Available at http://www.music.mcgill.ca~scherrer/MUMT611/a03/Scherrer07GMM.pdfS).

[143] D. A. Reynolds, "Speaker identification and verification using Gaussian mixture speaker models," *Speech Commun.*, vol. 17, no. 1-2, pp. 91-108, 1995.

[144] P. Bansal, K. Kant, S. Kumar, A. Sharda, and S. Gupta, "Improved Hybrid Model of HMM/GMM for speech recognition," *Intelligent Information and Engineering Systems*, vol. 2, no. 1, pp. 69-74, 2008.

[145] D. A. Reynolds and R. C. Rose, "Robust text-independent speaker identification using Gaussian mixture speaker models," *IEEE Transactions on Speech and Audio Processing*, vol. 3, no. 1, pp. 72-83, 1995.

[146] J. V. B. Soares and R. M. Cesar-Jr., *Segmentation of retinal vasculature using wavelets and supervised classification: Theory and implementation, In Automated Image Detection of Retinal Pathology, H. F. Jelinek and M. J. Cree, Eds., ed: CRC Press, 2007.*

[147] R. Bellman, *Adaptive control processes: a guided tour*: NJ: Princeton Univ. Press, 1961.

[148] T. Zhang, K. Huang, X. Li, J. Yang, and D. Tao, "Discriminative Orthogonal Neighborhood-Preserving Projections for Classification," *IEEE Transactions on Systems, Man, and Cybernetics, Part B: Cybernetics*, vol. 40, no. 1, pp. 253-263, 2010.

[149] D. Tao, X. Li, X. Wu, and S. J. Maybank, "Geometric Mean for Subspace Selection," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 31, no. 2, pp. 260-274, 2009.

[150] J. Sun, D. Tao, S. Papadimitriou, P. S. Yu, and C. Faloutsos, "Incremental tensor analysis: Theory and applications," *ACM Trans. Knowl. Discov. Data*, vol. 2, no. 3, pp. 1-37, 2008.

[151] T. Zhang, D. Tao, X. Li, and T. Yang, "A unifying framework for spectral

analysis based dimensionality reduction," in Proceedings of the IEEE International Joint Conference on Neural Networks (IJCNN 2008), pp. 1670-1677, 2008.

[152] X. Jiang, "Asymmetric Principal Component and Discriminant Analyses for Pattern Classification," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 31, no. 5, pp. 931-937, 2009.

[153] D. D. Lee and H. S. Seung, "Learning the parts of objects by nonnegative matrix factorization," *Nature*, vol. 401, no. pp. 788-791, 1999.

[154] G. Polzlbauer, T. Lidy, and A. Rauber, "Decision Manifolds&: A Supervised Learning Algorithm Based on Self-Organization," *IEEE Transactions on Neural Networks*, vol. 19, no. 9, pp. 1518-1530, 2008.

[155] L. M. Manevitz and M. Yousef, "One-class svms for document classification," *J. Mach. Learn. Res.*, vol. 2, no. 1, pp. 139-154, 2002.

[156] Y. Sun, A. K. C. Wong, M. S. Kamel, classification of imbalanced data: a review, Int. Journal of Artificial Intelligence and Pattern Recognition, vol. 23, no. 4, pp. 687-719, 2009.

[157] Y. Adini, Y. Moses, and S. Ullman, "Face recognition: the problem of compensating for changes in illumination direction," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 19, no. 7, pp. 721-732, 1997.

[158] A. M. Martinez, "Recognizing imprecisely localized, partially occluded, and expression variant faces from a single sample per class," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 24, no. 6, pp. 748-763, 2002.

[159] X. Tan, S. Chen, Z.-H. Zhou, and F. Zhang, "Face recognition from a single image per person: A survey," *Pattern Recognition*, vol. 39, no. 9, pp.

1725-1745, 2006.