COMPUTATIONAL INTELLIGENCE AND ITS APPLICATIONS SERIES

Biometric Image Discrimination Technologies

David Zhang, Xiaoyuan Jing, Jian Yang

Biometric Image Discrimination Technologies

David Zhang
Biometrics Research Centre,
The Hong Kong Polytechnic University, Hong Kong

Xiaoyuan Jing
Bio-Computing Research Centre,
ShenZhen Graduate School of Harbin Institute of Technology, China

Jian Yang
Biometrics Research Centre,
The Hong Kong Polytechnic University, Hong Kong

IDEA GROUP PUBLISHING
Hershey • London • Melbourne • Singapore

Acquisitions Editor: Michelle Potter
Development Editor: Kristin Roth
Senior Managing Editor: Amanda Appicello
Managing Editor: Jennifer Neidig
Copy Editor: Julie LeBlanc
Typesetter: Sharon Berger
Cover Design: Lisa Tosheff
Printed at: Yurchak Printing Inc.

Published in the United States of America by
Idea Group Publishing (an imprint of Idea Group Inc.)
701 E. Chocolate Avenue
Hershey PA 17033
Tel: 717-533-8845
Fax: 717-533-8661
E-mail: cust@idea-group.com
Web site: http://www.idea-group.com

and in the United Kingdom by
Idea Group Publishing (an imprint of Idea Group Inc.)
3 Henrietta Street
Covent Garden
London WC2E 8LU
Tel: 44 20 7240 0856
Fax: 44 20 7379 0609
Web site: http://www.eurospanonline.com

Library of Congress Cataloging-in-Publication Data

Zhang, David, 1949-
Biometric image discrimination technologies / David Zhang, Xiaoyuan Jing and Jian Yang.
p. cm.
Summary: "The book gives an introduction to basic biometric image discrimination technologies including theories that are the foundations of those technologies and new algorithms for biometrics authentication"--Provided by publisher.
Includes bibliographical references and index.
ISBN 1-59140-830-X (hardcover) -- ISBN 1-59140-831-8 (softcover) -- ISBN 1-59140-832-6 (ebook)
1. Pattern recognition systems. 2. Identification--Automation. 3. Biometric identification. I. Jing, Xiaoyuan. II. Yang, Jian. III. Title.
TK7882.P3Z 44 2006
006.4--dc22

2005032048

British Cataloguing in Publication Data
A Cataloguing in Publication record for this book is available from the British Library.

Biometric Image Discrimination Technologies

Table of Contents

Section II: Improved BID Technologies

Preface

Personal identification and verification both play a critical role in our society. Today, more and more business activities and work practices are computerized. E-commerce applications, such as e-banking, or security applications, such as building access, demand fast, real-time and accurate personal identification. Traditional knowledge-based or token-based personal identification or verification systems are tedious, time-consuming, inefficient and expensive.

Knowledge-based approaches use “something that you know” (such as passwords and personal identification numbers) for personal identification; token-based approaches, on the other hand, use “something that you have” (such as passports or credit cards) for the same purpose. Tokens (e.g., credit cards) are time-consuming and expensive to replace. Passwords (e.g., for computer login and e-mail accounts) are hard to remember. A company may spend $14 to $28 (U.S.) on handling a password reset, and about 19% of help-desk calls are related to the password reset problem. This may suggest that the traditional knowledge-based password protection is unsatisfactory. Since these approaches are not based on any inherent attribute of an individual in the identification process, they are unable to differentiate between an authorized person and an impostor who fraudulently acquires the “token” or “knowledge” of the authorized person. These shortcomings have led to biometrics identification or verification systems becoming the focus of the research community in recent years.

Biometrics, which refers to automatic recognition of people based on their distinctive anatomical (e.g., face, fingerprint, iris, etc.) and behavioral (e.g., online/off-line signature, voice, gait, etc.) characteristics, is a hot topic nowadays, since there is a growing need for secure transaction processing using reliable methods. Biometrics-based authentication can overcome some of the limitations of the traditional automatic personal identification technologies, but still, new algorithms and solutions are required.

After the Sept. 11, 2001 terrorist attacks, the interest in biometrics-based security solutions and applications has increased dramatically, especially in the need to spot potential criminals in crowds. This further pushes the demand for the development of different biometrics products. For example, some airlines have implemented iris recognition technology in airplane control rooms to prevent any entry by unauthorized persons. In 2004, all Australian international airports implemented passports using face recognition technology for airline crews, and this eventually became available to all Australian passport holders. A steady rise in revenues is predicted from biometrics for 2002-2007, from $928 million in 2003 to $4.035 million in 2007.

Biometrics involves the automatic identification of an individual based on his physiological or behavioral characteristics. In a non-sophisticated way, biometrics has existed for centuries. Parts of our bodies and aspects of our behavior have historically been used as a means of identification. The study of finger images dates back to ancient China; we often remember and identify a person by his or her face, or by the sound of his or her voice; and signature is the established method of authentication in banking, for legal contracts and many other walks of life.

However, automated biometrics has only 40 years' history. As everyone knows, matching finger images against criminal records is always an important way for law enforcers to find the criminal. But the manual process of matching is laborious and uses too much manpower. In late 1960s, the Federal Bureau of Investigation (FBI) began to automatically check finger images, and by the mid-1970s, a number of automatic finger scanning systems had been installed. Among these systems, Identimat is the first commercial system, as part of a time clock at Shearson Hamill, a Wall Street investment firm. This system measured the shape of hand and looked particularly at finger length. Though the production of Identimat ceased in late 1980s, its use pioneered the application of hand geometry and set a path for biometrics technologies as a whole. Besides finger and hand, some other biometrics techniques have also been developed. For example, fingerprint-based automatic checking systems were widely used in law enforcement by the FBI and other U.S. government departments. Advances in hardware, such as faster processing power and greater memory capacity, made biometrics more viable. Since the 1990s, iris, retina, face, voice, signature, DNA and palmprint technologies have joined the biometric family.

From 1996, and especially in 1998, more funds had been given to biometrics technology research and development. Therefore, research on biometrics became more active and exceeded the stage of separate research dispersed in pattern recognition, signal processing, image processing, computer vision, computer security and other subjects. By its distinguished features, such as live scan, identical person maximum likelihood and different person minimum likelihood, biometrics grew into an independent research field. A series of prominent events also shows that biometrics is garnering much more attention in both academia and industry. For example, in September 1997, *Proceedings of IEEE* published a special issue on automated biometrics; in April 1998, the BioAPI Consortium was formed to develop a widely available and accepted API (application program interface) that will serve for various biometrics technologies.

Today, biometrics-based authentication and identification are emerging as a reliable method in our international and interconnected information society. With rapid progress in electronics and Internet commerce, there has been a growing need for secure transaction processing using biometrics technology. This means that biometrics technology is no longer only the high-tech gadgetry of Hollywood science-fiction

movies. Many biometrics systems are being used for access control, computer security and law enforcement. The future of biometrics technology is promising. More and more biometrics systems will be deployed for different applications in our daily life. Several governments are now, or will soon be, using biometrics technology, such as the U.S. INSPASS immigration card or the Hong Kong ID card, both of which store biometric features for authentication. Also, banking and credit companies have applied biometrics technology to their business processes. In active use by some airports and airlines even before the Sept. 11, 2001 disaster, more are seriously considering the use of biometric authentication in the wake of these events. Now, biometrics technology not only protects our information and our property, but also safeguards our lives and our society.

Automated biometrics deal with image discrimination for a fingerprint, palmprint, iris, hand or face, which can be used to authenticate a person's claim to identity or establish an identity from a database. In other words, image discrimination is an elementary problem in the area of automated biometrics. With the development of biometrics and its applications, many classical discrimination technologies are borrowed and applied to deal with biometric images. Among them, principal component analysis (PCA, or K-L transform) and Fisher linear discriminant analysis (LDA) turns out to be very successful, in particular for face image recognition. Also, these methods have been greatly improved with respect to the specific biometric image analysis and applications. Recently, non-linear projection analysis technology represented by kernel principal component analysis (KPCA) and kernel Fisher discriminant (KFD), also show great potential in dealing with biometric problems. In fact, discrimination technologies can play an important role in the implementation of biometric systems. They provide methodologies for automated personal identification or verification. In turn, the applications in biometrics also facilitate the development of discrimination methodologies and technologies, making discrimination algorithms more suitable for image feature extraction and recognition. Since image discrimination is an elementary problem in the area of automated biometrics, *biometric image discrimination* (BID) should be developed. Now, many researchers not only apply the technology to BID, but also improve these useful approaches, even develop many related new methods. However, according to the authors' best knowledge, so far, very few books have been found exclusively devoted to such technology of BID.

In fact, BID technologies can be briefly defined as automated methods of feature extraction and recognition based on given biometric images. It should be stressed that the BID technologies are not the simple application of classical discrimination techniques to biometrics, but the improved or reformed discrimination techniques that are more suitable (e.g., more powerful in recognition performance or computational more efficient for feature extraction or classification) for biometrics applications. In other words, BID technologies should be with respect to the characteristics of BID problems, and find effective ways to solve these problems.

In general, BID problems have the following three characteristics: (1) *High dimensional* — This is due to the high-dimensional characteristic of biometric images, which make the direct classification in image space almost impossible, because the similarity calculation is computationally very expensive, as well as the large amounts of storage is required, let alone the performance of classification in varying lighting condition. So, a dimension reduction technique is necessary prior to recognition. (2) *Large scale* — In real-world applications, there are a number of typical large-scale BID prob-

lems. Given an input biometric sample, a large-scale BID identification system determines if the pattern is associated with any of a large number of enrolled identities, and these large-scale BID applications require high-quality BID technologies with good generalization power. (3) *Small sample size* — Differing from optical character recognition (OCR) problems, the training samples per class are always very limited, even one sample available for each individual, in real-world BID problems. The characteristics of high-dimensionality and small sample size make the BID problems become the so-called small sample size problems. In these problems, the within-class scatter matrix is always singular because the training sample size is generally less than the space dimension.

On BID problems, above all, we should determine how to represent the biometric images. The objectives of image representation are twofold. One is for a better identification (or verification), and the other is for an efficient similarity calculation. On the one hand, the sample points in image space generally lead to an unsatisfactory cluster, especially under the variations of illumination, time or other conditions. By virtue of feature extraction techniques, it can be expected to obtain a set of features, which is smaller in amount but more discriminative. These features may be more insensitive to the intra-class variations, such as that derived from varying lighting conditions. On the other hand, by feature extraction, the number of features is significantly reduced. This greatly benefits for the subsequent classification; less storage space is required and classification efficiency is improved.

Different biometric images, however, may have different representation methods. For example, we can get geometric features like the character of eyes, nose, mouth from face images; principal lines and wrinkles features from palmprint images; and minutiae points, ridges and singular points features from fingerprint images. These feature generation and representation methods depend on the specific category of biometric images.

It is necessary to work on exploring the common representation methods by virtue of discrimination technologies; that is, the methods applicable for any biometric images. Generally, there are two cases of applications of BID technologies for image representation. One is original image-based, and the other is feature-based. In the first case, BID technologies are used to derive the discriminative features directly from the original biometric images. In the second class, BID technologies are employed for the second feature extraction based on the features derived from other feature generation approaches (e.g., Fourier transform, wavelet transform, etc.). In a word, BID technologies suggest different ways to represent biometric images as its primary task. Besides, BID technologies also provide means to integrate different kinds of features for better recognition accuracy.

As active researchers, we have been devoted to BID research both in theory and in practice for a few years. A series of novel and effective BID technologies has been developed in the context of supervised and unsupervised statistical learning concepts. The class of new methods includes the following topics: (1) dual eigenfaces and hybrid neural methods for face image feature extraction and recognition; (2) improved LDA algorithms for face and palmprint discrimination; (3) new algorithms of complete LDA and K-L expansion for small sample size and high-dimensional problems like face recognition; (4) complex K-L expansion, complex PCA, and complex LDA or FLD for combined feature extraction and data fusion; (5) two-dimensional PCA (2DPCA or IMPCA) and image projection discriminant analysis (IMLDA or 2DLDA) that are used for supervised and unsupervised learning based on 2D image matrices; and (6) image discrimina-

tion technologies-based palmprint identification. These developed methods can be used for pattern recognition in general, and for biometric image discrimination in particular. Recently, we also developed a set of new kernel-based BID algorithms and found that KFD is equivalent to KPCA plus LDA in theory. This finding makes KFD more intuitive, more transparent and easier to implement. Based on this result and our work on LDA, a complete KFD algorithm is developed. This new method can take advantage of two kinds of discrimination information, and turned out to be more effective for image discrimination.

In this book, we focus our attention on linear projection analysis and develop some new algorithms that are verified to be more effective in biometrics authentication. This book will systematically introduce the relative BID technologies. But, this is not meant to suggest a low-relevance of the book to BID in general. Rather, the issues this book addresses are highly relevant to many fundamental concerns of both researchers and practitioners of BID in biometric applications. The materials in the book are the outgrowth of research the authors have conducted for many years, and present the authors' recent academic achievements made in the field, although, for the sake of completeness, related work of other authors will also be addressed.

The book is organized into three sections. As an introduction, we first describe the basic concepts necessary for a good understanding of BID and answer some questions about BID like why, what and how. Then, Section I focuses on fundamental BID technologies, where two original BID approaches, PCA and LDA, are defined. In addition, some typical biometric applications (such as face, palmprint gait, ear, voice, iris and signature) using these technologies are developed. Section II explores some improved BID technologies, including statistical uncorrelation analysis, some solutions of LDA for the small sample size problem, an improved LDA approach and a novel approach based on both DCT and linear discrimination technique. Other typical BID improvements, including dual eigenspaces method (DEM) and post-processing on LDA-based method, are also given. Section III states some advanced BID technologies. They deal with the complete KFDA, two-dimensional image matrix-based discriminator and two-directional PCA/LDA design, as well as feature fusion using complex discriminator.

There are 13 chapters in this book. Chapter I briefly introduces biometrics image discrimination (BID) technologies. We define and describe types of biometrics and biometric technologies. Some applications of biometrics are given , and we discuss biometric systems and discrimination technologies. We answer the question of what are BID technologies. Then, we outline the history and development of BID technologies, and provide an overview and taxonomy of appearance-based BID technologies. Finally, we highlight each chapter of this book.

In Chapter II, we first describe a basic concept of PCA, which is a useful statistical technique and can be used in some fields such as face patterns and other biometrics. We also introduce PCA definitions and some useful technologies. Then, the nonlinear PCA technologies are given. As a result, we obtain some useful conclusions.

Chapter III deals with issues related to LDA. First, we indicate some basic conceptions of LDA. The definitions and notations related to LDA are discussed. Then, the introduction to non-linear LDA and the chapter summary are given.

Some typical PCA/LDA applications in biometrics are shown in Chapter IV. Based on the introductions to both PCA and LDA mentioned in Chapters II and III, their simple descriptions are given. Then, we discuss seven significant biometrics applica-

tions, including face recognition, palmprint identification, gait verification, ear biometrics, speaker identification, iris recognition and signature verification. At the end of this chapter, we point out a brief but useful summary.

Chapter V indicates a new LDA approach called uncorrelated optimal discrimination vectors (UODV). After introduction, we first give some basic definitions. Then, a set of uncorrelated optimal discrimination vectors (UODV) is proposed, and we introduce an improved UODV approach. Some experiments and analysis are shown, and finally, we give some useful conclusions.

The solutions of LDA for small sample size problems are defined in Chapter VI. We first give an overview on the existing LDA regularization techniques. Then, a unified framework for LDA and a combined LDA algorithm for the small sample size problem are described. We provide the experimental results and some conclusions.

Chapter VII discusses an improved LDA approach — ILDA. After a short review and comparison of major linear discrimination methods, including the Eigenface method, Fisherface method, DLDA and UODV, we first introduce some definitions and notations. Then, the approach description of ILDA is presented. Next, we show some experimental results. Finally, we give some useful conclusions.

Chapter VIII provides a feature extraction approach, which combines the discrete cosine transform (DCT) with LDA. The DCT-based frequency-domain analysis technique is introduced. We describe the presented discriminant DCT approach and analyze its theoretical properties. Then, detailed experimental results and a summary are given.

In Chapter IX, we discuss some other typical BID improvements, including dual eigenspaces method (DEM) and post-processing on the LDA-based method for automated face recognition. After the introduction, we describe DEM. Then, post-processing on the LDA-based method is defined. Finally, we give some brief conclusions.

Chapter X introduces a complete kernel Fisher discriminant analysis that is a useful statistical technique applied to biometric application. We describe the theoretical perspective of KPCA. A new KFD algorithm framework — KPCA plus LDA — is given. We discuss the complete KFD algorithm, and, finally, offer experimental results and the chapter summary.

Chapter XI presents two straightforward image projection techniques — 2D image matrix-based PCA (IMPCA or 2DPCA) and 2D image matrix-based Fisher LDA (IMLDA or 2DLDA). After a brief introduction, we first introduce IMPCA. Then IMLDA technology is given. As a result, we offer some useful conclusions.

Chapter XII introduces a two-directional PCA/LDA approach that is a useful statistical technique applied to biometric authentication. We first describe both bi-directional PCA (BDPCA) method and BDPCA plus LDA methods. Some basic models and definitions related to two-directional PCA/LDA approach are given, and then we discuss two-directional PCA plus LDA. The experimental results and chapter summary are finally provided.

Chapter XIII describes the feature fusion techniques using complex discriminator. After the introduction, we first introduce serial and parallel feature fusion strategies. Then, the complex linear projection analysis methods, complex PCA and complex LDA, are developed. Some feature pre-processing techniques are given, and we analyze and reveal the symmetry property of parallel feature fusion. The proposed methods are applied to biometrics, related experiments are performed and detailed comparison analysis is exhibited. Finally, a summary is given.

In summary, this book is a comprehensive introduction to both theoretical analysis and ogies. It would serve as a textbook or as a useful reference for graduate students and researchers in the fields of computer science, electrical engineering, systems science and information technology. Researchers and practitioners in industry and research-and-development laboratories working security system design, biometrics, computer vision, control, image processing and pattern recognition would also find much interest in this book.

In the preparation of this book, David Zhang organized the contents of the book and was in charge of Chapters I, IV, IX and XII. Xiaoyuan Jing and Jian Yang handle Chapters II, III, V, VII, VIII and Chapters VI, X, XI and XIII, respectively. Finally, David Zhang looked through the whole book and examined all chapters.

Acknowledgments

Our sincere thank goes to professor Zhaoqi Bian of Tsinghua University, Beijing, and professor Jingyu Yang of Najing Polytechnic University, Najing, China, for their advice throughout this research. We would like to thank our team members, Dr. Hui Peng, Wangmeng Zuo, Dr. Guangming Lu, Dr. Xiangqian Wu, Dr. Kuanquan Wang and Dr. Jie Zhou for their hard work and unstinting support. In fact, this book is the common result of their many contributions. We would also like to express our gratitude to our research fellows, Michael Wong, Laura Liu and Dr. Ajay Kumar for their invaluable help and support. Thanks are also due to Martin Kyle, Dr. Zhizhen Liang, Miao Li and Xiaohui Wang for their help in the preparation of this book. The financial support of the CERG fund from the HKSAR Government, the central fund from the Hong Kong Polytechnic University and NFSC funds (No. 60332010 and No. 60402018) in China are, of course, also greatly appreciated. We owe a debt of thanks to Jan Travers and Kristin Roth of Idea Group Inc., for their initiative in publishing this volume.

***David Zhang**, Biometrics Research Centre*
The Hong Kong Polytechnic University, Hong Kong
E-mail: csdzhang@comp.polyu.edu.hk

***Xiaoyuan Jing**, Bio-Computing Research Centre*
ShenZhen Graduate School of Harbin Institute of Technology, China
E-mail address: jingxy_2000@yahoo.com

***Jian Yang**, Biometrics Research Centre*
The Hong Kong Polytechnic University, Hong Kong
E-mail address: csjyang@comp.polyu.edu.hk

Chapter I

An Introduction to Biometrics Image Discrimination (BID)

ABSTRACT

In this chapter, we briefly introduce biometrics image discrimination (BID) technologies. First, we define and describe types of biometrics and biometrics technologies. Then, some applications of biometrics are given. The next section discusses biometrics systems and discrimination technologies, followed by a definition of BID technologies. The history and development of BID technologies is offered, and an overview and taxonomy of appearance-based BID technologies, respectively, is provided. Finally, the last section highlights each chapter of this book.

DEFINITION OF BIOMETRICS TECHNOLOGIES

Biometrics image discrimination (BID) is a field of biometrics, the statistical analysis of biological characteristics. A common interest in biometrics is technologies that automatically recognize or verify individual identities using a measurable physiological or behavioral characteristic (Jain, Bolle, & Pankanti, 1999; Zhang, 2000a, 2000b). Physiological characteristics might include facial features, thermal emissions, features of the eye (e.g., retina and iris), fingerprints, palmprints, hand geometry, skin pores or veins in the wrists or hand. Behavioral characteristics include activities and their artefacts, such as handwritten signatures, keystrokes or typing, voiceprints, gaits and gestures.

Biometrics lays the foundation for an extensive array of highly secure authentication and reliable personal verification (or identification) solutions. The first commercial biometrics system, Identimat, was developed in the 1970s as part of an employee time clock at Shearson Hamill, a Wall Street investment firm (Miller, 1994). It measured the shape of the hand and the lengths of the fingers. At the same time, fingerprint-based automatic checking systems were widely used in law enforcement by the FBI and by United States (U.S.) government departments. Advances in hardware, such as faster processing power and greater memory capacity, made biometrics more viable. Since the 1990s, iris, retina, face, voice, palmprint, signature and DNA technologies have joined the biometrics family (Jain, Bolle, & Pankanti, 1999; Zhang, 2000b).

Rapid progress in electronics and Internet commerce has made more urgent need for secure transaction processing using biometrics technology. After the September 11, 2001 (9/11) terrorist attacks, the interest in biometrics-based security solutions and applications increased dramatically, especially in the need to identify individuals in crowds. Some airlines have implemented iris recognition technology in airplane control rooms to prevent entry by unauthorized persons. In 2004, all Australian international airports will implement passports using face recognition technology for airline crews, and this will eventually become available to all Australian passport holders (Zhang, 2004). As the costs, opportunities and threats of security breaches and transaction fraud increase, so does the need for highly secure identification and personal verification technologies.

The major biometrics technologies involve finger scan, voice scan, facial scan, palm scan, iris scan and signature scan, as well as integrated authentication technologies (Zhang, 2002a).

Finger-Scan Technology

Finger-scan biometrics is based on the distinctive characteristics of a human fingerprint. A fingerprint image is read from a capture device, the features are extracted from the image and a template is created. If appropriate precautions are followed, the result is a very accurate means of authentication. Fingerprint matching techniques can be placed into two categories: minutiae-based and correlation-based. Minutiae-based techniques first find minutiae points and then map their relative placements on the finger. However, there are some difficulties with this approach when the fingerprint image is of a low quality, because accurate extraction of minutiae points is difficult. Nor does this method take into account the global pattern of ridges and furrows. Correlation-based methods are able to overcome the problems of a minutiae-based approach. However, correlation-based techniques require the precise location of a registration point and are affected by image translation and rotation. Fingerprint verification may be a good choice for in-house systems that operate in a controlled environment, where users can be given adequate training. It is not surprising that the workstation access application area seems to be based almost exclusively on fingerprints, due to the relatively low cost, small size and ease of integration of fingerprint authentication devices.

Voice-Scan Technology

Of all the human traits used in biometrics, the one that humans learn to recognize first is the voice. Speech recognition systems can be divided into two categories: text-

dependent and text-independent. In text-dependent systems, the user is expected to use the same text (keyword or sentence) during training and recognition sessions. A text independent system does not use the training text during recognition sessions. Voice biometrics has the most potential for growth, because it does not require new hardware — most personal computers (PCs) nowadays already come with a microphone. However, poor quality and ambient noise can affect verification. In addition, the set-up procedure has often been more complicated than with other biometrics, leading to the perception that voice verification is not user-friendly. Therefore, voice authentication software needs to be improved. However, voice scanning may be integrated with finger-scan technology. Because many people see finger scanning as a higher form of authentication, voice biometrics will most likely be relegated to replacing or enhancing personal identification numbers (PINs), passwords or account names.

Face-Scan Technology

As with finger-scan and voice-scan biometrics, facial-scan technology uses various methods to recognize people. All the methods share certain commonalities, such as emphasizing those sections of the face that are less susceptible to alteration, including the upper outlines of the eye sockets, the areas surrounding the cheekbones and the sides of the mouth. Most technologies are resistant to moderate changes in hairstyle, as they do not utilize areas of the face located near the hairline. All of the primary technologies are designed to be robust enough to conduct one-to-many searches; that is, to locate a single face from a database of thousands, or even hundreds of thousands, of faces. Face authentication analyzes facial characteristics. This requires the use of a digital camera to capture a facial image. This technique has attracted considerable interest, although many people do not completely understand its capabilities. Some vendors have made extravagant claims, which are very difficult, if not impossible, to substantiate in practice. Because facial scanning needs an extra peripheral not customarily included with basic PCs, it is more of a niche market for use in network authentication. However, the casino industry has capitalized on this technology to create a facial database of fraudsters for quick detection by security personnel.

Palm-Scan Technology

Although research on the issues of fingerprint identification and voice recognition have drawn considerable attention over the last 25 years, and recently, issues in face recognition have been studied extensively, there still are some limitations to the existing applications. For example, some people's fingerprints are worn away due to the work they do with their hands, and some people are born with unclear fingerprints. Face-based and voice-based identification systems are less accurate and easier to overcome using a mimic. Efforts geared towards improving the current personal identification methods will continue, and meanwhile, new methods are under investigation. Unlike simple hand geometry that measures hand size and finger length, a palmprint approach is concerned with the inner surface of a hand, and looks in particular at line patterns and surface shape. A palm is covered with the same kind of skin as fingertips, and it is also larger; hence, it is quite natural to think of using a palmprint to recognize a person. Authentication of identity using a palmprint line is a challenging task, because line features (referred to as principle lines), wrinkles and ridges on a palm are not individually descriptive enough

for identification. This problem can be tackled by combining various features, such as texture, to attain a more robust verification. As a new attempt, and a necessary complement to existing biometrics techniques, palmprint authentication is considered part of the biometrics family.

Iris-Scan Technology

Iris authentication technology leverages the unique features of the human iris to provide an unmatched identification technology. The algorithms used in iris recognition are so accurate that the entire planet could be enrolled in an iris database with only a small chance of false acceptance or false rejection. Iris identification technology is a tremendously accurate biometrics. An iris-based biometrics involves analyzing features found in the colored ring of tissue that surrounds the pupil. The iris scan, which is undoubtedly the least intrusive of the eye-related biometrics, uses a fairly conventional camera and requires no close contact between the user and the iris reader. In addition, it has the potential for higher-than-average template-matching performance. Iris biometrics work with eyeglasses in place, and it is one of the few devices that can work well in identification mode. Ease of use and system integration has not traditionally been strong points with iris scanning devices, but people can expect improvements in these areas as new products emerge.

Signature-Scan Technology

Signature verification analyzes the way a user signs his or her name. Signing features, such as speed, velocity and pressure, are as important as the finished signature's static shape. Signature verification enjoys a synergy with existing processes that other biometrics do not. People are familiar with signatures as a means of (transaction-related) identity verification, and most people would think there was nothing unusual in extending this process by including biometrics. Signature verification devices are reasonably accurate, and are obviously acceptable in situations where a signature is already an accepted identifier. Surprisingly, compared with the other biometrics methodologies, relatively few significant signature applications have emerged as yet.

Multiple Authentication Technologies

From an application standpoint, widespread deployment of a user authentication solution requires support for an enterprise's heterogeneous environment. Often, this requires a multi-faceted approach to security, deploying security solutions in combination. An authentication solution should seamlessly extend the organization's existing security technologies. We are now interested in understanding both how to combine multiple biometrics technologies and what possible improvements these combinations can produce. One of the main problems for researchers into multiple biometrics is the scarcity of true multi-modal databases for testing their algorithms. Perhaps the most important resource available today is the extended M2VTS (multi-modal verification for teleservices and security applications) database, which is associated with the specific Lausanne protocol for measuring the performance of verification tasks. This database contains audio-visual material from 295 subjects.

APPLICATIONS OF BIOMETRICS

Many applications of biometrics are currently being used or considered worldwide. Most of these applications are still in the testing stage and are optional for end users. The accuracy and effectiveness of these systems need to be verified in the real-time operation environment. As an example, in this section, we will discuss various applications of personal authentication based on biometrics.

Any situation that allows an interaction between man and machine is capable of incorporating biometrics. Such situations may fall into a range of application areas. Biometrics is currently being used in areas such as computer desktops, networks, banking, immigration, law enforcement, telecommunication networks and monitoring the time and attendance of staff. Governments across the globe are tremendously involved in using and developing biometrics. National identity schemes, voting registration and benefit entitlement programs involve the management of millions of people and are rapidly incorporating biometrics solutions. Fraud is an ever-increasing problem, and security is becoming a necessity in many walks of life. Biometrics applications of personal authentication can be categorized simply, as follows (Zhang, 2000b):

Law Enforcement

The law enforcement community is perhaps the largest user of biometrics. Police forces throughout the world use AFIS (automated fingerprint identification systems) technology to process suspects, match finger images and process accused individuals. A number of biometrics vendors are earning significant revenues in this area, primarily using AFIS and palm-based technologies.

Banking

Banks have been evaluating a range of biometrics technologies for many years. Automated teller machines (ATMs) and transactions at the point of sale are particularly vulnerable to fraud and can be secured by biometrics. Other emerging markets, such as telephone banking and Internet banking, must also be totally secure for bank customers and bankers alike. A variety of biometrics technologies are now striving to prove themselves throughout this range of diverse market opportunities.

Computer Systems (or Logical Access Control)

Biometrics technologies are proving to be more than capable of securing computer networks. This market area has phenomenal potential, especially if the biometrics industry can migrate to large-scale Internet applications. As banking data, business intelligence, credit card numbers, medical information and other personal data become the target of attack, the opportunities for biometrics vendors are rapidly escalating.

Physical Access

Schools, nuclear power stations, military facilities, theme parks, hospitals, offices and supermarkets across the globe employ biometrics to minimize security threats. As security becomes more and more important for parents, employers, governments and other groups, biometrics will be seen as a more acceptable and, therefore, essential tool.

The potential applications are infinite. Cars and houses, for example, the sanctuary of the ordinary citizen, are under constant threat of theft. Biometrics — if appropriately priced and marketed — could offer the perfect security solution.

Benefit Systems

Benefit systems like welfare especially need biometrics to struggle with fraud. Biometrics is well placed to capitalize on this phenomenal market opportunity, and vendors are building on the strong relationship currently enjoyed with the benefits community.

Immigration

Terrorism, drug running, illegal immigration and an increasing throughput of legitimate travelers are putting a strain on immigration authorities throughout the world. It is essential that these authorities can quickly and automatically process law-abiding travelers and identify law breakers. Biometrics are being employed in a number of diverse applications to make this possible. The U.S. Immigration and Naturalization Service is a major user and evaluator of a number of biometrics. Systems are currently in place throughout the U.S. to automate the flow of legitimate travelers and deter illegal immigrants. Elsewhere, biometrics is capturing the imagination of countries such as Australia, Bermuda, Germany, Malaysia and Taiwan.

National Identity

Biometrics are beginning to assist governments as they record population growth, identify citizens and prevent fraud from occurring during local and national elections. Often, this involves storing a biometrics template on a card that in turn acts as a national identity document. Finger scanning is particularly strong in this area and schemes are already under way in Jamaica, Lebanon, the Philippines and South Africa.

Telephone Systems

Global communication has truly opened up over the past decade. While telephone companies are under attack from fraud, once again, biometrics is being called upon to defend against this onslaught. Speaker ID is obviously well suited to the telephone environment and is making in-roads into these markets.

Time, Attendance and Monitoring

Recording and monitoring the movement of employees as they arrive at work, have breaks and leave for the day were traditionally performed by time-card machines. Replacing the manual process with biometrics prevents any abuses of the system and can be incorporated with time management software to produce management accounting and personnel reports.

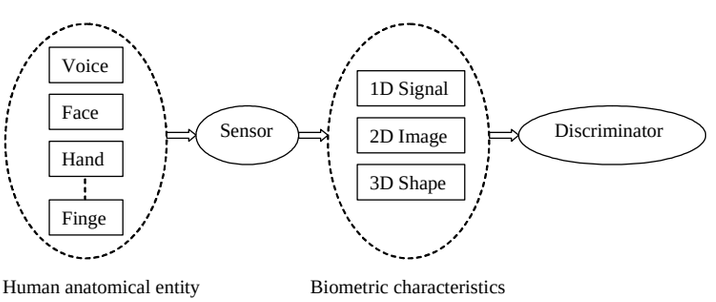
BIOMETRICS SYSTEMS AND DISCRIMINATION TECHNOLOGIES

A biometrics system is essentially a pattern recognition system that makes a personal identification by determining the authenticity of a specific physiological or behavioral characteristic possessed by the person (Pankanti, Bool, & Jain, 2000). Normally, personal characteristics such as voice waveforms, face images, fingerprints, or 3-D face or hand geometric shapes are obtained through a sensor and fed into a discriminator (pattern recognition engine) to return a result of success or failure. Figure 1.1 shows the architecture of a typical biometrics system.

In general, the first stage of biometrics systems is data acquisition. In this stage, the biometrics data (characteristics) of a person is obtained using data acquisition equipment. The biometrics data generally exists in the following three forms: 1D waveform (e.g., voice, signature data), 2D images (e.g., face images, fingerprints, palmprints or image sequences — i.e., video) and 3D geometric data (e.g., face or hand geometric shapes). After data acquisition and the corresponding data pre-processing, biometrics data are fed into a discriminator for feature extraction and matching. Finally, a matching score is obtained by matching an identification template against a master template. If the score is lower than a given threshold, the person is authenticated.

2D biometrics images are a very important form of biometrics data and are associated with many different biometrics technologies and systems, such as face recognition, fingerprint or palmprint verification, iris recognition, ear or tooth recognition, and gait recognition. With the development of biometrics and its applications, many classical discrimination technologies have been borrowed and applied to deal with biometrics images. Among them, principal component analysis (PCA, or K-L transform) and Fisher linear discriminant analysis (LDA) have been very successful, in particular for face image recognition. These methods have themselves been greatly improved with respect to specific biometrics image analyses and applications. Recently, non-linear projection analysis technology represented by kernel principal component analysis (KPCA) and kernel Fisher discriminant (KFD) has also shown great potential for dealing with biometrics problems.

Figure 1.1. Architecture of biometric systems



In summary, discrimination technologies play an important role in the implementation of biometrics systems. They provide methodologies for automated personal identification or verification. In turn, the applications in biometrics also facilitate the development of discrimination methodologies and technologies, making discrimination algorithms more suitable for image feature extraction and recognition.

Image discrimination, then, is an elementary problem area within automated biometrics. In this book, we will introduce readers to this area under the more specific heading of BID, systematically introducing relevant BID technologies. This book addresses fundamental concerns of relevance to both researchers and practitioners using BID in biometrics applications. The materials in the book are the product of many years of research on the part of the authors, and present the authors' recent academic achievements made in the field. For the sake of completeness, readers may rest assured that wherever necessary this book also addresses the relevant work of other authors.

WHAT ARE BID TECHNOLOGIES?

BID technologies can be briefly defined as automated methods of feature extraction and recognition based on given biometrics images. It should be stressed that BID technologies are not the simple application of classical discrimination techniques to biometrics, but are in fact improved or reformed discrimination techniques that are more suitable for biometrics applications; for example, by having a more powerful recognition performance or by being computationally more efficient for feature extraction or classification. In other words, the BID technologies are designed to be applied to BID problems, which are characteristically high dimensional, large scale, and offer only a small sample size. The following explains these characteristics more fully.

High Dimensionality

Biometrics images are high dimensional. For example, images with a resolution of 100×100 will produce a 10,000-dimensional image vector space. The central difficulty of high dimensionality is that it makes direct classification (e.g., the so-called correlation method that uses a nearest neighbor classifier) in image space almost impossible, first because the similarity (distance) calculation is very computationally expensive, second because it demands large amounts of storage. High dimensionality makes it necessary to use a dimension reduction technique prior to recognition.

Large Scale

Real-world biometrics applications are often large scale. Clear examples of this would include welfare disbursement, national ID cards, border control, voter ID cards, driver's licenses, criminal investigation, corpse identification, parenthood determination and the identification of missing children. Given an input biometrics sample, a large-scale BID identification system determines whether the pattern is associated with any of a large number (e.g., millions) of enrolled identities. These large-scale BID applications require high quality and very generalizable BID technologies.

Small Sample Size

Unlike, for example, optical character recognition (OCR) problems, the training samples per class that are available in real-world BID problems are always very limited. Indeed, there may be only one sample available for each individual. Combined with high dimensionality, small sample size creates the so-called small-sample-size (or under-sampled) problems. In these problems, the within-class scatter matrix is always singular, because the training sample size is generally less than the space dimension. As a result, the classical LDA algorithm becomes infeasible in image vector space.

In BID problems, representation of the biometrics images is centrally important. The objectives of image representation are twofold. One is for a better identification (or verification), and the other is for an efficient similarity calculation. On the one hand, the sample points in image space generally lead to unsatisfactory clusters, especially under the variations of illumination, time or other conditions. By virtue of feature extraction techniques, it can be expected to obtain a set of features smaller in number but more discriminative. These features may be more insensitive to intra-class variations, such as those derived from varying lighting conditions. On the other hand, feature extraction can significantly reduce the number of features. This greatly benefits subsequent classification, as it reduces storage requirements and improves classification efficiency.

Different biometrics images, however, may use different representation methods. For example, from face images, we can get geometric features such as eyes, nose and mouth. From palmprint images we can get principal lines and wrinkles features. From fingerprint images we can get minutiae points, ridges and singular points features. These feature generation and representation methods depend on the specific category of biometrics images.

The primary task of BID technologies is to enable different ways of representing biometrics images of integrating different kinds of features for better recognition accuracy. In this book, our focus is on exploring the common representation methods, the methods applicable to any biometrics image. Generally, there are two cases of applications of BID technologies for image representation: One is original image based, and the other is feature based. In the first case, BID technologies are used to derive the discriminative features directly from the original biometrics images. In the second, BID technologies are employed for the second-feature extraction based on the features derived from other feature generation approaches (e.g., Fourier transform, wavelet transform).

HISTORY AND DEVELOPMENT OF BID TECHNOLOGIES

Recent decades have seen the development of a number of BID technologies. Appearance-based BID techniques, represented by linear projection analysis, have been particularly successful. Linear projection analysis, including PCA, and LDA are classical and popular technologies that can extract the holistic features that have strong biometrics image discriminability. PCA was first used by Sirovich and Kirby (1987; Kirby & Sirovich, 1990) to represent images of human faces. Subsequently, Turk and Pentland applied PCA to face recognition and presented the well-known eigenfaces method (Turk

& Pentland, 1991a, 1991b). Since then, PCA has been widely investigated and has become one of the most successful approaches to face recognition (Pentland, Moghaddam, & Starner, 1994; Pentland, 2000; Zhao & Yang, 1999; Moghaddam, 2002; Zhang, Peng, Zhou, & Pal, 2002; Kim, Kim, Bang, & Lee, 2004). Independent component analysis (ICA) is currently popular in the field of signal processing; it has been developed recently as an effective feature extraction technique and has been applied to image discrimination. Bartlett, Yuen, Liu and Draper proposed using ICA for face representation and found that it was better than PCA when cosine was used as the similarity measure (Bartless, Movellan, & Sejnowski, 2002; Yuen & Lai, 2002; Liu & Wechsler, 2003; Draper, Baek, Bartlett, & Beveridge, 2003). Petridis and Perantonis revealed the relationship between ICA and LDA from the viewpoint of mutual information (Petridis & Perantonis, 2004).

To the best of our knowledge, Fisher LDA was first applied to image classification by Tian (Tian, Barbero, Gu, & Lee, 1986). Subsequently, Liu developed the LDA algorithm for small samples and applied it to face biometrics (Liu, Cheng, Yang, & Liu, 1992). Four years later, the most famous method, fisherface, appeared. This was based on a two-phase framework: PCA plus LDA (Swets & Weng, 1996; Belhumeur, Hespanha, & Kriengman, 1997). The theoretical justification for this framework has been laid recently (Yang & Yang, 2003). Many improved LDA algorithms have been developed (Jin, Yang, Hu, & Lou, 2001; Chen, Liao, Lin, Kao, & Yu, 2000; Yu & Yang, 2001; Lu, Plataniotis, & Venetsanopoulos, 2003; Liu & Wechsler, 2001, 2000; Zhao, Krishnaswamy, Chellappa, Swets, & Weng, 1998; Loog, Duin, & Haeb-Umbach, 2001; Duin & Loog, 2004; Ye, Janardan, Park, & Park, 2004; Howland & Park, 2004). Jin proposed an uncorrelated linear discriminant for face recognition (Jin, Yang, Hu, & Lou, 2001); Yu suggested a direct LDA algorithm for high-dimensional image data (Yu & Yang, 2001). Some researchers put forward enhanced LDA models to improve the generalization power of LDA in face-recognition applications (Lu, Plataniotis, & Venetsanopoulos, 2003; Liu & Wechsler, 2001, 2000; Zhao, Krishnaswamy, Chellappa, Swets, & Weng, 1998). Some investigators gave alternative LDA versions based on generalized Fisher criterions (Loog, Duin, & Haeb-Umbach, 2001; Duin & Loog, 2004; Ye, Janardan, Park, & Park, 2004). Howland suggested a generalized LDA algorithm based on generalized singular value decomposition (Howland & Park, 2004). Liu proposed a 2D image matrix-based algebraic feature extraction method for image recognition (Liu, Cheng, Yang, et al., 1993). As a new development of the 2D image matrix-based straightforward discrimination technique, a 2D PCA and uncorrelated image projection analysis were suggested for face representation and recognition (Yang, Zhang, Frangi, & Yang, 2004; Yang, Yang, Frangi, & Zhang, 2003). Recently, Ordowski and Meyer developed a geometric LDA for pattern recognition from a geometric point of view (Ordowski & Meyer, 2004). Hubert and Driessen suggested a robust discriminant analysis for dealing with data with outliers (Hubert & Driesen, n.d.). Others improve PCA or LDA from alternative viewpoints (Poston & Marchette, 1998; Du & Chang, 2001; Koren & Carmel, 2004).

Besides linear projection analysis technologies, non-linear projection analysis represented by both KPCA and KFD also has aroused considerable interest in the fields of pattern recognition and machine learning, and over the last few years have shown great potential in biometrics applications. KPCA was originally developed by Schölkopf (Schölkopf, Smola, & Müller, 1998), while KFD was first proposed by Mika (Mika, Ratsch, Weston, Schölkopf, & Müller, 1999; Mika, Ratsch, Schölkopf, Smola, Weston, & Müller, 1999). Subsequent research saw the development of a series of KFD algorithms (Baudat

& Anouar, 2000; Roth & Steinhage, 2000; Mika, Ratsch, & Müller, 2001; Mika, Smola, & Schölkopf, 2001; Mika, Ratsch, Weston, Schölkopf, Smola, & Müler, 2003; Yang, 2002; Lu, Plataniotis, & Venetsanopoulos, 2003; Xu, Zhang, & Li, 2001; Billings & Lee, 2002; Gestel, Suykens, Lanckriet, Lambrechts, De Moor, & Vanderwalle, 2002; Cawley & Talbot, 2003; Lawrence & Schölkopf, 2001). The KFD algorithms developed by Mika, Billings and Cawley are formulated for two classes, while those of Baudat, Roth and Yang are formulated for multiple classes. Because of its ability to extract the most discriminatory non-linear features, KFD has been found very effective in many real-world biometrics applications. Yang, Liu, Yang, and Xu used KPCA (KFD) for face feature extraction and recognition and showed that KPCA (KFD) outperforms the classical PCA (LDA) (Yang, 2002; Liu, 2004; Yang, Jin, Yang, Zhang, & Frangi, 2004; Yang, Frangi, & Yang, 2004; Xu, Yang, & Yang, 2004; Yang, Zhang, Yang, Zhong, & Frangi, 2005).

Over the last several years, we have been devoted to BID research both in theory and in practice. A series of novel and effective BID technologies has been developed in the context of supervised and unsupervised statistical learning concepts. The class of new methods includes:

- Dual eigenfaces and hybrid neural methods for face image feature extraction and recognition (Zhang, Peng, Zhou, & Pal, 2002)
- Improved linear discriminant analysis algorithms for face and palmprint discrimination (Jing, Zhang, & Jin, 2003; Jing & Zhang, 2003a; Jing, Zhang, & Yao, 2003; Jing, Zhang, & Tang, 2004; Yang, Yang, & Zhang, 2002; Jing & Zhang, 2004; Jing, Tang, & Zhang, 2005; Jing, Zhang, & Yang, 2003; Jing & Zhang, 2003b)
- New algorithms of complete LDA and K-L expansion for small-sample size and high-dimensional problems like face recognition (Yang & Yang, 2001, 2003; Yang, Zhang, & Yang, 2003; Yang, Ye, Yang, & Zhang, 2004)
- Complex K-L expansion, complex PCA, and complex LDA or FLD for combined feature extraction and data fusion (Yang & Yang, 2002; Yang, Yang, & Frangi, 2003; Yang, Yang, Zhang, & Lu, 2003)
- Two-dimensional PCA (2DPCA or IMPCA) and image projection discriminant analysis (IMLDA or 2DLDA) that are used for supervised and unsupervised learning based on 2D image matrices (Yang & Yang, 2002; Yang, Zhang, Frangi, & Yang, 2004; Yang, Yang, Frangi, & Zhang, 2003)
- Image discrimination technologies-based Palmprint identification (Lu, Zhang, & Whang, 2003; Wu, Zhang, & Wang, 2003).

These methods developed can be used for pattern recognition in general, and for biometrics image discrimination in particular.

Recently, we developed a set of new kernel-based BID algorithms and found that KFD is in theory equivalent to KPCA plus LDA (Yang, Jin, Yang, Zhang, & Frangi, 2001, 2005). This finding makes KFD more intuitive, more transparent and easier to implement. Based on this result and our work on LDA, a complete KFD algorithm was developed (Yang, Zhang, Yang, Zhong, & Frangi, 2005). This new method can take advantage of two kinds of discrimination information and has turned out to be more effective for image discrimination.

OVERVIEW: APPEARANCE-BASED BID TECHNOLOGIES

In biometrics applications, appearance-based methods play a dominant role in image representation. These methods have in common the property that they allow efficient characterization of a low-dimensional subspace within the overall space of raw image measurements. Once a low-dimensional representation of the target class (face, hand, etc.) has been obtained, then standard statistical methods can be used to learn the range of appearance that the target exhibits in the new, low-dimensional coordinate system. Because of the lower dimensionality, relatively few examples are required to obtain a useful estimate of discriminant functions (or discriminant directions).

BID technologies are either machine-learning technologies or image-transformation technologies. The machine-learning technologies can in turn be divided into supervised and unsupervised methods. Supervised methods employ the class label information of training samples in the learning process, while the unsupervised methods do not. The image-transformation technologies can be categorized into two groups: linear and non-linear methods. Linear methods use linear transforms (projections) for image dimensional reduction, while the non-linear methods use non-linear transforms for the same purpose.

BID technologies can also be categorized according to whether their input data is 1D or 2D. In recognition problems, 2D input data (matrix) can be processed in two ways. The first way — the 1D- (or vector-) based method — is to transform the data into 1D vectors by stacking the columns of the matrix, as we usually do, and then to use classical vector-based methods for further discrimination. The second way — the 2D- (or matrix-) based method — skips the matrix-to-vector conversion process and performs discrimination analysis directly on the 2D matrix.

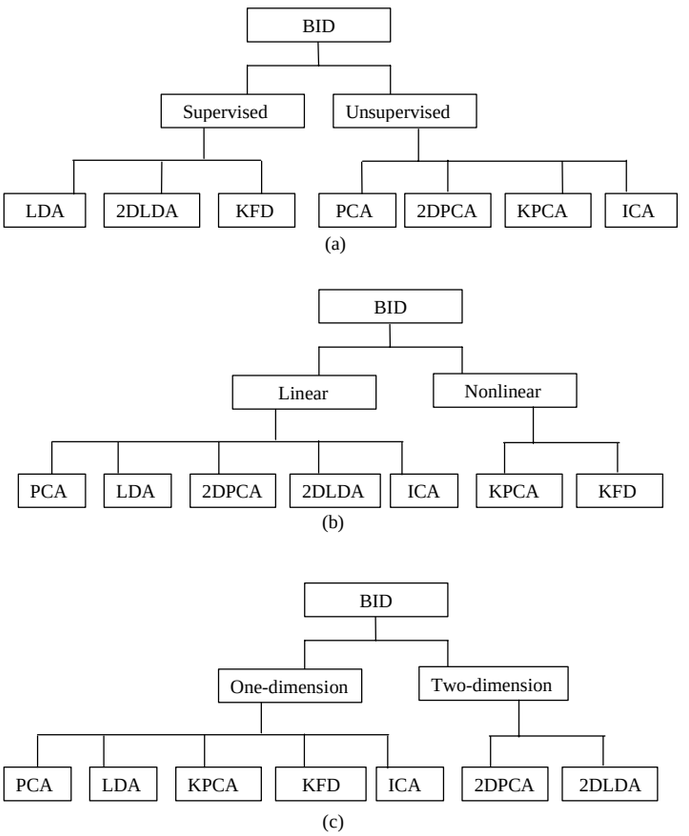
These three taxonomies are outlined below and illustrated in Figure 1.2.

- **Supervised/unsupervised.** The supervised BID technologies include Fisher LDA, 2DLDA and KFD. The unsupervised BID technologies include: PCA, 2DPCA, KPCA and ICA.
- **Linear/non-linear.** The linear BID technologies include: PCA, LDA, 2DPCA, 2DLDA and ICA. The non-linear BID technologies include: KPCA and KFD. Note that ICA is regarded as a linear method because its determined image transform is linear, although it needs to solve a non-linear optimization problem in the learning process.
- **1D/2D.** 2D-based methods include 2DPCA and 2DLDA. The others, such as PCA, LDA, ICA, KPCA and KFD, are 1D-based.

BOOK PERSPECTIVE

This book is organized into three main sections. Chapter I first described the basic concepts necessary for a good understanding of BID and answered some questions about BID, like why, what and how. Section I focuses on fundamental BID technologies.

Figure 1.2. Illustration of three BID technology taxonomies: (a) supervised/ unsupervised; (b) linear/non-linear; and (c) one-dimensional/two-dimensional

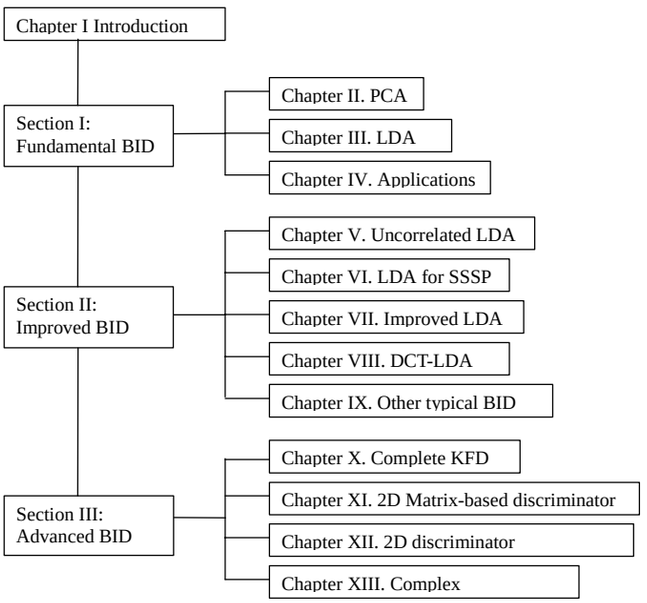


Chapters II and III, respectively, define two original BID approaches, PCA and LDA. Chapter IV provides some typical biometrics applications that use these technologies.

Section II explores some improved BID technologies. Chapter V describes statistical uncorrelated discriminant analysis. In Chapter VI, we develop some solutions of LDA for small sample-size problems. As we know, when LDA is used for solving small sample-size problems like face identification, the difficulty we always encounter is that the within-class scatter matrix is singular. In this chapter, we try to address this problem in theory and build a general framework for LDA in singular cases. Chapter VII defines an improved LDA approach. Chapters VIII and IX, respectively, introduce a DCT-LDA and dual eigenspaces method.

Section III states some advanced BID technologies. Chapter X indicates the complete KFD. In this chapter, a new complete kernel Fisher discriminant analysis (CKFD) algorithm is developed. CKFD is based on a two-phase framework; that is, KPCA

Figure 1.3. Overview of the book



plus LDA, which is more transparent and simpler than the previous ones. CKFD can make full use of two kinds of discriminant information and be more powerful for discrimination. Two-dimensional image matrix-based discriminator and two-directional PCA/LDA architectures are discussed in Chapters XI and XII, respectively. Chapter XI develops two straightforward image projection analysis techniques, termed 2D image matrix-based PCA (IMPCA or 2DPCA) and 2D image matrix-based Fisher LDA (IMLDA or 2DLDA), which can learn the projector directly based on the 2D input data. Chapter XII goes further on the techniques suggested in Chapter XI and realizes a double-directional (horizontal and vertical) compression on the original 2D data. In addition, Chapter XIII summarizes some complex discrimination techniques that can be used for feature fusion. The complex vector is utilized to represent the parallel combined features, and the linear projection analysis methods such as PCA and LDA are generalized for feature extraction in the complex feature space. The architecture of book is illustrated in Figure 1.3.

REFERENCES

Bartlett, M. S., Movellan, J. R., & Sejnowski, T. J. (2002). Face recognition by independent component analysis. *IEEE Trans. Neural Networks*, *13*(6), 1450-1464.

Baudat, G., & Anouar, F. (2000). Generalized discriminant analysis using a kernel approach. *Neural Computation*, *12*(10), 2385-2404.

Belhumeur, P. N., Hespanha, J. P., & Kriengman, D. J. (1997). Eigenfaces vs. fisherfaces: Recognition using class specific linear projection. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, *19*(7), 711-720.

Billings, S. A., & Lee, K. L. (2002). Nonlinear Fisher discriminant analysis using a minimum squared error cost function and the orthogonal least squares algorithm. *Neural Networks*, *15*(2), 263-270.

Cawley, G. C., & Talbot, N. L. C. (2003). Efficient leave-one-out cross-validation of kernel Fisher discriminant classifiers. *Pattern Recognition*, 36(11), 2585-2592.

Chen, L. F., Liao, H. Y., Lin, J. C., Kao, M. D., & Yu, G. J. (2000). A new LDA-based face recognition system which can solve the small sample size problem. *Pattern Recognition*, *33*(10), 1713-1726.

Draper, B. A., Baek, K., Bartlett, B. S., & Beveridge, J. R. (2003). Recognizing faces with PCA and ICA. *Computer vision and image understanding: Special issue on face recognition*, *91*(1/2), 115-137.

Du, Q., & Chang, C. I. (2001). A linear constrained distance-based discriminant analysis for hyperspectral image classification. *Pattern Recognition, 34*(2), 361-373.

Duin, R. P. W., & Loog, M. (2004). Linear dimensionality reduction via a heteroscedastic extension of LDA: the Chernoff criterion. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, *26*(6), 732-739.

Gestel, T. V., Suykens, J. A. K., Lanckriet, G., Lambrechts, A., De Moor, B., & Vanderwalle, J. (2002). Bayesian framework for least squares support vector machine classifiers, gaussian processs and kernel Fisher discriminant analysis. *Neural Computation*, *15*(5), 1115-1148.

Howland, P., & Park, H. (2004). Generalizing discriminant analysis using the generalized singular value decomposition. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, *26*(8), 995-1006.

Hubert, M., & Driessen, K. V. (2002). Fast and robust discriminant analysis. *Computational Statistics and Data Analysis*, *45,* 301-320.

Jain, A., Bolle, R., & Pankanti, S. (1999). *Biometrics: Personal identification in networked society*. Boston: Kluwer Academic Publishers.

Jin, Z., Yang, J. Y., Hu, Z. S., & Lou, Z. (2001). Face recognition based on uncorrelated discriminant transformation. *Pattern Recognition*, *34*(7), 1405-1416.

Jing, X., Tang, Y., & Zhang, D. (2005). A Fourier-LDA approach for image recognition. *Pattern Recognition, 38*(3), 453-457.

Jing, X., & Zhang, D. (2003a). Face recognition based on linear classifiers combination. *Neurocomputing, 50*, 485-488.

Jing, X., & Zhang, D. (2003b). Improvements on the uncorrelated optimal discriminant vectors. *Pattern Recognition, 36*(8), 1921-1923.

Jing, X., & Zhang, D. (2004). A face to palmprint recognition approach based on discriminant DCT featrue extraction. *IEEE Transactions on Systems, Man and Cybernetics, Part B, 34*(6), 2405-2415.

Jing, X., Zhang, D., & Jin, Z. (2003). UODV: Improved algorithm and generalized theory. *Pattern Recognition, 36*(11), 2593-2602.

Jing, X., Zhang, D., & Tang, Y. (2004). An improved LDA approach. *IEEE Transactions on SMC-B, 34*(5), 1942-1951.

Jing, X., Zhang, D., & Yang, J-Y. (2003). Face recognition based on a group decision-making combination approach. *Pattern Recognition, 36*(7), 1675-1678.

Jing, X., Zhang, D., & Yao, Y. (2003). Improvements on linear discrimination technique with application to face recognition. *Pattern Recognition Letters, 24*(15), 2695-2701.

Kim, H. C., Kim, D., Bang, S. Y., & Lee, S. Y. (2004). Face recognition using the second-order mixture-of-eigenfaces method. *Pattern Recognition*, *37*(2), 337-349.

Kirby, M., & Sirovich, L. (1990). Application of the KL procedure for the characterization of human faces. *IEEE Transactions on Pattern Analysis and Machine Intelligence*,*12*(1), 103-108.

Koren, Y., & Carmel, L. (2004). Robust linear dimensionality reduction. *IEEE Transactions on Visualization and Computer Graphics*, *10*(4), 459-470.

Lawrence, N. D., & Schölkopf, B. (2001). Estimating a kernel Fisher discriminant in the presence of label noise. *Proceedings of the 18th International Conference on Machine Learning* (pp. 306-313).

Liu, C. (2004). Gabor-based kernel PCA with fractional power polynomial models for face recogn*IEEE Transactions on Pattern Analysis and Machine Intelligence*, *26*(5), 572-581.

Liu, C., & Wechsler, H. (2000). Robust coding schemes for indexing and retrieval from large face databases. *IEEE Transactions on Image Processing*, *9*(1), 132-137.

Liu, C., & Wechsler, H. (2001). A shape- and texture-based enhanced Fisher classifier for face recognition. *IEEE Transactions on Image Processing*, *10*(4), 598-608.

Liu, C., & Wechsler, H. (2003). Independent component analysis of Gabor features for face recognition. *IEEE Transactions Neural Networks*, *14*(4), 919-928.

Liu, K., Cheng, Y-Q., & Yang, J-Y. (1993). Algebraic feature extraction for image recognition based on an optimal discriminant criterion. *Pattern Recognition*, *26*(6), 903-911.

Liu, K., Cheng, Y-Q., Yang, J-Y., & Liu, X. (1992). An efficient algorithm for Foley-Sammon optimal set of discriminant vectors by algebraic method. *International Journal of Pattern Recognition and Artificial Intelligence, 6*(5), 817-829.

Loog, M., Duin, R. P. W., & Haeb-Umbach, R. (2001). Multiclass linear dimension reduction by weighted pairwise Fisher criteria. *IEEE Transactions on Pattern Analysis and Machine Intelligence, 23*(7), 762-766.

Lu, G., Zhang, D., & Wang, K. (2003). Palmprint recognition using eigenpalms features. *Pattern Recognition Letters, 24*, 9-10, 1463-1467.

Lu, J., Plataniotis, K. N., & Venetsanopoulos, A. N. (2003). Face recognition using kernel direct discriminant analysis algorithms. *IEEE Transactions on Neural Networks*, *14*(1), 117-126.

Lu, J., Plataniotis, K. N., & Venetsanopoulos, A. N. (2003). Face recognition using LDA-based algorithms. *IEEE Trans. Neural Networks*, *14*(1), 195-200.

Mika, S., Rätsch, G., & Müller, K. R. (2001). A mathematical programming approach to the kernel Fisher algorithm. In T. K. Leen, T. G. Dietterich, & V. Tresp (Eds.), *Advances in neural information processing systems 13* (pp. 591-597). Cambridge: MIT Press.

Mika, S., Rätsch, G., Schölkopf, B., Smola, A., Weston, J., & Müller, K. R. (1999). Invariant feature extraction and classification in kernel spaces. In *Advances in neural information processing systems 12*. Cambridge, MA: MIT Press.

Mika, S., Rätsch, G., Weston, J., Schölkopf, B., & Müller, K. R. (1999). Fisher discriminant analysis with kernels. *IEEE International Workshop on Neural Networks for Signal Processing IX Madison* (pp. 41-48).

Mika, S., Rätsch, G., Weston, J., Schölkopf, B., Smola, A., & Müller, K. R. (2003). Constructing descriptive and discriminative non-linear features: Rayleigh coefficients in kernel feature spaces. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, *25*(5), 623-628.

Mika, S., Smola, A. J., & Schölkopf, B. (2001). An improved training algorithm for kernel Fisher discriminants. In T. Jaakkola & T. Richardson (Eds.), *Proceedings of AISTATS 2001,* 98-104.

Miller, B. (1994). Vital signs of identity. *IEEE Spectrum*, *31*(2), 22-30.

Moghaddam, B. (2002). Principal manifolds and probabilistic subspaces for visual recognition. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, *24*(6), 780-788.

Ordowski, M., & Meyer, G. (2004). Geometric linear discriminant analysis for pattern recognition. *Pattern Recognition*, *37*, 421-428.

Pankanti, S., Bolle, R., & Jain, A. (2000). Biometrics: The future of identification. *IEEE Computer*, *33*(2), 46-49.

Pentland, A. (2000). Looking at people: Sensing for ubiquitous and wearable computing. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, *22*(1), 107-119.

Pentland, A., Moghaddam, B., & Starner, T. (1994). View-based and mododular eigenspaces for face recognition. *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition* (pp. 84-91).

Petridis, S., & Perantonis, S. T. (2004). On the relation between discriminant analysis and mutual information for supervised linear feature extraction. *Pattern Recognition*, *37*(5), 857-874.

Poston, W. L., &. Marchette, D. J. (1998). Recursive dimensionality reduction using Fisher's linear discriminant. *Pattern Recognition, 31*, 881-888.

Roth, V., & Steinhage, V. (2000). Nonlinear discriminant analysis using kernel functions. In S. A. Solla, T. K. Leen, & K.-R. Mueller (Eds.), *Advances in neural information processing systems 12* (pp. 568-574). Cambridge, MA: MIT Press.

Schölkopf, B., Smola, A., & Müller, K. R. (1998). Nonlinear component analysis as a kernel eigenvalue problem. *Neural Computation*, *10*(5), 1299-1319.

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

Swets, D. L., & Weng, J. (1996). Using discriminant eigenfeatures for image retrieval. *IEEE Transactions on Pattern Analysis and Machine Intelligence*,*18*(8), 831-836.

Tian, Q., Barbero, M., Gu, Z. H., & Lee, S. H. (1986). Image classification by the Foley-Sammon transform. *Optical Engineering, 25*(7), 834-839.

Turk, M., & Pentland, A. (1991a). Eigenfaces for recognition. *Journal of Cognitive Neuroscience*, *3*(1), 71-86.

Turk, M., & Pentland, A. (1991b). Face recognition using eigenfaces. *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition* (pp. 586-591).

Wu, X., Zhang, D., & Wang, K. (n.d.) Fisherpalms based palmprint recognition. *Pattern Recognition Letters, 24*(15), 2829-2838.

Xu, J., Zhang, X, & Li, Y. (2001). Kernel MSE algorithm: A unified framework for KFD, LS-SVM, and KRR. In *Proceedings of the International Joint Conference on Neural Networks*, 1486-1491.

Xu, Y., Yang, J-Y, & Yang, J. (2004). A reformative kernel Fisher discriminant analysis. *Pattern Recognition*, *37*(6), 1299-1302.

Yang, J., Frangi, A.F., & Yang, J-Y. (2004). A new Kernel Fisher discriminant algorithm with application to face recognition. *Neurocomputing*, *56*, 415-421.

Yang, J., Jin, Z., Yang, J-Y., Zhang, D., & Frangi, A. F. (2004). Essence of kernel Fisher discriminant: KPCA plus LDA. *Pattern Recognition*, *37*(10), 2097-2100.

Yang, J., & Yang, J-Y. (2001). Optimal FLD algorithm for facial feature extraction. *SPIE Proceedings of the Intelligent Robots and Computer Vision XX: Algorithms technique and Active Vision*, 4572, 438-444.

Yang, J., & Yang, J-Y. (2002a). From image vector to matrix: A straightforward image projection technique — IMPCA vs. PCA. *Pattern Recognition*, *35*(9), 1997-1999.

Yang, J., & Yang, J-Y. (2002b). Generalized K-L transform based combined feature extraction. *Pattern Recognition*, *35*(1), 295-297.

Yang, J., & Yang, J-Y. (2003). Why can LDA be performed in PCA transformed space? *Pattern Recognition*, *36*(2), 563-566.

Yang, J., Yang, J-Y., & Frangi, A.F. (2003). Combined fisherfaces framework. *Image and Vision Computing*, *21*(12), 1037-1044.

Yang, J., Yang, J-Y., Frangi, A. F., & Zhang, D. (2003). Uncorrelated projection discriminant analysis and its application to face image feature extraction. *International Journal of Pattern Recognition and Artificial Intelligence*, *17*(8), 1325-1347.

Yang, J., Yang, J-Y., & Zhang, D. (2002). What's wrong with the Fisher criterion?. *Pattern Recognition*, *35*(11), 2665-2668.

Yang, J., Yang, J-Y., Zhang, D., & Lu, J. F. (2003). Feature fusion: Parallel strategy vs. serial strategy. *Pattern Recognition*, *36*(6), 1369-1381.

Yang, J., Ye, H., Yang, J-Y., & Zhang, D. (2004). A new LDA-KL combined method for feature extraction and its generalization. *Pattern Analysis and Application*, *7*(1), 40-50.

Yang, J., Zhang, D., Frangi, A.F., & Yang, J-Y. (2004). Two-dimensional PCA: A new approach to face representation and recognition. *IEEE Transactions on Pattern Analysis and Machine Intelligence, 26*(1), 131-137.

Yang, J., Zhang, D., & Yang, J-Y. (2003). A generalized K-L expansion method which can deal with small sample size and high-dimensional problems. *Pattern Analysis and Application*, *6*(1), 47-54.

Yang, J., Zhang, D., Yang, J-Y., Zhong, J., & Frangi, A. F. (2005). KPCA plus LDA: A complete kernel Fisher discriminant framework for feature extraction and recognition. *IEEE Trans. Pattern Anal. Machine Intell.*, *27*(2), 230-244.

Yang, M. H. (2002). Kernel eigenfaces vs. kernel fisherfaces: Face recognition using kernel methods. *Proceedings of the Fifth IEEE International Conference on Automatic Face and Gesture Recognition (RGR'02),* 215-220.

Ye, J. P., Janardan, R., Park, C. H., & Park, H. (2004). An optimization criterion for generalized discriminant analysis on undersampled problems. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, *26*(8), 982-994.

Yu, H., & Yang, J. (2001). A direct LDA algorithm for high-dimensional data – with application to face recognition. *Pattern Recognition*, *34*(10) 2067-2070.

Yuen, P. C., & Lai, J. H. (2002). Face representation using independent component analysis. *Pattern Recognition*, *35*(6), 1247-1257.

Zhang, D. (2002a). *Biometrics solutions for authentication in an e-world*. Boston: Kluwer Academic Publishers.

Zhang, D. (2000b). *Automated biometrics: Technologies and systems*. Boston: Kluwer Academic Publishers.

Zhang, D. (2004). *Palmprint authentication*. Boston: Kluwer Academic Publishers.

Zhang, D., Kong, W. K., You, J., & Wong, M. (2003). On-line palmprint identification. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, *25*(9), 1041-1050.

Zhang, D., Peng, H., Zhou, J., & Pal, S.K. (2002b). A novel face recognition system using hybrid neural and dual eigenfaces methods. *IEEE Transactions on Systems, Man, and Cybernetics, Part A, 32*(6) 787-793.

Zhao, L., & Yang, Y. (1999). Theoretical analysis of illumination in PCA-based vision systems. *Pattern Recognition*, *32*(4), 547-564.

Zhao, W., Krishnaswamy, A., Chellappa, R., Swets, D., & Weng, J. (1998). Discriminant analysis of principal components for face recognition. In H. Wechsler, P. J. Phillips, V. Bruce, F. F. Soulie, & T. S. Huang (Eds.), *Face recognition: From theory to applications* (pp. 73-85). Berlin & Heidelberg: Springer-Verlag.

Section I

BID Fundamentals

Chapter II

Principal Component Analysis

ABSTRACT

In this chapter, we first describe some basic concepts of PCA, a useful statistical technique that can be used in many fields, such as face patterns and other biometrics. Then, we introduce PCA definitions and related technologies. Following, we discuss non-linear PCA technologies. Finally, some useful conclusions are summarized.

INTRODUCTION

PCA is a classical feature extraction and data representation technique widely used in pattern recognition and computer vision (Duda, Hart, & Stork, 2000; Yang, Zhang, Frangi, & Yang, 2004; Anderson, 1963; Kim, n.d.; Boser, Guyon, & Vapnik, 1992). Sirovich and Kirby first used PCA to efficiently represent pictures of human faces (Sirovich & Kirby, 1987; Kirby & Sirovich, 1990). They argued that any face image could be reconstructed approximately as a weighted sum of a small collection of images that define a facial basis (eigenimages) and a mean image of the face. Since eigenpictures are fairly good at representing face images, one could consider using the projections along them as classification features for recognizing human faces. Within this context, Turk and Pentland presented the well-known eigenfaces method for face recognition in 1991 (Turk & Pentland, 1991). They developed the well-known face recognition method, where the

eigenfaces correspond to the eigenvectors associated with the dominant eigenvalues of the face covariance matrix. The eigenfaces define a feature space, or "face space," which drastically reduces the dimensionality of the original space, and face detection and identification are carried out in the reduced space (Zhang, 1997). Since then, PCA has been widely investigated and become one of the most successful approaches in face recognition (Pentland, 2000; Grudin, 2000; Cottrell & Fleming, 1990; Valentin, Abdi, O'Toole, & Cottrell, 1994). Penev and Sirovich discussed the problem of the dimensionality of the "face space" when eigenfaces are used for representation (Penev & Sirovich, 2000). Zhao and Yang tried to account for the arbitrary effects of illumination in PCA-based vision systems by generating an analytically closed form formula of the covariance matrix for the case with a special lighting condition and then generalizing to an arbitrary illumination via an illumination equation (Zhao & Yang, 1999). However, Wiskott, Fellous, Krüger and von der Malsburg (1997) pointed out that PCA could not capture even the simplest invariance unless this information is explicitly provided in the training data. They proposed a technique known as elastic bunch graph matching to overcome the weaknesses of PCA. In this chapter, we will show you some basic definitions of PCA.

DEFINITIONS AND TECHNOLOGIES

Mathematical Background of PCA

This section will attempt to give some elementary background mathematical skills required to understand the process of PCA (Smith, 2002; Vapnik, 1995).

Eigenvectors and Eigenvalues

Given a d-by-d matrix $\mathbf{M}$, a very important class of equation is of the form (Duda, Hart, & Stork, 2000):

$$\mathbf{Mx} = \lambda\mathbf{x} \tag{2.1}$$

for scalar λ, which can be written:

$$(\mathbf{M} - \lambda\mathbf{I})\mathbf{x} = \mathbf{0} \tag{2.2}$$

where $\mathbf{I}$ is the identity matrix and $\mathbf{0}$ is the zero vector. The solution vector $\mathbf{x} = \mathbf{e}_i$ and corresponding scalar $\lambda = \lambda_i$ are called the *eigenvector* and associated *eigenvalue*, respectively. If $\mathbf{M}$ is real and symmetric, there are d (possibly nondistinct) solution vectors $\{\mathbf{e}_1, \mathbf{e}_2, \ldots, \mathbf{e}_d\}$, each with an associated eigenvalue $\{\lambda_1, \lambda_2, \ldots, \lambda_d\}$. Under multiplication by $\mathbf{M}$ the eigenvectors are changed only in magnitude, not direction:

$$\mathbf{Me}_j = \lambda_j\mathbf{e}_j \tag{2.3}$$

If $\mathbf{M}$ is diagonal, then the eigenvectors are parallel to the coordinate axes.

One method of finding the eigenvalues is to solve the *characteristic equation* (or *secular equation*):

$$\left|\mathbf{M}-\lambda\mathbf{I}\right| = \lambda^{d} + a_1\lambda^{d-1} + \cdots + a_{d-1}\lambda + a_d = 0 \tag{2.4}$$

for each of its d (possibly nondistinct) roots λ_j. For each such root, we then solve a set of linear equations to find its associated eigenvector $\mathbf{e}_j$.

Finally, it can be shown that the trace of a matrix is just the sum of the eigenvalues and the determinant of a matrix is just the product of its eigenvalues:

$$\text{tr}[\mathbf{M}] = \sum_{i=1}^{d}\lambda_i \text{ and } \left|\mathbf{M}\right| = \prod_{i=1}^{d}\lambda_i \tag{2.5}$$

If a matrix is diagonal, then its eigenvalues are simply the nonzero entries on the diagonal, and the eigenvectors are the unit vectors parallel to the coordinate axes.

Expectations, Mean Vectors and Covariance Matrices

The expected value of a vector is defined as the vector whose components are the expected values of the original components (Duda, Hart, & Stork, 2000). Thus, if $\mathbf{f}(\mathbf{x})$ is an n-dimensional, vector-valued function of the d-dimensional random vector $\mathbf{x}$:

$$\mathbf{f}(\mathbf{x}) = \begin{bmatrix} f_1(\mathbf{x}) \\ f_2(\mathbf{x}) \\ \vdots \\ f_n(\mathbf{x}) \end{bmatrix} \tag{2.6}$$

then the expected value of $\mathbf{f}$ is defined by:

$$\mathrm{E}[\mathbf{f}] = \begin{bmatrix} \mathrm{E}[f_1(\mathbf{x})] \\ \mathrm{E}[f_2(\mathbf{x})] \\ \vdots \\ \mathrm{E}[f_n(\mathbf{x})] \end{bmatrix} = \sum_{\mathbf{x}} \mathbf{f}(\mathbf{x})P(\mathbf{x}) \tag{2.7}$$

In particular, the d-dimensional mean vector μ is defined by:

$$\mu = \mathrm{E}[\mathbf{x}] = \begin{bmatrix} \mathrm{E}[x_1] \\ \mathrm{E}[x_2] \\ \vdots \\ \mathrm{E}[x_d] \end{bmatrix} = \begin{bmatrix} \mu_1 \\ \mu_2 \\ \vdots \\ \mu_d \end{bmatrix} = \sum_{\mathbf{x}} \mathbf{x}P(\mathbf{x}) \tag{2.8}$$

Similarly, the covariance matrix $\mathbf{\Sigma}$ is defined as the (square) matrix whose *ij*th element σ_{ij} is the covariance of x_i and x_j:

$$\sigma_{ij} = \sigma_{ji} = \mathrm{E}\left[\left(x_i - \mu_i\right)\left(x_j - \mu_j\right)\right] \qquad i, j = 1...d \tag{2.9}$$

We can use the vector product $(\mathbf{x}\text{-}\mu)(\mathbf{x}\text{-}\mu)^T$ to write the covariance matrix as:

$$\Sigma = \mathrm{E}\left[\left(\mathbf{x} - \mu\right)\left(\mathbf{x} - \mu\right)^T\right] \tag{2.10}$$

Thus, $\mathbf{\Sigma}$ is symmetric, and its diagonal elements are just the variances of the individual elements of $\mathbf{x}$, which can never be negative; the off-diagonal elements are the covariances, which can be positive or negative. If the variances are statistically independent, the covariances are zero, and the covariance matrix is diagonal. The analog to the Cauchy-Schwarz inequality comes from recognizing that if $\mathbf{w}$ is any *d*-dimensional vector, then the variance of $\mathbf{w}^t\mathbf{x}$ can never be negative. This leads to the requirement that the quadratic form $\mathbf{w}^t\mathbf{\Sigma}\mathbf{x}$ never be negative. Matrices for which this is true are said to be *positive semidefinite*; thus, the covariance matrix $\mathbf{\Sigma}$ must be positive semidefinite. It can be shown that this is equivalent to the requirement that none of the eigenvalues of $\mathbf{\Sigma}$ can be negative.

Background Mathematics

We begin by considering the problem of representing all of the vectors in a set of *n d*-dimensional samples $\mathbf{x}_1, \ldots, \mathbf{x}_n$ by a single vector $\mathbf{x}_0$ (Duda, Hart, & Stork, 2000). To be more specific, suppose that we want to find a vector $\mathbf{x}_0$ such that the sum of the squared distances between $\mathbf{x}_0$ and the various $\mathbf{x}_k$ is as small as possible. We define the squared-error criterion function $J_0(\mathbf{x}_0)$ by:

$$J_0\left(\mathbf{x}_0\right) = \sum_{k=1}^{n} \| \mathbf{x}_0 - \mathbf{x}_k \|^2 \tag{2.11}$$

and seek the value of $\mathbf{x}_0$ that minimizes J_0. It is simple to show that the solution to this problem is given by $\mathbf{x}_0 = \mathbf{m}$, where $\mathbf{m}$ is the mean vector of sample:

$$\mathbf{m} = \frac{1}{n}\sum_{k=1}^{n} \mathbf{x}_k \tag{2.12}$$

This can be easily verified by writing:

$$\begin{aligned} J_0(\mathbf{x_0}) &= \sum_{k=1}^{n}\left\|\mathbf{x_0} - \mathbf{m}\right\|^2 - 2\sum_{k=1}^{n}\left(\mathbf{x_0} - \mathbf{m}\right)^T\left(\mathbf{x}_k - \mathbf{m}\right) + \sum_{k=1}^{n}\left\|\mathbf{x}_k - \mathbf{m}\right\|^2 \\ &= \sum_{k=1}^{n}\left\|\mathbf{x_0} - \mathbf{m}\right\|^2 - 2\left(\mathbf{x_0} - \mathbf{m}\right)^T\sum_{k=1}^{n}\left(\mathbf{x}_k - \mathbf{m}\right) + \sum_{k=1}^{n}\left\|\mathbf{x}_k - \mathbf{m}\right\|^2 \\ &= \sum_{k=1}^{n}\left\|\mathbf{x_0} - \mathbf{m}\right\|^2 + \underbrace{\sum_{k=1}^{n}\left\|\mathbf{x}_k - \mathbf{m}\right\|^2}_{\text{independent of } \mathbf{x}_0} \end{aligned} \tag{2.13}$$

Since the second sum is independent of $\mathbf{x}_0$, this expression is obviously minimized by the choice $\mathbf{x}_0 = \mathbf{m}$.

The mean vector of samples is a zero-dimensional representation of the data set. It is simple, but it does not reveal any of the variability in the data. We can obtain a more interesting, one-dimensional representation by projecting the data onto a line running through the sample mean. Let $\mathbf{e}$ be a unit vector in the direction of the line. Then the equation of the line can be written as:

$$\mathbf{x} = \mathbf{m} + a\,\mathbf{e} \tag{2.14}$$

where the scalar a (which takes on any real value) corresponds to the distance between any point $\mathbf{x}$ and the mean $\mathbf{m}$. If we represent $\mathbf{x}_k$ by $\mathbf{m} + a\,\mathbf{e}$, we can find an "optimal" set of coefficients a_k by minimizing the squared-error criterion function:

$$\begin{aligned} J(a_1, \ldots, a_n, \mathbf{e}) &= \sum_{k=1}^{n} \left\| (\mathbf{m} + a_k \mathbf{e}) - \mathbf{x}_k \right\|^2 \\ &= \sum_{k=1}^{n} \left\| a_k \mathbf{e} - (\mathbf{x}_k - \mathbf{m}) \right\|^2 \\ &= \sum_{k=1}^{n} a_k \left\| \mathbf{e} \right\|^2 - 2\sum_{k=1}^{n} a_k \mathbf{e}^T (\mathbf{x}_k - \mathbf{m}) + \sum_{k=1}^{n} \left\| \mathbf{x}_k - \mathbf{m} \right\|^2 \end{aligned} \tag{2.15}$$

Recognizing that $||\mathbf{e}|| = 1$, partially differentiating with respect to a_k, and setting the derivative to zero, we obtain:

$$a_k = \mathbf{e}^T(\mathbf{x}_k - \mathbf{m}) \tag{2.16}$$

Geometrically, this result merely says that we obtain a least-squares solution by projecting the vector $\mathbf{x}_k$ onto the line in the direction of $\mathbf{e}$ that passes through the sample mean.

This brings us to the more interesting problem of finding the best direction $\mathbf{e}$ for the line. The solution to this problem involves the so-called scatter matrix $\mathbf{S}_t$ defined by:

$$\mathbf{S}_t = \sum_{k=1}^{n} (\mathbf{x}_k - \mathbf{m})(\mathbf{x}_k - \mathbf{m})^T. \tag{2.17}$$

The scatter matrix should look familiar: It is merely n – 1 times the sample covariance matrix. It arises here when we substitute a_k found in Equation 2.16 into Equation 2.15 to obtain:

$$\begin{aligned} J_1(\mathbf{e}) &= \sum_{k=1}^{n} a_k^{\,2} - 2\sum_{k=1}^{n} a_k^{\,2} + \sum_{k=1}^{n} \left\| \mathbf{x}_k - \mathbf{m} \right\|^2 \\ &= -\sum_{k=1}^{n} \left[\mathbf{e}^T (\mathbf{x}_k - \mathbf{m}) \right]^2 + \sum_{k=1}^{n} \left\| \mathbf{x}_k - \mathbf{m} \right\|^2 \\ &= -\sum_{k=1}^{n} \mathbf{e}^T (\mathbf{x}_k - \mathbf{m})(\mathbf{x}_k - \mathbf{m})^T \mathbf{e} + \sum_{k=1}^{n} \left\| \mathbf{x}_k - \mathbf{m} \right\|^2 \\ &= -\mathbf{e}^T \mathbf{S}_t \mathbf{e} + \sum_{k=1}^{n} \left\| \mathbf{x}_k - \mathbf{m} \right\|^2 \end{aligned} \tag{2.18}$$

Clearly, the vector **e** that minimizes J_1 also maximizes $\mathbf{e}^T\mathbf{S}_t\mathbf{e}$. We use the method of Lagrange multipliers to maximize $\mathbf{e}^T\mathbf{S}_t\mathbf{e}$ subject to the constraint that ||**e**|| = 1. Letting λ be the undetermined multiplier, we differentiate:

$$u = \mathbf{e}^T\mathbf{S}_t\mathbf{e} - \lambda(\mathbf{e}^T\mathbf{e} - 1) \tag{2.19}$$

with respect to **e** to obtain:

$$\frac{\partial u}{\partial e} = 2\mathbf{S}_t\,\mathbf{e} - 2\lambda\mathbf{e} \tag{2.20}$$

Setting this gradient vector equal to zero, we see that **e** must be an eigenvector of the scatter matrix:

$$\mathbf{S}_t\,\mathbf{e} = \lambda\mathbf{e} \tag{2.21}$$

In particular, because $\mathbf{e}^T\mathbf{S}_t\mathbf{e} = \lambda\,\mathbf{e}^T\mathbf{e} = \lambda$, it follows that to maximize $\mathbf{e}^T\mathbf{S}_t\mathbf{e}$, we want to select the eigenvector corresponding to the largest eigenvalue of the scatter matrix. In other words, to find the best one-dimensional projection of the data (best in the least-sum-of-squared-sense), we project the data onto a line through the sample mean in the direction of the eigenvector of the scatter matrix having the largest eigenvalue.

This result can be readily extended from a one-dimensional projection to a *d'* dimensional projection. In place of Equation 2.14, we write:

$$\mathbf{x} = \mathbf{m} + \sum_{i=1}^{d'} a_i\mathbf{e}_i \tag{2.22}$$

where *d'* d" *d*. It is not difficult to show that the criterion function:

$$J_{d'} = \sum_{k=1}^{n}\left\|\left(\mathbf{m} + \sum_{i=1}^{d'} a_{k_i}\mathbf{e}_i\right) - \mathbf{x}_k\right\|^2 \tag{2.23}$$

is minimized when the vectors $\mathbf{e}_1, \ldots, \mathbf{e}_{d'}$ are the d' eigenvectors of the scatter matrix having the largest eigenvalues. Because the scatter matrix is real and symmetric, these eigenvectors are orthogonal. They form a natural set of basis vectors for representing any feature vector **x**. The coefficients a_i in Equation 2.22 are the components of **x** in that basis, and are called the *principal components.* Geometrically, if we picture the data points $\mathbf{x}_1, \ldots, \mathbf{x}_n$ as forming a *d*-dimensional, hyperellipsoidally shaped cloud, then the eigenvectors of the scatter matrix are the dimensionality of feature space by restricting attention to those directions along which the scatter of the cloud is greatest.

Principal Component Analysis (PCA)

Finally, we come to PCA (Smith, 2002; Zhao, Krishnaswamy, Chellappa, Swets, & Weng, 1998; Chellappa & Sirohey, 1995). As mentioned above, this is a way of identifying

patterns in data, and expressing the data in such a way as to highlight their similarities and differences. Once you have found these patterns in the data, it is possible to compress the data — that is, by reducing the number of dimensions — without much loss of information (Duda, Hart, & Stork, 2000; Smith, 2002; Joliffe, 1986). This technique is used in image compression, as we will see in a later section (Gonzalez & Wintz, 1987). This section will take you through the steps needed to perform a PCA on a set of data. We are not going to describe exactly why the technique works, but we will try to provide an explanation of what is happening at each point so that you can make informed decisions when you try to use this technique.

Method

- **Step 1: Get some data**
  In this simple example, we are going to use a made-up data set that is found in Figure 2.1 (Smith, 2002). It only has two dimensions, so we can provide plots of the data to show what the PCA analysis is doing at each step.
- **Step 2: Subtract the mean**
  All the *x* values have $\overline{x}$ (the mean of the *x* values of all the data points) subtracted, and all the *y* values have $\overline{y}$ subtracted from them. This produces a data set whose mean is zero.
- **Step 3: Calculate the covariance matrix**

Figure 2.1. PCA example data, original data on the left, data with the means subtracted on the right, and a plot of the data (Smith, 2002)

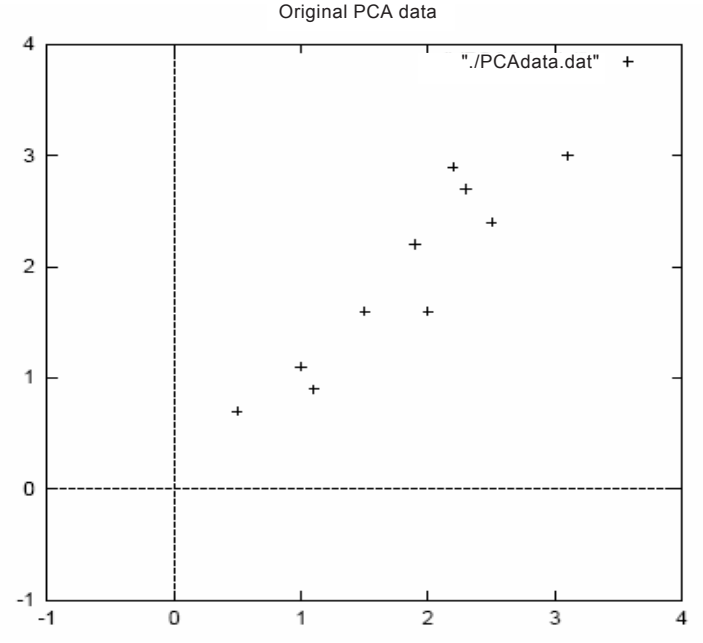
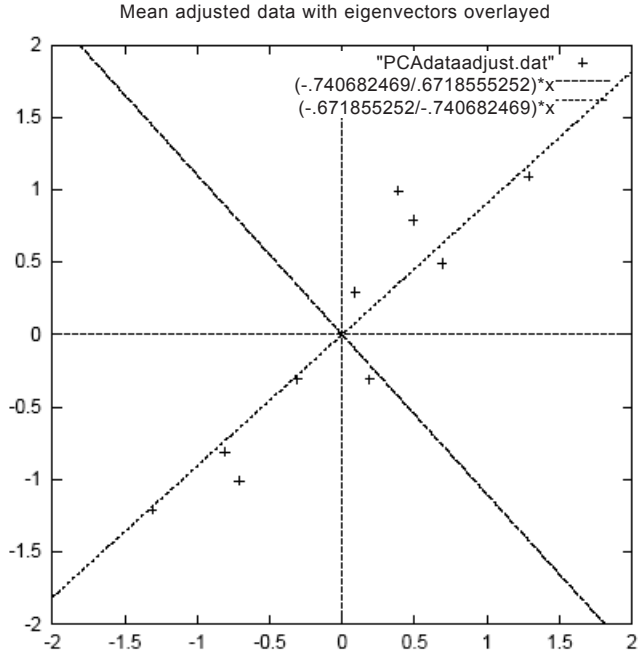


Table 2.1. 2-dimensional data set (Smith, 2002) (a) and data adjust (b)

(a)		(b)	
x	*y*	*x*	*y*
2.5	2.4	0.69	0.49
0.5	0.7	-1.31	-1.21
2.2	2.9	0.39	0.99
1.9	2.2	0.09	0.29
3.1	3.0	1.29	1.09
2.3	2.7	0.49	0.79
2	1.6	0.19	-0.31
1	1.1	-0.81	-0.81
1.5	1.6	-0.31	-0.31
1.1	0.9	-0.71	-1.01

Figure 2.2. A plot of the normalized data (mean subtracted) with the eigenvectors of the covariance matrix overlaid on top (Smith, 2002)



This is done in exactly the same way discussed earlier. The result:

$$cov = \begin{pmatrix} 0.616555556 & 0.615444444 \\ 0.615444444 & 0.716555556 \end{pmatrix} \tag{2.24}$$

- **Step 4: Calculate the eigenvectors and eigenvalues of the covariance matrix**
  Here are the eigenvectors and eigenvalues:

$$eigenvalues = \begin{pmatrix} 0.0490833989 \\ 1.28402771 \end{pmatrix} \tag{2.25}$$

$$eigenvectors = \begin{pmatrix} -0.735178656 & -0.677873399 \\ 0.677873399 & -0.735178656 \end{pmatrix}$$

  So, since the non-diagonal elements in this covariance matrix are positive, we should expect that both the x and y variables increase together.
  It is important to note that both of these eigenvectors are *unit* eigenvectors; that is, both of their lengths are 1. This is very important for PCA. Fortunately, most math packages, when asked for eigenvectors, will give you unit eigenvectors.
  The data plotted in Figure 2.2 has quite a strong pattern. As expected from the covariance matrix, the two variables do indeed increase together. On top of the data, I have also plotted both eigenvectors. They appear as diagonal dotted lines. As stated in the eigenvector section, they are perpendicular to each other, but more importantly, they provide us with information about the patterns in the data. See how one of the eigenvectors goes through the middle of the points, like a line of best fit? That eigenvector is showing us how these two data sets are related along that line. The second eigenvector gives us the other, less important, pattern in the data: that all the points follow the main line, but are off to the side of the main line by some amount.
  So, by this process of taking the eigenvectors of the covariance matrix, we have been able to extract lines that characterize the data. The remaining steps involve transforming the data so it is expressed in terms of these lines.
- **Step 5: Choose components and form a feature vector**
  Here is where the notion of data compression and reduced dimensionality comes in (Smith, 2002). If you look at the eigenvectors and eigenvalues from the previous section, you will notice that the eigenvalues are quite different. In fact, the eigenvector with the highest eigenvalue is the principal component of the data set. In our example, the eigenvector with the largest eigenvalue was the one that pointed down the middle of the data. This is the most significant relationship between the data dimensions.
  In general, once eigenvectors are found from the covariance matrix, the next step is to order them by eigenvalue, highest to lowest. This gives the components in order of significance. Now, if you like, you can decide to *ignore* the components of less significance. You do lose some information, but, if the eigenvalues are small, you don't lose much. If you leave out some components, the final data set will have fewer dimensions than the original. To be precise, if you originally have n dimensions in your data, and you calculate n eigenvectors and eigenvalues and then choose only the first p eigenvectors, then the final data set has only p dimensions.

You must now form a feature vector, also known as a matrix of vectors. This is constructed by taking the eigenvectors that you want to keep and forming a matrix with these eigenvectors in the columns.

$$Feature\,Vector = \left(eig_1 \quad eig_2 \quad eig_3 \quad \cdots \quad eig_n\right) \tag{2.26}$$

Given our example set of data, and the fact we have two eigenvectors, we have two choices: We can either form a feature vector with both of the eigenvectors:

$$\begin{pmatrix} -0.677873399 & -0.735178656 \\ -0.735178656 & 0.677873399 \end{pmatrix}$$

or, we can choose to leave out the smaller, less-significant component and only have a single column:

$$\begin{pmatrix} -0.677873399 \\ -0.735178656 \end{pmatrix}$$

We shall see the result of each of these in the next section.

- **Step 6: Derive the new data set**
  This is the final step in PCA, and is also the easiest (Smith, 2002). Once we have chosen the components (eigenvectors) that we wish to keep in our data and formed a feature vector, we simply take the transpose of the vector and multiply it on the remainder of the original data set, transposed.

$$FinalData = RowFeatureVector \times RowDataAdjust \tag{2.27}$$

RowFeatureVector is the matrix with the eigenvectors in the columns *transposed* so that the eigenvectors are now in the rows, with the most significant eigenvector at the top; and *RowDataAdjust* is the mean-adjusted data *transposed* (i.e., the data items are

Table 2.2. Transformed data (Smith, 2002)

x	*Y*
-0.827970186	-0.175115307
1.77758033	0.142857227
-0.992197494	0.384374989
-0.274210416	0.130417207
-1.67580142	-0.209498461
-0.912949103	0.175282444
0.0991094375	-0.349824698
1.14457216	0.0464172582
0.438046137	0.0177646297
1.22382056	-0.162675287

in each column, with each row holding a separate dimension). The equations from here on are easier if we take the transpose of the feature vector and the data first, rather than having a little T symbol above their names. *FinalData* is the final data set, with data items in columns and dimensions along rows.

What will this give us? It will give us the original data solely in terms of the vectors we chose. Our original data set had two axes, x and *y*, so our data was in terms of them. It is possible to express data in terms of any two axes that you like. The expression is the most efficient if these axes are perpendicular. This is why it is important that eigenvectors always be perpendicular to each other. We have changed our data from being in terms of the axes *x* and *y* to be in terms of our two eigenvectors. In the case of when the new data set has reduced dimensionality – that is, we have left some of the eigenvectors out – the new data is only in terms of the vectors that we decided to keep.

To show this on our data, we have done the final transformation with each of the possible feature vectors. We have taken the transpose of the result in each case to render the data in a table-like format. We have also plotted the final points to show how they relate to the components.

If we keep both eigenvectors for the transformation, we get the data and the plot found in Figure 2.3. This plot is basically the original data, rotated so that the eigenvectors are the axes. This is understandable, since we have lost no information in this decomposition.

The other transformation we can make is by taking only the eigenvector with the largest eigenvalue. The table of data resulting from that is found in Figure 2.4. As

Figure 2.3. The table of data by applying the PCA analysis using both eigenvectors, and a plot of the new data points (Smith, 2002)

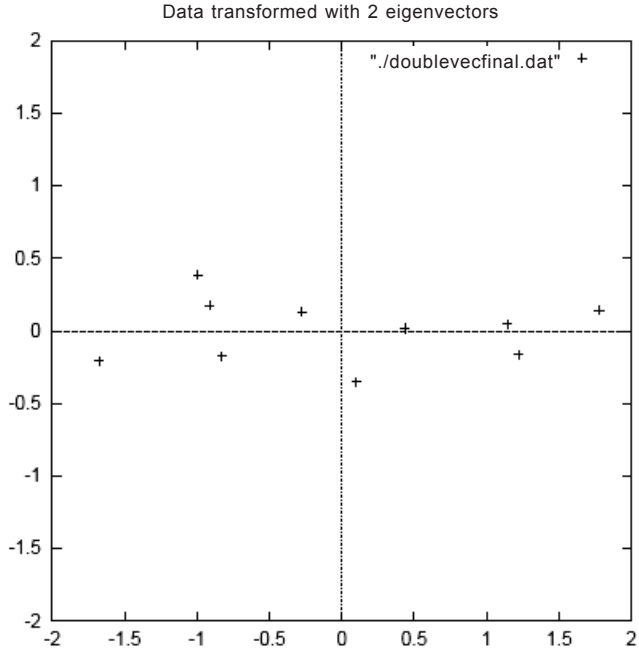


Figure 2.4. The data after transforming using only the most significant eigenvector (Smith, 2002)

Transformed Data (Single eigenvector)

x
-0.827970186
1.77758033
-0.992197494
-0.274210416
-1.67580142
-0.912949103
0.0991094375
1.14457216
0.438046137
1.22382056

expected, it has only a single dimension. If you compare this data set with the one resulting from using both eigenvectors, you will notice that this data set is exactly the first column of the other. So, if you were to plot this data, it would be 1-dimensional, and would be points on a line in exactly the *x* positions of the points in the plot in Figure 2.3. We have effectively thrown away the whole other axis, which is the other eigenvector.

Basically, we have transformed our data so that it is expressed in terms of the patterns between them, where the patterns are the lines that most closely describe the relationships between the data. This is helpful because we have now classified our data point as a combination of the contributions from each of those lines. Initially, we had the simple *x* and *y* axes. This is fine, but the *x* and *y* values of each data point don't really tell us exactly how that point relates to the rest of the data. Now, the values of the data points tell us exactly where (i.e., above/below) the trend lines the data point sits. In the case of the transformation using both eigenvectors, we have simply altered the data so it is in terms of those eigenvectors instead of the usual axes. But the single-eigenvector decomposition has removed the contribution due to the smaller eigenvector and left us with data that is only in terms of the other.

Getting the Old Data Back

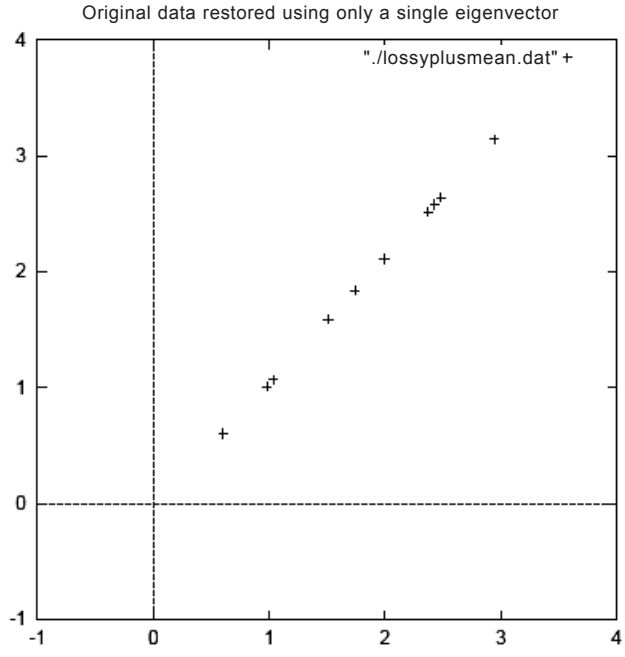
Wanting to get the original data back is obviously of great concern if you are using the PCA transform for data compression (we will see an example in the next section) (Smith, 2002). Before we do that, remember that only if we took all the eigenvectors in our transformation will we get back exactly the original data. If we have reduced the number of eigenvectors in the final transformation, then the retrieved data has lost some information.

See the final transform in Equation 2.27, which can be turned around so that, to get the original data back:

$$RowDataAdjust = RowFeatureVector^{-1} \times FinalData \tag{2.28}$$

where $RowFeatureVector^{-1}$ is the inverse of $RowFeatureVector$. However, when we take *all* the eigenvectors in our feature vector, it turns out that the inverse of our feature

Figure 2.5. The reconstruction from the data that was derived using only a single eigenvector (Smith, 2002)



vector is actually equal to the transpose of our feature vector. This is only true because the elements of the matrix are all the unit eigenvectors of our data set. This makes the return trip to our data easier, because the equation becomes:

$$RowDataAdjust = RowFeatureVector^{T} \times FinalData \tag{2.29}$$

But, to get the actual original data back, we need to add on the mean of that original data (remember, we subtracted it at the start). So, for completeness:

$$RowOriginalData = (RowFeatureaVector^{T} \times FinalData) + OriginalMean \tag{2.30}$$

This formula is also applied when you do not have all the eigenvectors in the feature vector. So even when you leave out some eigenvectors, the above equation still makes the correct transform.

We will not perform the data re-creation using the *complete* feature vector, because the result is exactly the data we started with. However, I will do it with the reduced feature vector to show how information has been lost. Figure 2.5 shows this plot. Compare it to the original data plotted in Figure 2.1 and you will notice how, while the variation along the principle eigenvector (see Figure 2.2 for the eigenvector overlaid on top of the mean-adjusted data) has been kept, the variation along the other component (the other eigenvector that we left out) has gone.

NON-LINEAR PCA TECHNOLOGIES

An Introduction to Kernel PCA

As previously mentioned, PCA is a classical linear feature extraction technique (Joliffe, 1986; Diamantaras & Kung, 1996). In recent years, the nonlinear feature extraction methods, such as kernel principal component analysis (KPCA), have been of wide concern (Zwald, Bousquet, & Blanchard, n.d.; Schölkopf, Smola, & Müller; 1998; Schölkopf, Smola, & Müller, 1998, 1999; Liu, Lu, & Ma, 2004).

KPCA is a technique for non-linear feature extraction closely related to methods applied in Support Vector Machines (Schölkopf, Burges, & Smola, 1999). It has proven useful for various applications, such as de-noising (Kearns, Solla, & Cohn, 1999) and as a pre-processing step in regression problems (Rosipal, Girolami, & Trejo, 2000). KPCA has also been applied by Romdhani, Gong, and Psarrou (1999) to the construction of non-linear statistical shape models of faces, but we will argue that their approach to constraining shape variability is not generally valid (Huang, 2002).

The kernel trick is demonstrated to be able to efficiently represent complicated nonlinear relations of the input data, and recently kernel-based nonlinear analysis methods have been given more attention. Due to their versatility, kernel methods are currently very popular as data-analysis tools. In such algorithms, the key object is the so-called kernel matrix (the Gram matrix built on the data sample), and it turns out that its spectrum can be related to the performance of the algorithm. Studying the behavior of eigenvalues of kernel matrices, their stability and how they relate to the eigenvalues of the corresponding kernel integral operator is thus crucial for understanding the statistical properties of kernel-based algorithms.

The kernel trick first maps the input data into an implicit feature space F with a nonlinear mapping, and then the data are analyzed in F. KPCA was originally developed by Schölkopf. Schölkopf proposed to combine the kernel trick with PCA and developed KPCA for feature representation (Schölkopf, Smola, & Müller, 1998; Schölkopf & Smola, 2002; Schölkopf, Mika, Burges, et al., 1999). First, the input data is mapped into an implicit feature space F with the kernel trick, and then linear PCA is performed in F to extract nonlinear principal components of the input data. This can also be called a nonlinear subspace analysis method. It was reported that KPCA outperformed PCA for face recognition in Yang's work (Yang, Ahuja, & Driegman, 2000), and better results were given in Kim, Jung and Kim (2002) by combining KPCA with an SVM classifier. However, like PCA, KPCA is designed to minimize the overall variance of the input data, and it is not necessarily optimal for discriminating purposes.

Background Mathematics

Recall the set of n d-dimensional samples $\mathbf{x}_1, \ldots, \mathbf{x}_n$ by a single vector $\mathbf{x}_0$ mentioned earlier, $\mathbf{x}_k = [x_{k1}, x_{k2}, \ldots, x_{kd}]^T \in \mathbb{R}^d$, PCA aims to find the projection directions that maximize the variance, S_t, which is equivalent to finding the eigenvalues from the covariance matrix (Yang, n.d.):

$$\mathbf{S}_t \mathbf{e} = \lambda \mathbf{e} \tag{2.31}$$

For eigenvalues $\lambda \geq 0$ and eigenvectors $\mathbf{e} \in \mathbb{R}^d$. In KPCA, each vector $\mathbf{x}$ is projected from the input space, $\mathbb{R}^d$, to a high-dimensional feature space $\mathcal{F}$, by a nonlinear mapping function: Φ: $\mathbb{R}^d \rightarrow \mathcal{F}$. Note that the dimensionality of the feature space can be arbitrarily large. In $\mathcal{F}$, the corresponding eigenvalue problem is:

$$\mathbf{S}_t^{\Phi} \mathbf{e}^{\Phi} = \lambda \mathbf{e}^{\Phi} \tag{2.32}$$

where $\mathbf{S}_t^{\Phi}$ is a covariance matrix. All solutions $\mathbf{e}^{\Phi}$ with $\lambda \neq 0$ lie in the span of $\Phi(\mathbf{x}_1), ..., \Phi(\mathbf{x}_n)$, namely, there exist coefficients α_i such that:

$$\mathbf{e}^{\Phi} = \sum_{i=1}^{n} \alpha_i \Phi(\mathbf{x}_i) \tag{2.33}$$

Denoting an $n \times n$ matrix K by:

$$K_{ij} = k(\mathbf{x}_i, \mathbf{x}_j) = \Phi(\mathbf{x}_i) \bullet \Phi(\mathbf{x}_j) \tag{2.34}$$

The KPCA problem becomes:

$$n\lambda K\alpha = K^2\alpha \tag{2.35}$$

$$n\lambda\alpha = K\alpha \tag{2.36}$$

where α denotes a column vector with entries $\alpha_1, \ldots, \alpha_2$. The above derivations assume that all the projected samples $\Phi(\mathbf{x})$ are centered in $\mathcal{F}$.

Note that conventional PCA is a special case of KPCA with a polynomial kernel of the first order. In other words, KPCA is a generalization of conventional PCA, since different kernels can be utilized for different nonlinear projections.

We can now project the vectors in $\mathcal{F}$ to a lower dimensional space spanned by the eigenvectors $\mathbf{e}^{\Phi}$. Let $\mathbf{x}$ be a test sample whose projection is $\Phi(\mathbf{x})$ in $\mathcal{F}$, then the projection of $\Phi(\mathbf{x})$ onto the eigenvectors $\mathbf{e}^{\Phi}$ is the nonlinear principal components corresponding to Φ:

$$\mathbf{e}^{\Phi} \bullet \Phi(\mathbf{x}) = \sum_{i=1}^{n} \alpha_i \left(\Phi(\mathbf{x}_i) \bullet \Phi(\mathbf{x}) \right) = \sum_{i=1}^{n} \alpha_i k(\mathbf{x}_i, \mathbf{x}) \tag{2.37}$$

In other words, we can extract the first q ($1 \leq q \leq n$) nonlinear principal components (i.e., eigenvectors $\mathbf{e}^{\Phi}$) using the kernel function without the expensive operation that explicitly projects the samples to a high dimensional space $\mathcal{F}$. The first q components correspond to the first q non-increasing eigenvalues of Equation 2.36. For face recognition where each $\mathbf{x}$ encodes a face image, we call the extracted nonlinear principal components kernel eigenfaces.

Methods

To perform KPCA (Figure 2.6), the following steps have to be carried out: First, we compute the matrix $K_{ij} = k(\mathbf{x}_i, \mathbf{x}_j) = \Phi(\mathbf{x}_i) \bullet \Phi(\mathbf{x}_j)$. Next, we solve Equation 2.36 by diagonalizing K, and normalize the eigenvector expansion coefficients α_i by requiring $\lambda_n(\alpha_n \bullet \alpha_n) = 1$ (Yang, n.d.). To extract the principal components (corresponding to the kernel k) of a test point $\mathbf{x}$, we then compute projections onto the eigenvectors by (Equation 2.37, Figure 2.7):

$$\mathbf{e}^{\Phi} \bullet \Phi(\mathbf{x}) = \sum_{i=1}^{n} \alpha_i k(\mathbf{x}_i, \mathbf{x}) \tag{2.38}$$

If we use a kernel satisfying Mercer's conditions, we know that this procedure exactly corresponds to standard PCA in some high-dimensional feature space, except that we do not need to perform expensive computations in that space.

Properties of KPCA

For Mercer kernels, we know that we are in fact doing a standard PCA in F. Consequently, all mathematical and statistical properties of PCA carry over to KPCA, with the modifications that they become statements about a set of points $\Phi(\mathbf{x}_i)$, $i = 1, \ldots, n$, in $\mathcal{F}$ rather than in $\mathbb{R}^N$ (Zwald, Bousquet, & Blanchard, n.d.; Schölkopf, Smola, & Müller, 1998a, 1998b). In $\mathcal{F}$, we can thus assert that PCA is the orthogonal basis transformation with the following properties (assuming that the eigenvectors are sorted in descending order of the eigenvalue size):

- The first q ($q \in [1, n]$) principal components — that is, projections on eigenvectors — carry more variance than any other q orthogonal directions
- The mean-squared approximation error in representing the observations by the first q principal components is minimal
- The principal components are uncorrelated
- The first q principal components have maximal mutual information with respect to the inputs (this holds under Gaussianity assumptions, and thus depends on the particular kernel chosen and on the data)

Figure 2.6 shows that in some high-dimensional feature space $\mathcal{F}$ (bottom right), we are performing linear PCA, just as a PCA in input space (top). Since $\mathcal{F}$ is nonlinearly related to input space (via Φ), the contour lines of constant projections onto the principal eigenvector (drawn as an arrow) become nonlinear in input space. Note that we cannot draw a pre-image of the eigenvector in input space, as it may not even exist. Crucial to KPCA is the fact that there is no need to perform the map into $\mathcal{F}$: all necessary computations are carried out by the use of a kernel function k in input space (here: $\mathbb{R}^2$) (Schölkopf, Smola, & Müller, 1998b).

To translate these properties of PCA in $\mathcal{F}$ into statements about the data in input space, they must be investigated for specific choices of kernels.

Figure 2.6. The basic idea of KPCA

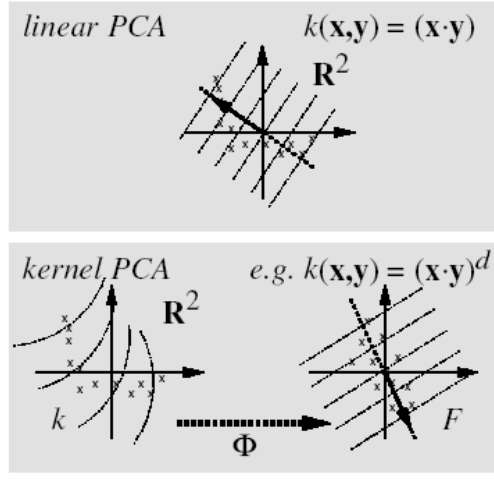
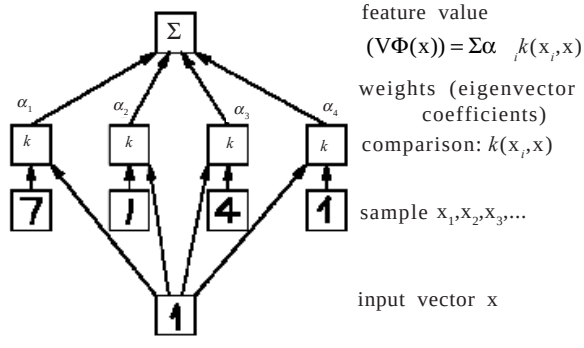


Figure 2.7. Feature extractor constructed by KPCA



We conclude this section with a characterization of KPCA with polynomial kernels. It was explained how using polynomial kernels $(\mathbf{x},\mathbf{y})^d$ corresponds to mapping into a feature space whose dimensions are spanned by all possible *d*-th order monomials in input coordinates. The different dimensions are scaled with the square root of the number of ordered products of the respective *d* pixels. These scaling factors precisely ensure invariance of KPCA under the group of all orthogonal transformations (rotations and mirroring operations). This is a desirable property: It ensures that the features extracted do not depend on which orthonormal coordinate system we use for representing our input data.

Theorem 2.1 (Invariance of Polynomial Kernels). Up to a scaling factor, kernel PCA with $k(\mathbf{x},\mathbf{y}) = (\mathbf{x},\mathbf{y})^d$ *is the only PCA in a space of all monomials of degree d that is invariant under orthogonal transformations of input space.*

This means that even if we could compute all monomials of degree p for the data at hand and perform PCA on the monomials, with the additional requirement of not implying any preferred directions, we would obtain multiples of the results generated by KPCA.

In the first layer, the input vector is compared to the sample via a kernel function, chosen a priori (e.g., polynomial, Gaussian or sigmoid). The outputs are then linearly combined using weights found by solving an eigenvector problem. As shown in the text, the depicted network's function can be thought of as the projection onto an eigenvector of a covariance matrix in a high-dimensional feature space. As a function on input space, it is nonlinear (Schölkopf, Smola, & Müller, 1998).

SUMMARY

The principal component analysis or *Karhunen-Loeve transform* is a mathematical way of determining the linear transformation of a sample of points in d-dimensional space. PCA is a linear procedure for finding the direction in input space where most of the energy of the input lies. In other words, PCA performs feature extraction. The projections of these components correspond to the eigenvalues of the input covariance matrix.

PCA is a well-known method for orthogonalizing data. It converges very fast and the theoretical method is well understood. There are usually fewer features extracted than there are inputs, so the unsupervised segment provides a means of data reduction. The main use of PCA is to reduce the dimensionality of a data set while retaining as much information as possible. It computes a compact and optimal description of the data set.

KPCA is a nonlinear generalization of PCA in the sense that it is performing PCA in feature spaces of arbitrarily large (possibly infinite) dimensionality, and if we use the kernel $k(\mathbf{x},\mathbf{y}) = (\mathbf{x},\mathbf{y})$, we recover the original PCA algorithm. Compared to the above approaches, KPCA has the main advantage that no nonlinear optimization is involved — it is essentially linear algebra, as simple as standard PCA. In addition, we need not specify in advance the number of components we want to extract. Compared to neural approaches, KPCA could be disadvantageous if we need to process a very large number of observations, as this results in a large matrix K. Compared to principal curves, KPCA is much harder to interpret in input space; however, at least for polynomial kernels, it has a very clear interpretation in terms of higher-order features.

REFERENCES

Anderson, T. W. (1963). A symptotic theory for principal component analysis. *Annals of Mathematical Statistics, 34*, 122-148.

Boser, B. E., Guyon, I.M., & Vapnik, V. N. (1992). A training algorithm for optimal margin classifiers. In D. Haussler (Ed.), *Proceedings of the 5th Annual ACM Workshop on Computational Learning Theory* (pp. 144-152). New York: ACM Press.

Chellappa, R. W., C. L., & Sirohey, S. (1995). Human and machine recognition of faces: A survey. *Proceedings of the IEEE* (Vol. 83, pp. 705-140).

Cottrell, G. W., & Fleming, M. K. (1990). Face recognition using unsupervised feature extraction. *Proceedings of the International Neural Network Conference* (pp. 322-325).

Diamantaras, K. I., & Kung, S. Y. (1996). *Principal component neural networks*. New York: Wiley.

Duda, R. O., Hart, P. E., & Stork, D. G. (2000). *Pattern classification* (2nd ed.). New York: John Wiley Press.

Gonzalez, R. C., & Wintz, P. (1987). *Digital image processing*. MA: Addison-Wesley.

Grudin, M. A. (2000). On internal representations in face recognition systems. *Pattern Recognition*, *33*(7), 1161-1177.

Huang, M-H. (2002). Kernel eigenfaces vs. kernel fisherfaces: Face recognition using kernel methods. *Proceedings of the Fifth IEEE International Conference on Automatic Face and Gesture Recognition* (pp. 215-220).

Joliffe, I. T. (1986). *Principal component analysis. Springer series in statistics*. New York: Springer Verlag.

Kearns, M. S., Solla, S. A., & Cohn, D. A. (Eds.). (1999). *Advances in neural information processing systems* (pp. 536-542). Cambridge, MA: MIT Press.

Kim, K. (n.d.). *Face recognition using principal component analysis.* College Park: Department of Computer Science, University of Maryland.

Kim, K. I., Jung, K., & Kim, H. J. (2002). Face recognition using kernel principal component analysis. *IEEE Signal Processing Let.*, *9*, 40-42.

Kirby, M., & Sirovich, L. (1990). Application of the KL Procedure for the characterization of human faces. *IEEE Transactions on Pattern Analysis and Machine Intelligence, 12*(1), 103-108.

Liu, Q. S., Lu, H. Q., & Ma, S. D. (2004). Improving kernel Fisher discriminant analysis for face recognition. *IEEE Transactions on Circuits and Systems for Video Technology, 14*(1).

Penev, P. S., & Sirovich, L. (2000). The global dimensionality of face space. *Proceedings of the Fourth IEEE International Conference on Automatic Face and Gesture Recognition* (pp. 264-270).

Pentland, A. (2000). Looking at people: Sensing for ubiquitous and wearable computing. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, *22*(1), 107-119.

Romdhani, S., Gong, S., & Psarrou, A. (1999). A multi-view nonlinear active shape model using kernel PCA. In T. Pridmore & D. Elliman (Eds.), *Proceedings of the 10th British Machine Vision Conference (BMVC99)* (pp. 483-492). London: BMVA Press.

Rosipal, R., Girolami, M., & Trejo, L. J. (2000). *Kernel PCA for feature extraction and denoising in non-linear regression.* Technical Report No. 4, Department of Computing and Information Systems. UK: University of Paisley.

Schölkopf, B., Burges, C. J. C., & Smola, A. J. (Eds.). (1999). *Advances in kernel methods – support vector learning* (pp. 327-352). Cambridge, MA: MIT Press.

Schölkopf, B., Mika, S., Burges, C. J. C., Knirsch, P., Müller, K-R., Ratsch, G., & Smola, A. (1999). Input space versus feature space in kernel-based methods. *IEEE Transactions on Neural Networks*, *10*(5), 1000-1017.

Schölkopf, B., & Smola, A. (2002). *Learning with kernels: Support vector machines, regularization, optimization and beyond*. Cambridge, MA: MIT Press.

Schölkopf, B., Smola, A., & Müller, K-R. (1998). Nonlinear component analysis as a kernel eigenvalue problem. *Neural Computation, 10*, 1299-1319.

Schölkopf, B., Smola, A. J., & Müller, K. R. (1998). Kernel principal component analysis. *Neural Computation, MIT Press, 10*(5), 1299-1319.

Schölkopf, B., Smola, A., & Müller, K-R. (1999). Kernel principal component analysis. In B. Schölkopf, C. J. C. Burges, & A. Smola (Eds.), *Advances in kernel methods – Support vector learning* (pp. 327-352). Cambridge, MA: MIT Press.

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

Smith, L. I. (2002). *A tutorial on principal components analysis.* Retrieved from http://www.cs.otago.ac.nz/cosc453/student.tutorials/principal_component.pdf

Turk, M., & Pentland, A. (1991). Eigenfaces for recognition. *Journal of Cognitive Neuroscience, 3*(1).

Valentin, D., Abdi, J., O'Toole, A. J., & Cottrell, G. W. (1994). Connectionist models of face processing: A survey. *Pattern Recognition*, *27*(9), 1209-1230.

Vapnik, V. (1995). *The nature of statistical learning theory*. New York: Springer Verlag.

Wiskott, L., Fellous, J. M., Krüger, N., & von der Malsburg, C. (1997). Face recognition by elastic bunch graph matching. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, *19*(7), 775-779.

Yang, J., Zhang, D., Frangi, A. F., & Yang, J-Y. (2004). Two-dimensional PCA: A new approach to appearance-based face representation and recognition. *IEEE Transactions on Pattern Analysis and Machine Intelligence, 26*(1),131-137.

Yang, M-H. (2002). Face recognition using kernel methods. In T. G. Dietterich, S. Becker, & Z. Ghahramani (Eds.), *Advances in neural information processing systems* (p. 14). Cambridge, MA: MIT Press.

Yang, M-H., Ahuja, N., & Kriegman, D. (2000). Face recognition using kernel eigenfaces. *Proceedings of the 2000 IEEE International Conference on Image Processing* (Vol. 1, pp. 37-40). Vancouver, Canada.

Zhang, J. (1997). Face recognition: Eigenface, elastic matching, and neural nets. *Proceedings of the IEEE, 85*(9).

Zhao, L., & Yang, Y. (1999). Theoretical analysis of illumination in PCA-based vision systems. *Pattern Recognition*, *32*(4), 547-564.

Zhao, W. Y., Krishnaswamy, A., Chellappa, R., Swets, D. L., & Weng, J. (1998). Discriminant analysis of principal components for face recognition. *Proceedings of International Conference on Automatic Face and Gesture Recognition* (pp. 336-341).

Zwald, L., Bousquet, O., & Blanchard, G. (2004). *Statistical properties of kernel principal component analysis.* The 17th Annual Conference on Learning Theory (COLT'04), Alberta, Canada.

Chapter III

Linear Discriminant Analysis

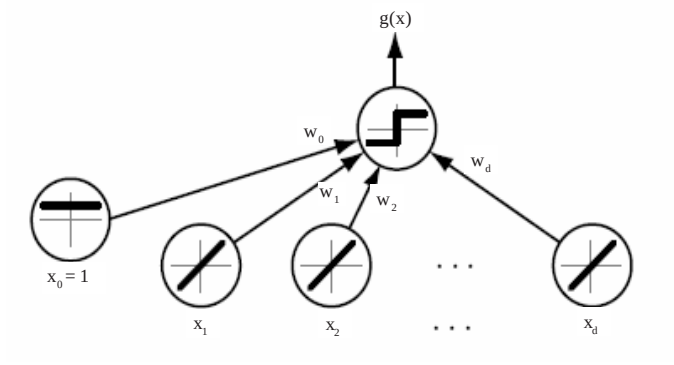
ABSTRACT

This chapter deals with issues related to linear discriminant analysis (LDA). In the introduction, we indicate some basic conceptions of LDA. Then, the definitions and notations related to LDA are discussed. Finally, the introduction to non-linear LDA and the chapter summary are given.

INTRODUCTION

Although PCA finds components useful for representing data, there is no reason to assume these components must be useful for discriminating between data in different classes. As was said in Duda, Hart and Stork (2000), if we pool all of the samples, the directions that are discarded by PCA might be exactly the directions needed for distinguishing between classes. For example, if we had data for the printed uppercase letters O and Q, PCA might discover the gross features that characterize Os and Qs, but might ignore the tail that distinguishes an O from a Q. Whereas PCA seeks directions that are efficient for representation, *discriminant analysis* seeks directions that are efficient for discrimination (McLachlan, 1992; Chen, Liao, Ko, Lin, & Yu, 2000; Hastie, Buja, & Tibshirani, 1995). In the previous chapter we introduced algebraic considerations for dimensionality reduction which preserve variance. We can see that variance preserving dimensionality reduction is equivalent to (1) de-correlating the training sample data, and (2) seeking the d-dimensional subspace of $\mathbb{R}^n$ that is the closest (in the least-squares

Figure 3.1. A simple linear classifier having d input units, each corresponding to the values of the components of an input vector.



Each input feature value x_i is multiplied by its corresponding weight w_i; the output unit sums all these products and emits a +1 if $\mathbf{w}^T\mathbf{x} + w_0 > 0$ or a 1 otherwise (Duda, Hart, & Stork, 2000)

sense) possible to the original training sample. In this chapter, we extend the variance preserving approach for data representation for labeled data sets. In this section, we will focus on two-class sets and look for a separating hyperplane (Yang & Yang, 2001; Xu, Yang, & Jin, 2004; Etemad & Chellappa, 1997):

$$g(\mathbf{x}) = \mathbf{w}^T\mathbf{x} + w_0 \tag{3.1}$$

such that **x** belongs to the first class if $g(\mathbf{x}) > 0$ and **x** belongs to the second class if $g(\mathbf{x}) < 0$. In the statistical literature, this type of function is called a linear discriminant function. The decision boundary is given by the set of points satisfying $g(\mathbf{x}) = 0$ which is a hyperplane. Fisher's LDA is a variance preserving approach for finding a linear discriminant function (Duda, Hart, & Stork, 2000; McLachlan, 1992; Chen, Liao, Ko, Lin, & Yu, 2000).

The Two-Category Case

A discriminant function that is a linear combination of the components of **x** can be written as Equation 3.1, where **w** is the *weight vector* and w_0 the *bias* or *threshold weight*. A two-category threshold weight linear classifier implements the following decision rule: Decide ω_1 if $g(\mathbf{x}) > 0$ and ω_2 if $g(\mathbf{x}) < 0$. Thus, **x** is assigned to ω_1 if the inner product $\mathbf{w}^T\mathbf{x}$ exceeds the threshold $-w_0$ and ω_2 otherwise. If $g(\mathbf{x}) = 0$, **x** can ordinarily be assigned to either class, but in this chapter we shall leave the assignment undefined. Figure 3.1 shows a typical implementation, a clear example of the general structure of a pattern recognition system we saw (Duda, Hart, & Stork, 2000).

The equation $g(\mathbf{x}) = 0$ defines the decision surface that separates points assigned to ω_1 from points assigned to ω_2. When $g(\mathbf{x})$ is linear, this decision surface is a *hyperplane* (Burges, 1996; Evgeniou, Pontil, & Poggio, 1999; Ripley, 1994; Suykens & Vandewalle,

1999; Van Gestel, Suykens, & De Brabanter, 2001). If $\mathbf{x}_1$ and $\mathbf{x}_2$ are both on the decision surface, then:

$$\mathbf{w}^T\mathbf{x}_1 + w_0 = \mathbf{w}^T\mathbf{x}_2 + w_0 \tag{3.2}$$

or:

$$\mathbf{w}^T(\mathbf{x}_1 \bullet \mathbf{x}_2) = 0 \tag{3.3}$$

and this shows that **w** is normal to any vector lying in the hyperplane. In general, the hyperplane *H* divides the feature space into two halfspaces, decision region R_1 for ω_1 and region R_2 for ω_2. Since $g(\mathbf{x}) > 0$ if **x** is in R_1, it follows that the normal vector **w** points into R_1. It is sometimes said that any **x** in R_1 is on the *positive* side of *H*, and any **x** in R_2 is on the *negative* side.

The discriminant function g(**x**) gives an algebraic measure of the distance from **x** to the hyperplane. Perhaps the easiest way to see this is to express **x** as:

$$\mathbf{x} = \mathbf{x}_p + r\frac{\mathbf{w}}{\|\mathbf{w}\|} \tag{3.4}$$

where $\mathbf{x}_p$ is the normal projection of **x** onto *H*, and *r* is the desired algebraic distance — positive if **x** is on the positive side and negative if **x** is on the negative side. Then, since $g(\mathbf{x}_p) = 0$, we have:

$$g(\mathbf{x}) = \mathbf{w}^T\mathbf{x} + w_0 = r\|\mathbf{w}\| \tag{3.5}$$

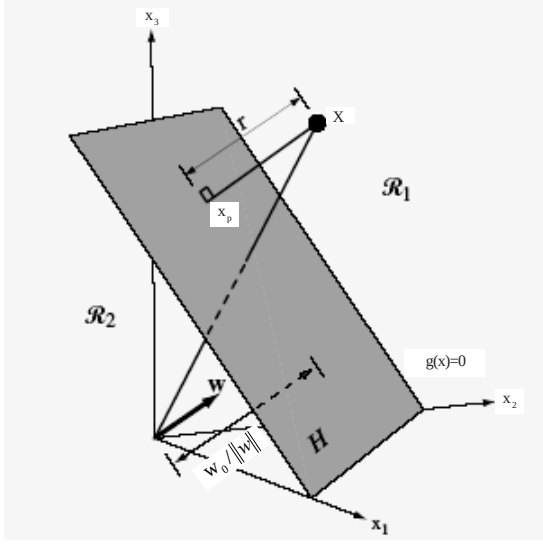
or:

$$r = \frac{g(\mathbf{x})}{\|\mathbf{w}\|} \tag{3.6}$$

In particular, the distance from the origin to *H* is given by $w_0/\|\mathbf{w}\|$. If $w_0 > 0$ the origin is on the positive side of *H*, and if $w_0 < 0$ it is on the negative side. If $w_0 = 0$, then $g(\mathbf{x})$ has the homogeneous form $\mathbf{w}^T\mathbf{x}$, and the hyperplane passes through the origin. A geometric illustration of these algebraic results is given in Figure 3.2.

To summarize, a linear discriminant function divides the feature space by a hyperplane decision surface. The orientation of the surface is determined by the normal vector **w**, and the location of the surface is determined by the bias w_0. The discriminant function $g(\mathbf{x})$ is proportional to the signed distance from **x** to the hyperplane, with $g(\mathbf{x}) > 0$ when **x** is on the positive side, and $g(\mathbf{x}) < 0$ when **x** is on the negative side (McLachlan, 1992; Chen, Liao, Ko, Lin, & Yu, 2000; Hastie, Buja, & Tibshirani, 1995; Mika, Ratsch, & Müller, 2001; Yang & Yang, 2001, 2003; Xu, Yang, & Jin, 2004).

Figure 3.2. The linear decision boundary H, where $g(\mathbf{x}) = \mathbf{w}^T\mathbf{x} + w_0 = 0$, separates the feature space into two halfspaces, $\mathbf{R}_1$ (where $g(\mathbf{x}) > 0$) and $\mathbf{R}_2$ (where $g(\mathbf{x}) < 0$) (Duda, Hart, & Stork, 2000)



The Multicategory Case

There is more than one way to devise multicategory classifiers employing linear discriminant functions. For example, we might reduce the problem to $c - 1$ two-class problems, where the ith problem is solved by a linear discriminant function that separates points assigned to ω_i from those not assigned to ω_i. A more extravagant approach would be to use $c(c - 1)/2$ linear discriminants, one for every pair of classes. As illustrated in Figure 3.3, both of these approaches can lead to regions in which the classification is undefined. We shall avoid this problem by adopting the approach defining c linear discriminant functions (Burges, 1996; Evgeniou, Pontil, & Poggio, 1999; Ripley, 1994; Suykens & Vandewalle, 1999; Zheng, Zhao, & Zou, 2002):

$$g_i(\mathbf{x}) = \mathbf{w}^{\mathrm{T}}\mathbf{x}_i + w_{i0} \qquad i = 1, \ldots, c \tag{3.7}$$

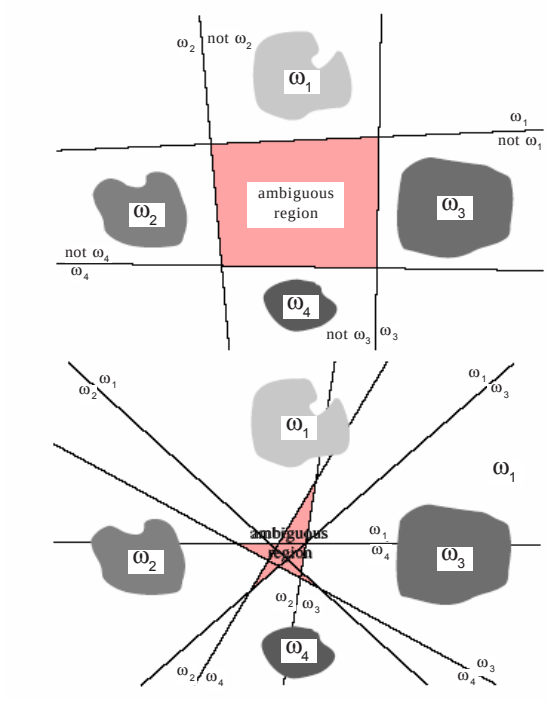
and assigning $\mathbf{x}$ to ω_i if $g_i(\mathbf{x}) > g_j(\mathbf{x})$ for all $j \neq i$; in case of ties, the classification is left undefined. The resulting classifier is called a *linear machine*. A linear machine divides the feature space into c decision regions, with $g_i(\mathbf{x})$ being the largest discriminant if $\mathbf{x}$ is in region R_i. If R_i and R_j are contiguous, the boundary between them is a portion of the hyperplane H_{ij} defined by:

$$g_i(\mathbf{x}) = g_j(\mathbf{x}) \tag{3.8}$$

or:

$$(\mathbf{w}_i - \mathbf{w}_j)^{\mathrm{T}}\mathbf{x} + (w_{i0} - w_{j0}) = 0 \tag{3.9}$$

Figure 3.3. Linear decision boundaries for a four-class problem



The top figure shows w_i/not w_i dichotomies while the bottom figure shows w_i/w_j dichotomies. The pink regions have ambiguous category assignments (Duda, Hart, & Stork, 2000)

It follows at once that $\mathbf{w}_i - \mathbf{w}_j$ is normal to H_{ij}, and the signed distance from $\mathbf{x}$ to H_{ij} is given by $(g_i - g_j)/||\mathbf{w}_i - \mathbf{w}_j||$. Thus, with the linear machine it is not the weight vectors themselves but their *differences* that are important. While there are $c(c-1)/2$ pairs of regions, they need not all be contiguous, and the total number of hyperplane segments appearing in the decision surfaces is often fewer than $c(c-1)/2$, as shown in Figure 3.4.

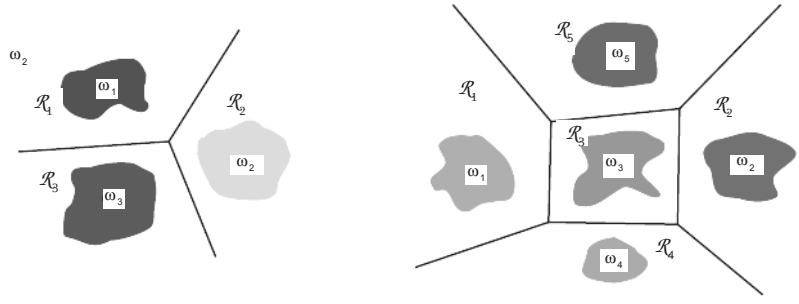
It is easy to show that the decision regions for a linear machine are convex and this restriction surely limits the flexibility and accuracy of the classifier. In particular, for good performance every decision region should be singly connected, and this tends to make the linear machine most suitable for problems for which the conditional densities $p(\mathbf{x}|\omega_i)$ are unimodal.

Generalized Linear Discriminant Functions

The linear discriminant function $g(\mathbf{x})$ can be written as:

$$g(\mathbf{x}) = w_0 + \sum_{i=1}^{d} w_i x_i \tag{3.10}$$

Figure 3.4. Decision boundaries produced by a linear machine for a three-class problem and a five-class problem (Duda, Hart, & Stork, 2000)



where the coefficients w_i are the components of the weight vector $\mathbf{w}$. By adding additional terms involving the products of pairs of components of $\mathbf{x}$, we obtain the *quadratic discriminant function* (Burges, 1996; Friedman, 1989; Baudat & Anouar, 2000; Mika, Ratsch, & Müller, 2001):

$$g(\mathbf{x}) = w_0 + \sum_{i=1}^{d} w_i x_i + \sum_{i=1}^{d}\sum_{j=1}^{d} w_{ij} x_i x_j \tag{3.11}$$

Since $x_i x_j = x_j x_i$, we can assume that $w_{ij} = w_{ji}$ with no loss in generality. Thus, the quadratic discriminant function has additional $d(d+1)/2$ coefficients at its disposal with which to produce more complicated separating surfaces. The separating surface defined by $g(\mathbf{x}) = 0$ is a second-degree, or *hyperquadric,* surface. The linear terms in $g(\mathbf{x})$ can be eliminated by translating the axes. We can define $\mathbf{W} = [w_{ij}]$, a symmetric, nonsingular matrix and then the basic character of the separating surface can be described in terms of the scaled matrix $\overline{\mathbf{W}} = \mathbf{W}/(\mathbf{w}^T\mathbf{W}^{-1}\mathbf{w} - 4w_0)$. If $\overline{\mathbf{W}}$ is a positive multiple of the identity matrix, the separating surface is a *hypersphere*. If $\overline{\mathbf{W}}$ is positive definite, the separating surfaces is a *hyperellipsoid*. If some of the eigenvalues of $\overline{\mathbf{W}}$ are positive and others are negative, the surface is one of the varieties of types of *hyperboloids*. As observed in Chapter II, these are the kinds of separating surfaces that arise in the general multivariate Gaussian case.

By continuing to add terms such as $w_{ijk} x_i x_j x_k$, we can obtain the class of *polynomial discriminant functions*. These can be thought of as truncated series expansions of some arbitrary $g(\mathbf{x})$, and this in turn suggests the *generalized linear discriminant function*:

$$g(\mathbf{x}) = \sum_{i=1}^{\hat{d}} a_i y_i(\mathbf{x}) \tag{3.12}$$

or:

$$g(\mathbf{x}) = \mathbf{a}^T\mathbf{y} \tag{3.13}$$

where **a** is a $\hat{d}$-dimensional weight vector, and where the $\hat{d}$ functions $y_i(\mathbf{x})$ — sometimes called φ functions — can be arbitrary functions of **x**. Such functions might be computed by a feature detecting subsystem. By selecting these functions judiciously and letting $\hat{d}$ be sufficiently large, one can approximate any desired discriminant function by such an expansion. The resulting discriminant function is not linear in **x**, but it is linear in **y**. The $\hat{d}$ functions $y_i(\mathbf{x})$ merely map points in d-dimenional **x**-space to points in $\hat{d}$-dimensional **y**-space. The homogeneous discriminant $\mathbf{a}^T\mathbf{y}$ separates points in this transformed space by a hyperplane passing through the origin. Thus, the mapping from **x** to **y** reduces the problem to one of finding a homogeneous linear discriminant function.

Unfortunately, the curse of dimensionality often makes it hard in practice to capitalize on this flexibility. A complete quadratic discriminant function involves $\hat{d} = (d + 1)(d + 2)/2$ terms. If d is modestly large, say $d = 50$, this requires the computation of a great many terms. The inclusion of cubic and higher orders leads to $O(\hat{d}^3)$ terms. Furthermore, the $\hat{d}$ components of the weight vector **a** must be determined from training samples. If we think of $\hat{d}$ as specifying the number of degrees of freedom for the discriminant function, it is natural to require that the number of samples be not less than the number of degrees of freedom. Clearly, a general series expansion of $g(\mathbf{x})$ can easily lead to completely unrealistic requirements for computation and data. We saw in previous sections that this drawback, however, could be accommodated by imposing a constraint of large margins, or bands between the training patterns. In this case, we are not technically speaking fitting all the free parameters. Rather, we are relying on the assumption that the mapping to a high-dimensional space does not impose any spurious structure or relationships among the training points. Alternatively, multilayer neural networks approach this problem by employing multiple copies of a single nonlinear function of the input features, as was shown in Chapter II.

While it may be hard to realize the potential benefits of a generalized linear discriminant function, we can at least exploit the convenience of being able to write $g(\mathbf{x})$ in the homogeneous form $\mathbf{a}^T\mathbf{y}$. In the particular case of the linear discriminant function:

$$g(\mathbf{x}) = w_0 + \sum_{i=1}^{d} w_i x_i = \sum_{i=0}^{d} w_i x_i \tag{3.14}$$

where we set $x_0 = 1$. Thus, we can write:

$$\mathbf{y} = \begin{bmatrix} 1 \\ x_1 \\ \vdots \\ x_d \end{bmatrix} = \begin{bmatrix} 1 \\ \mathbf{x} \end{bmatrix} \tag{3.15}$$

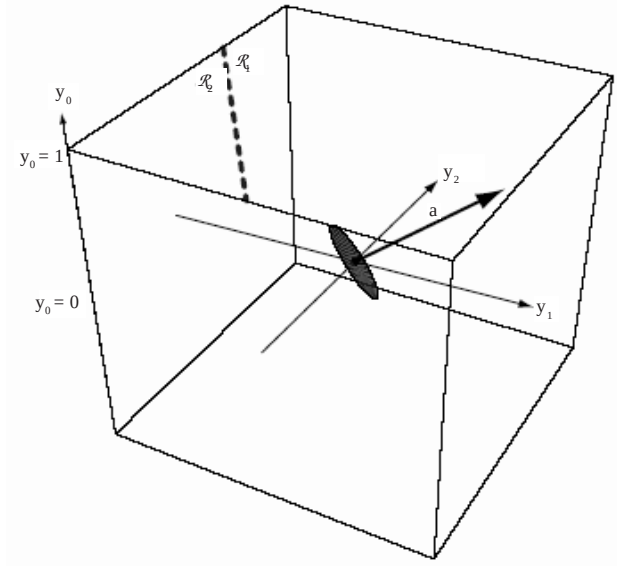
and **y** is sometimes called an *augmented feature vector*. Likewise, an *augmented weight vector* can be written as:

$$\mathbf{a} = \begin{bmatrix} w_0 \\ w_1 \\ \vdots \\ w_d \end{bmatrix} = \begin{bmatrix} w_0 \\ \mathbf{w} \end{bmatrix} \tag{3.16}$$

This mapping from d-dimensional **x**-space to (d+1)-dimensional **y**-space is mathematically trivial but quite convenient. The addition of a constant component to **x** preserves all the distance relationships between samples. The resulting **y** vectors all lie in a d-dimensional subspace, which is the **x**-space itself. The hyperplane decision surface $\hat{H}$ defined by $\mathbf{a}^T\mathbf{y} = 0$ passes through the origin in **y**-space, even though the corresponding hyperplane H can be in any position in **x**-space. The distance from **y** to $\hat{H}$ is given by $|\mathbf{a}^T\mathbf{y}|/||\mathbf{a}||$, or $|g(\mathbf{x})|/||\mathbf{a}||$. Since $||\mathbf{a}|| > ||\mathbf{w}||$, this distance is less than, or at most equal to, the distance from **x** to H. By using this mapping, we reduce the problem of finding a weight vector **w** and a threshold weight w_0 to the problem of finding a single weight vector **a** (Fig. 3.5).

Figure 3.5 shows the set of points for which $\mathbf{a}^T\mathbf{y} = 0$ is a plane (or more generally, a hyperplane) perpendicular to **a** and passing through the origin of **y** space, as indicated by the red disk. Of course, such a plane need not pass through the origin of the two-dimensional **x**-space at the top, as shown by the dashed line. Thus there exists an augmented weight vector **a** that will lead to any straight decision line in **x**-space (Duda, Hart, & Stork, 2000).

*Figure 3.5. A three-dimensional augmented feature space **y** and augmented weight vector **a** (at the origin)*



LDA DEFINITIONS

Fisher Linear Discriminant

One of the recurring problems encountered in applying statistical techniques to pattern recognition problems has been called the "curse of dimensionality." Procedures that are analytically or computationally manageable in low-dimensional spaces can become completely impractical in a space of 50 or 100 dimensions. Pure fuzzy methods are particularly ill-suited to such high-dimensional problems, since it is implausible that the designer's linguistic intuition extends to such spaces. Thus, various techniques have been developed for reducing the dimensionality of the feature space in the hope of obtaining a more manageable problem.

We can reduce the dimensionality from *d* dimensions to one dimension if we merely project the *d*-dimensional data onto a line. However, by moving the line around, we might find an orientation for which the projected samples are well separated. This is exactly the goal of classical discriminant analysis (Duda, Hart, & Stork, 2000; McLachlan, 1992; Chen, Liao, Ko, Lin, & Yu, 2000; Friedman, 1989; Hastie, Tibshirani, & Buja, 1994).

Suppose that we have a set of *n d*-dimensional samples $\mathbf{x}_1,\ldots,\mathbf{x}_n$, n_1 in the subset D_1 labeled ω_1 and n_2 in the subset D_2 labeled ω_2. If we form a linear combination of the components of $\mathbf{x}$, we obtain the scalar dot product:

$$y = \mathbf{w}^T\mathbf{x} \tag{3.17}$$

and a corresponding set of *n* samples $y_1, \ldots, y_n$ divided into the subsets Y_1 and Y_2. Geometrically, if $\mathbf{w} = 1$, each y_i is the projection of the corresponding $\mathbf{x}_i$ onto a line in the direction of $\mathbf{w}$. Actually, the magnitude of $\mathbf{w}$ is of no real significance, since it merely scales *y*. The direction of $\mathbf{w}$ is important, however. If we imagine that the samples labeled ω_1 fall more or less into one cluster while those labeled ω_2 fall in another, we want the projections falling onto the line to be well separated, not thoroughly intermingled. Figure 3.6 illustrates the effect of choosing two different values for $\mathbf{w}$ for a two-dimensional example. It should be abundantly clear that if the original distributions are multimodal and highly overlapping, even the "best" $\mathbf{w}$ is unlikely to provide adequate separation, and thus, this method will be of little use.

We now turn to the matter of finding the best such direction $\mathbf{w}$, and one we hope will enable accurate classification. A measure of the separation between the projected points is the difference of the sample means. If $\mathbf{m}_i$ is the *d*-dimensional sample mean given by:

$$\mathbf{m}_i = \frac{1}{n_i}\sum_{\mathbf{x}\ D_i}\mathbf{x} \tag{3.18}$$

then the sample mean for the projected points is given by:

$$\tilde{m}_i = \frac{1}{n_i}\sum_{y\in Y_i} y \tag{3.19}$$

$$\frac{1}{n_i}\sum_{y\in Y_i} y = \frac{1}{n_i}\sum_{\mathbf{x}\ D_i} \mathbf{w}^{\mathrm{T}}\mathbf{x} \tag{3.20}$$

$$\frac{1}{n_i}\sum_{\mathbf{x}\ D_i} \mathbf{w}^{\mathrm{T}}\mathbf{x} = \mathbf{w}^{\mathrm{T}}\mathbf{m}_i \tag{3.21}$$

and is simply the projection of $\mathbf{m}_i$.

It follows that the distance between the projected means is:

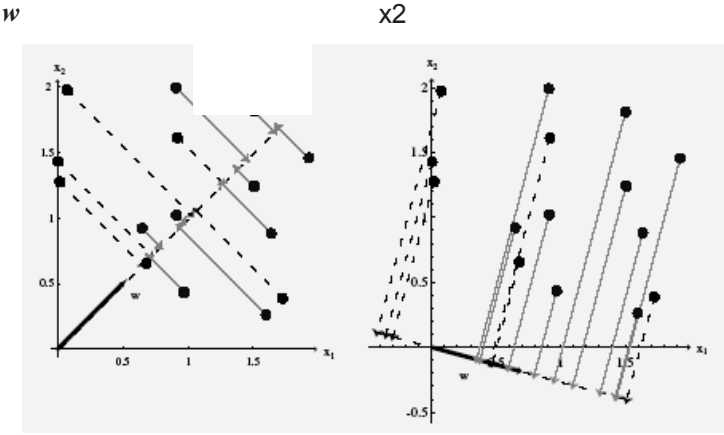
$$|\tilde{m}_1 - \tilde{m}_2| = |\mathbf{w}^{\mathrm{T}}(\mathbf{m_1} - \mathbf{m_2})| \tag{3.22}$$

and that we can make this difference as large as we wish merely by scaling $\sum_i x_i$. Of course, to obtain good separation of the projected data, we really want the difference between the means to be large relative to some measure of the standard deviations for each class. Rather than forming sample variances, we define the *scatter* for projected samples labeled ω_i by:

$$\tilde{s}_i^2 = \sum_{y\in Y_i} (y - \tilde{m}_i)^2 \tag{3.23}$$

Thus, $(1/n)(\tilde{s}_1^2 + \tilde{s}_2^2)$ is an estimate of the variance of the pooled data, and $\tilde{s}_1^2 + \tilde{s}_2^2$ is called the total *within-class scatter* of the projected samples. The *Fisher linear discriminant* employs linear function $\mathbf{w}^T\mathbf{x}$ for which the criterion function:

Figure 3.6. Projection of the same set of samples onto two different lines in the directions marked ***w***



The figure on the right shows greater separation between the red and black projected points (Duda, Hart, & Stork, 2000)

$$\boldsymbol{J}(\mathbf{w}) = \frac{|\tilde{m}_1 - \tilde{m}_2|^2}{\tilde{s}_1^2 + \tilde{s}_2^2} \tag{3.24}$$

is maximum (and independent of $\mathbf{w}$). While the maximizing $\boldsymbol{J}(\cdot)$ leads to the best separation between the two projected sets (in the sense just described), we will also need a threshold criterion before we have a true classifier. We first consider how to find the optimal $\mathbf{w}$, and later turn to the issue of thresholds.

To obtain $\boldsymbol{J}(\cdot)$ as an explicit function of $\mathbf{w}$, we define the *scatter matrices* $\mathbf{S}_i$ and scatter matrices $\mathbf{S}_w$ by:

$$\mathbf{S}_i = \sum_{\mathbf{x} \in D_i} (\mathbf{x} - \mathbf{m}_i)(\mathbf{x} - \mathbf{m}_i)^{\mathrm{T}} \tag{3.25}$$

and:

$$\mathbf{S}_w = \mathbf{S}_1 + \mathbf{S}_2 \tag{3.26}$$

Then we can write:

$$\begin{aligned} \tilde{s}_i^2 &= \sum_{\mathbf{x} \in D_i} (\mathbf{w}^{\mathrm{T}}\mathbf{x} - \mathbf{w}^{\mathrm{T}}\mathbf{m}_i)^2 \\ &= \sum_{\mathbf{x} \in D_i} \mathbf{w}^{\mathrm{T}}(\mathbf{x} - \mathbf{m}_i)(\mathbf{x} - \mathbf{m}_i)^{\mathrm{T}}\mathbf{w} \\ &= \mathbf{w}^{\mathrm{T}}\mathbf{S}_i\mathbf{w} \end{aligned} \tag{3.27}$$

Therefore, the sum of these scatters can be written:

$$\tilde{s}_1^2 + \tilde{s}_2^2 = \mathbf{w}^{\mathrm{T}}\mathbf{S}_w\mathbf{w} \tag{3.28}$$

Similarly, the separation of the projected means obeys:

$$\begin{aligned} (\tilde{m}_1 - \tilde{m}_2)^2 &= (\mathbf{w}^{\mathrm{T}}\mathbf{m}_1 - \mathbf{w}^{\mathrm{T}}\mathbf{m}_2) \\ &= \mathbf{w}^{\mathrm{T}}(\mathbf{m}_1 - \mathbf{m}_2)(\mathbf{m}_1 - \mathbf{m}_2)^{\mathrm{T}}\mathbf{w} \\ &= \mathbf{w}^{\mathrm{T}}\mathbf{S}_b\mathbf{w} \end{aligned} \tag{3.29}$$

where:

$$\mathbf{S}_b = (\mathbf{m}_1 - \mathbf{m}_2)(\mathbf{m}_1 - \mathbf{m}_2)^{\mathrm{T}} \tag{3.30}$$

We call $\mathbf{S}_w$ the *within-class scatter matrix*. It is proportional to the sample covariance matrix for the pooled *d*-dimensional data. It is symmetric and positive semi-definite, and is usually nonsingular if $n > d$. Likewise, $\mathbf{S}_b$ is called the *between-class scatter matrix*. It is also symmetric and positive semidefinite, but because it is the outer

product of two vectors, its rank is at most one. In particular, for any $\mathbf{w}$, $\mathbf{S}_b\mathbf{w}$ is in the direction of $\mathbf{m}_1 - \mathbf{m}_2$, and $\mathbf{S}_b$ is quite singular.

In terms of $\mathbf{S}_b$ and $\mathbf{S}_w$, the criterion function $\boldsymbol{J}(\cdot)$ can be written as:

$$\boldsymbol{J}(\mathbf{w}) = \frac{\mathbf{w}^{\mathrm{T}}\mathbf{S}_b\mathbf{w}}{\mathbf{w}^{\mathrm{T}}\mathbf{S}_w\mathbf{w}} \tag{3.31}$$

This expression is well known in mathematical physics as the generalized Rayleigh quotient. It is easy to show that a vector $\mathbf{w}$ that maximizes $\boldsymbol{J}(\cdot)$ must satisfy:

$$\mathbf{w}^{\mathrm{T}}\mathbf{S}_w\mathbf{w} = c \neq 0$$

and define Lagrange function $\boldsymbol{L}(\mathrm{w}, \lambda)$:

$$\boldsymbol{L}(\mathrm{w}, \lambda) = \mathbf{w}^{\mathrm{T}}\mathbf{S}_b\mathbf{w} - \lambda(\mathbf{w}^{\mathrm{T}}\mathbf{S}_w\mathbf{w} - c)$$

$$\frac{\partial \boldsymbol{L}(\mathbf{w},\lambda)}{\partial \mathbf{w}} = \mathbf{S}_b\mathbf{w} - \lambda\mathbf{S}_w\mathbf{w}$$

$$\mathbf{S}_b\mathbf{w} - \lambda\mathbf{S}_w\mathbf{w} = 0$$

$$\mathbf{S}_b\mathbf{w} = \lambda\mathbf{S}_w\mathbf{w} \tag{3.32}$$

for some constant », which is a generalized eigenvalue problem. This can also be seen informally by noting that at an extremum of $\boldsymbol{J}(\mathbf{w})$, a small change in $\mathbf{w}$ in Equation 3.31 should leave unchanged the ratio of the numerator to the denominator. If $\mathbf{S}_w$ is nonsingular, we can obtain a conventional eigenvalue problem by writing:

$$\mathbf{S}_w^{-1}\mathbf{S}_b\mathbf{w} = \lambda\mathbf{w} \tag{3.33}$$

In our particular case, it is unnecessary to solve for the eigenvalues and eigenvectors of $\mathbf{S}_w^{-1}\mathbf{S}_b$ due to the fact that $\mathbf{S}_b\mathbf{w}$ is always in the direction of $\mathbf{m}_1 - \mathbf{m}_2$. Since the scale factor for $\mathbf{w}$ is immaterial, we can immediately write the solution for the $\mathbf{w}$ that optimizes $\boldsymbol{J}(\cdot)$:

$$\mathbf{w} = \mathbf{S}_w^{-1}(\mathbf{m}_1 - \mathbf{m}_2) \tag{3.34}$$

Thus, we have obtained $\mathbf{w}$ for Fisher's linear discriminant — the linear function yielding the maximum ratio of between-class scatter to within-class scatter (Yu & Yang, 2001; Zhao, Chellappa, & Phillips, 1999; Chen, Liao, Lin, Ko, & Yu, 2000; Huang, Liu, Lu, & Ma, 2002). (The solution $\mathbf{w}$ given by Equation 3.34 is sometimes called the *canonical variate*.) Thus, the classification has been converted from a *d*-dimensional problem to a hopefully more manageable one-dimensional problem. This mapping is many-to-one,

and in theory cannot possibly reduce the minimum achievable error rate if we have a very large training set. In general, one is willing to sacrifice some of the theoretically attainable performance for the advantages of working in one dimension. All that remains is to find the threshold; that is, the point along the one-dimensional subspace separating the projected points. When the conditional densities $p(\mathbf{x}|\ w_i)$ are multivariate normal with equal covariance matrices $\mathbf{\Sigma}$, we can calculate the threshold directly. In that case, recall from Chapter II that the optimal decision boundary has the equation:

$$\mathbf{w}^T\mathbf{x}+\mathbf{w}_0=0 \tag{3.35}$$

where:

$$\mathbf{w} = \sum^{-1} (\mu_1 - \mu_2) \tag{3.36}$$

and where w_0 is a constant involving $\mathbf{w}$ and the prior probabilities. If we use sample means and the sample covariance matrix to estimate μ_i and Σ, we obtain a vector in the same direction as the $\mathbf{w}$ of Equation 3.36 that maximized $\boldsymbol{J}(\cdot)$. Thus, for the normal, equal-covariance case, the optimal decision rule is merely to decide: ω_1 if Fisher's linear discriminant exceeds some threshold, and ω_2 otherwise. More generally, if we smooth the projected data, or fit it with a univariate Gaussian, we then should choose w_0 where the posteriors in the one dimensional distributions are equal.

The computational complexity of finding the optimal $\mathbf{w}$ for the Fisher linear discriminant (Equation 3.34) is dominated by the calculation of the within-category total scatter and its inverse, an $O(d^2n)$ calculation.

Multiple Discriminant Analysis

For the c-class problem, the natural generalization of Fisher's linear discriminant involves $c-1$ discriminant functions. Thus, the projection is from a d-dimensional space to a $(c-1)$-dimensional space, and it is tacitly assumed that $d \geq c$. The generalization for the within-class scatter matrix is obvious (Duda, Hart, & Stork, 2000; McLachlan, 1992; Chen, Liao, Ko, Lin, & Yu, 2000; Friedman, 1989; Yu & Yang, 2001; Zho, Chellappa, & Phillips, 1999; Chen, Liao, Lin, Ko, & Yu, 2000; Huang, Liu, Lu, & Ma, 2002):

$$\mathbf{S}_w = \sum_{i=1}^{c} \mathbf{S}_i \tag{3.37}$$

where, as before:

$$\mathbf{S}_i = \sum_{\mathbf{x}\ D_i} (\mathbf{x}-\mathbf{m}_i)(\mathbf{x}-\mathbf{m}_i)^T \tag{3.38}$$

and:

$$\mathbf{m}_i = \frac{1}{n_i} \sum_{\mathbf{x}\ D_i} \mathbf{x} \tag{3.39}$$

The proper generalization for $\mathbf{S}_b$ is not quite so obvious. Suppose that we define a *total mean vector* $\mathbf{m}$ and a *total scatter matrix* $\mathbf{S}_t$ by:

$$\mathbf{m} = \frac{1}{n}\sum_{\mathbf{x}} \mathbf{x} = \frac{1}{n}\sum_{i\ 1}^{c} n_i m_i \tag{3.40}$$

and:

$$\mathbf{S}_t = \sum_{\mathbf{x}} (\mathbf{x} - \mathbf{m})(\mathbf{x} - \mathbf{m})^{\mathrm{T}} \tag{3.41}$$

Then it follows that:

$$\begin{aligned}\mathbf{S}_t &= \sum_{i=1}^{c}\sum_{x\in D_i} (\mathbf{x}\text{-}\mathbf{m}_i + \mathbf{m}_i\text{-}\mathbf{m})(\mathbf{x}\text{-}\mathbf{m}_i + \mathbf{m}_i\text{-}\mathbf{m})^{\mathrm{T}} \\ &= \sum_{i=1}^{c}\sum_{x\in D_i} (\mathbf{x}\text{-}\mathbf{m}_i)(\mathbf{x}\text{-}\mathbf{m}_i)^{\mathrm{T}} + \sum_{i=1}^{c}\sum_{x\in D_i} (\mathbf{m}_i\text{-}\mathbf{m})(\mathbf{m}_i\text{-}\mathbf{m})^{\mathrm{T}} \\ &= \mathbf{S}_w + \sum_{i=1}^{c}\sum_{x\in D_i} n_i(\mathbf{m}_i\text{-}\mathbf{m})(\mathbf{m}_i\text{-}\mathbf{m})^{\mathrm{T}}\end{aligned} \tag{3.42}$$

It is natural to define this second term as a general between-class scatter matrix, so that the total scatter is the sum of the within-class scatter and the between-class scatter:

$$\mathbf{S}_b = \sum_{i=1}^{c} n_i(\mathbf{m}_i\text{-}\mathbf{m})(\mathbf{m}_i\text{-}\mathbf{m})^{\mathrm{T}} \tag{3.43}$$

and:

$$\mathbf{S}_t = \mathbf{S}_w + \mathbf{S}_b \tag{3.44}$$

If we check the two-class case, we find that the resulting between-class scatter matrix is $n_1 n_2 / n$ times our previous definition.

The projection from a d-dimensional space to a $(c - 1)$-dimensional space is accomplished by $c - 1$ discriminant functions:

$$y_i = \mathbf{w}_i^{\mathrm{T}}\mathbf{x} \quad i = 1, \ldots, c-1 \tag{3.45}$$

If the y_i are viewed as components of a vector $\mathbf{y}$, and the weight vectors $\mathbf{w}_i$ are viewed as the columns of a d-by-$(c - 1)$ matrix $\mathbf{W}$, then the projection can be written as a single matrix equation:

$$\mathbf{y} = \mathbf{W}^{\mathrm{T}}\mathbf{x} \tag{3.46}$$

The samples $\mathbf{x}_1, \ldots, \mathbf{x}_n$ project to a corresponding set of samples $\mathbf{y}_1, \ldots, \mathbf{y}_n$, which can be described by their own mean vectors and scatter matrices. Thus, if we define:

$$\tilde{m}_i = \frac{1}{n_i} \sum_{\mathbf{y} \in Y_i} \mathbf{y} \tag{3.47}$$

$$\tilde{m}_i = \frac{1}{n} \sum_{i=1}^{c} n_i \tilde{m}_i \tag{3.48}$$

$$\tilde{\boldsymbol{S}}_w = \sum_{i=1}^{c} \sum_{\mathbf{y} \in Y_i} (\mathbf{y} - \tilde{\mathbf{m}}_i)(\mathbf{y} - \tilde{\mathbf{m}}_i)^{\mathrm{T}} \tag{3.49}$$

and:

$$\tilde{\boldsymbol{S}}_b = \sum_{i=1}^{c} n_i (\tilde{\mathbf{m}}_i - \tilde{\mathbf{m}})(\tilde{\mathbf{m}}_i - \tilde{\mathbf{m}})^{\mathrm{T}} \tag{3.50}$$

It is a straightforward matter to show that:

$$\tilde{\boldsymbol{S}}_w = \mathbf{W}^{\mathrm{T}} \mathbf{S}_w \mathbf{W} \tag{3.51}$$

and:

$$\tilde{\boldsymbol{S}}_b = \mathbf{W}^{\mathrm{T}} \mathbf{S}_b \mathbf{W} \tag{3.52}$$

These equations show how the within-class and between-class scatter matrices are transformed by the projection to the lower dimensional space (Figure 3.7). What we seek is a transformation matrix **W** that in some sense maximizes the ratio of the between-class scatter to the within-class scatter. A simple scalar measure of scatter is the determinant of the scatter matrix. The determinant is the product of the eigenvalues, and hence is the product of the "variances" in the principal directions, thereby measuring the square of the hyperellipsoidal scattering volume. Using this measure, we obtain the criterion function:

$$\boldsymbol{J}(\mathbf{W}) = \frac{\left| \tilde{\boldsymbol{S}}_b \right|}{\left| \tilde{\boldsymbol{S}}_w \right|} = \frac{| \mathbf{W}^{\mathrm{T}} \mathbf{S}_b \mathbf{W} |}{| \mathbf{W}^{\mathrm{T}} \mathbf{S}_w \mathbf{W} |} \tag{3.53}$$

The problem of finding a rectangular matrix **W** that maximizes $\boldsymbol{J}(\cdot)$ is tricky, though fortunately it turns out that the solution is relatively simple. The columns of an optimal **W** are the generalized eigenvectors that correspond to the largest eigenvalues in:

$$\mathbf{S}_b\mathbf{w}_i=\lambda_i\mathbf{S}_w\mathbf{w}_i. \quad (3.54)$$

A few observations about this solution are in order. First, if $\mathbf{S}_w$ is non-singular, this can be converted into a conventional eigenvalue problem as before. However, this is actually undesirable, since it requires an unnecessary computation of the inverse of $\mathbf{S}_w$. Instead, one can find the eigenvalues as the roots of the characteristic polynomial:

$$|\mathbf{S}_b-\lambda_i\mathbf{S}_w|=0 \quad (3.55)$$

and then solve:

$$(\mathbf{S}_b-\lambda_i\mathbf{S}_w)\mathbf{w}_i=0 \quad (3.56)$$

directly for the eigenvectors $\mathbf{w}_i$. Because $\mathbf{S}_b$ is the sum of c matrices of rank one or less, and because only $c-1$ of these are independent, $\mathbf{S}_b$ is of rank $c-1$ or less. Thus, no more than $c-1$ of the eigenvalues are nonzero, and the desired weight vectors correspond to these nonzero eigenvalues. If the within-class scatter is isotropic, the eigenvectors are merely the eigenvectors of $\mathbf{S}_b$, and the eigenvectors with nonzero eigenvalues span the space spanned by the vectors $\mathbf{m}_i-\mathbf{m}$. In this special case, the columns of $\mathbf{W}$ can be found simply by applying the Gram-Schmidt orthonormalization procedure to the $c-1$ vectors $\mathbf{m}_i-\mathbf{m}$, $i=1,\ldots,c-1$. Finally, we observe that in general the solution for $\mathbf{W}$ is not unique. The allowable transformations include rotating and scaling the axes in various ways. These are all linear transformations from a $(c-1)$-dimensional space to a $(c-1)$-dimensional space, however, and do not change things in any significant way; in particular, they leave the criterion function $\boldsymbol{J}(\mathbf{W})$ invariant and the classifier unchanged.

As in the two-class case, multiple discriminant analysis primarily provides a reasonable way of reducing the dimensionality of the problem. Parametric or nonparametric techniques that might not have been feasible in the original space may work well in the lower-dimensional space. In particular, it may be possible to estimate separate covariance matrices for each class and use the general multivariate normal assumption after the transformation, where this could not be done with the original data. In general, if the transformation causes some unnecessary overlapping of the data and increases the theoretically achievable error rate, then the problem of classifying the data still remains. However, there are other ways to reduce the dimensionality of data, and we shall encounter this subject again in later chapters.

NON-LINEAR LDA TECHNOLOGIES

Discriminant analysis addressed the following question: Given a data set with two classes, say, which is the best feature or feature set (either linear or non-linear) to discriminate the two classes? Classical approaches tackle this question by starting with the (theoretically) optimal Bayes classifier and, by assuming normal distributions for the classes, it is possible to derive standard algorithms like quadratic or LDA, among them the famous Fisher discriminant (Devijver & Kitter, 1989; Fukunaga, 1990). Of course, any other model different from a Gaussian for the class distributions could be assumed, but

this often sacrifices the simple closed-form solution. Several modifications towards more general features have been proposed (e.g., Devijver & Kitter, 1982); for an introduction and review on existing methods see, for example Devijver and Kitter (1982), Fukunaga (1998), Aronszajn (1950), Ripley (1996), and Liu, Huang, Lu, and Ma (2002).

In this section, we propose to use the kernel idea (Chen, Liao, Ko, Lin, & Yu, 2000; Aizerman, Braverman, & Rozonoer, 1964; Aronszajn, 1950), originally applied in Support Vector Machines (SVMs) (Burges, 1996; Schölkopf, Burges, & Smola, 1999; Vapnik, 1995; Hastie, Tibshirani, & Friendman, 2001), KPCA (Schölkopf, Smola, & Müller, 1998) and other kernel-based algorithms (cf. Schölkopf, Burges, & Smola, 1999; Mika, Smola, & Schölkopf, 2001; Rosipal & Trejo, 2001; Mika, Ratsch, & Müller, 2001; Smola & Schölkopf, 1998) to define a non-linear generalization of Fisher's discriminant. KFD uses kernel feature spaces, yielding a highly flexible algorithm that turns out to be competitive with SVMs (Evgeniou, Pontil, & Poggio, 1999; Suykens & Vandewalle, 1999; Van Gestel, Suykens, & De Brabanter, 2001).

Note that there exist a variety of methods called *kernel discriminant analysis* (McLachlan, 1992; Chen, Liao, Ko, Lin, & Yu, 2000; Aizerman, Braverman, & Rozonoer, 1964). Most of them aim at replacing the parametric estimate of class conditional distributions by a non-parametric kernel estimate.

Here, we restrict ourselves to finding non-linear directions by first mapping the data non-linearly into some feature space $\boldsymbol{F}$ and computing Fisher's linear discriminant there, thus implicitly yielding a non-linear discriminant in input space (Baudat & Anouar, 2000):

$$\mathbf{S}_w = \sum_{i=1,2} \sum_{\mathbf{x} \in X_i} (\mathbf{x} - \mathbf{m}_i)(\mathbf{x} - \mathbf{m}_i)^{\mathrm{T}}$$

Let Φ be a non-linear mapping to some feature space $\boldsymbol{F}$ to find the linear discriminant in $\boldsymbol{F}$ we need to maximize:

$$\boldsymbol{J}(\mathbf{w}) = \frac{\mathbf{w}^T \mathbf{S}_b^{\Phi} \mathbf{w}}{\mathbf{w}^T \mathbf{S}_w^{\Phi} \mathbf{w}} \tag{3.57}$$

where now $\mathrm{w} \in \boldsymbol{F}$, $\mathbf{S}_b^{\Phi}$ and $\mathbf{S}_w^{\Phi}$ are the corresponding matrices in $\boldsymbol{F}$; that is:

$$\mathbf{S}_b^{\Phi} = (\mathbf{m}_1^{\Phi} - \mathbf{m}_2^{\Phi})(\mathbf{m}_1^{\Phi} - \mathbf{m}_2^{\Phi})^{\mathrm{T}}$$

and:

$$\mathbf{S}_w^{\Phi} = \sum_{i=1,2} \sum_{x \in X_i} (\boldsymbol{\Phi}(\mathbf{x}) - \mathbf{m}_i^{\Phi})(\boldsymbol{\Phi}(\mathbf{x}) - \mathbf{m}_i^{\Phi})^{\mathrm{T}} \tag{3.58}$$

with:

$$\mathbf{m}_i^{\Phi} = \frac{1}{l_i} \sum_{j=1}^{l_i} \boldsymbol{\Phi}(\mathbf{x}_j^i)$$

Introducing kernel functions: Clearly, if F is very high or even infinitely dimensional, this will be impossible to solve directly. To overcome this limitation, we use the same trick as in KPCA (Schölkopf, Smola, & Müller, 1998) or SVMs. Instead of mapping the data explicitly, we seek a formulation of the algorithm that uses only dot-products ($\Phi(\mathbf{x}) \cdot \Phi(\mathbf{y})$) of the training patterns. As we are then able to compute these dot-products efficiently, we can solve the original problem without ever mapping explicitly to F. This can be achieved using Mercer kernels (Saitoh, 1998). These kernels $k(\mathbf{x}, \mathbf{y})$ compute a dot-product in some feature space F; that is, $k(\mathbf{x}, \mathbf{y}) = (\Phi(\mathbf{x}) \cdot \Phi(\mathbf{y}))$. Possible choices for k that have proven useful — for example, in SVMs or KPCA — are Gaussian RBF, $k(\mathbf{x}, \mathbf{y}) = \exp(-\|\mathbf{x}-\mathbf{y}\|^2/c)$ or polynomial kernels, $k(\mathbf{x}, \mathbf{y}) = (\mathbf{x} \cdot \mathbf{y})^d$, for some positive constants c and d respectively (Roth & Steinhage, 2000).

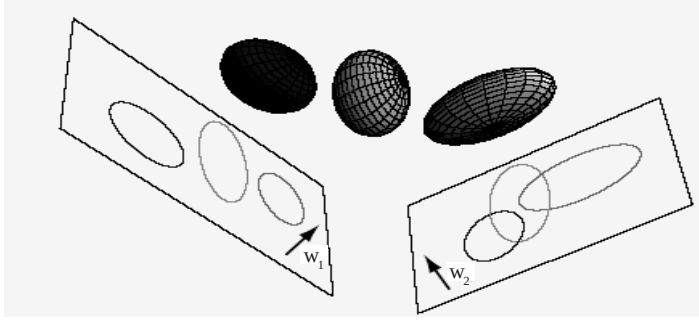
To find Fisher's discriminant in the feature space $\boldsymbol{F}$, we first need a formulation of Equation 3.57 in terms of only dot-products if input patterns, which we then replace by some kernel function. From the theory of reproducing kernels, we know that any solution $\mathbf{w} \in \boldsymbol{F}$ must lie in the span of all training samples in $\boldsymbol{F}$. Therefore, we can find an expansion for w of the form:

$$\mathbf{w} = \sum_{i=1}^{l} \alpha_i \mathbf{\Phi}(\mathbf{x}_i) \tag{3.59}$$

Using the expansion Equation 3.59 and the definition of $\mathbf{m}_i^{\Phi}$ we write:

$$\mathbf{w}^{\mathrm{T}} \mathbf{m}_i^{\Phi} = \frac{1}{l} \sum_{j=1}^{l} \sum_{k=1}^{l_i} \alpha_j k(\mathbf{x}_j, \mathbf{x}_k^i) = \boldsymbol{\alpha}^{\mathbf{T}} \mathbf{M}_i \tag{3.60}$$

Figure 3.7. Three three-dimensional distributions are projected onto two-dimensional subspaces, described by a normal vectors $\boldsymbol{w}_1$ and $\boldsymbol{w}_2$



Informally, multiple discriminant methods seek the optimum such subspace; that is, the one with the greatest separation of the projected distributions for a given total within-scatter matrix, here as associated with $\boldsymbol{w}_1$ (Duda, Hart, & Stork, 2000)

where we defined $(\mathbf{M}_i)_j = \frac{1}{l_i}\sum_{k=1}^{l_i} k(\mathbf{x}_j, \mathbf{x}_k^i)$ and replaced the dot-products by the kernel function. Now consider the numerator of Equation 3.57. By using the definition of $\mathbf{S}_b^\Phi$ and Equation 3.60, it can be rewritten as:

$$\mathbf{w}^T \mathbf{S}_b^\Phi \mathbf{w} = \boldsymbol{\alpha}^T \mathbf{M} \boldsymbol{\alpha} \tag{3.61}$$

where $\mathbf{M} = (\mathbf{M}_1 - \mathbf{M}_2)(\mathbf{M}_1 - \mathbf{M}_2)^T$. Considering the denominator, using Equation 3.59, the definition of $\mathbf{m}_i^\Phi$ and a similar transformation as in Equation 3.61, we find:

$$\mathbf{w}^T \mathbf{S}_w^\Phi \mathbf{w} = \boldsymbol{\alpha}^T \mathbf{N} \boldsymbol{\alpha} \tag{3.62}$$

where we set $\mathbf{N} = \sum_{j=1,2} \mathbf{K}_j (\boldsymbol{I} - 1_{li}) \mathbf{K}_j^T$, $\mathbf{K}_j$ is a $l \times l_j$ matrix with $(\mathbf{K}_j)nm = k(\mathbf{x}_n, \mathbf{x}_m^j)$ (this is the kernel matrix for class j), $\boldsymbol{I}$ is the identity and $\mathbf{1}_{lj}$ the matrix with all entries $1/l_j$.

Combining Equations 3.61 and.62, we can find Fisher's linear discriminant in $\boldsymbol{F}$ by maximizing:

$$\boldsymbol{J}(\boldsymbol{\alpha}) = \frac{\boldsymbol{\alpha}^T \mathbf{M} \boldsymbol{\alpha}}{\boldsymbol{\alpha}^T \mathbf{N} \boldsymbol{\alpha}} \tag{3.63}$$

This problem can be solved (analogously to the algorithm in the input space) by finding the leading eigenvector of $\mathbf{N}^{-1}\mathbf{M}$. We will call this approach (non-linear) KFD. The projection of a new pattern $\mathbf{x}$ onto $\mathbf{w}$ is given by:

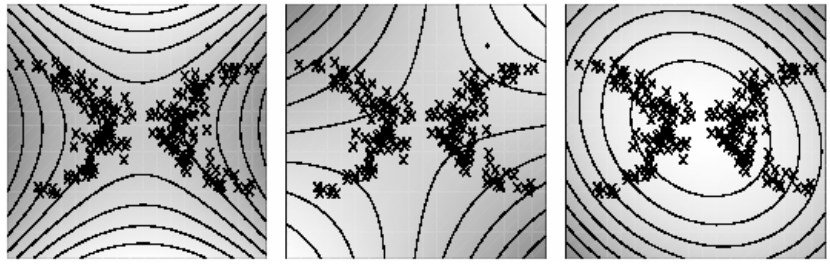
$$(\mathbf{w} \cdot \Phi(\mathbf{x})) = \sum_{i=1}^{l} \alpha_i k(\mathbf{x}_i, \mathbf{x}) \tag{3.64}$$

Numerical issues and regularization: Obviously, the proposed setting is ill-posed. We are estimating l dimensional covariance structures from l samples. Besides numerical problems, which cause the matrix $\boldsymbol{N}$ to be not positive, we need a way of capacity control in $\boldsymbol{F}$. To this end, we simply add a multiple of identity matrix to $\boldsymbol{N}$; that is, replace $\boldsymbol{N}$ by $\mathbf{N}_\mu$, where:

$$\mathbf{N}_\mu = \mathbf{N} + \boldsymbol{\mu} \boldsymbol{I} \tag{3.65}$$

This can be viewed in different ways: (1) it clearly makes the problem numerically more stable, as for $\boldsymbol{\mu}$ large enough $\mathbf{N}_\mu$ will become positive definite; (2) it can be seen in analogy to Friedman, (1989) and Hastie, Tibshirani and Buja (1994), decreasing the bias in sample-based estimation of eigenvalues; (3) it imposes a regularization on $\|\boldsymbol{\alpha}\|^2$ (remember that we are maximizing Equation 3.63), favoring solutions with small expansion coefficients. Although the real influence in this setting of the regularization is not yet fully understood, it shows connections to those used in SVMs (Burges, 1996; Schölkopf, Burges, & Smola, 1999; Suykens & Vandewalle, 1999; Van Gestel, Suykens, & De

Figure 3.8. Comparison of feature found by KFD (left) and those found by KPCA: First (middle) and second (right) (Mika, Rätsch, Weston, Scholkopf, & Muller, 1999)



Brabanter, 2001). Furthermore, one might use other regularization type additives to N; for example, penalizing $||\mathbf{w}||^2$ in analogy to SVM (by adding the full kernel matrix $K_{ij}=k(\mathbf{x}_i,\mathbf{x}_j)$).

Figure 3.8 shows an illustrative comparison of the feature found by KFD and the first and second (non-linear) feature found by KPCA (Schölkopf, Smola, & Müller, 1998) on a toy data set. For both, we used a polynomial kernel of degree two and for KFD the regularized with class scatter Equation 3.65, where $\boldsymbol{\mu}= 10^{-3}$. Depicted are the two classes (crosses and dots), the feature value (indicated by grey level) and contour lines of identical feature value. Each class consists of two noisy parabolic shapes mirrored at the *x* and *y* axis, respectively. We see that the KFD feature discriminates the two classes in a nearly optimal way, whereas the KPCA features, albeit describing interesting properties of the data set, do not separate the two classes well (although higher-order KPCA features might also be discriminating).

To evaluate the performance of this new approach, Mika performed an extensive comparison to other state-of-art classifiers (Mika, Ratsch, Weston, Schölkopf, & Müller, 1999). The experimental setup was chosen in analogy to Ratsch, Onoda, and Müller (1998) and they compared the KFD to AdaBoost, regularized AdaBoost (Ratsch, Onoda, & Müller, 1998) and SVMs (with Gaussian kernel). For KFD, they used Gaussian kernels, too, and the regularized within-class scatter from Equation 3.65. After the optimal direction $w \in \boldsymbol{F}$ was found, they computed projections onto it using Equation 3.64. To estimate an optimal threshold on the extracted feature, one may use any classification technique; for example, it is as simple as fitting a sigmoid (Platt, 1999). They used a linear SVM. They used 13 artificial and real-world datasets from the UCI, DELVE and STATLOG benchmark repositories (except for banana). Then 100 partitions into test and training set (about 60:40) were generated. On each of these data sets they trained and tested all classifiers (see Rätsch, Onoda, & Müller, 1998 for details). The results in Table 3.1 show the average test error over these 100 runs and the standard deviation. To estimate the necessary parameters, they ran five-fold cross validation on the first five realizations of the training sets and took the model parameters to be the median over the five estimates.

The experiments show that the KFD (plus an SVM to estimate the threshold) is competitive or in some cases even superior to the other algorithms on almost all data sets (an exception being image). Furthermore, there is still much room for extensions and further theory, as LDA is an intensively studied field and many ideas preciously

Table 3.1. Comparison between KFD, a single RBF classifier, AdaBoost (AB), regularized AdaBoost (AB_R) and SVM. Best method in bold face, second-best emphasized (Mika, Ratsch, Weston, Scholkopf, & Muller, 1999)

	RBF	AB	AB_R	SVM	KFD
Banana	**10.8 ± 0.6**	12.3 ± 0.7	*10.9 ± 0.4*	11.5 ± 0.7	**10.8 ± 0.5**
B.Cancer	27.6 ± 4.7	30.4 ± 4.7	26.5 ± 4.5	*26.0 ± 4.7*	**25.8 ± 4.6**
Diabetes	24.3 ± 1.9	26.5 ± 2.3	23.8 ± 1.8	*23.5 ± 1.7*	**23.2 ± 1.6**
German	24.7 ± 2.4	27.5 ± 2.5	24.3 ± 2.1	**23.6 ± 2.1**	*23.7 ± 2.2*
Heart	17.6 ± 3.3	20.3 ± 3.4	16.5 ± 3.5	**16.0 ± 3.3**	*16.1 ± 3.4*
Image	3.3 ± 0.6	**2.7 ± 0.7**	**2.7 ± 0.6**	*3.0 ± 0.6*	4.8 ± 0.6
Ringnorm	1.7 ± 0.2	1.9 ± 0.3	*1.6 ± 0.1*	1.7 ± 0.1	1.5 ± 0.1
F.Sonar	34.4 ± 2.0	35.7 ± 1.8	34.2 ± 2.2	**32.4 ± 1.8**	*33.2 ± 1.7*
Splice	*10.0 ± 1.0*	10.1 ± 0.5	**9.5 ± 0.7**	10.9 ± 0.7	10.5 ± 0.6
Thyroid	4.5 ± 2.1	*4.4 ± 2.2*	4.6 ± 2.2	4.8 ± 2.2	**4.2 ± 2.1**
Titanic	23.3 ± 1.3	*22.6 ± 1.2*	*22.6 ± 1.2*	**22.4 ± 1.0**	23.2 ± 2.0
Twonorm	2.9 ± 0.3	3.0 ± 0.3	*2.7 ± 0.2*	3.0 ± 0.2	**2.6 ± 0.2**
Waveform	10.7 ± 1.1	10.8 ± 0.6	**9.8 ± 0.8**	*9.9 ± 0.4*	*9.9 ± 0.4*

developed in the input space carry over to feature space. Note that while the complexity of SVMs scales with the number of Support Vectors, KFD does not have a notion of Support Vectors, and its complexity scales with the number of training patterns. On the other hand, we speculate that some of the superior performance of KFD over SVM might be related to the fact that KFD uses all training samples in the solution, not only the difficult ones; that is, the Support Vectors.

SUMMARY

Fisher's LDA is a classical multivariate technique both for dimension reduction and classification. The data vectors are transformed into a low-dimensional subspace such that the class centroids are spread out as much as possible. Fisher's linear discriminant finds a good subspace in which categories are best separated. Other techniques can then be applied in the subspace. Fisher's method can be extended to cases with multiple categories projected onto subspaces of higher dimension than a line.

Although Fisher's discriminant is one of the standard linear techniques in statistical data analysis, linear methods are often too limited, and several approaches have been made in the past to derive more general class separability criteria (Fukunaga, 1990; Hastie & Tibshirani, 1994; Aronszajn, 1950). However, in many applications, the linear boundaries do not adequately separate the classes. Non-linear LDA, which uses the kernel trick of representing dot products by kernel function, was presented. We are still able to find closed-form solutions and maintain the theoretical beauty of Fisher's dicriminant analysis. Furthermore, different kernels allow for high flexibility due to the wide range of non-linearities possible. Experiments of Mika (Mika, Ratsch, Weston, Schölkopf, & Müller, 1999) show that KFD is competitive to other state-of-the-art classification techniques.

REFERENCES

Aizerman, M., Braverman, E., & Rozonoer, L. (1964). Theoretical foundations of the potential function method in pattern recognition learning. *Automation and Remote Control, 25*, 821-839.

Aronszajn, N. (1950). Theory of reproducing kernels. *Transactions of the American Mathematical Society*, *686*, 337-404.

Baudat, G., & Anouar, F. (2000). Generalized discriminant analysis using a kernel approach. *Neural Computation, 12*, 2385-2404.

Burges, C. (1996). Simplified support vector decision rules. In L. Saitta (Ed.), *Proceedings of the 13th International Conference on Maching Learning, 13th ICML,* San Mateo, CA (pp. 71-77).

Chen, L., Liao, H., Ko, M., Lin, J., & Yu, G. (2000). A new LDA-based face recognition system which can solve the small sample size problem. *Pattern Recognition, 33*, 1713-1726.

Devijver, P., & Kitter, J. (1982). *Pattern recognition: A statistical approach*. NJ: Prentice Hall.

Duda, R. O., Hart, P. E., & Stork, D. G. (2000). *Pattern classification* (2nd ed.). New York: John Wiley Press.

Etemad, K., & Chellappa, R. (1997). Discriminant analysis for recognition of human face images. *Journal of Optical Society of America, A*, 1724-1733.

Evgeniou, T., Pontil, M., & Poggio, T. (1999). Regularization networks and support vector machines. *Advances in Computational Mathematics*, *13,* 1-50.

Friedman, J. (1989). Regularized discriminant analysis. *Journal of the American Statistical Association, 84*(405), 165-175.

Fukunaga, K. (1990). *Introduction to statistical pattern recognition* (2nd ed.). San Diego, CA: Academic Press.

Hastie, T., Buja, A., & Tibshirani, R. (1995). Penalized discriminant analysis. *Annals of Statistics*, *23*, 73-102.

Hastie, T., & Tibshirani, R. (1994). Discriminant analysis by Gaussian mixtures. *Journal of the American Statistical Association*.

Hastie, T., Tibshirani, R., & Buja, A. (1994). Flexible discriminant analysis by optimal scoring. *Journal of the American Statistical Association, 89,* 1255-1270.

Hastie, T., Tibshirani, R., & Friendman, J. (2001).*The elements of statistical learning: Data mining, inference, and prediction*. New York: Springer.

Huang, R., Liu, Q. S., Lu, H. Q., & Ma, S. D. (2002). Solving the small sample size problem of LDA. *Proceedings of the International Conference of Pattern Recognition* (Vol. 3, pp. 29-32).

Liu, Q. S., Huang, R., Lu, H. Q., & Ma, S. D. (2002). Face recognition using kernel based Fisher discriminant analysis. *Proceedings of the International Conference of Automatic Face and Gesture Recognition* (pp. 197-201).

McLachlan, G. (1992). *Discriminant analysis and statistical pattern recognition.* New York: John Wiley & Sons.

Mika, S., Rätsch, G., & Müller, K-R. (2001). A mathematical programming approach to the kernel Fisher algorithm. *Advances in Neural Information Processing Systems*, *13*, 591-597.

Mika, S., Rätsch, G., Weston, J., Scholköpf, B., & Müller, K-R. (1999). Fisher discriminant analysis with kernels. In Y.-H. Hu, J. Larsen, E. Wilson, & S. Douglas (Eds.), *Processing of Neural Networks for Signal Processing Workshop,* Madison, WI (pp. 41-48).

Mika, S., Smola, A., & Scholköpf, B. (1994). An improved training algorithm for kernel Fisher discriminants. *Proceedings of Artificial Intelligence and Statistics* (pp. 98-104). San Franciscco: Morgan Kaufmann.

Platt, J. (1999). Probabilistic outputs for support vector machines and comparison to regularized likelihood methods. In A. J. Smola, P. Bartlett, B. Scholköpf, & D. Schuurmans (Eds.), *Advances in large margin classifiers*. MIT Press.

Rätsch, G., Onoda, T., & Müller, K-R. (1998). *Soft margins for adaboost (Technical Report NC-TE-1998-021)*. London: Royal Holloway College, University of London.

Ripley, B. (1996). *Pattern recognition and neural networks*. London: Cambridge University Press.

Ripley, B. D. (1994). Neural networks and related methods for classification. *Journal of the Royal Statistical Society Series B, 56*, 409-456.

Rosipal, R., & Trejo, L. J. (2001). Kernel partial least squares regression in reproducing kernel Hilbert space. *Journal of Machine Learning Research*, *2*, 97-123.

Roth, V., & Steinhage, V. (2000). Nonlinear discriminant analysis using kernel functions. *Advances in Neural Information Processing Systems*, *12*, 568-574.

Saitoh, S. (1988). *Theory of reproducing kernels and its applications*. Harrow: Longman Scientific & Technical.

Scholköpf, B., Burges, C., & Smola, A. (Eds.). (1999). *Advances in kernel methods-support vector learning*. MIT Press.

Scholköpf, B., Mika, S., Burges, C., Knirsch, P., Müller, K-R., Rätsch, G., & Smola, A. (1999). Input space vs. feature space in kernel-based methods. *IEEE Transactions on Neural Networks: Special Issue on VC Learning Theory and Its Application.*

Scholköpf, B., Smola, A., & Müller, K-R. (1998). Nonlinear component analysis as a kernel eigenvalue problem. *Neural Computation*, *10*, 1299-1319.

Shashua, A. (1999). On the relationship between the support vector machine for classification and sparsified fisher's linear discriminant. *Neuaral Procressing Letters*, *9*(2), 129-139.

Smola, A., & Scholköpf, B. (1998). On a kernel-based method for pattern recognition, regression, approximation and operator inversion. *Algorithmica*, *22*, 211-231.

Suykens, J. A. K., & Vandewalle, J. (1999). Least squares support vector machine classifiers. *Neural Processing Letters*, *9*, 293-300.

Tong, S., & Koller, D. (n.d.). Bayes optimal hyperplanes-maximal margin hyperplanes. Submitted to *IJCAI'99 Workshop on Support Vector Machines*.

Van Gestel, T., Suykens, J. A. K., & De Brabanter, J. (2001). Least squares support vector machine regression for discriminant analysis. *Processing International Joint INNS-IEEE Conference on Neural Networks (INNS 2001),* Washington, DC (pp. 14-19).

Vapnik, V. (1995). *The nature of statistical learning theory*. New York: Springer Verlag.

Vapnik, V. (1998). *Statistical learning theory*. New York: Wiley.

Xu, Y., Yang, J-Y., & Jin, Z. (2004). A novel method for Fisher discriminant analysis. *Pattern Recognition*, *37*, 381-384.

Yang, J., & Yang, J-Y. (2001). Optimal FLD algorithm for facial feature extraction. In *SPIE Proceedings of the Intelligent Robots and Computer Vision XX: Algorithms, Techniques, and Active Vision,* 4572 (pp. 438-444).

Yang, J., & Yang, J-Y. (2003). Why can LDA be performed in PCA transformed space? *Pattern Recognition, 36*(2), 563-566.

Yu, H., & Yang, J. (2001). A direct LDA algorithm for high-dimensional data – with application to face recognition. *Pattern Recognition*, *34*(10), 2067-2070.

Zhao, W., Chellappa, R., & Phillips, P. J. (1999). *Subspace linear discriminant analysis for face recognition (Tech Report CAR-TR-914)*. Center for Automation Research, University of Maryland.

Zheng, W. M., Zhao, L., & Zou, C. (2002). Recognition using extended multiple discriminant analysis (EMDA) method. *Proceedings of the First International Conference on Machine Learning and Cybernetic* (Vol. 1, pp. 121-125). Beijing.

Chapter IV

PCA/LDA Applications in Biometrics

ABSTRACT

In this chapter, we show some PCA/LDA applications in biometrics. Based on the introductions to both PCA and LDA mentioned in Chapters II and III, their simple descriptions are given first. Then, we indicate a significant application in face recognition. The next sections discuss palmprint identification and gait verification, respectively. For other applications, ear biometrics, speaker identification, iris recognition and signature verification are respectively described. At the end of this chapter, we point out a brief but useful summary.

INTRODUCTION

PCA is famous for its dimension-reducing ability. It uses the least number of dimensions but keeps most of the facial information. Romdhani puts forward the usage of this algorithm and shows that there exists a subspace of image space called face space (Romdhani, Gon, & Psarrou, 1999). Faces, after being transformed into this space, have the smallest least square error after we compare the image before and after it is reconstructed. We can use this characteristic for detecting face. Feraud showed the advantages and disadvantages between PCA, ANN and estimation functions (Feraud, 1997). Moghaddam uses the outputs of PCA to provide a probability matching function for face recognition (Moghaddam, Wahid, & Pentland, 1998). They used the EM algorithm to analyze the output data, and made the recognition rate more reliable.

LDA is a statistical method. It was defined by Fisher in 1937 (Fisher, 1936), and can maximize the difference between classes by using within-class scatter and between-class scatter. The distance between classes will be enlarged after a training procedure. Georphiades et al. defined a space they called fisherspace derived from LDA (Georphiades, Belhumeur, & Kriegman, n.d.). The fisherspace can not only improve the accuracy of face recognition, but also reduce the influence of the lighting problem. We describe this algorithm in the next section. Additionally, we also discuss the limitation of LDA when it does face recognition in this chapter.

FACE RECOGNITION

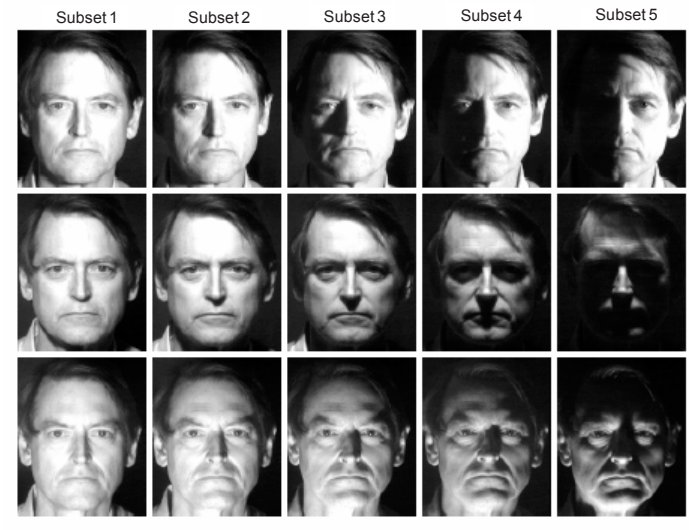
Previous researchers have developed numerous tools to increase the signal-to-noise ratio (Lin, 1997). To deal with complex image background, the recognizer requires a good face detector to isolate the real faces from other parts of the image. Illumination is often a major factor in the obstruction of the recognition process. To alleviate the influence of the illumination effect, people may take conventional image enhancement techniques (dynamic thresholding, histogram equalization) or train a neural network for feature extraction. Another approach to reduce the illumination effect is using the eigenface method. As will be mentioned later, the eigenface algorithm reduces the high-dimensional feature space into a low-dimensional subspace where most of the energy resides (i.e., eigenspace). According to the literature (Moghaddam, Wahid, & Pentland, 1998; Turk & Pentland, 1991a, 1991b; Lin, 1997), one or a few eigenfaces (terminology for the eigenvectors in the eigenface algorithm) could be used to represent the "illumination effect" on facial images. Therefore, putting lower weighting on those eigenfaces when doing the recognition reduces the illumination effect. Yet another remedy for illumination variation is using the fisherface algorithm. The fisherface algorithm is a refinement of the eigenface algorithm. It further reduces the eigenspace by the Fisher's linear discriminant (FLD). FLD selects the subspace in such a way that the ratio of the between-class scatter and the within-class scatter is maximized. It is reported that the fisherface algorithm outperforms the eigenface algorithm on the facial database with wide variation in lighting condition (Belhumeur, Hespanha, & Kriegman, 1997). (The detail of the fisherface algorithm will not be covered in this chapter. Interested readers please refer to Belhumeur, Hespanha, & Kriegman, 1997.)

In the following sections, we examine four pattern classification techniques for solving the face recognition problem (Belhumeur, Hespanha, & Kriegman, 1997), comparing methods that have become quite popular in the face recognition literature; that is, correlation and eigenface methods, with alternative methods developed by the authors. We approach this problem within the pattern classification paradigm, considering each of the pixel values in a sample image as a coordinate in a high-dimensional space (the image space).

Eigenface

The eigenface method is also based on linearly projecting the image space to a low-dimensional feature space (Belhumeur, Hespanha, & Kriegman, 1997; Sirovitdh & Kirby, 1987; Turk & Pentland, 1991a, 1991b). However, the eigenface method, which uses PCA

Figure 4.1. The same person seen under varying lighting conditions can appear dramatically different



Images are taken from the Harvard database (Belhumeur, Hepanha, & Kriegman, 1997)

for dimensionality reduction, yields projection directions that maximize the total scatter across all classes; that is, all images of all faces. In choosing the projection that maximizes total scatter, PCA retains some of the unwanted variations due to lighting and facial expression. As illustrated in Figure 4.1 and stated by Moses, Adini, and Ullman, the variations between the images of the same face due to illumination and viewing direction are almost always larger than image variations due to change in face identity (1997). Thus, while the PCA projections are optimal for reconstruction from a low-dimensional basis, they may not be optimal from a discrimination standpoint.

As mentioned, a technique now commonly used for dimensionality reduction in computer vision — particularly in face recognition — is PCA (Hallinan, 1994; Sirovitdh & Kirby, 1987; Turk & Pentland, 1991a, 1991b). PCA techniques, also known as Karhunen-Loeve methods, choose a dimensionality-reducing linear projection that maximizes the scatter of all projected samples.

More formally, let us consider a set of N sample images $\{x_1, x_2, \ldots, x_N\}$ taking values in an n-dimensional feature space, and assume that each image belongs to one of c classes $\{\chi_1, \chi_2, \ldots, \chi_c\}$. Let us also consider a linear transformation mapping the original n-dimensional feature space into an m-dimensional feature space, where $m < n$. Denoting by $W \in \Re^{n \times m}$ a matrix with orthonormal columns, the new feature vectors $y_k \in \Re^m$ are defined by the following linear transformation:

$$y_k = W^T x_k, \qquad k = 1, 2, \ldots, N \tag{4.1}$$

Let the total scatter matrix S_t be defined as Equation 3.41 mentioned in Chapter III.

Note that after applying the linear transformation, the scatter of the transformed feature vectors $\{y_1, y_2, \ldots, y_N\}$ is W^TSW. In PCA, the optimal projection W_{opt} is chosen to maximize the determinant of the total scatter matrix of the projected samples; that is:

$$W_{opt} = \arg\max_W \left| W^T S_t W \right| \tag{4.2}$$

Suppose that:

$$W_{opt} =: [\omega_1, \omega_2, ..., \omega_m] \tag{4.3}$$

where $\{\omega_i \mid i = 1, 2, ..., m\}$ is the set of n-dimensional eigenvectors of S_t corresponding to the m largest decreasing eigenvalues.

A drawback of this approach is that the scatter being maximized is not only due to the between-class scatter that is useful for classification, but also the within-class scatter that, for classification purposes, is unwanted information. Recall the comment by Moses et al.: Much of the variation from one image to the next is due to illumination changes (Craw, Tock, & Bennet, 1991). Thus, if PCA is presented with images of faces under varying illumination, the projection matrix W_{opt} will contain principal components (i.e., eigenfaces) that retain, in the projected feature space, the variation due lighting. Consequently, the points in projected space will not be well clustered and, worse, the classes may be smeared together.

It has been suggested that by throwing out the first several principal components, the variation due to lighting is reduced. The hope is that if the first principal components capture the variation due to lighting, then better clustering of projected samples is achieved by ignoring them. Yet, it is unlikely that the first several principal components correspond solely to variation in lighting; as a consequence, information that is useful for discrimination may be lost.

Fisherface

In this section we outline the fisherface method — one that is insensitive to extreme variations in lighting and facial expressions (Belhumeur, Hespanha, & Kriegman, 1997). Note that lighting variability includes not only intensity, but also direction and number of light sources. As seen in Figure 4.1, the same person, with the same facial expression, seen from the same viewpoint can appear dramatically different when light sources illuminate the face from different directions. Our approach to face recognition exploits two observations:

1. For a Lambertian surface without self-shadowing, all of the images of a particular face from a fixed viewpoint will lie in a 3-D linear subspace of the high-dimensional image space (Little & Boyd, 1998).
2. Because of expressions, regions of self-shadowing and specularity, the above observation does not exactly apply to faces. In practice, certain regions of the face may have variability from image to image that often deviates drastically from the linear subspace and, consequently, are less reliable for recognition.

We make use of these observations by finding a linear projection of the faces from the high-dimensional image space to a significantly lower-dimensional feature space that is insensitive both to variation in lighting direction and facial expression. We choose projection directions that are nearly orthogonal to the within-class scatter, projecting away variations in lighting and facial expression while maintaining discriminability. Fisherface maximizes the ratio of between-class scatter to that of within-class scatter.

We should point out that FLD (Fisher, 1936) is a "classical" technique in pattern recognition (Duda, Hart, & Stork, 2000) developed by Robert Fisher in 1936 for taxonomic classification. Depending on the features being used, it has been applied in different ways in computer vision and even in face recognition. Cui, Swets, and Weng applied Fisher's discriminator (using different terminology, they call it the most discriminating feature — MDF) in a method for recognizing hand gestures (Cui, Swets, & Weng, 1995). Though no implementation is reported, they also suggest that the method can be applied to face recognition under variable illumination.

We should also point out that we have made no attempt to deal with variation in pose. An appearance-based method such as ours can be easily extended to handle limited pose variation using either a multiple-view representation such as Pentland, Moghaddam and Starner's view-based eigenspace (Pentland, Moghaddam, & Starner, 1994) or Murase and Nayar's Appearance Manifolds. Other approaches to face recognition that accommodate pose variation include Beymer (1994). Furthermore, we assume that the face has been located and aligned within the image, as there are numerous methods for finding faces in scenes (Chen, Wu, & Yachida, 1995; Craw, Tock, & Bennet, 1992).

The *linear subspace* algorithm takes advantage of the fact that under ideal conditions, the classes are linearly separable (Little & Boyd, 1998; Belhumeur, Hespanha, & Kriegman, 1997). Yet, one can perform dimensionality reduction using linear projection and still preserve linear separability; error-free classification under any lighting conditions is still possible in the lower-dimensional feature space using linear decision boundaries. This is a strong argument in favor of using linear methods for dimensionality reduction in the face recognition problem, at least when one seeks insensitivity to lighting conditions.

Here we argue that by using class-specific linear methods for dimensionality reduction and simple classifiers in the reduced feature space, one gets better recognition rates in substantially less time than with the linear subspace method. Since the learning set is labeled, it makes sense to use this information to build a more reliable method for reducing the dimensionality of the feature space. (FLD, described in Chapter III, is an example of a class-specific method in the sense that it tries to "shape" the scatter to make it more reliable for classification. This method selects W in such a way that the ratio of the between-class scatter and the within-class scatter is maximized. Let the between-class scatter matrix $\mathbf{S}_b$ and the within-class scatter matrix $\mathbf{S}_w$ be defined as Equations 3.43 and 3.37, mentioned in Chapter III. If $\mathbf{S}_w$ is nonsingular, the optimal projection W_{opt} is chosen as that which maximizes the ratio of the determinant of the between-class scatter matrix of the projected samples to the determinant of the within-class scatter matrix of the projected samples; that is:

$$W_{opt} = \arg\max_{W} \frac{\left|W^T \mathbf{S}_b W\right|}{\left|W^T \mathbf{S}_w W\right|} = \left[w_1, w_2, ..., w_m\right] \tag{4.4}$$

where $\{w_i \mid i = 1, 2, ..., m\}$ is the set of generalized eigenvectors of $\mathbf{S}_b$ and $\mathbf{S}_w$ corresponding to a set of decreasing generalized eigenvalues $\{\lambda_i \mid i = 1, 2, ..., m\}$, which are mentioned in Equation 3.32. Note that an upper bound on m is c-1 where c is the number of classes.

To illustrate the benefits of the class-specific linear projections, we constructed a low-dimensional analogue to the classification problem in which the samples from each class lie near a linear subspace. Figure 4.2 is a comparison of PCA and FLD for a two-class problem in which the samples from each class are randomly perturbed in a direction perpendicular to the linear subspace. For this example, N = 20, n = 2 and m = 1. So the samples from each class lie near a line in the 2-D feature space. Both PCA and FLD have been used to project the points from 2-D down to 1-D. Comparing the two projections in the figure, PCA actually smears the classes together so that they are no longer linearly separable in the projected space. It is clear that although PCA achieves a larger total scatter, FLD achieves greater between-class scatter, and consequently, classification becomes easier.

In the face recognition problem, one is confronted with the difficulty that the within-class scatter matrix $\mathbf{S}_w \in \mathbb{R}^{n \times n}$ is always singular. This stems from the fact that the rank of $\mathbf{S}_w$ is less than $N - c$ and, in general, the number of pixels in each image (n) is much larger than the number of images in the learning set (N). This means that it is possible to choose the matrix W such that the within-class scatter of the projected samples can be made exactly zero.

To overcome the complication of a singular $\mathbf{S}_w$, we propose an alternative to the criterion in Equation 4.4. This method, which we call fisherfaces, avoids this problem by projecting the image set to a lower-dimensional space so that the resulting within-class scatter matrix $\mathbf{S}_w$ is nonsingular. This is achieved by using PCA to reduce the dimension of the feature space to $N - c$, and then applying the standard FLD defined by Equation 4.3 to reduce the dimension to $c - 1$. More formally, W_{opt} is given by:

$$W_{opt} = W_{fld} W_{pca} \tag{4.5}$$

where:

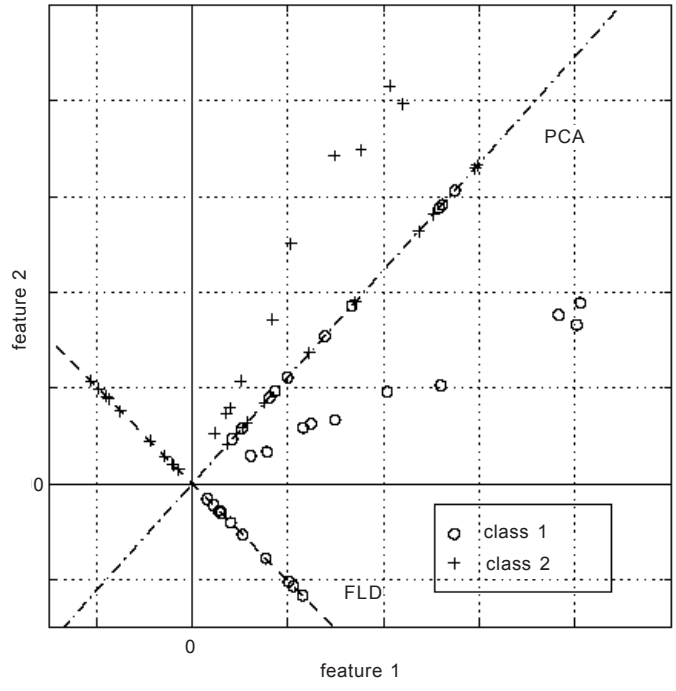
$$W_{fld} = \arg\max_W \left| W^T \mathbf{S}_t W \right| \tag{4.6}$$

$$W_{fld} = \arg\max_W \frac{\left| W^T W_{pca}^T \mathbf{S}_b W_{pca} W \right|}{\left| W^T W_{pca}^T \mathbf{S}_w W_{pca} W \right|} \tag{4.7}$$

Note that in computing W_{pca} we have thrown away only the smallest c principal components.

There are certainly other ways of reducing the within-class scatter while preserving between-class scatter. For example, a second method we are currently investigating chooses W to maximize the between-class scatter of the projected samples after having first reduced the within-class scatter. Taken to an extreme, we can maximize the between-

Figure 4.2. A comparison of PCA and FLD for a two-class problem, where data for each class lies near a linear subspace (Belhumeur, Hespanha, & Kriegman, 1997)



class scatter of the projected samples subject to the constraint that the within-class scatter is zero, that is:

$$W_{opt} = \arg\max_{W \in \mathrm{W}} \left| W^T \mathbf{S}_b W \right| \tag{4.8}$$

where W is the set of $n \times m$ matrices contained in the kernel of $\mathbf{S}_w$.

The fisherface method appears to be the best at extrapolating and interpolating over variations in lighting, although the linear subspacemethod is a close second. Removing the initial three principal components does improve the performance of the eigenface method in the presence of lighting variations, but does not alleviate the problem. In the limit as more principal components are used in the eigenface method, performance approaches that of correlation. Similarly, when the first three principal components have been removed, performance improves as the dimensionality of the feature space is increased. Note, however, that performance seems to level off at about 45 principal components. Sirovitch and Kirby found a similar point of diminishing returns when using eigenfaces to represent face images (Sirovitdh & Kirby, 1987). The fisherface method appears to be the best at simultaneously handling variations in lighting and expression. As expected, the linear subspacemethod suffers when confronted with variations in facial expression.

Experimental Results

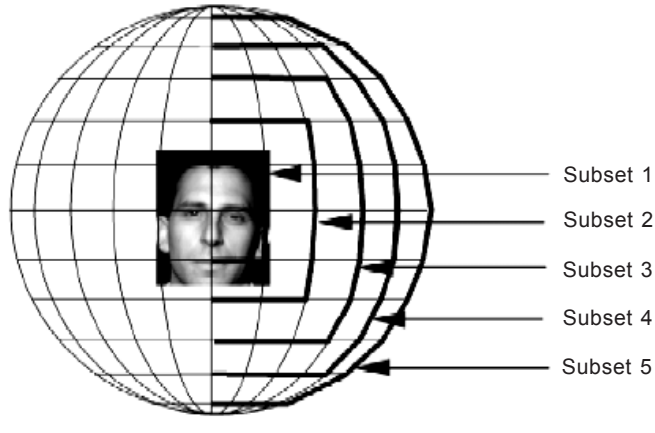
In this section, we present and discuss each of the aforementioned face recognition techniques using two different databases (Belhumeur, Hepanha, & Kriegman, 1997). Because of the specific hypotheses we wanted to test about the relative performance of the considered algorithms, many of the standard databases were inappropriate. So, we have used a database of 500 images from the Harvard Robotics Laboratory, in which lighting has been systematically varied. Second, we constructed a database of 176 images at Yale that includes variation in both facial expression and lighting.

Variation in Lighting

The first experiment was designed to test the hypothesis that under variable illumination, face recognition algorithms will perform better if they exploit the fact that images of a Lambertian surface lie in a linear subspace (Belhumeur, Hepanha, & Kriegman, 1997). More specifically, the recognition error rates for all four algorithms described earlier are compared using an image database constructed by Hallinan at the Harvard Robotics Laboratory (Hallinan, 1994, 1995). In each image in this database, a subject held his or her head steady while being illuminated by a dominant light source. The space of light source directions, which can be parameterized by spherical angles, was then sampled in 15º increments (see Figure 4.3). From this database, we used 330 images of five people (66 of each). We extracted five subsets to quantify the effects of varying lighting. Sample images from each subset are shown in Figure 4.1.

- **Subset 1:** Contains 30 images for which both the longitudinal and latitudinal angles of light source direction are within 15º of the camera axis, including the lighting direction coincident with the camera's optical axis.

Figure 4.3. The highlighted lines of longitude and latitude indicate the light source directions for Subsets 1 through 5

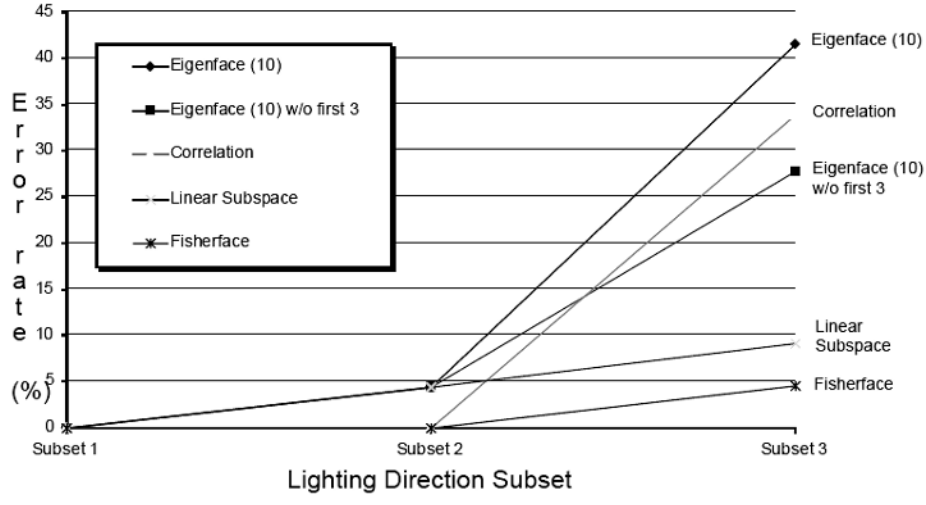


Each intersection of a longitudinal and latitudinal line on the right side of the illustration has a corresponding image in the database (Belhumeur, Hepanha, & Kriegman, 1997).

- **Subset 2:** Contains 45 images for which the greater of the longitudinal and latitudinal angles of light source direction are 30º from the camera axis.
- **Subset 3:** Contains 65 images for which the greater of the longitudinal and latitudinal angles of light source direction are 45º from the camera axis.
- **Subset 4:** Contains 85 images for which the greater of the longitudinal and latitudinal angles of light source direction are 60º from the camera axis.
- **Subset 5:** Contains 105 images for which the greater of the longitudinal and latitudinal angles of light source direction are 75º from the camera axis.

For all experiments, classification was performed using a nearest-neighbor classifier. All training images of an individual were projected into the feature space. The images were cropped within the face so that the contour of the head was excluded. For the eigenface and correlation tests, the images were normalized to have zero mean and unit

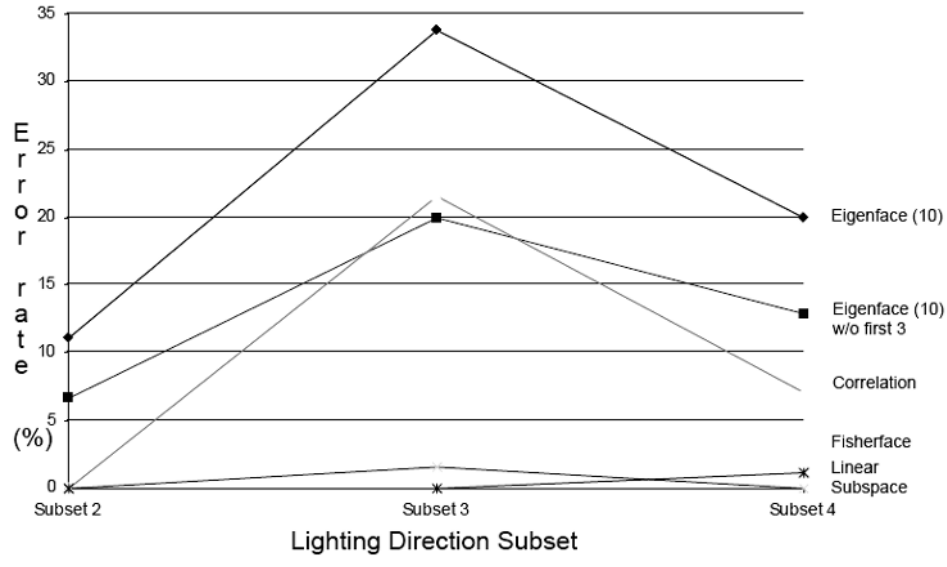
Figure 4.4. Extrapolation



Extrapolating from Subset 1				
Method	Reduced space	Error Rate (%)		
		Subset 1	Subset 2	Subset 3
Eigenface	4	0.0	31.1	47.7
	10	0.0	4.4	41.5
Eigenface w/o 1st 3	4	0.0	13.3	41.6
	10	0.0	4.4	27.7
Correlation	29	0.0	0.0	33.9
Linear Subspace	15	0.0	4.4	9.2
Fisherface	4	0.0	0.0	4.6

When each of the methods is trained on images with near frontal illumination (Subset 1), the graph and corresponding table show the relative performance under extreme light source conditions (Belhumeur, Hepanha, & Kriegman, 1997)

Figure 4.5. Interpolation



Interpolating between Subsets 1 and 5				
Method	Reduced space	Error Rate (%)		
		Subset 2	Subset 3	Subset 4
Eigenface	4	53.3	75.4	52.9
	10	11.11	33.9	20.0
Eigenface w/o 1 st 3	4	31.11	60.0	29.4
	10	6.7	20.0	12.9
Correlation	129	0.0	21.54	7.1
Linear Subspace	15	0.0	1.5	0.0
Fisherface	4	0.0	0.0	1.2

When each of the methods is trained on images from both near-frontal and extreme lighting (Subsets 1 and 5), the graph and corresponding table show the relative performance under intermediate lighting conditions (Belhumeur, Hespanha, & Kriegman, 1997)

variance, as this improved the performance of these methods. For the eigenface method, results are shown when 10 principal components were used. Since it has been suggested that the first three principal components are primarily due to lighting variation and that recognition rates can be improved by eliminating them, error rates are also presented using principal components four through 13.

We performed two experiments on the Harvard Database: extrapolation and interpolation. In the extrapolation experiment, each method was trained on samples from Subset 1 and then tested using samples from Subsets 1, 2 and 3. Since there are 30 images in the training set, correlation is equivalent to the eigenface method using 29 principal components. Figure 4.4 shows the results from this experiment.

In the interpolation experiment, each method was trained on Subsets 1 and 5 and then tested the methods on Subsets 2, 3 and 4. Figure 4.5 shows the results from this experiment.

These two experiments reveal a number of interesting points:

1. All of the algorithms perform perfectly when lighting is nearly frontal. However, as lighting is moved off axis, there is a significant performance difference between the two class-specific methods and the eigenface method.
2. The eigenface method is equivalent to correlation when the number of eigenfaces equals the size of the training set (Murase & Nayar, 1995) and since performance increases with the dimension of the eigenspace, the eigenface method should do no better than correlation (Brunelli & Poggio, 1993). This is empirically demonstrated as well.
3. In the eigenface method, removing the first three principal components results in better performance under variable lighting conditions.
4. While the linear subspacemethod has error rates competitive with the fisherface method, it requires storing more than three times as much information and takes three times as long.
5. The fisherface method had error rates lower than the eigenface method and required less computation time.

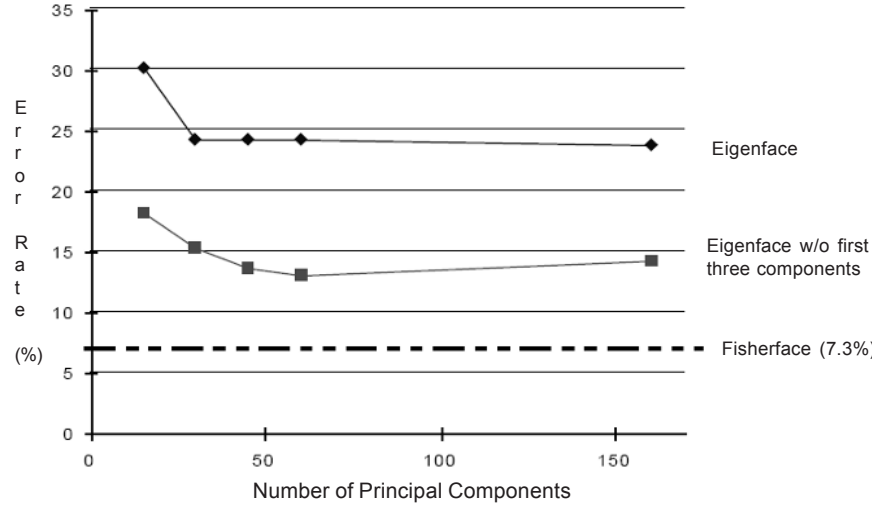
Variation in Facial Expression, Eye Wear and Lighting

Using a second database constructed at the Yale Center for Computational Vision and Control, we designed tests to determine how the methods compared under a different range of conditions (Belhumeur, Hespanha, & Kriegman, 1997). For 16 subjects, 10 images were acquired during one session in front of a simple background. Subjects included females and males; some with facial hair and some wore glasses. Figure 4.6 shows 10 images of one subject. The first image was taken under ambient lighting in a neutral facial expression and the person wore glasses. In the second image, the glasses

Figure 4.6. The Yale database contains 160 frontal face images covering 16 individuals taken under 10 different conditions

A normal image under ambient lighting, one with or without glasses, three with different point light sources, and five different facial expressions (Belhumeur, Hespanha, & Kriegman, 1997)

Figure 4.7. As demonstrated on the Yale database, the variation in performance of the eigenface method depends on the number of principal components retained



Dropping the first three appears to improve performance (Belhumeur, Hespanha & Kriegman, 1997)

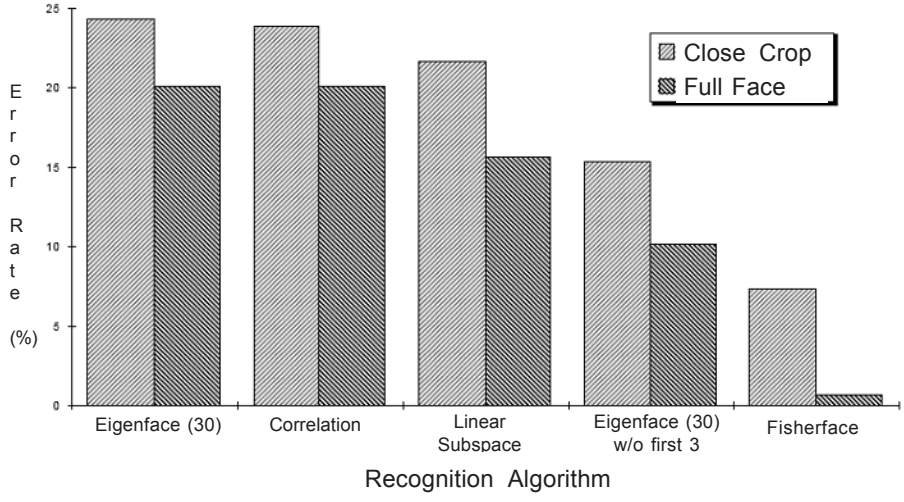
were removed. If the person normally wore glasses, those were used; if not, a random pair was borrowed. Images 3 through 5 were acquired by illuminating the face in a neutral expression with a Luxolamp in three positions. The last five images were acquired under ambient lighting with different expressions (happy, sad, winking, sleepy and surprised).

For the eigenface and correlation tests, the images were normalized to have zero mean and unit variance, as this improved the performance of these methods. The images were manually centered and cropped to two different scales: The larger images included the full face and part of the background, while the closely cropped ones included internal structures such as the brow, eyes, nose, mouth and chin, but did not extend to the occluding contour.

In this test, error rates were determined by the "leaving-one-out" strategy (Duda & Hart, 1973): To classify an image of a person, that image was removed from the data set and the dimensionality reduction matrix W was computed. All images in the database, excluding the test image, were then projected down into the reduced space to be used for classification. Recognition was performed using a nearest-neighbor classifier. Note that for this test, each person in the learning set is represented by the projection of 10 images, except for the test person, who is represented by only nine.

In general, the performance of the eigenface method varies with the number of principal components. Thus, before comparing the linear subspace and fisherface methods with the eigenface method, we first performed an experiment to determine the number of principal components yielding the lowest error rate. Figure 4.7 shows a plot of error rate vs. the number of principal components for the closely cropped set, when the initial three principal components were retained and when they were dropped.

Figure 4.8. The graph and corresponding table show the relative performance of the algorithms when applied to the Yale database, which contains variations in facial expression and lighting (Belhumeur, Hespanha, & Kriegman, 1997)



"Leaving-One-Out: of Yale Database			
Method	Reduced space	Error Rate (%)	
		Close Crop	Full Face
Eigenface	30	24.4	19.4
Eigenface w/o 1st 3	30	15.3	10.8
Correlation	160	23.9	20.0
Linear Subspace	48	21.6	15.6
Fisherface	15	7.3	0.6

The relative performance of the algorithms is self-evident in Figure 4.8. The fisherface method had error rates that were better than half that of any other method. It seems that the fisherface method chooses the set of projections that perform well over a range of lighting variation, facial expression variation and presence of glasses.

Note that the linear subspace method faired comparatively worse in this experiment than in the lighting experiments in the previous section. Because of variation in facial expression, the images no longer lie in a linear subspace. Since the fisherface method tends to discount those portions of the image that are not significant for recognizing an individual, the resulting projections *W* tend to mask the regions of the face that are highly variable. For example, the area around the mouth is discounted, since it varies quite a bit for different facial expressions. On the other hand, the nose, cheeks and brow are stable over the within-class variation and are more significant for recognition. Thus, we conjecture that fisherface methods, which tend to reduce within-class scatter for all

classes, should produce projection directions that are also good for recognizing other faces besides the ones in the training set.

All of the algorithms performed better on the images of the full face. Note that there is a dramatic improvement in the fisherface method, where the error rate was reduced from 7.3% to 0.6%. When the method is trained on the entire face, the pixels corresponding to the occluding contour of the face are chosen as good features for discriminating between individuals; that is, the overall shape of the face is a powerful feature in face identification. As a practical note, however, it is expected that recognition rates would have been much lower for the full-face images if the background or hair styles had varied and may even have been worse than the closely cropped images.

Glasses Recognition

When using class-specific projection methods, the learning set can be divided into classes in different manners (Belhumeur, Hespanha, & Kriegman, 1997). For example, rather than selecting the classes to be individual people, the set of images can be divided into two classes: "wearing glasses" and "not wearing glasses." With only two classes, the images can be projected to a line using the fisherface methods. Using PCA, the choice of the eigenfaces is independent of the class definition.

In this experiment, the data set contained 36 images from a superset of the Yale database, half with glasses. The recognition rates were obtained by cross validation; that is, to classify the images of each person, all images of that person were removed from the database before the projection matrix W was computed. Table 4.1 presents the error rates for two different methods.

PCA had recognition rates near chance, since, in most cases, it classified both images with and without glasses to the same class. On the other hand, the fisherface methods can be viewed as deriving a template suited for finding glasses and ignoring other characteristics of the face. This conjecture is supported by observing the fisherface in Figure 4.9 corresponding to the projection matrix W. Naturally, it is expected that the same techniques could be applied to identifying facial expressions where the set of training images is divided into classes based on the facial expression.

Remarks

The experiments suggest a number of conclusions (Belhumeur, Hespanha, & Kriegman, 1997):

1. All methods perform well if presented with an image in the test set that is similar to an image in the training.
2. The fisherface method appears to be the best at extrapolating and interpolating over variation in lighting, although the linear subspacemethod is a close second.
3. Removing the largest three principal components does improve the performance of the eigenface method in the presence of lighting variation, but does not achieve error rates as low as some of the other methods described here.
4. In the limit, as more principal components are used in the eigenface method, performance approaches that of correlation. Similarly, when the first three principal components have been removed, performance improves as the dimensionality of the feature space is increased. Note, however, that performance seems to level off

at about 45 principal components. Sirovitch and Kirby found a similar point of diminishing returns when using eigenfaces to represent face images (Sirovitdh & Kirby, 1987).

5. The fisherface method appears to be the best at simultaneously handling variation in lighting and expression. As expected, the linear subspacemethod suffers when confronted with variation in facial expression.

Even with this extensive experimentation, interesting questions remain: How well does the fisherface method extends to large databases. Can variation in lighting conditions be accommodated if some of the individuals are only observed under one lighting condition?

Additionally, current face detection methods are likely to break down under extreme lighting conditions, such as Subsets 4 and 5 in Figure 4.1, and so new detection methods are needed to support the algorithms presented in this chapter. Finally, when shadowing dominates, performance degrades for all of the presented recognition methods, and techniques that either model or mask the shadowed regions may be needed. We are currently investigating models for representing the set of images of an object under *all possible* illumination conditions, and have shown that the set of *n*-pixel images of an

Figure 4.9. The left image is an image from the Yale database of a person wearing glasses. The right image is the fisherface used for determining if a person is wearing glasses. (Belhumeur, Hespanha, & Kriegman, 1997)

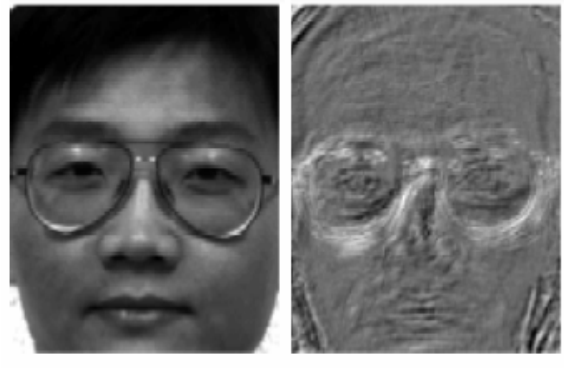


Table 4.1. Comparative recognition error rates for glasses/no glasses recognition using the Yale database (Belhumeur, Hespanha, & Kriegman, 1997)

Glasses Recognition		
Method	Reduced Space	Error Rate (%)
PCA	10	52.6
Fisherface	1	5.3

object of any shape and with an arbitrary reflectance function, seen under all possible illumination conditions, forms a convex cone in R^n (Belhumeur & Kriegman, 1996). Furthermore, and most relevant to this section, it appears that this convex illumination cone lies close to a low-dimensional linear subspace (Hallinan, 1994).

PALMPRINT IDENTIFICATION

Fisherpalm

In this section, a novel method for palmprint recognition, called fisherpalm (Wu, Zhang, & Wang, 2003), is introduced. In this method, each pixel of a palmprint image is considered as a coordinate in a high-dimensional image space. A linear projection based on Fisher's linear discriminant is used to project palmprints from this high-dimensional original palmprint space to a significantly lower-dimensional feature space (fisherpalm space), in which the palmprints from the different palms can be discriminated much more efficiently. The relationship between the recognition accuracy and the resolution of the palmprint image is also investigated. The experimental results show that, in this introduced method, the palmprint images with resolution 32×32 are optimal for medium-security biometric systems, while those with resolution 64×64 are optimal for high-security biometric systems. High accuracies (>99%) have been obtained by the proposed method, and the speed of this method (responding time ≤0.4s) is rapid enough for real-time palmprint recognition.

Introduction to Palmprint

Computer-aided personal recognition is becoming increasingly important in our information society. Biometrics is one of the most important and reliable ways in this field. Palmprint (Zhang, 2000; Zhang & Shu, 1999; Shu, Rong, Bain, & Zhang, 2001; Shu & Zhang, 1998), as a new biometric feature, has several advantages: low-resolution imaging can be employed; low-cost capture devices can be used; it is very difficult, if not impossible, to fake a palmprint; the line features of the palmprints are stable; and so forth. It is for these reasons that palmprint recognition has recently attracted an increasing amount of attention from researchers. There are many approaches for palmprint recognition in various literature, most of which are based on structural features (Zhang & Shu, 1999; Duda, Hart, & Stork, 2001), statistical features (Li, Zhang, & Xu, 2002) or the hybrid of these two types of features (You, Li, & Zhang, 2002). However, structural features, such as principal lines, wrinkles, delta points, minutiae (Zhang, 2000; Zhang & Shu, 1999), feature points (Duta, Jain, & Mardi, 2001) and interesting points (You, Li, & Zhang, 2002), are difficult to be extracted, represented and compared, while the discriminability of statistical features such as texture energy (Zhang, 2000; You, Li, & Zhang, 2002) is not strong enough for palmprint recognition. To overcome these problems, another type of features, called algebraic features (Liu, Cheng, & Yang, 1993), is extracted from palmprints for identity recognition in this section. Algebraic features, which represent intrinsic attributions of an image, can be extracted based on various algebraic transforms or matrix decompositions (Liu, Cheng, & Yang, 1993). FLD (Duta, Jain, & Mardia, 2001) is an efficient approach to extract the algebraic features that have strong discriminability.

FLD, which is based on linear projections, seeks the projection directions that are advantageous for discrimination. In other words, the class separation is maximized in these directions. Figure 4.2 illustrates intuitively the principle of FLD. In this figure, the samples, which are in the 2D feature space, are from two classes. Obviously, it is difficult to tell apart them in the original 2D space. However, if we use FLD to project these data from 2D to 1D, we can easily to discriminate them in such 1D space. This approach has been widely used in pattern recognition. Fisher (1936) first proposed this approach for taxonomic classification. Cui, Swets, and Weng (1995) employed a similar algorithm (MDF) for hand sign recognition. Liu et al. (1993) adopted FLD to extract the algebraic features of handwritten characters. Belhumeur et al. (1997) developed a very efficient approach (fisherface) for face recognition. In this section, a novel palmprint recognition method, called Fishpalm, is proposed based on FLD. In this method, each palmprint image is considered as a point in a high-dimensional image space. A linear projection based on FLD is used to project palmprints from this high-dimensional space to a significantly lower-dimensional feature space, in which the palmprints from the different palms can be discriminated much more efficiently. The relationship between the recognition accuracy and the resolution of the palmprint image is also investigated.

When palmprints are captured, the position, direction and stretching degree of a palm may vary from time to time. Therefore, even palmprints from the same palm may have a little rotation and shift. Furthermore, the sizes of palms are different from one another. Hence, palmprint images should be oriented and normalized before feature extraction and matching. In our CCD-based palmprint capture device, there are three pegs between the first finger and middle finger, between the middle finger and the third finger, and between the third finger and little finger to limit the palm's shift and rotation. These pegs make the fingers stretch so that they do not touch each other; thus, three holes are formed between these fingers. In this section, the centers of gravities of these holes are used to align the palmprints, and the central part of the image, whose size is 128×128, is cropped to represent the whole palmprint (Li, Zhang, & Xu, 2002).

Fisherpalms Extraction

An $N \times N$ palmprint image can be considered as an N^2 vector and each pixel corresponds to a component. That is, $N \times N$ palmprint images can be regarded as points in a high-dimensional space (N^2-dimensional space), called the *original palmprint space* (OPS). Generally, the dimension of the OPS is too high to be used directly. For example, the dimension of the original 128×128 palmprint image space is 16,384. We should, therefore, reduce the dimension of the palmprint image and, at the same time, improve or keep the discriminability between palmprint classes. A linear projection based on FLD, thus, is selected for this purpose. Let us consider a set of N palmprints $\{x_1, x_2, \ldots, x_N\}$ taking values in an n-dimensional OPS, and assume that each image is captured from one of c palms $\{X_1, X_2, \ldots, X_c\}$ and the number of images from X_i ($i = 1, 2, \ldots, c$) is N_i. FLD tries to find a linear transformation W_{opt} to maximize the Fisher criterion (Duta, Jain, & Mardia, 2001); we also can see it from Equation 3.53, and according to Equations 3.37 through 3.43, we can get Equations 4.9 to 4.12:

$$J(W) = \frac{|W^T S_b W|}{|W^T S_w W|} \tag{3.53}$$

where S_b and S_w are the between-class scatter matrix and the within-class scatter matrix, and according to Equations 3.37 through 3.43, we can get Equations 4.9 to 4.12:

$$S_b = \sum_{i=1}^{c} N_i(\mu_i - \mu)(\mu_i - \mu)^T \tag{4.9}$$

$$S_w = \sum_{i=1}^{c} \sum_{x_k \in X_i} (x_k - \mu_i)(x_k - \mu_i)^T \tag{4.10}$$

$$\mu_i = \frac{1}{N_i} \sum_{x_l \in X_i} x_l \tag{4.11}$$

$$\mu = \frac{1}{N} \sum_{j=1}^{N} x_j \tag{4.12}$$

The optical linear transformation W_{opt} can be obtained from Equation 4.4 as follows:

$$W_{opt} = \arg\max \frac{|W^T S_b W|}{|W^T S_w W|} = |w_1, w_2, \cdots, w_m| \tag{4.4}$$

where $\{w_i \mid i=1,2,\ldots,m\}$ is the set of generalized eigenvectors of S_b and S_w corresponding to the m nonzero generalized eigenvalues $\{\lambda_i \mid i=1, 2, \ldots, m\}$, like Equation 3.32:

$$S_b w_i = \lambda_i S_w w_i, \quad i=1,2,\ldots,m$$

There are at most c-1 nonzero generalized eigenvalues (Duta, Jain, & Mardi, 2001; hence, the upper bound of m is c - 1, where c is the number of palmprint classes.

Obviously, up to now, all this discussion is based on the assumption that the denominator of Equation 3.53 does not equal zero; that is, S_w is of full rank. However, in general, S_w is not a full rank matrix. This stems from the fact that the rank of S_w is at most N-c, and, in general, the number of images in the training set N is much smaller than the number of pixels in each image n. This means that it is possible to choose the matrix W_{opt} such that the within-class scatter of the projected samples — that is, the denominator of Equation 3.53 — can be made exactly zero. Thus, we cannot use Equations 3.53, 4.4 and 3.32 to obtain W_{opt} directly. To overcome this problem, the original palmprint images are first projected to a lower-dimensional space by using Karhunen-Loeve (K-L) transformation so that the resulting within-class scatter matrix is nonsingular. Then, the standard FLD is employed to process the projected samples. This method, which has been used efficiently in face recognition (Belhumeur et al., 1997), is described as below:

1. Compute the transformation matrix of K–L transformation U_{KL}:

$$U_{KL} = \arg\max_{U} |U^T S_t U| = |u_1, u_2, \cdots, u_n| \tag{4.13}$$

where $\{u_i | i = 1, 2, \ldots, n\}$ is the set of eigenvector of S_t corresponding to the nonzero eigenvalues and S_T is the total scatter matrix defined as:

$$S_t = \sum_{k=1}^{N} (x_k - \mu)(x_k - \mu)^T \tag{4.14}$$

2. Compute the transformed within-class scatter matrix S'_w, which is a full rank matrix:

$$S'_w = U_{KL}^T S_w U_{KL} \tag{4.15}$$

3. Compute the transformed between-class scatter matrix S'_b:

$$S'_b = U_{KL}^T S_b U_{KL} \tag{4.16}$$

4. The standard FLD defined by Equation 4.3 is used to the transformed samples to obtain W_{fld}, and we can also see Equation 4.6:

$$W_{fld} = \arg\max_W \frac{|W^T S'_b W|}{|W^T S'_w W|} = \arg\max_W \frac{|W^T U_{KL}^T S_b U_{KL} W|}{|W^T U_{KL}^T S_w U_{KL} W|}$$

5. Compute W_{opt}:

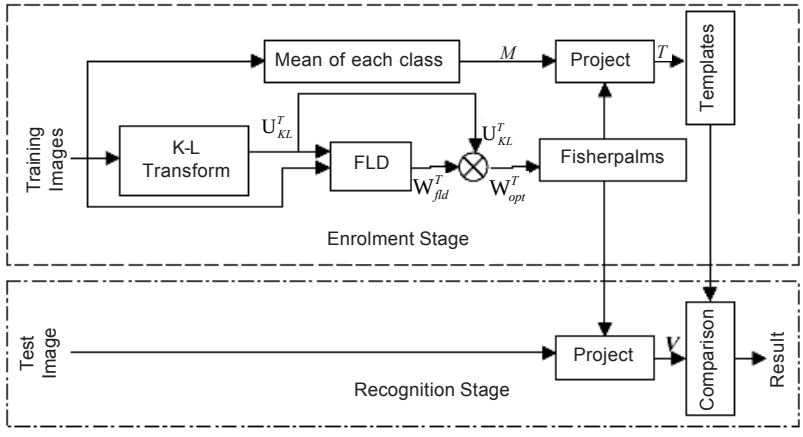
$$W_{opt}^T = W_{fld}^T U_{KL}^T \tag{4.17}$$

The columns of W_{opt} $\{w_1, w_2, \ldots, w_m\}$ $(m \le c - 1)$ are orthonormal vectors. The space spanned by these vectors is called the fisherpalm space (FPS), and the vector w_i (i =1,

Figure 4.10. An example of the fisherpalm in the case of two palmprint classes

(a) and (b) are samples in the class one and two, respectively, and (c) is the fisherpalm used for classification

Figure 4.11. Block diagram of the fisherpalms-based palmprint recognition



$2, \ldots, m; m \leq c-1$) is called a fisherpalm. Figure 4.10 shows an example of the fisherpalm in the case of two palmprint classes. In this case, there is only one fisherpalm in W_{opt}. In this figure, (a) and (b) are samples in class one and two, respectively, and (c) is the fisherpalm used for classification.

The block diagram of the fisherpalms-based palmprint recognition is shown in Figure 4.11. There are two stages in our system: the enrollment stage and the recognition stage. In the enrollment stage, the fisherpalms W_{opt} are computed by using the training samples (Equation 4.26-4.41) and stored as an FPS at first, and then the mean of each palmprint class is projected onto this FPS:

$$T = W_{opt}^{W} M \tag{4.18}$$

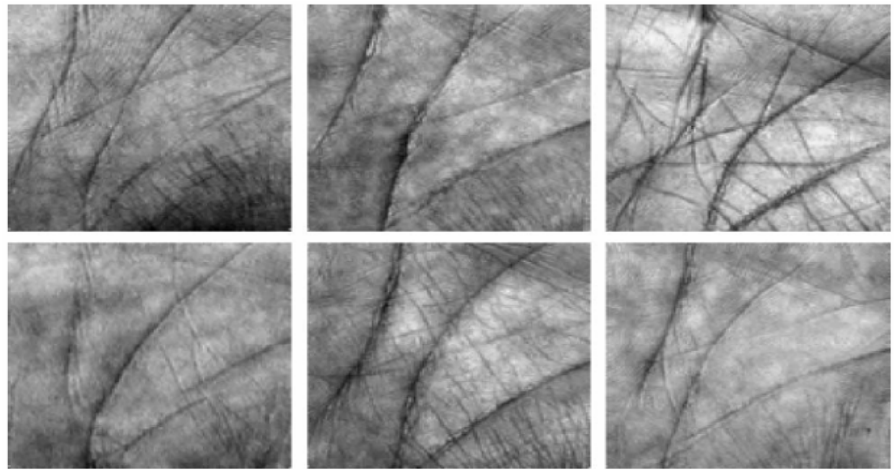
where $M = \{m_1, m_2, \ldots, m_c\}$, c is the number of palmprint classes and each column of M, m_i (i=1, 2, . . . , c) is the mean of the ith class palmprints. T is stored as the template for each palmprint class. In the recognition stage, the input palmprint image is projected onto the stored FPS to get its feature vector V, and then V is compared with the stored templates to obtain the recognition result.

Experimental Results and Analyses

We have collected palmprint images from 300 different palms using our CCD-based palmprint device to establish a palmprint database. Each palm is captured 10 times. Therefore, there are 3000 palmprints in our database. The resolution of all original palmprint images is 384×284 pixels at 75 dpi. By using the preprocessing approach (Li, Zhang, & Xu, 2002), palmprints are oriented, and the central part of the image, whose size is 128×128, is cropped to represent the whole palmprint. Some samples in our database are shown in Figure 4.12. Obviously, the dimension of the OPS is 128×128=16,384.

In biometric systems, there exist two limitations of this high dimension (Yuela, Dai, & Feng, 1998): First, the recognition accuracy will decrease dramatically when the number

Figure 4.12. Some typical samples from our database



of image classes increases. In face recognition, the typical size of training images is around 200. Second, it results in high computation complication, especially when the number of the classes is large. To overcome these limitations, we should reduce the resolution of the palmprint images. In face recognition, the image with 16×16 resolution is sufficient for distinguishing a human face (Yuela, Dai, & Feng, 1998; Harmon, 1973). To investigate the relationship between the recognition accuracy and the resolution of palmprint images, the original images are decomposed into a Gaussian pyramid (see Figure 4.13) and the images at each level are tested. The original image is at the bottom of the pyramid (0th level) and the images at ith level (i =1,…,5) of the pyramid are obtained as follows: convolve the images at $(i-1)$th level with a Gaussian kernel and then subsample the convolved images. The resolution of ith level image is $2^{7-i}\times 2^{7-i}$, where i =0,1,…,5. At each level, six images of each palmprint class are randomly chosen as training samples to form the template, and the remaining four images are used as test samples. All of the experiments are conducted using Microsoft Windows 2000 and Matlab 6.1 with image processing toolbox on a personal computer with an Intel Pentium III processor (900 MHz).

To analyze the relationship between the performance of the proposed method and image resolution, the feature vector of each testing palmprint is matched against each stored template at each level. A genuine matching is defined as the matching between the palmprints from the same palm and an imposter matching is the matching between the palmprints from different palms. A total of 360,000 (4×300×300) comparisons are performed at each level, in which 1,200 (4×300) comparisons are genuine matching. The genuine and impostor distributions at each pyramid level are plotted in Figure 4.14. It can be seen from this figure that there exists two peaks in the distributions at each level. One peak corresponds to genuine matching and the other corresponds to impostor matching. When the distance threshold is set as the one corresponding to the intersection of genuine and impostor distribution curves, the total error reaches the minimum, and the corresponding threshold, false accept rate (FAR), false reject rate (FRR) and half total

Figure 4.13. An example of Gaussian pyramid decomposition of a palmprint image

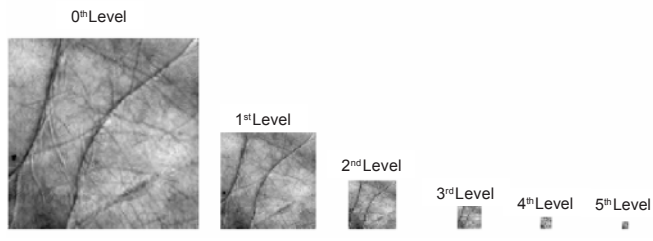


Table 4.2. FAR, FRR and HTER of each pyramid level

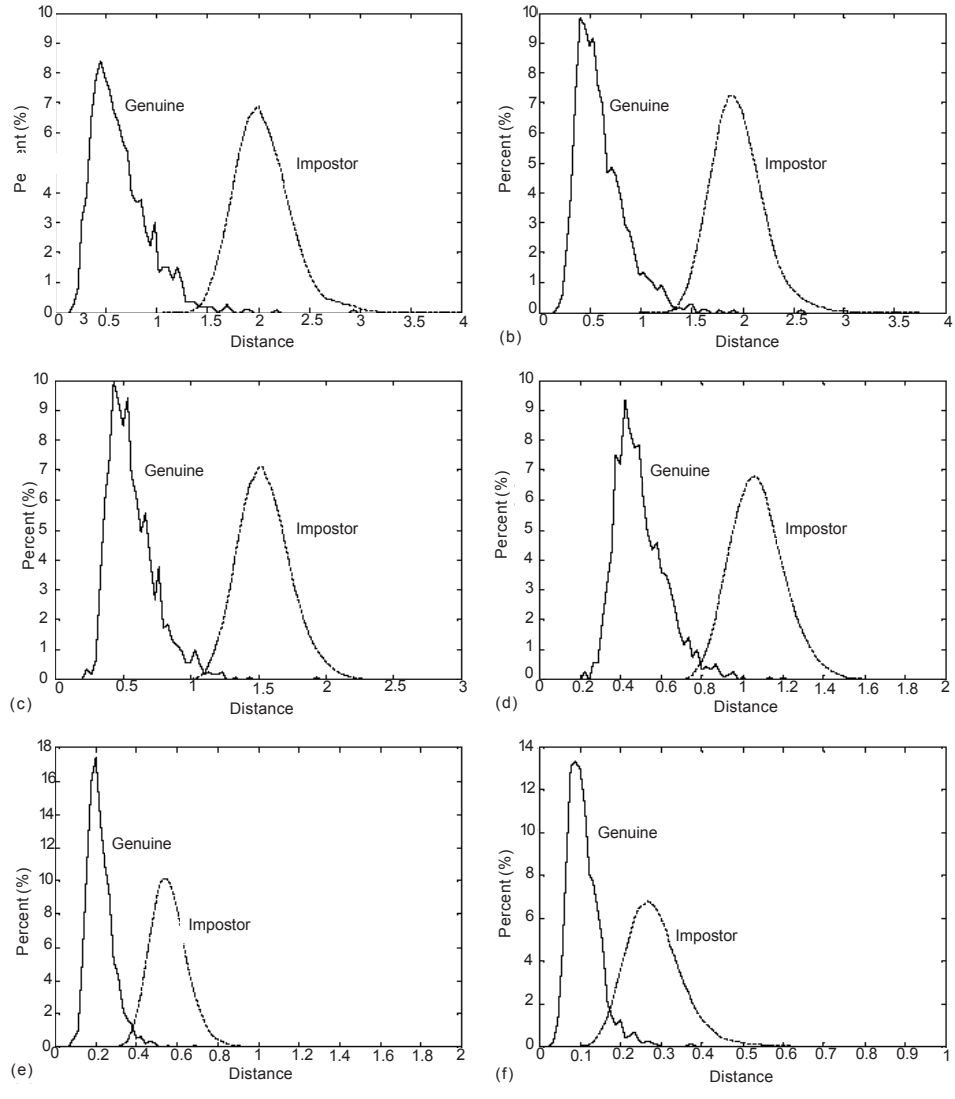
Pyramid level	0th	1st	2nd	3rd	4th	5th
Distance threshold	1.41	1.33	1.09	0.79	0.38	0.17
FAR (%)	0.1628	0.0814	0.1800	0.6491	1.4507	3.2302
FAA (%)	1.4167	1.2500	1.2238	2.5833	1.9167	6.3333
HTER (%)	0.7897	0.6657	0.7019	1.6162	1.6837	4.7817

error rate (HTER, which equals (FAR+FRR)/2) (Bengio, Mariethoz, & Maroel, 2001) at each level are listed in Table 4.2. According to this table, HTERs of 0th, 1st and 2nd level are much less than those of other levels. In other words, the palmprint image with 128×128, 64×64 and 32×32 resolution are more suitable for fisherpalms-based palmprint recognition than the other resolutions. Because the differences of HTERs at 0th, 1st and 2nd level are very little (<0.1), it is difficult to decide which level is optimal for identity recognition.

A further analysis of the images at these three levels (0th, 1st and 2nd) is made by considering their receiver operating characteristic (ROC) curves, which plots the FAR against the FRR (Bengio, Mariethoz, & Maroel, 2001). Figure 4.15 plots the ROC curves of 0th, 1st and 2nd level and the corresponding equal error rate (EER, where FAR = FRR). From this figure, the whole curve of 1st level is below that of 0th level. Hence, the palmprints at 1st level are better than those at 0th level in the proposed method. This figure also shows that the curve of the 2nd level is below that of the 1st level when the FAR is in the interval [0.55, 1.87] (between the magenta dotted lines in Figure 4.14, the corresponding FRR is in [0.67, 1.0]), and the curve of the 1st level is below that of the 2nd level when the FAR is smaller than 0.55 (the corresponding FRR is larger than 1.0). Therefore, images at the 2nd level (32×32 resolution) are optimal for medium-security systems, such as some civil systems, in which FRR should be low; while the images at the 1st level (64×64 resolution) are optimal for high-security systems, such as some military systems, in which FAR should be low. The EER of the 0th, 1st and 2nd level are 1.00%, 0.95% and 0.82%, respectively (see Table 4.3).

Other experiments about palmprint identification (1-to-300 matching) are also done by using the images at the 0th, 1st and 2nd level. A nearest-neighbor classifier based on

Figure 4.14. Genuine and imposter distributions at each pyramid level



(a) 0th level, (b) 1st level, (c) 2nd level, (d) 3rd level, (e) 4th level and (f) 5th level

Euclidean distance is employed. The identification rates of 0th, 1st and 2nd level are 99.20%, 99.25% and 99.75%, respectively. The testing time and identification accuracies of these levels are listed in Table 4.3. The rates and respond time can meet the requirement of an online palmprint recognition system. The training time of these levels is also listed in Table 4.3.

Comparisons have been conducted among our method, Duta's approach (Duda, Gart, & Stork, 2001) and Li's algorithm (Li, Zhang, & Xu, 2002). In Duta's approach, the lines of a palmprint were first extracted by directly binarizing the off-line palmprint images

Figure 4.15. ROC curves at the 0th, 1st and 2nd level

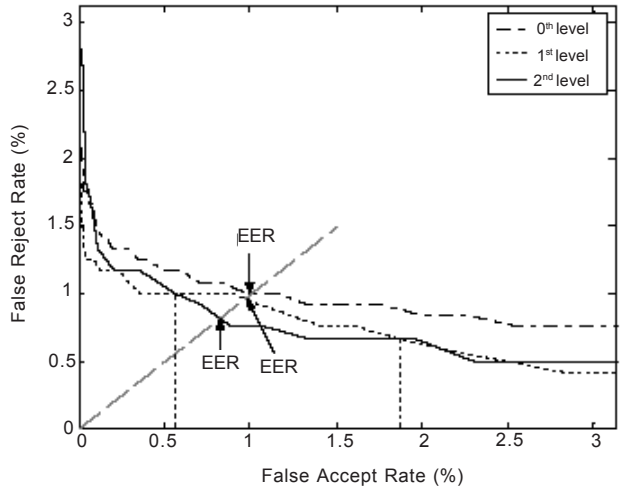


Table 4.3. Performance of the proposed method at 1st and 2nd pyramid level

Pyramid level		0th	1st	2nd
Resolution		128*128	64*64	32*32
train	Training times	1070	47	12
One-to-one matching test	Equal error rate (%)	1.00	0.95	0.82
One-to-300 matching test	Accuracy (%)	99.20	99.25	99.75
	Testing times (s)	0.40	0.36	0.34

(which were obtained by pressing an inked palm on a paper) with an interactively chosen threshold, and then some feature points and their orientation were extracted from these lines to verify the identity. Thirty off-line images with 400×300 resolution captured from three persons were used, and 95% accuracy was obtained in the one-to-one matching test in their experiments. The feature points in this approach belonged to structural features of palmprints.

It is evident that the recognition accuracies of this approach depend heavily on the result of the line extraction. Because of noise and unexpected disturbance — such as the movement of the hand, lighting, settings and so forth — the online palmprints (which are captured online by a CCD camera-based device) have much worse quality than off-line images. Thus it is much more difficult to extract lines from the online palmprint images. There is no effective line extraction method for online palmprints yet. Therefore, we only use Duta's (Duda, Hart, & Stork, 2001) experimental results here for comparison. In Li's algorithm (2002), the R feature and h feature of the palmprint, which belonged to statistical

Table 4.4. Comparison of different palmprint recognition methods

Method	Duta's approach(2001)	Li's algorithm(2002)			Our method		
Database size	30 images (from 3 palm)	3000 images (from 300 palms)			3000 images (from 300 palms)		
Feature extraction	Feature points	R feature and θ feature			Fisherpalms		
Feature type	Structural feature	Statistical feature			Fisherpalms		
Image resolution	400*300	128*128	64*64	32*32	128*128	64*64	32*32
One-to-one matching accurate rates (%)	95.00	96.40	95.2	93.24	99.00	99.05	99.18
One-to-many matching accurate rates (%)	Not presented	94.67	93.00	90.03	99.20	99.25	99.75

features, were extracted from the frequency domain to identify different persons. R features showed the intensity of the lines of a palmprint and h features showed the direction of these lines. However, all of these features could not reflect the spatial position of these lines, since they were extracted in frequency domain. Thus, their abilities to discriminate palms were not strong. Li's algorithm for 128×128, 64×64 and 32×32 resolutions has been implemented in our database. The corresponding accuracies in the one-to-one matching are 96.40%, 95.02% and 93.24%, respectively. The accuracies in the 1-to-300 matching are 94.67%, 93.00% and 90.33%, respectively. Obviously, the results of our method are much better than those of Duta's and Li's. This is because the fisherpalms method, which is based on algebraic transforms and matrix decompositions, has none of the mentioned shortcomings of Duta's approach and Li's algorithm and, at the same time, the extracted algebraic features can represent the intrinsic attributions of palmprints. Table 4.4 summarizes the feature of our method and these two approaches with respect to database size, image resolution, feature type, feature extraction and accuracy.

Conclusions and Future Work

Palmprint is an important complement of personal identification. There are many features in a palmprint, such as structural features, statistical features and so forth. In this section, we try to extract another type features, algebraic features, from palmprint. The novel palmprint recognition method proposed in this section is called fisherpalms. FLD is used to project the palmprint image from the very high-dimensional OPS to the very low-dimensional FPS, in which the ratio of the determinant of the between-class scatter to that of the within-class scatter is maximized. It shows that, in the fisherpalms-based palmprint recognition system, the images with resolution 32×32 are optimal for medium-security biometric systems, while those with resolution 64×64 are optimal for high-security biometric systems. For palmprints with resolution 32×32, accuracies of 99.18% and 99.75% are obtained in one-to-one matching test and one-to-300 matching tests, respectively. For palmprints with resolution 64×64, these accuracies are 99.05% and 99.25%. And for palmprints with resolution 128×128, accuracies of 99.00% and

99.20% are obtained in one-to-one matching and one-to-300 matching tests, respectively. The average testing time for the images with these resolutions in 1-to-300 matching is not more than 0.4s, which is short enough for real-time palmprint recognition. Some more features of the proposed method, such as the robustness to rotation and translation, the effect of noises and illuminations and so forth, will be investigated in future work.

Eigenpalm

In this section, we propose a palmprint recognition method based on eigenspace technology. By means of the Karhunen–Loeve transform, the original palmprint images are transformed into a small set of feature space, called "eigenpalms" (Lu, Zhang, & Wang, 2003), which are the eigenvectors of the training set and can represent the principle components of the palmprints quite well. Then, the eigenpalm features are extracted by projecting a new palmprint image into the subspace spanned by the "eigenpalms", and applied to palmprint recognition with a Euclidean distance classifier. Experimental results illustrate the effectiveness of our method in terms of the recognition rate.

Definitions and Notations

Compared with other biometrics technologies, palmprint has become an important complement to personal identification because of its advantages, such as low resolution, low cost, non-intrusiveness and stable structure features (Duta, Jain, & Mardia, 2002; You, Li, & Zhang, 2002).

The palm, the inner surface of the hand between the wrist and the fingers, consists of three parts: the finger-root region, inside region and outside region. There are three principle lines made by flexing the hand and wrist in the palm, which are usually defined as life line, heart line and head line (Shu & Zhang, 1998). The previous work on palmprint recognition focused on two aspects: (1) extracting the principle lines and creases in the spatial domain (Zhang & Shu, 1999; Duta, Jain, & Mardia, 2002; You, Li, & Zhang, 2002); and (2) transforming the palmprint images into the frequency domain to obtain the energy distribution feature (Wang, Ning, Tan, & Hu, 2004). In the first approach, the lines and creases of a palm are sometimes difficult to extract directly from a given palmprint image with low resolution. The recognition rates and computational efficiency are also not sufficient. In the second approach, the abundant textural details of a palm are ignored and the extracted features are greatly affected by the lighting conditions. The problems with these two approaches suggest that new methods are required for palmprint recognition.

The concept of an eigenspace has been widely used in face recognition. That work shows that the extracted "eigenfaces" can effectively represent the principal components of the faces (Peng & Zhang, 1997; Turk & Pentland, 1991b). In this section, we find that it also offers good characteristics for palmprint recognition. Based on the K-L transform, the original palmprint images used in training are transformed into a small set of characteristic feature images, called "eigenpalms," which are the eigenvectors of the training set. Then, feature extraction is performed by projecting a new palmprint image into the subspace spanned by the "eigenpalms."

When capturing a palmprint, the position, direction and stretching degree may vary from time to time. As a result, even the palmprints from the same palm could have a little

rotation and shift. Also, the sizes of palms are different from one another. It is necessary to align all palmprints and normalize their sizes for further feature extraction and matching (Wang, Ning, Tanand, & Hu, 2004). In our biometrics research laboratory, a palmprint input device can capture online palmprint images. Both the rotation and translation are corrected by the capture device panel, which can locate the palms by six pillars. Subimages with a fixed size (128×128) are extracted from the captured palmprint images (384×284) so that different palmprints are converted into the same image size for further processing.

Eigenpalms: Feature Extraction

Usually, a palmprint image is described as a 2-D array *(N×N)*. In the eigenspace method, this can be defined as a vector of length N^2, called a "palm vector." A sub palmprint image is fixed with a resolution of 128×128; hence, a vector can be obtained that represents a single point in the 16,384-dimensional space. Since palmprints have similar structures (usually three main lines and creases), all "palm vectors" are located in a narrow image space; thus, they can be described by a relatively low-dimensional space. As the most optimal orthonormal expansion for image compression, the K-L transform can represent the principle components of the distribution of the palmprints or the eigenvectors of the covariance matrix of the set of palmprint images. Those eigenvectors define the subspace of the palmprints, which are called "eigenpalms." Then, each palmprint image in the training set can be exactly represented in terms of a linear combination of the "eigenpalms." Let the training samples of the palmprint images be $x_1, x_2, \ldots, x_M$, where M is the number of images in the training set. The average palmprint image of the training set is defined by:

$$\mu = \frac{1}{M}\sum_{i=1}^{M} x_i \tag{4.19}$$

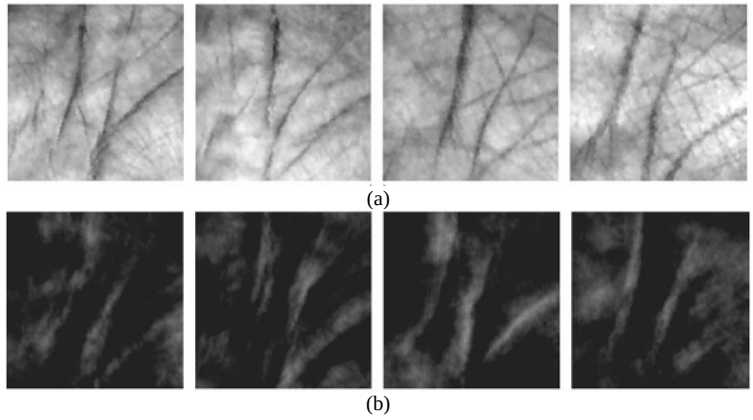
The difference between each palmprint image and the average image is given by $\varphi_i = x_i - \mu$. Then, we can obtain the covariance matrix of $\{x_i\}$ as follows:

$$C = \frac{1}{M}\sum_{i=1}^{M}(x_i - \mu)(x_i - \mu)^T = \frac{1}{M}XX^T \tag{4.20}$$

where the matrix $X=[\varphi_1 \varphi_2 \ldots \varphi_M]$. Obviously, the matrix C is of dimensions $N^2 \times N^2$. It is evident that the eigenvectors of C can span an algebraic eigenspace and provide an optimal approximation for those training samples in terms of the mean square error. However, determining the eigenvectors and eigenvalues of the matrix C ($C \in \Re^{N^2 \times N^2}$) is an intractable task for a typical image size. Therefore, we need to find an efficient method to calculate the eigenvectors and eigenvalues. It is well-known that the following formula is satisfied for the matrix C:

$$Cu_k = \lambda_k u_k \tag{4.21}$$

Figure 4.16. (a) Sub-palmprint samples in our training set, (b)the eigenpalms derived from the above samples



where u_k refers to the eigenvector of the matrix C, and λ_k is the correlative eigenvalue of matrix C. In practice, the number of the training samples, M, is relatively small. The eigenvectors (v_k) and eigenvalues (a_k) of matrix $L = X^T X (L \in \Re^{M \times M})$ are much easier to calculate. Therefore, we have:

$$X^T X v_k = a_k v_k \tag{4.22}$$

and we multiply each side of the Equation 4.22 by X:

$$XX^T (Xv_k) = a_k (Xv_k) \tag{4.23}$$

Then, we can get the eigenvectors of matrix C:

$$u_k = Xv_k \tag{4.24}$$

By using this method, the calculations are greatly reduced, where $U = \{u_k, k=1,2,\ldots,M\}$ denotes the basis vectors that correspond to the original palmprint images and span an algebraic subspace called unitary eigenspace of the training set. Resizing each of the eigenvectors into the image domain ($N \times N$), we find that they are like palmprints in appearance and can represent the principle characters (especially, the main lines) of the palmprints, which are referred as "eigenpalms." Figure 4.16 shows some of the eigenpalms derived from the samples in the training set.

Since each palmprint in the training set can be represented by an eigenvector, the number of the eigenpalms is equal to the number of the samples in the training set. However, the theory of PCA states that it does not need to choose all of the eigenvectors as the base vectors; just those eigenvectors that correspond to the largest eigenvalues can represent the characteristic of the training set quite well. Then the M' significant

eigenvectors ($u_k^{'}$) with the largest associated eigenvalues are selected to be the components of the eigenpalms ($U^{'} = \{u_k^{'}, k = 1, \cdots, M^{'}\}$), which can span an M' dimensional subspace of all possible palmprint images. A new palmprint image is transformed into its "eigenpalms" components by the following operation:

$$f_i = U^{'}(x_i - \mu), (i = 1, \cdots, M) \tag{4.25}$$

where the weights of the projection $f_i (f_i \in \Re^{M^{'} \times 1})$ refer to the standard feature vector of each person, and M' is called the feature length.

Experimental Results

Palmprint images were collected in our laboratory from 191 people using our self-designed capture device. Since the two palmprints (right-hand and left-hand) of each person are different, we captured both and treated them as palmprints from different people. Eight samples were captured for each palm with different rotation and translations. Thus, a palmprint database of 382 classes was created, which included a total of 3,056 (=191×2×8) images with 384×284 pixels in 256 gray levels. Four kinds of experiment schemes were designed: one (two, three or four) sample(s) of each person was randomly selected for training, and the other four samples were used for authentication, respectively. During the experiments, the features are extracted by using the proposed eigenspace method with length 50, 100, 150 and 200. The weighted Euclidean distance is used to cluster those features (Zhu & Tan, 2000),

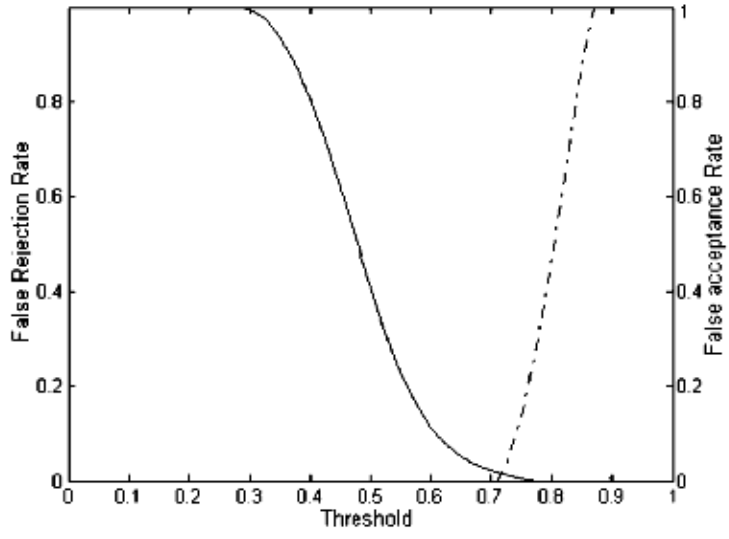
$$d_k = \sum_{i=1}^{N} \frac{(f(i) - f_k(i))^2}{(s_k)^2} \tag{4.26}$$

where f is the feature vector of the unknown palmprint, f_k and s_k denote the kth feature vector and its standard deviation, and N is the feature length. Based on these schemes, the matching is separately conducted and the results are listed in Table 4.5. A high recognition rate (99.149%) was achieved for the fourth scheme, with feature length of 100. It is evident that feature length can play an important role in the matching process. Long feature lengths lead to a high recognition rate. However, this principle only holds to a certain point, as the experimental results show that the recognition rate remains unchanged, or even becomes worse, when the feature length is extended further.

Table 4.5. Testing results of the three matching schemes with different feature lengths

Recognition rate (%)		Feature length			
		50	100	150	200
Training samples	1	94.175	95.550	95.175	93.128
	2	96.073	97.186	96.924	95.942
	3	97.186	98.429	98.822	97.971
	4	97.840	99.149	99.084	98.691

Figure 4.17. The FRR and FAR of the proposed algorithm



Further analysis of the fourth scheme was made by calculating the standard error rates (FAR and FRR) (Zhang & Shu, 1999). Obviously, for an effective method, both rates must be as low as possible, but they are actually antagonists, and lowering these errors is part of an intricate balancing act. For example, if you make a system more difficult to enter for an impostor (reducing FAR), you also make the system more difficult to enter for a valid enrollee (i.e., FRR raised). This process operates in the reverse sense, too. For a given system, this becomes a question of probabilities, and a company deploying such a system will generally adjust the matching threshold depending on the level of security needed. For instance, a bank needs a very secure system, so it would adjust the threshold very low to reach an FAR close to zero. However, the bank's employees will have to accept false rejections, and they may have to try several times to enter the system. The curves for the FRR and FAR of the fourth scheme are shown in Figure 4.17. When the threshold value is set to 0.71, the palmprint recognition method can achieve an ideal result with an FRR –1% and an FAR – 0.03%, respectively.

Compared with the approach in Duta, Jain, and Mardia (2002), which used a set of feature points along the prominent palm lines and the associated line orientation of palmprint images to identify the individuals, a matching rate about 95% was achieved. But only 30 palmprint samples from three persons were collected for testing. It seems that the testing set is too small to cover the distribution of all palmprints. An average recognition rate of 91% was achieved by the technology proposed in You, Li and Zhang (2002), which involved a hierarchical palmprint recognition fashion. The global texture energy features were used to guide the dynamic selection for a small set of similar candidates from the database at coarse level for further processing. An interesting point-based image matching was performed on the selected similar patterns at fine levels for the final confirmation. Since multiple feature extraction methods and matching algorithms

Table 4.6. Comparison of different palmprint recognition methods

	Method		
	Feature points (Duta et al., 2002)(Anil K.Jain)	Hierarchical identification (You et al., 2002) (Jane You)	Eigenpalm proposed
Database (samples)	30	200	3056
Features	Feature points	Global textures& feature points	Eigenpalms
Recognition rate (%)	95	91	99.149

are needed, the whole process of recognition is more complex. Nevertheless, the recognition rate of our method is more efficient, as illustrated in Table 4.6.

Remarks

In this section, the eigenpalm method is developed for palmprint recognition by using the K-L transform algorithm, which can represent the principal components of palmprints fairly well. The features are extracted by projecting palmprint images into an eigenpalms subspace. To assess the efficiency of our method, the weighted Euclidean distance classifier is applied. A correct recognition rate of up to 99% can be obtained using our approach.

GAIT APPLICATION

Gait is a newly emergent biometric feature that offers the ability of identifying people at a distance. Gait can be advantageous in some aspects over other forms of biometric features in the following ways (Wang, Ning, Tan, & Hu, 2004):

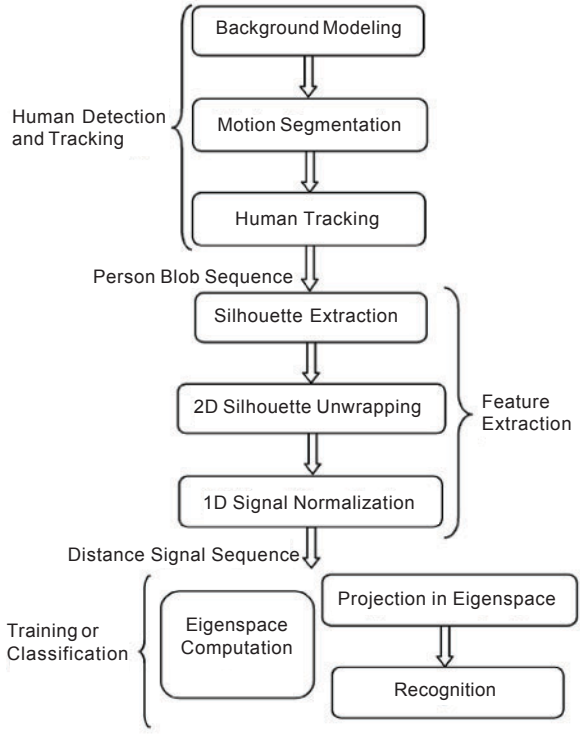
1. Gait seems to be unique. That each person seems to have a distinctive way of walking is easily understood from a biomechanics viewpoint. Human walking is a complex action of locomotion, involving synchronized integrated movements of body parts, joints and the interaction among them. It is the distinguishable variations among the properties of body structures, weights of limbs and actions of different subjects that may provide a unique cue for identity recognition.
2. Gait is unobtrusive. Most biometric features usually require physical touch or proximal sensing, while using gait would avoid such problems, since it does not require the user's interaction. Also, gait can be easily extracted from great distances secretly, which naturally advances acceptance of the users.
3. Gait can be used for recognition at a distance. Established biometric features, such as face and fingerprint, are limited in such a capability because they usually require sensing the cooperative users at close ranges. However, at a distance, these biometric features are hardly applicable. Fortunately, gait is still visible in this case. So, from the surveillance point of view, gait is a very attractive modality for recognition at a distance.

As stated above, gait has many advantages, especially unobtrusive identification at a distance, making it very attractive. Gait recognition, as a combination of human motion analysis and biometrics, aims essentially to discriminate people by the way they walk. An ongoing research project, the Human Identification at a Distance (Human ID) program1 sponsored by DARPA, aims to develop a full range of multi-modal surveillance technologies for detecting, classifying and identifying humans from a great distance to enhance protection from terrorist attacks. Its focus is on dynamic face recognition and recognition from body dynamics, including gait.

Overview of Approach

The introduction aims to establish an automatic gait recognition method based upon spatiotemporal silhouette analysis measured during walking. Gait includes both the body appearance and the dynamics of human walking motion (Lee & Grimson, 2002). Intuitively, recognizing people by gait depends greatly on how the silhouette shape of an individual changes over time in an image sequence. So, we may consider gait motion to be composed of a sequence of static body poses and expect that some distinguishable signatures with respect to those static body poses can be extracted and used for recognition by considering temporal variations of those observations. Also, eigenspace transformation based on PCA has actually been demonstrated to be a potent metric in

Figure 4.18. Overview of the proposed method (Wang et al., 2003)



face recognition (i.e., eigenface) and gait analysis (Murase & Sakai, 1996; Huang, Harris, & Nixon, 1999; Johnson & Bobick, 2002; BenAbdelkader, Culter, Nanda, & Davis, 2001; Bobick & Johnson, 2001; Winter, 1990; Vega & Sarkar, 2002). Based on these observations, this chapter proposes a silhouette analysis-based gait recognition algorithm using the traditional PCA. The algorithm implicitly captures the structural and transitional characteristics of gait. Although it is very simple in essence, the experimental results are surprisingly promising (Wang, Tan, Ning, & Hu, 2003). The overview of the proposed algorithm is shown in Figure 4.18 (Wang et al., 2003).

Feature Extraction

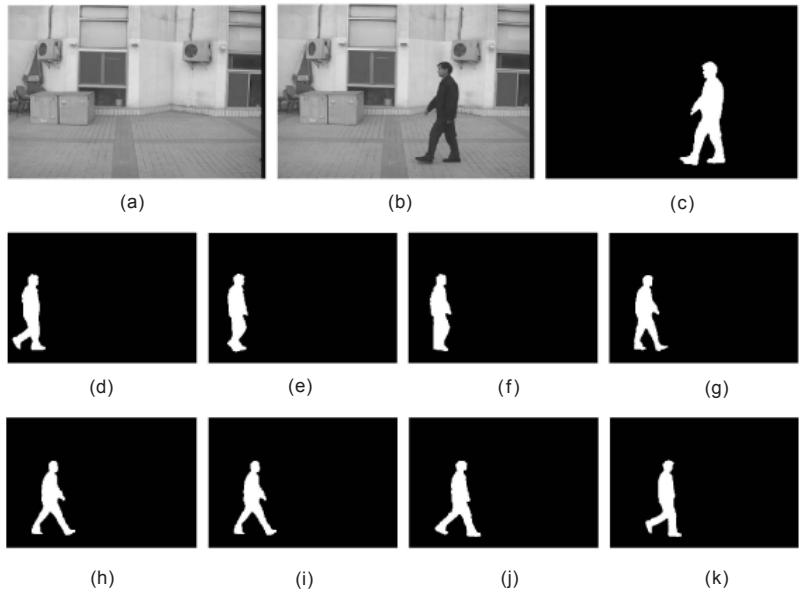
Before training and recognition, each image sequence including a walking figure is converted into an associated temporal sequence of distance signals at the preprocessing stage.

Human Detection and Tracking

Background Modeling

Background subtraction has been widely used in foreground detection, where a fixed camera is usually used to observe dynamic scenes. How to reliably generate the

Figure 4.19. Examples of moving silhouette extraction and tracking



(a) Background image constructed by the LMedS method, (b) an original image, (c) the extracted silhouette from (b), and (d)-(k) temporal changes of moving silhouettes in a gait pattern (frame 17 to frame 24) (Wang et al., 2003)

background image from video sequences is critical. Here, the least median of squares (LMedS) (Yang & Levine, 1992) method is used to construct the background from a small portion of image sequences, even including moving objects. Let I represent a sequence including N images. The resulting background b_{xy} can be computed by (Yang & Levine, 1992):

$$b_{xy} = \min_{p} med_t (I_{xy}^t - p)^2 \tag{4.27}$$

where p is the background brightness value to be determined for the pixel location (x,y), *med* represents the median value, and t represents the frame index ranging within $1 - N$. It is found that N over 60 is sufficient for our data set to generate a reliable background.

Differencing

We use the following extraction function to indirectly perform differencing (Kumo, Watanabe, Shimosakoda, & Nakagawa, 1996):

$$f(a,b) = 1 - \frac{2\sqrt{(a+1)(b+1)}}{(a+1)+(b+1)} \cdot \frac{2\sqrt{(256-a)(256-b)}}{(256-a)+(256-b)} \tag{4.28}$$

where $a(x,y)$ and $b(x,y)$ are the brightness of current image and the background at the pixel position (x,y), respectively, $0 \le a(x,y), b(x,y) \le 255$, $0 \le f(a,b) < 1$. This function can detect the change sensitivity of the difference value according to the brightness level of each pixel in the background image. For each image I_{xy}, the distributions of the above extraction function $f(a(x,y), b(x,y))$ over x and y can be easily obtained. Then, the moving pixels can be extracted by comparing such a distribution against a threshold value decided by the conventional histogram method.

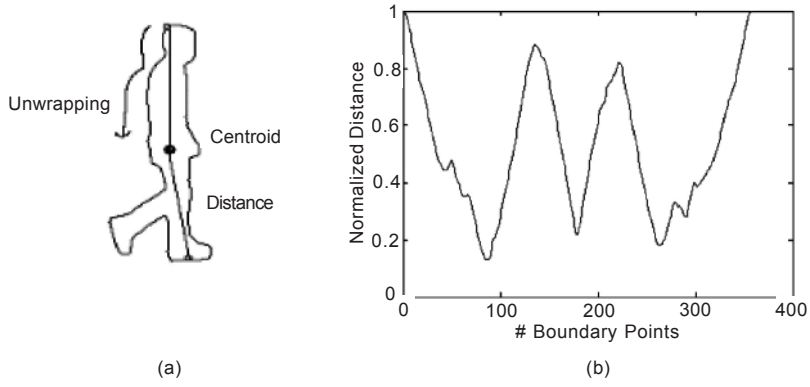
Postprocessing and Tracking

To eliminate inaccuracy due to segmentation error, each foreground region is then tracked from frame to frame by a simple correspondence method based on the overlap of their respective bounding boxes in any two consecutive frames (Haritaoglu, Harwood, & David, 2000). That is, we perform a binary edge correlation between the current and previous silhouette profiles over a small set of displacements (Haritaoglu, Harwood & David, 2000). An example of motion segmentation and the tracking process is shown in Figure 4.19, from which we can see that the human detection and tracking procedure performs well on our data as a whole. It absolutely does not affect the following feature extraction process, though there are small portions of silhouette distortions, such as partial missing of body parts (e.g., invisible arms in Figures 4.19d, 4.19j and 4.19k) and the cross of two slightly separated legs (e.g., in Figure 4.19f).

Silhouette Representation

An important cue in determining underlying motion of a walking figure is temporal changes of the walker's silhouette. To make the proposed method insensitive to changes of color and texture of clothes, we use only the binary silhouette. Additionally, for the

Figure 4.20. Silhouette representation



(a) Illustration of boundary extraction and counterclockwise unwrapping, and (b) the normalized distance signal consisting of all distances between the centroid and the pixels on the boundary (Wang et al., 2003)

sake of computational efficiency, we convert these 2D silhouette changes into an associated sequence of 1D signal to approximate temporal pattern of gait. This process is illustrated in Figure 4.20. After the moving silhouette of a walking figure has been tracked, its outer contour can be easily obtained using a border following algorithm. Then, we may compute its shape centroid (x_c, y_c). By choosing the centroid as a reference origin, we unwrap the outer contour counterclockwise to turn it into a distance signal $S = \{d_1, d_2, \ldots, d_i, \ldots, d_{Nb}\}$ that is composed of all distances d_i between each boundary pixel (x_i, y_i) and the centroid.

$$d_i = \sqrt{(x_i - x_c)^2 + (y_i - y_c)^2} \tag{4.29}$$

This signal indirectly represents the original 2D silhouette shape in the 1D space.

Training and Projection

PCA Training

PCA, also known as eigenanalysis, is a technique used to reduce the dimensionality of data and examine the relationship between a set of correlated variables. PCA has been used successfully before in both gait and face recognition techniques. Dimensionality reduction is vital to recognition purposes, because the size of recognition matrices can be vast and very computationally expensive or infeasible. The purpose of PCA training is to obtain several principal components to represent the original gait features from a high-dimensional measurement space to a low-dimensional eigenspace. The training process similar to Haung, Harris, and Nixon (1999) is illustrated as follows:

Given s classes for training, each class represents a sequence of distance signals of one subject's gait. Multiple sequences of each person can be freely added for training. Let $D_{i,j}$ be the jth distance signal in class i and N_i the number of such distance signals in the ith class. The total number of training samples is $N_t=N_1+N_2+,\ldots,N_s$, and the whole training set can be represented by $[D_{1,1}, D_{1,2}, \ldots, D_{1,N1}, D_{2,1}, \ldots D_{s,Ns}]$. We can easily obtain the mean m_d and the global covariance matrix $\mathbf{\Sigma}$ of such a data set by:

$$m_d = \frac{1}{N_t}\sum_{i=1}^{s}\sum_{j=1}^{N_i} D_{i,j} \tag{4.30}$$

$$\sum = \frac{1}{N_t}\sum_{i=1}^{s}\sum_{j=1}^{N_i} (D_{i,j} - m_d)(D_{i,j} - m_d)^T \tag{4.31}$$

If the rank of the matrix $\mathbf{\Sigma}$ is N, then we can compute N nonzero eigenvalues $\lambda_1, \lambda_2, \ldots, \lambda_N$ and the associated eigenvectors $e_1, e_2, \ldots, e_N$ based on SVD. Generally speaking, the first few eigenvectors correspond to large changes in training patterns. Therefore, for the sake of memory efficiency in practical applications, we may ignore those small eigenvalues and their corresponding eigenvectors using a threshold value T_s

$$W_k = \sum_{i=1}^{k}\lambda_i \Big/ \sum_{i=1}^{N}\lambda_i > T_s \tag{4.32}$$

where W_k is the accumulated variance of the first k largest eigenvalues with respect to all eigenvalues. In our experiments, T_s is chosen as 0.95 for obtaining steady results.

Projection

Taking only the $k < N$ largest eigenvalues and their associated eigenvectors, the transform matrix $E=[e_1, e_2, \ldots, e_k]$ can be constructed to project an original distance signal $D_{i,j}$ into a point $P_{i,j}$ in the k-dimensional eigenspace.

$$P_{i,j}=[e_1,e_2,\ldots,e_k]^{\mathrm{T}}D_{i,j} \tag{4.33}$$

Accordingly, a sequential movement of gait can be mapped into a manifold trajectory in such a parametric eigenspace.

It is well known that k is usually much smaller than the original data dimension N. That is to say, eigenspace analysis can drastically reduce the dimensionality of input samples. For each training sequence, the projection centroid C_i in the eigenspace is accordingly given by averaging all single projections corresponding to each frame in the sequence.

$$C_i = \frac{1}{N_i}\sum_{j=1}^{N_i} P_{i,j} \tag{4.34}$$

Recognition

Gait recognition is a traditional pattern classification problem that can be solved by measuring similarities between reference patterns and test samples in the parametric eigenspace.

Similarity Measures

Spatiotemporal Correlation

Gait is a kind of spatiotemporal motion pattern, so we use *spatial-temporal correlation* (STC, an extension of 2D image correlation to 3D correlation in the space and time domain (Murase & Sakai, 1996)) to better capture its spatial structural and temporal transitional characteristics.

For two input sequences, we can first convert them into a sequence of distance signal $I_1(t)$ and $I_2(t)$ at the preprocessing stage, as described in feature extraction. Then, they are respectively projected into a trajectory $P_1(t)$ *and* $P_2(t)$ in the eigenspace using Equation 4.33. The similarity measure between two such input vector sequences can be computed by (Murase & Sakai, 1996):

$$d^2 = \min_{ab} \sum_{t=1}^{T} \| P_1(t) - P_2^{'}(at+b) \|^2 \tag{4.35}$$

where $P_2^{'}(at+b)$ is a dynamic time warping vector from $P_2(t)$ with respect to time stretching and shifting for an approximation of the temporal alignment between the two sequences. The selection of parameters a and b depends on the relative stride frequency and phase difference within a stride (two steps), respectively. Let f_1 and f_2 denote the frequencies of the two gait sequences; then $a = f_2/f_1$. By cropping a subsequence of length f_2 from the second sequence vector repeatedly and stretching it with a, we may obtain its correlation with $P_1(t)$. The average minimum of all prominent valleys of the correlation results determines their similarity. Gait period analysis has been explored in previous work (BenAbdelkader, Culter, & Davis, 2002; Collins, Gross, & Shi, 2002), which serves to determine the frequency and phase of each observed sequence so as to align sequences before matching.

The Normalized Euclidean Distance

Note that the computational cost will increase quickly if the comparison is performed in the spatiotemporal domain, especially when time stretching and shifting is taken into account (Murase & Sakai, 1996). Here, we turn to use the *normalized Euclidean distance* (NED) between the projection centroids of two gait sequences for the similarity measure to eliminate such matching problems.

Assuming that the trajectories of any two sequences in the eigenspace are $P_1(t)$ and $P_2(t)$, respectively, we can easily obtain their associated projection centroids C_1 and C_2 using Equation 4.34. Each projection centroid implicitly represents a principal structural shape of certain subject in the eigenspace. The normalized Euclidean distance between the two sequential projection centroids can be defined by (Wang et al., 2003):

$$d^2 = \left\| \frac{C_1}{\| C_1 \|} - \frac{C_2}{\| C_2 \|} \right\| \tag{4.36}$$

Furthermore, for multiple sequences of the same subject, we may also obtain its exemplar projection centroid by further averaging the projection centroids of those single sequences as a reference template for that class. This exemplar centroid will also be used for gait classification in Wang's (Wang et al., 2003) experiments.

Classifier

The classification process is carried out via two simple classification methods; namely, the *nearest-neighbor* classifier (NN) and the nearest-neighbor classifier with respect to class exemplars (ENN) derived from the mean projection centroid of those training sequences for a given subject. Let T represent a test sequence and R_i represent the ith reference sequence. We may classify this test sequence into class c that can minimize the similarity distance between the test sequence and all reference patterns by (Wang et al., 2003):

$$c = \arg\min_i d_i(T, R_i) \tag{4.37}$$

where d is the similarity measure described in Equation 4.29. Note that d can only choose NED if ENN is used. No doubt, a more sophisticated classifier could be employed, but the main interest here is to evaluate the genuine discriminatory ability of the extracted features in our method.

Experiments

Extensive experiments are carried out to verify the effectiveness of the proposed algorithm. The following describes the details of the experiments.

Table 4.7. Overview of some typical databases used in the literature (Wang et al., 2003)

Database	UCSD	NLPR	US(1)	CMU	MIT	UMD(1)	UMD(2)	GVU	USF
Environment	O	O	I	I	I	O	O	I or O	O
Walk surface	G1	G1	F	T	F	G1	G1	G1 or F	G2 or C
#Subjects	6	20	12	25	24	25	55	20	74
#Sequences	40	240	48	-	194	100	-	-	452
#Views	1	3	1	6	1	2	2	1	2
Synchronized	N/A	N/A	N/A	Y	N/A	N/A	N	N/A	N
#Walk styles	1	1	1	4	1	1	1	1	1
Frame rate	30	25	25	30	15	20	20	25	30

Figure 4.21. Some sample images in the NLPR gait database

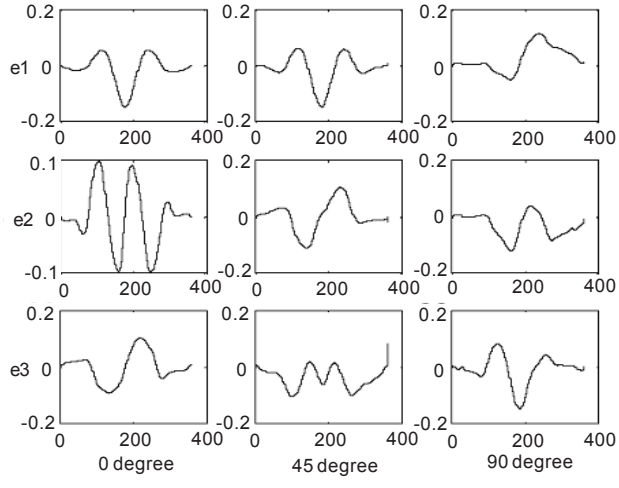
(a) (b) (c)

(a) Lateral view, (b) oblique view and (c) frontal view (Wang et al., 2003)

Data Acquisition

A new gait database, called the NLPR database, is established for our experiments. A digital camera (Panasonic NV-DX100EN) fixed on a tripod is used to capture gait sequences on two different days in an outdoor environment. All subjects walk along a straight-line path at free cadences in three different views with respect to the image plane; namely, laterally (0˚), obliquely (45˚) and frontally (90˚). The resulting NLPR database includes 20 subjects and four sequences for each viewing angle per subject. For instance, when the subject is walking laterally to the camera, the direction of walking is from left to right for two of the four sequences, and from right to left for the remaining two. The

Figure 4.22. The first three eigenvectors for each viewing angle obtained by PCA training



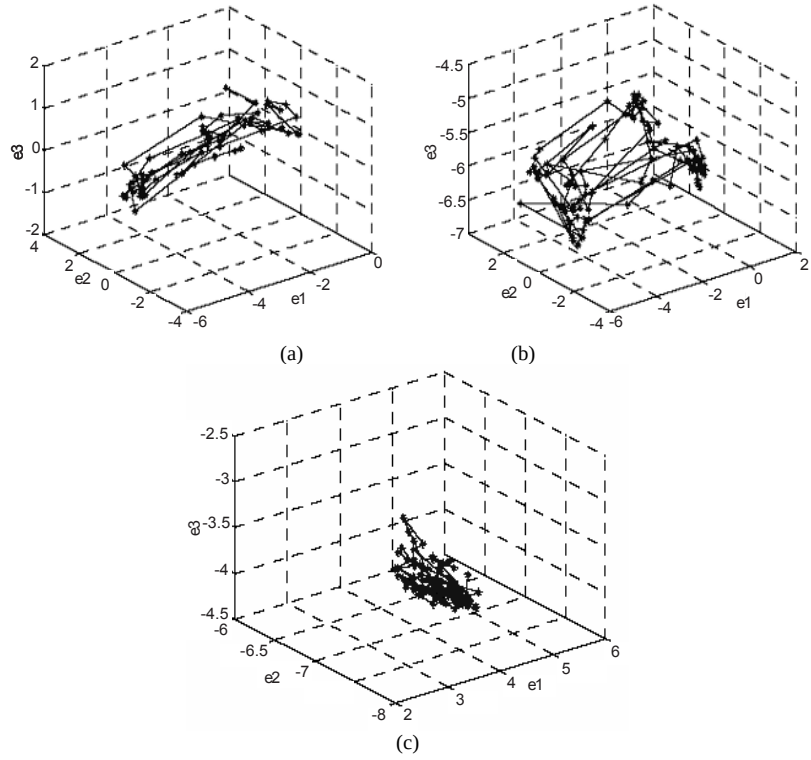
(a) Lateral view, (b) oblique view and (c) frontal view (Wang, Tan, Ning & Hu, 2003)

database, therefore, includes a total of 240 gait sequences (20×4×3). These sequence images with 24-bit full color are captured at a rate of 25 frames per second and the original resolution is 352×240. The length of each image sequence varies with the pace of the walker, but the average is about 90 frames. To the best of our knowledge, Wang's (Wang et al., 2003) database is probably one of the concurrent gait databases available in the public domain, which is reasonably sized (see Table 4.7 for a summary of major gait databases currently in use). Some sample images are shown in Figure 4.21, where the white line with arrow represents the walking path.

Preprocessing and Training

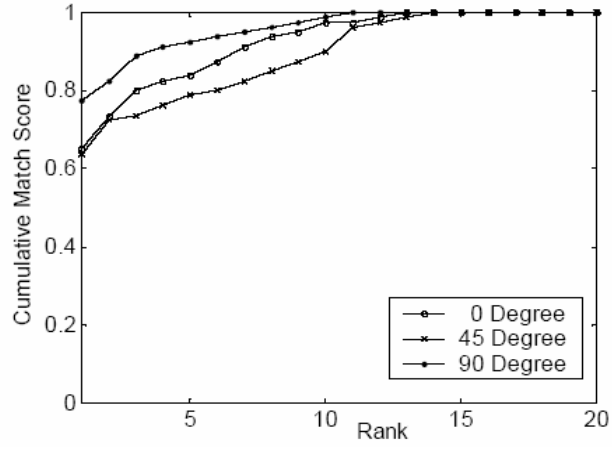
In Wang et al. (2003), they choose a small portion of such distance signal sequences, including all classes for training. And we keep the first 15 eigenvalues and their associated eigenvectors to form the eigenspace transformation matrix. Figure 4.22 gives the first three eigenshapes for each viewing angle. From Figure 4.22, we can see that these eigen-curves are either odd symmetric or even symmetric, which reveals that gait has a characteristic of symmetry.

Figure 4.23. The projection trajectories of three training gait sequences (only the 3D eigenspace is used here for clarity)

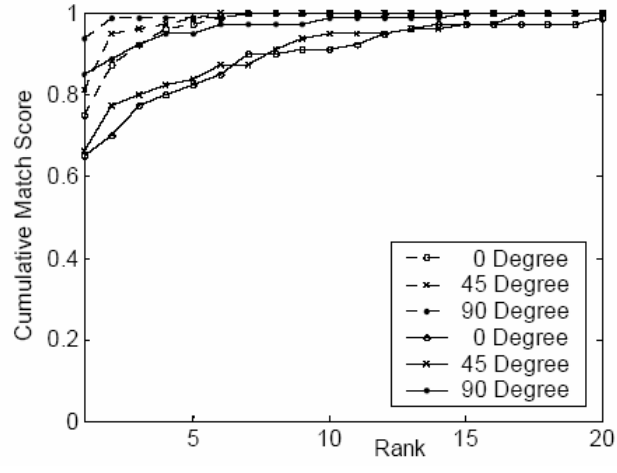


(a) Lateral view, (b) oblique view and (c) frontal view (Wang et al., 2003)

Figure 4.24. Identification performance based on cumulative match scores



(a)



(b)

(a) Classifier based on STC and(b) classifiers based on NED with respect to single projection centroid (solid line) and exemplar projection centroid (dotted line), respectively (Wang et al., 2003)

Once the eigenspace is obtained, each distance signal derived from each silhouette image can be represented by a linear combination of these 15 principal eigenvectors. That is, each distance signal can be mapped into one point in a 15-dimensional eigenspace. Each gait sequence will be accordingly projected into a manifold trajectory in the eigenspace. The projection trajectories of three trained sequences with respect to lateral view, oblique view and frontal view, respectively, are shown in Figure 4.23, where only 3D eigenspace is used for visualization.

Results and Analysis

Identification Mode

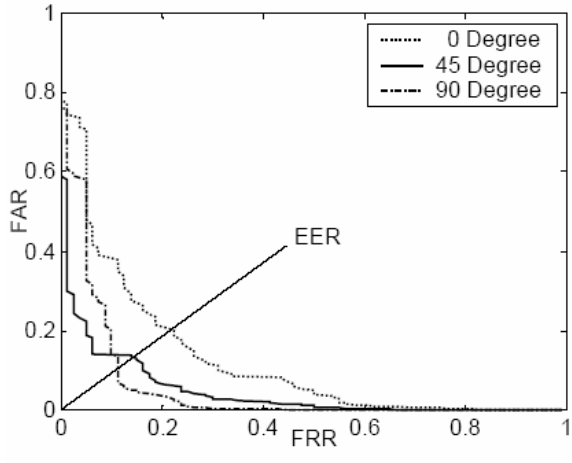
In Wang et al. (2003), a useful classification performance measure was introduced by the FERET protocol for the evaluation of face recognition algorithms (Phillips, Moon, Rizvi, & Rause, 2000). It is defined as the cumulative probability $p_{(k)}$ that the real class of a test measurement is among its top k matches (Phillips, Moon, Rizvi, & Rause, 2000). The performance statistics are reported as the cumulative match scores. The rank k is plotted along the horizontal axis, and the vertical axis is the percentage of correct matches (Phillips, Moon, Rizvi, & Rause, 2000).

Here, Wang et al. (2003) uses the leave-one-out cross-validation rule with the NLPR database to estimate the performance of the proposed method. Each time, we leave out one image sequence as a test sample and train on the remainder. After computing the similarity differences between the test sample and the training data, the NN or ENN is applied for classification. Figure 4.24 shows the cumulative match scores for ranks up to 20, where Figure 4.24a uses the STC similarity measure and Figure 4.24b uses the NED similarity measure with respect to projection centroids (solid line) and exemplar projection centroids (dotted line), respectively. It is noted that the correct classification rate is equivalent to $p_{(1)}$ (i.e., Rank=1). That is, for side view, oblique view and frontal view, the correct classification rates are, respectively, 65%, 63.75% and 77.5% with NN and STC; 65%, 66.25% and 85% with NN and NED; and 75%, 81.25% and 93.75% with ENN and NED.

Verification Mode

For completeness, Wang et al. (2003) also estimates FAR and FRR via the leave-one-out rule in verification mode. That is, we leave out one example, train the classifier using the remaining, and then verify the left-out sample on all 20 classes. Note that in each of

Figure 4.25. ROC curves of gait classifier based on NED with respect to three viewing angles



these 80 iterations for each viewing angle, there is one genuine attempt and 19 imposters, since the left-out sample is known to belong to one of the 20 classes. Figure 4.25 shows the ROC curve using the NED similarity measure with exemplar projection centroids, from which we can see that the EERs are about 20%, 13% and 9% for 0-, 45- and 90-degree views, respectively. Here, the verification performance of frontal view is also better than those of the other views.

Remarks

The lack of generality of viewing angles is a limitation to most gait recognition algorithms. This present method is view-dependent, like most previous work, so a useful experiment would be to determine the sensitivity of the features to different views, whose results would enable a multi-camera tracking system to select an optimal view for recognition (Little & Boyd, 1998). Another obvious way to generalize the algorithm itself is to store training sequences taken from multiple viewpoints and to classify both the subject and the viewpoint (Collins, Gross, & Shi, 2002).

It is more sufficient for recognition to extract dynamic information, such as the oscillatory trajectories of joints. Therefore, 3D human body modeling and tracking might prove to be of benefit. Future work may try to combine both static and dynamic features of gait, such as posture, arm/leg/hip swing and so forth. Also, seeking better similarity measures, designing more sophisticated classifiers, gait segmentation and the evaluation of different scenarios deserve more attention in future work.

EAR BIOMETRICS

Introduction

The ear has been proposed as a biometric (Lammi, n.d.; Victor, Bowyer, & Sarkar, 2000). Using the ear in person identification has been interesting at least 100 years (Lammi, n.d.). The difficulty is that we have several adjectives to describe, for example, faces, but almost none for ears. We all can recognize people from faces, but we can hardly recognize anyone from ears. But a famous work among ear identification was made by Alfred Iannarelli in 1989, when he gathered up more than 10,000 ears and found that they all were different (Burge & Burger, 2000; Victor, Bowyer, & Sarkar, 2002; Chang, Bowyer, Sarkar, & Victor, 2003; Hoogstrate, Van den Heuvel, & Huyben, 2000; Iannarelli, 1989). Already in 1906, Imhofer found that in the set of 500 ears, only four characteristics were needed to state the ears unique (Hoogstrate, Van den Heuvel, & Huyben, 2000).

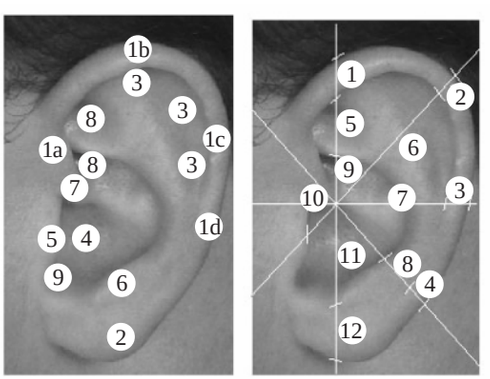
There are at least three methods for ear identification:

1. Taking a *photo* of an ear (see Figure 4.26). Research supports the hypothesis about ear uniqueness. Even identical twins had similar, but not identical, ear physiological features (Burge & Burger, 1998; Chang, Bowyer, Sarkar, & Victor, 2003; Hoogstrate, Van den Heuvel, & Huyben, 2000; Victor, Bowyer, & Sarkar, 2002).
2. Taking "*earmarks.*" By pushing an ear against a flat glass, the earmarks are used mainly in crime solving. Even though some judgments are made based on the

Table 4.8. Permanence of different biometrics over time. The best permanence has most 0-symbols and the worst, the least. (Bromba GmbH, 2003; Burge & Burger, 2000)

Biometric Trait	***Permanence over time***
Fingerprint (Minutia)	000000
Signature(dynamic)	0000
Facial Structure	00000
Iris Pattern	000000000
Retina	00000000
Hand Geometry	0000000
Finger Geometry	0000000
Vein structure of the back of the hand	000000
Ear Form	000000
Voice(Tone)	000
DNA	000000000
Odor	000000?
Keyboard Strokes	0000
Comparison: Password	**00000**

Figure 4.26. (a) Anatomy and (b) measurements



(a) 1 Helix Rim, 2 Lobule, 3 Antihelix, 4 Concha, 5 Tragus, 6 Antitragus, 7 Crus of Helix, 8 Triangular Fossa, 9 Incisure Intertragica; (b) Locations of the anthropometric measurements used in the "Iannarelli System" (Burge & Burger, 1998)

earmarks, currently they are not accepted in courts in some countries (Bamber, 2001; Forensic Evidence News, 2000).

3. Taking *thermogram pictures* of the ear (Lammi, n.d.).

Taking a photo of the ear is the most commonly used method in research. The photo is taken and combined with previously taken photos for identifying a person. Since Iannarelli does not have academic background for his studies (Morgan, 1999; Pun & Moon, 2004; Iannarelli, 1989; Yan & Bowyer, n.d.), Victor et al. (2002) and Chang et al.

(2003) have used PCA and FERET evaluation protocol for their research about ears. We will later focus on this research. Moreno et al. presented a multiple identification method, which combines the results from several neural classifiers using feature outer ear points, information obtained from ear shape and wrinkles, and macro features extracted by compression network (Moreno, Sanchez, & Velez, 1999). Burge and Burger have researched automating ear biometrics with a Voronoi diagram of its curve segments (Burge & Burger, 1998, 2000). Hurley, Nixon and Carter have used force-field transformations for ear recognition (Hurley, Nixon, & Carter, 2000a, 2000b). The image is treated as an array of Gaussian attractors that act as the source of the force field.

This section focuses on the PCA algorithm in ear recognition, shown with two different cases.

PCA in Ear Recognition

Eigenears

Victor, Bowyer and Sarkar have made a comparison between face and ear recognition (Lammi, n.d.; Victor, Bowyer, & Sarkar, 2002). They used PCA (also known as "eigenfaces"), which is a dimensionality-reduction technique in which variation in the dataset is preserved. The classification is done in eigenspace, which is a lower-dimension space defined by principal components or the eigenvectors of the data set. The process consists of three steps: (1) Preprocessing, (2) Normalization and (3) Identification (see Figure 4.27 for details).

In the preprocessing step the ear images are cropped to a size of 400×500 pixels (face images to 768×1024). Coordinates of two distinct points are supplied to the normalization routine: Triangular Fossa and the Antitragus. The normalization step includes geometric normalization, masking and photometric normalization. In this phase, all the images are

Figure 4.27. Steps of PCA method (Victor et al., 2002)

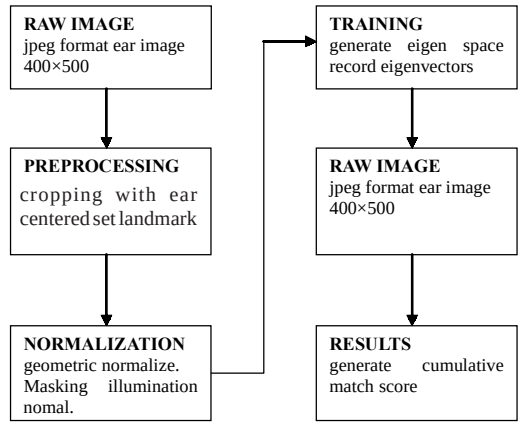


Table 4.9. Summary of comparison between eigen-faces and eigen-ears (Victor et al., 2002)

Experiment#	Face/Ear compared		Expected Result	Result
1	Same day, different expression	Same day, opposite ear	Greater variation in expressions than ears; ears perform better	Face performs better
2	Different day, similar expression	Different day, same ear	Greater variation in expression across days; ears perform better	Face performs better
3	Different day, different expression	Different day, opposite ear	Greater variation in face expression than ear; ears perform better	Face perform better

scaled to a standard 130×150 size. Next, all non-ear areas, like hair, background and so forth, are masked. Different levels of masking are experimented with for finding the best one to get as good performance as possible for the algorithm. Finally, the images are normalized for illumination. There are two phases in the identification phase: training and testing. In the training phase, the eigenvalues and eigenvectors of the training set are extracted and the eigenvectors are chosen based on the top eigenvalues. Victor, Bowyer and Sarkar (2002) have decided not to use any specific gallery but have a general representation of both ears and faces. Training set is a set of clean images without any duplicates. In the testing phase, the algorithm is provided a set of known ears and faces and a set of unknown ears and faces as the probe set. The algorithm matches each probe to its possible identity in the gallery. The ear and face images were collected at the University of South Florida. There were 294 subjects with 808 ear images in the experiment, of which half of the ear pictures were the left ear and half the right ear. Some of the images were from the same person, but taken on different days for testing the day variation of the ears. Every subject had a face image in the database and a corresponding ear image taken under the same conditions as the face image. This is a requirement for reasonable comparison and evaluation. Victor, Bowyer, and Sarkar (2002) refer to an article by Philips, Moon, Rizvi, and Rauss (2000) when stating that all the lighting arrangements and positions of light, cameras and subject follow the FERET face image acquisition protocol.

In the training session, 207 images were used for both ears and faces. The number of eigenvectors used in testing was 82. The null hypothesis was that the used set of experiments doesn't give significant performance difference between using the ear or face as biometric.

There were three experiments performed to test this hypothesis: (1) gallery and probe images taken same day with different expression, (2) gallery and probe images taken in different days with normal expression, and (3) gallery and probe images taken different day as normal and different expressions (Table 4.9).

Face-based recognition gives better performance than ear-based recognition in all three experiments (Victor et al., 2002).

Figure 4.28. The same ear can look different depending on, for example, day, lighting or pose variation (Chang et al., 2003)

Gallery image Day variation Lighting variation Pose variation

Table 4.10. Sources of verification error

Misspoken or misread prompted phases
Extreme emotional states (e.g., stress or duress)
Time varying (intra-or intersession) microphone placement
Poor or inconsistent room acoustics (e.g., multipath and noise)
Channel mismatch (e.g., using different microphones for enrollment and verification)
Sickness (e.g., head colds can alter the vocal tract)
Aging (the vocal tract can drift away from models with age)

Another Evaluation

Chang, Bowyer, Fellow, and Sarkar have made another comparison between ears and face images in appearance-based biometrics (Lammi, n.d.; Chang, Boyer, Sarkar, & Victor, 2003). The process is same as in the research of Victor et al. (see Figure 4.28). PCA was used and the evaluation was done as in the FERET approach. There were 197 subjects in the training set; each had both face image and ear image taken under the same conditions and at the same image acquisition session. If the face or ear was covered in the picture, they were left out from this research.

There were three experiments: (1) day variation experiment, (2) lighting condition variation experiment and (3) pose variation experiment with 22.5-degree rotation. The null hypothesis was that there is no significant difference between using the face or the ear as a biometric when using the same PCA-based algorithm, same subject pool and controlled variation in the used images.

The final result was that the recognition rate for ears was 71.6%; for face, 70.5%. The difference is not statistically significant using a McNemar test (Chang et al., 2003).

Remarks

In this section, we have given a short overview about ear biometrics. We focused on the PCA with photo, which is called the eigenear method. It is an established method in the face recognition domain. Although ear and face shows close resemblance with each other, there are inherent differences between the two. For instance, ear lacks the different

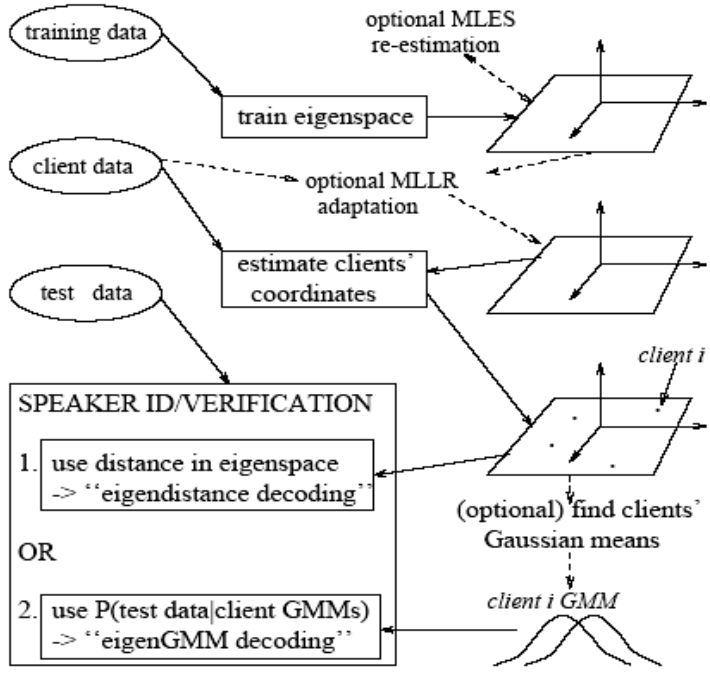
features that face possesses (e.g., eyes, nose, mouth, etc.). Such difference may call for an adaptation of face recognition algorithms or new approaches that cater to the unique features of ears. Further issues, like alternate face recognition algorithms, should also be studied for their suitability of application to the ear domain. These questions warrant an in-depth study in themselves.

SPEAKER IDENTIFICATION

Introduction

Voice capture is unobtrusive and voice print is an acceptable biometric in almost all societies (Jain et al., 1998; Furui, 1997). Some applications entail authentication of identity over telephone. In such situations, voice may be the only feasible biometric. Table 4.10 lists some of the human and environmental factors that contribute to these errors. General overviews of speaker recognition have been given by Atal, Doddington, Furui, O'Shaughnessy, Rosenberg, Soong, Sutherland, and Jack (Atal, 1976; Doddington, 1985; Furui, 1991; O'Shaughnessy, 1987; Rosenberg, 1976; Rosenberg & Soong, 1992; Sutherland & Jack, 1988).

Figure 4.29. The eigenvoice approach (Thyes, Kuhn, Nguyen, & Junqua, n.d.)



The focus of this section is on PCA and LDA applications of speaker recognition. Gaussian mixture models (GMMs) have been successfully applied to the tasks of speaker ID and verification when a large amount of enrollment data is available to characterize client speakers (Rosenberg, 1976; Thyes, Kuhn, Nguyen, & Junqua, n.d.; Forsyth, 1995; Kuhn, Nguyen, Junqua, et al., 1998, 1999; Kuhn, Junqua, Nguyen, & Niedzielski, 2000; Legetter & Woodland, 1995; Reynolds, 1995; Sukkar, Gandhi, & Setlur, 2000). A possible solution is the "eigenvoice" approach, in which client and test speaker models are confined to a low-dimensional linear subspace obtained previously from a different set of training data. One advantage of the approach is that it does away with the need for impostor models for speaker verification. The eigenvoice approach is described as follows (see Figure 4.29): First, we obtain a set of models for training speakers (in the experiments described here, these models were conventional GMMs) (Thyes, Kuhn, Nguyen, & Junqua, n.d.); Next, we apply a technique such as PCA or LDA to the means of the training speaker GMMs to obtain a low-dimensional eigenspace made up of "eigenvoice" basis vectors.

Eigenspace Training Techniques

PCA discovers the directions that account for the largest variability among training speakers (Thyes, Kuhn, Nguyen, & Junqua, n.d.; Kuhn, Nguyen, Junqua, et al., 1998, 1999; Kung, Junqua, Nguyen, & Niedzielski, 2000). In the experiments reported here, each training speaker's Gaussian means were concatenated to form a "supervector" of dimension D. PCA was applied to the set of T supervectors obtained from the T training speakers, yielding T-1 eigenvoice vectors ordered by the magnitude of their contribution to the between-speaker scatter matrix. This matrix is:

$$S_B = \sum_{s=1}^{T} N_s (\mu_s - \mu)(\mu_s - \mu)^T \tag{4.38}$$

which is similar to the between-class scatter matrix $\mathbf{S}_b$ defined as Equation 3.43. In Equation 4.38, N_s is the number of training utterances of speaker s, μ_s is the mean of all N_s samples and μ is the overall mean. Typically, we discard the higher-order eigenvoices (which mainly contain noise) to obtain an eigenspace of dimension less than T-1.

To better model the speaker space, we can apply maximum likelihood eigenspace (MLES) estimation (Nguyen, Wellekens, & Junqua, 1999), which re-estimates the initial PCA eigenspace so as to maximize the likelihood of the training data, given the speaker's identity: that is, $P(O_s | \lambda_s)$ is maximized (where O_s and λ_s represent an observation and the GMM of a given speaker respectively).

LDA is particularly relevant to speaker ID and verification, since it tries to increase discrimination between classes (in our case, a class consists of all speech from a given speaker) (Belhumeur et al., 1997). For other recent work applying LDA to this task (though in a completely different way), see Sukkar, Gandhi, and Setlur (2000). LDA was much less relevant to our earlier work on speaker adaptation for speech recognition systems, since no one cares whether an adapted recognizer distinguishes between speakers if it performs well for the current speaker. Consider an orthogonal transformation W mapping each D-dimensional supervector x_k into eigenspace:

$$y_k = W^T x_k \tag{4.39}$$

(where y_k is the transformed vector of dimension T). The transformation matrix W is selected so as to maximize the ratio between the between-class scatter S_B and the within-class scatter S_W similarly defined as Equation 3.37.

$$S_W = \sum_{s=1}^{T} \sum_{x_k \in X_s} (x_k - \mu_s)(x_k - \mu_s)^T \tag{4.40}$$

where μ_s is the mean of speaker s. The optimal transformation matrix W_{lda} will then be chosen so as to maximize the ratio of the determinant of $\tilde{S}_B = W_{lda} S_B W_{lda}$ of the projected samples to the determinant of $\tilde{S}_W = W_{lda} S_B W_{lda}{}^T$ of the projected samples:

$$W_{lda} = \arg\max_W \frac{\left|W^T S_B W\right|}{\left|W^T S_W W\right|} = \left[e(1)e(2)...e(K)\right] \tag{4.41}$$

where $\{e(i) \mid i = 1, \ldots, K\}$ are the generalized eigenvectors of S_B and S_W corresponding to the K largest eigenvalues $\{\lambda_i \mid i = 1, \ldots, K\}$:

$$\begin{aligned} & S_B e(i) = \lambda_i S_W e(i), \quad i = 1,...,K \\ \Leftrightarrow\ & S_W{}^{-1} S_B e(i) = \lambda_i e(i). \end{aligned} \tag{4.42}$$

The rank of S_W is at most N-T, where N is the total number of utterances in the training database and T the number of speakers. Thus, for each GMM used to build the eigenspace W_{lda}, we require more than D sample utterances (D is the dimension of the supervectors).

Experiments

Two databases were used in these experiments: the YOHO Speaker Verification database of "combination lock" phrases and the TIMIT database of acoustically varied continuous speech (Thyes et al., n.d.; Linguistic Data Consortium, n.d.). To obtain eigenspaces, speaker-dependent GMMs were initialized on a simple "SILENCE speech SILENCE" segmentation obtained by means of a silence model and a speaker-independent model. The sampling rate was 8 kHz (TIMIT data were downsampled to 8 kHz). There were 26 MFCC acoustic features (13 static, 13 dynamic), to which cepstral filtering was applied.

Results for Abundant Enrollment Data

In an initial set of experiments on YOHO, we tried several speaker ID approaches on 82 speakers with 360 seconds of enrolment data per client (Thyes, Kuhn, Nguyen, & Junqua, n.d.). When 5 seconds of test speech not used for enrollment was presented for each of the 82 clients, the conventional GMM approach with 32 Gaussians yielded 98.8%

Figure 4.30. Speaker ID: 10-second enrollment data, 5-second test data (Thyes et al., n.d.)

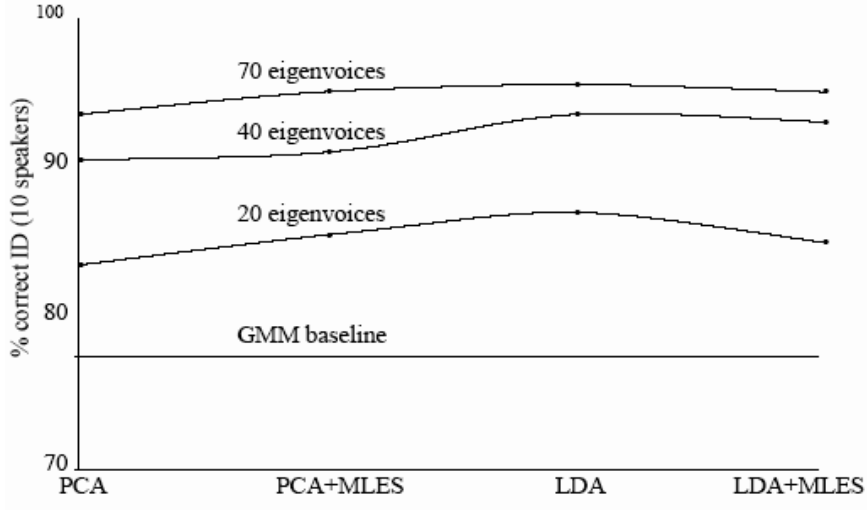


Table 4.11. Speaker verification (ERR): 64-GMM, 40 eigenvoices, YOHO training, enrollment and testing (Thyes et al., n.d.)

5 seconds enrolment		
Best GMM baseline (4G)		21.5%
Decoding	PCA	LDA
Euclidian Distance	9.6%	7.0%
GMM Decoding	11.0%	9.9%
10 seconds enrolment		
Best GMM baseline (8G)		14.4%
Decoding	PCA	LDA
Euclidian Distance	7.1%	6.4%
GMM Decoding	10.0%	9.0%

correct identification. For the eigenvoice approaches, the eigenspace was obtained from 72 of the 82 client speakers (implying that the maximum possible dimensionality of the eigenspace is 71). The best eigenvoice result, 98.0%, occurred in the case where LDA was used for eigenspace training, the dimensionality of the eigenspace was set to 70 and eigen-GMM decoding was used. Among all the eigenvoice approaches, training the eigenspace with LDA and carrying out eigen-GMM decoding always contributed to better performance than other methods.

Results for Sparse Enrollment Data

Figure 4.30 shows speaker ID results for 5 seconds of test speaker data and sparse enrollment data: 10 seconds enrollment for each of 10 clients (Thyes et al., n.d.). Clearly, eigenspace dimensionality has a powerful impact on performance. Note that LDA always

Table 4.12. Comparison: 64 CMM, YOHO vs. TIMIT vs. MLLR-adapted TIMIT for eigenspace training (Thyes et al., n.d.)

Eigenvoice dimension	20	40	70
YOHO eigenspace			
PCA without MLES	84.3%	89.0%	93.0%
PCA with MLES	86.8%	89.3%	92.8%
LDA	87.8%	94.3%	95.0%
TIMIT eigenspace			
PCA without MLES	76.5%	86.0%	91.5%
PCA with MLES	79.0%	85.5%	92.0%
LDA	77.3%	83.5%	82.8%
MLLR-adapted TIMIT eigenspace			
PCA without MLES	78.5%	88.5%	92.3%
PCA with MLES	79.3%	88.8%	92.5%
LDA	79.3%	86.8%	84.0%

performs better than any other method, beating PCA, PCA initialization with MLES re-estimation and LDA with MLES re-estimation.

Experimental results for speaker verification (using a speaker-independent impostor model for eigenGMM decoding) are shown in Table 4.11 for a 40-dimensional eigenspace on 64 GMMs obtained from 72 speakers (disjoint from the 10 client speakers).

Eigenspace Adaptation

We trained an eigenspace for 64 GMMs on the 630 TIMIT speakers, each supplying 10 sentences, and carried out enrollment and testing on YOHO. The adaptation was performed on the enrollment data from the 10 clients; we observed no significant difference when much larger amounts of adaptation data were used.

Remarks

The eigenvoice approach forces models for the client and test speakers to be confined to a low-dimensional subspace obtained from training data (Thyes et al., n.d.). For sparse amounts of enrollment data (5-10 seconds), this approach consistently outperforms conventional GMM training. For larger amounts of enrollment data, the loss of degrees of freedom caused by restriction to eigenspace leads to inferior performance. For speaker verification, an advantage of the approach is that, in its "eigendistance decoding" variant, it dispenses with the need for impostor models.

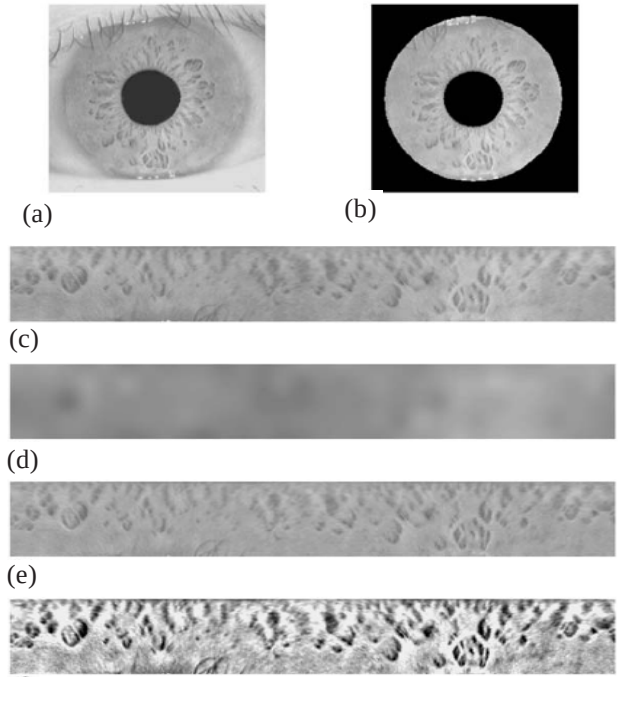
Of the eigenspace training methods tested, LDA appears to be the most promising. However, all the eigenvoice methods may run into difficulty when trained on acoustically diverse databases with small amounts of data per speaker. For instance, speaker-dependent variability in TIMIT is less important than phoneme identity, channel effects and phonetic context (Kajarekar, Malayath, & Hermansky, 1999); this makes it likely that eigenspaces trained on TIMIT and similar databases will confound speaker-dependent information with these other types of information. Clearly, the top priority for future work is the development of more robust eigenspace training techniques.

IRIS RECOGNITION

Introduction

As a physiological biometric, iris recognition aims to identify persons using iris characteristics of human eyes. Recently, iris recognition has received more attention due to its high reliability (Mansfield, Kelly, Chandler, & Kane, 2001; Daugman, 2001; Wildes, 1997). This section makes an attempt to reflect shape information of the iris, analyzing local intensity variations of an iris image. In the framework, a set of 1D intensity signals is constructed to contain the most important local variations of the original 2D iris image. Gaussian-Hermite moments of such intensity signals reflect to a large extent their various spatial modes and are used as distinguishing features. A resulting high-dimensional feature vector is mapped into a low-dimensional subspace using FLD, and then the nearest center classifier based on cosine similarity measure is adopted for classification. Extensive experimental results show that the proposed method is effective and encouraging.

Figure 4.31. Iris image preprocessing



(a) Original image, (b) localized image, (c) normalized image, (d) estimated background illumination, (e) lighting corrected image, and (f) enhanced image (Ma, Tan, Wang, & Zhang, 2004)

The human iris, an annular part between the pupil (generally appearing black in an image) and the white sclera (as shown in Figure 4.33a), has an extraordinary structure and provides many interlacing minute characteristics, such as freckles, coronas, stripes, furrows, crypts and so forth. These visible characteristics, generally called the texture of the iris, are unique to each subject (Daugman, 2001; Wildes, 1997; Adler, 1965; Davision, 1962; Johnson, 1991; Bertillon, 1885; Siedlarz, 1994; Daugman, 1993; Daugman, 2003; Wildes, Asmuth, Green, Hsu, Kolczynski, Matey, & McBride, 1996; Flom & Safir, 1987). Individual differences that exist in the development of anatomical structures in the body result in such uniqueness. Some research work (Wildes, 1997; Daugman, 1993, 1994; Flom & Safir, 1987; Wildes, Asmuth, Hsu, Kolczynski, Matey, & McBride, 1996) has also stated that the iris is essentially stable through a person's life. Furthermore, since the iris is an internal organ as well as externally visible, iris-based personal identification systems can be non-invasive to their users (Daugman, 2001; Wildes, 1997; Daugman, 1993; daugman, 2003; Wildes, Asmuth, Green, Hsu, Kolczynski, Matey, & McBride, 1996; Daugman, 1994; Wildes, smuth, Hsu, Kolczynski, Matey, & McBride, 1996), which is important for practical applications. Iris recognition relies greatly on how to accurately represent local details of the iris. Different from previous work on iris recognition (Daugman, 2001; Wildes, 1997; Dagman, 1993, 2003; Wildes et al., 1996; Boles & Boashash, 1998; Lim, Lee, Byeon, & Kim, 2001; Zhu, Tan, & Wang, 2000; Ma, Wang, & Tan, 2002; Ma, Wang, & Tan, 2002), this algorithm analyzes local intensity variations to reflect shape information of the iris.

Image Preprocessing

An iris image, as shown in Figure 4.31a, contains not only the iris but some irrelevant parts (e.g., eyelid, pupil, etc.). A change in the camera-to-eye distance may also result in variations in the size of the same iris. Furthermore, the brightness is not uniformly distributed because of non-uniform illumination. Therefore, before feature extraction, the original image needs to be preprocessed to localize and normalize the iris, and reduce the influence of the factors mentioned above.

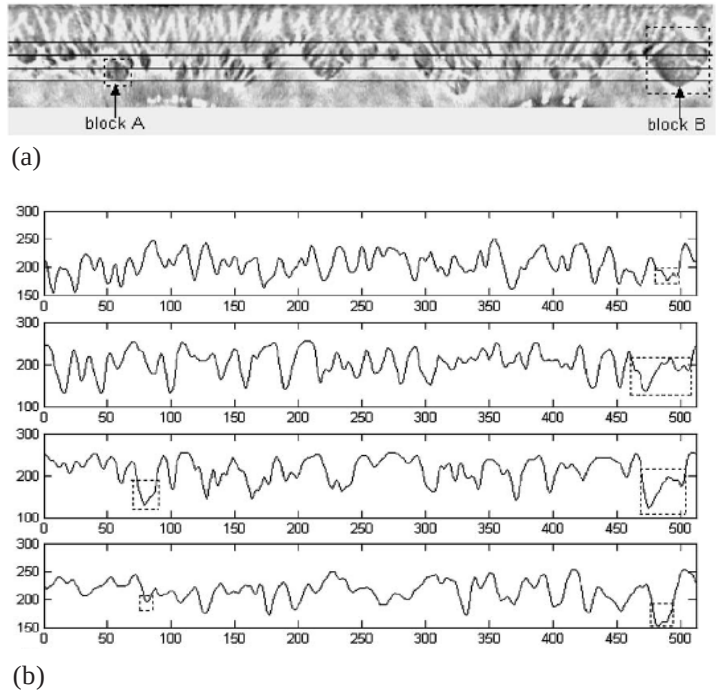
Feature Extraction

As is known, the shape is generally characterized by the object contours (namely, image edges). However, it is difficult to well segment the irregular iris blocks of a very small size in gray images. Such irregular blocks cause noticeable local intensity variations in iris images. Therefore, we approximately reflect shape information of the iris characteristics by analyzing the resulting local variations in the iris image. Figure 4.32 gives a brief illustration of the relationship between local intensity variations and iris images.

As Figure 4.32a shows, the iris comprises a large number of small irregular blocks (i.e., irregular regions in an image). Two crown-shaped regions (block A and B) marked by the dotted box in the Figure are used to illustrate how we analyze shape information of the irregular iris blocks. In gray images, local intensity variations in the boundary of a region are generally sharper than those in the inside of a region.

This can be observed in the intensity signals plotted in Figure 4.32. The segments circumscribed by the dotted box in the four plots denote the intensity variations of the crown-shaped regions in the horizontal direction. That is, an irregular iris block can cause

Figure 4.32. Illustration of the relationship between local intensity variations and iris images



(a) A normalized iris image, (b) four intensity signals, which, respectively, correspond to gray values of four rows (denoted by four black lines in (a) of the normalized image (Ma, Tan, Wang, & Zhang, 2004)

significant local variations in the intensity signals. The shape of an iris block determines both the number of the intensity signals that this block can affect and the interval of significant local variations. The interval of significant local variations caused by the same iris block is also different among intensity signals. Therefore, we expect to approximately reflect shape information of the iris blocks by analyzing local variations of the intensity signals. The moment-based method has been widely used to represent local characteristics of images in pattern recognition and image processing (Ma, Wang, & Tan, 2002; Prokop & Reeves, 1992; Liao & Pawlak, 1996; Loncaric, 1998; Shen, 1997; Shen, Shen, & Shen, 2000). Here, Gaussian-Hermite moments are adopted to characterize local variations of the intensity signals.

Generation of 1D Intensity Signals

Generally, local details of the iris spread along the radial direction in the original image corresponding to the vertical direction in the normalized image (see Figure 4.31). Therefore, information density in the angular direction corresponding to the horizontal direction in the normalized image is much higher than than in other directions (Daugman, 2001; Ma, Wang, & Tan, 2002).

In experiments, Ma, Tan, Wang, and Zhang (2004) found that the iris region closer to the pupil provides the most discriminating information for recognition and is also rarely occluded by eyelids and eyelashes. So they extracted features only in the region closer to the pupil. This region takes up about 80% of the normalized image.

Gaussian-Hermite Moments

Moments have been widely used in pattern recognition and image processing, especially in various shape-based applications. More recently, the orthogonal moment-based method has been one of the active research topics in shape analysis. Unlike commonly used geometric moments, orthogonal moments use orthogonal polynomial functions as transform kernels, which produces minimal information redundancy. The detailed study on the different moments and their behavior evaluation may be found in Liao and Pawlak (1996) and Shen, Shen, and Shen (2000). Here, Gaussian-Hermite moments are used for feature extraction due to their mathematical orthogonality and effectiveness for characterizing local details of the signal (Shen, 1997; Shen, Shen, & Shen, 2000). The *n*th order 1D Gaussian-Hermite moment $M_n(x)$ of a signal $S(x)$ is defined as:

$$M_n(x) = \int_{-\infty}^{+\infty} K_n(t) S(x+t)dt, n = 0,1,2,\cdots, \tag{4.43}$$
$$K_n = g(t,\sigma)H(t/\sigma),$$
$$H_n(t) = (-1)^n \exp(t^2)\frac{d^n \exp(-t^2)}{dt^n},$$

where $g(t;\sigma)$ is a Gaussian function, $H_n(t)$ is a nth order Hermite polynomial function, and the kernel K_n(t) is a product of these two functions. Figure 4.33 shows the spatial responses of the Gaussian-Hermite moment kernels of different orders and their corresponding Fourier transforms.

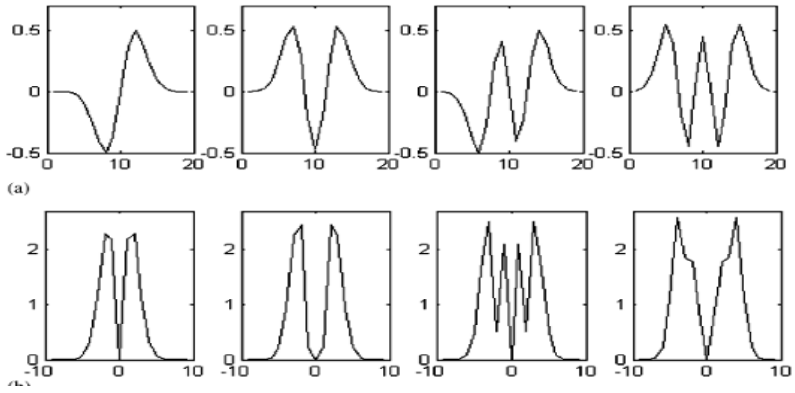
Feature Vector

For each signal S_i, we can calculate its Gaussian-Hermite moment $M_{i,n}$ of order n according to Equation 4.43. In our experiments, we generate 10 intensity signals, $i \in \{1, \ldots, 10\}$, and use four different order Gaussian-Hermite moments, $n \in \{1, 2, 3, 4\}$. In addition, the space constant of the Gaussian function in Equation 4.43 affects the shape of the Gaussian-Hermite moment kernels. In the experiments, it is set to 2.5. Since the outputs $M_{i,n}$ denote different local features derived using different moment kernels, we concatenate all these features together to form an integrated feature vector:

$$V = [M_{1,1}, M_{1,2}, \ldots, M_{10,3}, M_{10,4}]^T \tag{4.44}$$

where T is the transpose operator. Since the length of each intensity signal is 512, the feature vector V includes 20,480 (512×10×4) components. To reduce the space dimension and the subsequent computational complexity, we can "downsample" each moment $M_{i,n}$ by a factor d before the concatenation. Here, downsampling means replacing d succes-

Figure 4.33. Gaussian-Hermite moment kernels



(a) Spatial responses of Gaussian-Hermite moment kernels, order 1 to 4, (b) the Fourier spectra of (a)

sive feature elements by their average. So, the downsampled feature vector V^d can be rewritten as follows:

$$V^d = [M^d_{1,1}, M^d_{1,2}, \cdots, M^d_{10,3}, M^d_{10,4}]^{\mathrm{T}} \tag{4.45}$$

Invariance

It is desirable to obtain an iris representation invariant to translation, scale and rotation. Invariance to translation is intrinsic to our algorithm, since feature extraction is based on a set of intensity signals instead of the original image. To achieve approximate scale invariance, we normalize an input image to a rectangular block of a fixed size. We can also provide approximate rotation invariance by downsampling each moment $M_{i,n}$. That is, each moment $M_{i,n}$ is circularly shifted before downsampling.

Matching (Using LDA)

By feature extraction, an iris image can be represented as a high-dimensional feature vector, depending on the downsampling factor *d*. To reduce the computational cost and improve the classification accuracy, Fisher linear discriminant is first used to generate a new feature vector with salient information of the original feature vector, and then the nearest center classifier is adopted for classification in a low-dimensional feature subspace.

Two popular methods for dimensionality reduction are PCA and FLD. Compared with PCA, FLD not only utilizes information of all samples but also shows interest in the underlying structure of each class. In general, the latter can be expected to outperform the former (Belheumer et al., 1997; Zhao, Chellappa, & Phillips, 1999). FLD searches for projected vectors that best discriminate different classes in terms of maximizing the ratio of between-class to within-class scatter, which can be described by the following equation:

$$W = \arg\max_{w} \frac{|W^{\mathrm{T}} S_B W|}{|W^{\mathrm{T}} S_W W|} = [w_1 w_2 \cdots w_m], \tag{4.46}$$

$$S_B = \sum_{i=1}^{c} (\mu_i - \mu)(\mu_i - \mu)^{\mathrm{T}}$$

$$S_W = \sum_{i=1}^{c} \sum_{j=1}^{N_I} (x_j^i - \mu_i)(x_j^i - \mu_i)^{\mathrm{T}},$$

where c is the total number of classes, μ is the mean of all samples, μ_i is the mean of the ith class, N_i is the number of samples of the ith class, x_{ij} is the jth sample of the ith class, S_B is the between-class scatter matrix and S_W is the within-class scatter matrix. In our experiments, an EFM is utilized for the solution to the optimal projective matrix W. The EFM method (Li, Wang, & Zhang, 2004) improves the generalization capability of FLD using a more effective numerical solution approach. Further details of FLD may be found in Belhumeur et al. (1997); Zhao, Chellappa, and Phillips (1999); Liu and Wechsler (2002); Swets and Weng (1996); and Fukunaga (1991).

The new feature vector is defined as:

$$f = W^T V^d \tag{4.47}$$

where V^d is the original feature vector. The proposed algorithm makes use of the nearest center classifier defined in Equation 4.48 for classification in a low-dimensional feature subspace constructed by the optimal projective matrix W.

$$j = \arg\min_{1 \le i \le c} d(f, f_i), \tag{4.48}$$

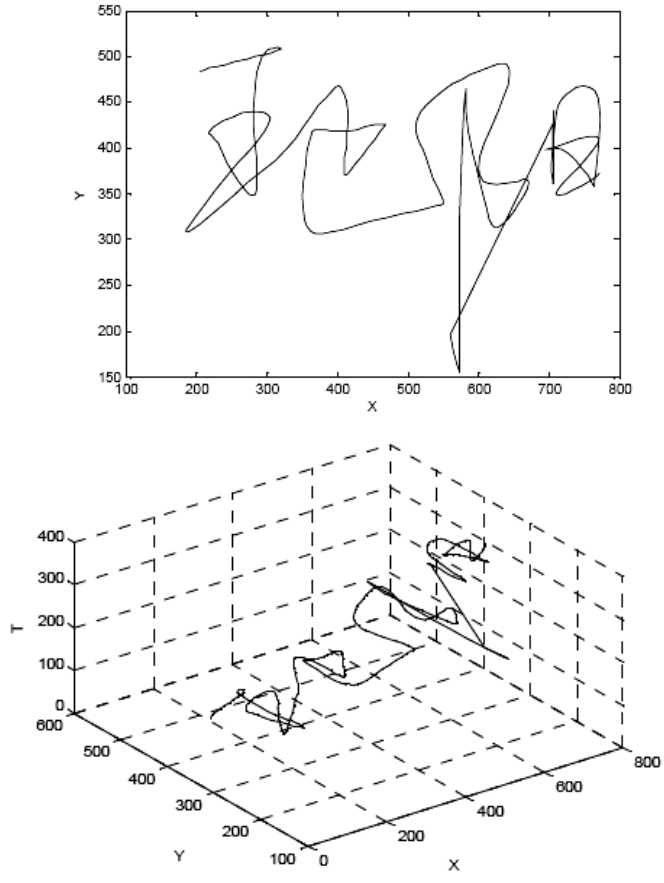
$$d(f, f_i) = 1 - \frac{f^{\mathrm{T}} f_i}{\| f \| \| f_i \|}$$

where f is the feature vector of an unknown sample, f_i is the feature vector of the ith class, c is the total number of classes, $\|\cdot\|$ denotes the Euclidean norm and $d(f, fi)$ is cosine similarity measure. The feature vector f is classified into the jth class, the closest mean, using the similarity measure $d(f, fi)$.

Table 4.13. The typical operating states of the proposed method (Ma, Tan, Wang, & Zhang, 2004)

False match rate(%)	False non-match rate(5)
0.001	1.13
0.01	1.05
0.1	0.65

Figure 4.34. An original signature is shown in 2D and 3D with time information (Li, Wang, & Zhan, 2004)



Remarks

Among existing methods for iris recognition, those proposed by Daugman (2001; Ma, Wang, & Tan, 2002; Prokop & Reeves, 1992), Wildes et al. (1996) and Boles et al. (Davision, 1962), respectively, are the best known. Moreover, they characterize local details of the iris from different viewpoints; that is, phase-based approach; texture analysis-based approach and zero-crossing representation method. The results in experiments (Ma, Tan, Wang, & Zhang, 2004) show that FDA can be used in iris recognition very perfectly. See the results in Table 4.13.

SIGNATURE VERIFICATION

Introduction

Signature verification is commonly used to approbate the contents of a document or to authenticate a financial transaction. With the rapid development and wide appli-

cation of network, online signature verification becomes a hot topic in the field of Internet security. Many people engage in the research of online signature verification, and a wide range of methods have been presented in the past 10 years.

Dr. Nalwa presented a very detailed approach by relying primarily on pen dynamics during the production of the signature instead of the detailed shape of a signature (Nalwa, 1997). Jain et al. introduced an online signature verification system based on string matching and writer-dependent threshold (Jain, Griess, & Conell, 2002). Martens et al., Munich et al. and Yong et al. discussed an online signature verification system based on dynamic time warping (DTW), which is originated from the field of speech recognition (Martens & Claesen, 1997; Munich & Perona, 1999; Yong & Jian, 1999). Other methods, such as hidden Markov models (HMMs), artificial neural networks (ANN) and genetic algorithm (GA) applied to signature verification are also introduced in some literature. In this chapter, a kind of online signature verification method based on PCA and minor component analysis (MCA) is introduced. We divide a signature into several segments, according to predefined rules, and search an optimal path in which two signatures can be well corresponded by using DTW. Reference signatures are used to produce principal component (PC) and minor component (MC) with K-L transform. Signature verification will be based on PCs and MCs. Contrasted with other applications of PCA, both PC and MC are used in this section, and MC, especially, plays a very important role to verification.

Signature Processing and Segmentation

In this system, a common and cheap tablet is used as a capture device. With a fixed sample frequency, a signature can be described by a series of points (x_i, y_i, t_i). An original signature captured by a general device is shown in Figure 4.34.

We can divide a signature into two sequences, (x_i, t_i) and (y_i, t_i), corresponding to x- and y- coordinates. Since the noise coming from the capture device and handshake, it is necessary to normalize the signature and smooth it with a sliding window. In this section, the Gaussian function is used to smooth x- and y-curves of a signature. Curves about x- and y-axis after preprocessing are shown in Figure 4.35. Now we define a set of features to describe these curves of a signature. The sequence of inflexions in troughs and crests are detected and marked in each curve, and then a series of features for each pair of neighboring inflexions to describe a curve is defined as follows:

The length of these two neighboring inflexions in x-coordinates is:

$$l_x^i = x_i - x_{i-1} \tag{4.49}$$

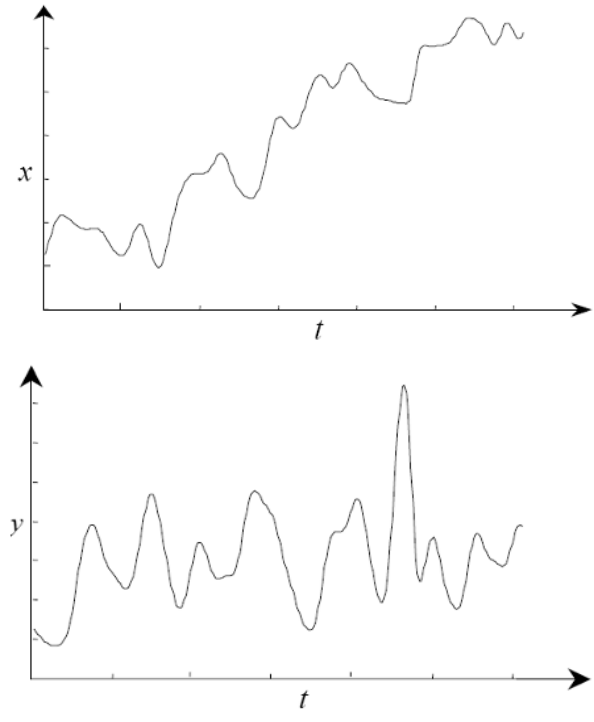
The obliquity of the line connecting these neighboring inflexions is:

$$\theta^i = \arctan\left(\frac{y_i - y_{i-1}}{x_i - x_{i-1}}\right) \tag{4.50}$$

The position of the inflexion is:

$$p^i = \frac{x_i - x_1}{L_s} \tag{4.51}$$

Figure 4.35. Curves of a signature about x- and y-axis after preprocessing



The mean velocity between these two neighboring inflexions in *x*-coordinates is:

$$V_{mean}^{i} = \frac{l_x^i}{T_i} \tag{4.52}$$

Deviation of velocity is:

$$V_d^i = \sqrt{\frac{\sum_{j=1}^{M_i} (V_j^i - V_{mean}^i)^2}{M_i - 1}} \tag{4.53}$$

where (x_i, y_i) is the i^{th} inflexion, x_1 is the *x*-coordinate of the first inflexion and *Ls* is the length of the whole signature in *x*-coordinates. M_i is the number of sample points of i^{th} segment. Now, we have a sequence of vectors to describe a whole signature about the *x*-axis.

$$H_x = (h_1, h_2, \cdots, h_N), h_i = (l_x^i, \theta^i, p^t, V_{mean}^i, V_d^i) \tag{4.54}$$

The sequence of vectors about *y*-axis can be deduced in the same way.

Flexible Matching and Stable Segments Extraction

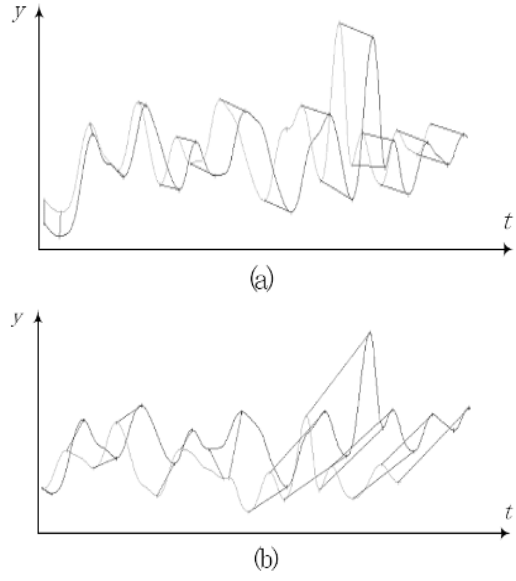
As we know, the segment numbers of two signatures produced by the same signer are often different. So we must try to match the segments of a signature with another one correctly. DTW is a technique well suited for this matching. Using DTW, we get an optimal path in which the summation of distances of all corresponding segments between two curves is minimum. Yet, not all features of h_i (defined in Equation 4.54) are suitable in this matching process, because dynamic features (V^i_{mean} and V^i_d) are more unstable than static features (l^i_x, θ^i and p^i). So we define a sequence S of static features for DTW.

$$S = (s_1, \cdots, s_N), s_i = (l^i_x, \theta^i, p^i) \tag{4.55}$$

To find the optimal path of two sequences $S^p = (s^p_1, \cdots, s^p_I)$ and $S^q = (s^q_1, \cdots, s^q_J)$, which describe the static information of two 1-D signature curves, a DTW algorithm is introduced, as follows:

$$D_{i,j} = \min \begin{pmatrix} D_{i,j} \\ D_{i+1,j} + d_{2,1} \\ D_{i,j+1} + d_{1,2} \end{pmatrix}, 1 \le i \le I, 1 \le j \le J \tag{4.56}$$

Figure 4.36. Matching two signatures about y-axis by DTW



(a) Matching between two genuine signatures, (b) matching between a genuine signature and a skilled forgery

where $D_{i,j}$ is the distance of static features between s_i^p and s_j^q, and $d_{1,2}$ and $d_{2,1}$ are punishments. After DTW matching, the optimal path is recorded (Figure 4.36). Since K-L transform needs all vectors with the same length, we must make the segment number of each signature the same. For this reason, after matching each signature with others in reference, we search those segments that appear in each matching paths and mark them as stable segments. The vectors of stable segments in each reference signature will be the same length. With both static and dynamic features of these stable segments, a feature vector for a reference can be described as:

$$H^m = (h_1^m, h_2^m, \cdots, h_n^m), m = 1, \cdots, M \tag{4.57}$$

For a test signature, we also search its stable segments in the same way. If there is a matched segment in the test, it will be marked and both static and dynamic features will be added into a feature vector H'; otherwise, a predefined null segment will be added into H'. After searching all stable segments in references, a feature vector H' of a test signature whose length is the same with the reference will be produced.

PCA and MCA

PCA is an essential technique for data compression and feature extraction, and has been widely used. In most applications of PCA, people usually throw away MCs and only care about PCs, as PCs contain most information of the reference data. In this section, both PCs and MCs are used. Even the MC plays a more important role than the PC. The feature vector $H = (h_1, h_2, \ldots, h_n)$ of each signature is reshaped to a 1D vector s whose length is $N = 5 \times n$ (5 characters are used to represent a segment). Then M reference vector $(s_i^r - \bar{s})$, $i = 1,...,M$ are combined in a $N \times M$ reference matrix S:

$$S = (s_1^r - \bar{s}, s_2^r - \bar{s}, \cdots, s_M^r - \bar{s}), \bar{s} = \frac{1}{M}\sum_{i=1}^{M} s_i^r \tag{4.58}$$

where M is the number of reference signatures.

The eigenvector u_i and eigenvalue λ_i, $i = 1,...,5 \times N$ of the covariance matrix $\mathbf{\Sigma}$ of S can be deduced by K-L transform. The space constituted by fore M - 1 eigenvectors contains all the information of reference signatures. We call these eigenvectors with large eigenvalues as PCs and eigenvectors with very small and zero eigenvalues as MCs.

We separate eigenvectors $U = \{u_i, i = 1,..., 5 \times N\}$ into two parts: One is U_P constituted of PCs and the other is U_M constituted of MCs, defined by

$$U_P = \{u_i, i = 1, \cdots, M-1\}, U_M = \{u_i, i = M, \cdots, 5 \times N\} \tag{4.59}$$

Considering the effect of different eigenvectors in U_P, large coefficients are given to the eigenvectors with small eigenvalues, and small coefficients are given to the eigenvectors with large eigenvalues. U_P is non-linearly transformed into:

$$\hat{U}_P = \left\{ \frac{u_1}{\sqrt{\lambda_1}}, \frac{u_2}{\sqrt{\lambda_2}}, \cdots, \frac{u_{M-1}}{\sqrt{\lambda_{M-1}}} \right\}, u_i \in U_P \tag{4.60}$$

where u_i is the eigenvector with λ_i.

The reference s_i^r and the test s^t can be transformed into new space of $\hat{U}_P$.

$$\hat{s}_i^r = s_i^r \cdot \hat{U}_P, \hat{s}^t = s^t \cdot \hat{U}_P \tag{4.61}$$

where $\hat{s}_i^r$ and $\hat{s}^t$ is new vectors in the space of $\hat{U}_P$, and their length is $M - 1$.

Now we turn to introduce MCA into signature verification. We give a concept: energy of a signature in U_M. Since all the information of references is contained in the space of U_P, the energy of references in the space of U_M is very small or zero. So, we can judge a test signature as genuine signature or forgery by its energy in the space of U_M. The less energy of a test signature in the space of U_M is, the more similar it is with the references. The energy G in the space of U_M can be defined as:

$$G = \| (s^t - \bar{s}) \cdot U_M \| \tag{4.62}$$

where s^t is the test signature and $\bar{s}$ is the mean vector of references.

From the above applications of PCs and MCs, a distance to judge a test signature as genuine signature or forgery can be defined as:

$$Dis = \frac{1}{M} \sum_{i=1}^{M} \| s^t - s_i^r \| \cdot C_1 + \| (s^t - \bar{s}) \cdot U_E \| \cdot C_2 \tag{4.63}$$

where C_1 and C_2 are weights of these two parts, which come from PCA and MCA.

The effects of PC and MC in signature verification can be understood as follows: In the feature space of reference signatures, some parts are stable, which can be used to represent a genuine signature well, and other parts are unstable, which represent the inner varieties of reference signatures. With K-L transform and resizing the space of PCs non-linearly (Equation 4.60), the inner variety can be restrained well. Furthermore, since the energy of reference signature in the space of MCs is very small or zero, the less energy of a test signature in the space of MCs is, the more similar it is with references. Taking the above advantage, PCA and MCA can be well applied into online signature verification.

Experiment Results

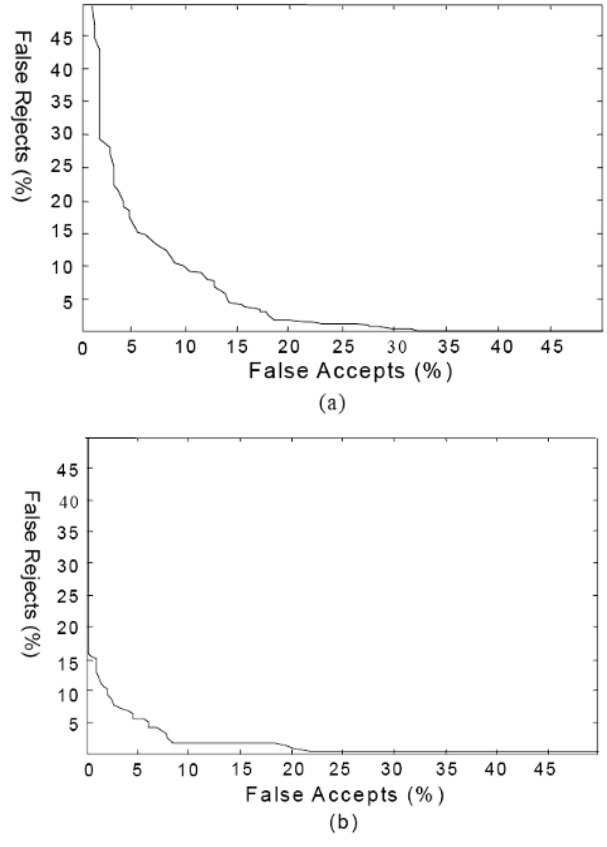
The proposed method has been implemented and evaluated with 1,215 signatures. Based on a database containing 810 genuine signatures and 405 skilled forgeries of 81 signers, the experiments were carried out. Each signer was asked to write his or her signature 10 times, of which five signatures were used as references and the other five

signatures were used for testing. Observed during the whole producing course of a genuine signature, some forgers were asked to imitate five signatures of each genuine signer. To compare my method with normal DTW, in my experiments, we only use the local features of signature as defined earlier. The tradeoff curve using DTW and the discriminance of Euclidean distance is shown in Figure 4.37a, and the tradeoff curve using DTW and PCA/MCA is shown in Figure 4.37b. The EER with DTW and the discriminance of Euclidean distance is about 10%, and the EER of my method is about 5%.

Remarks

An online signature verification method based on DTW and PCA/MCA is proposed in this section. Taking advantage of PCA and MCA, the stable and unstable information of reference signatures can be well analyzed and applied in signature verification. During this course, the unstable parts are restrained and the stable parts are impelled. The MC plays a very important role, though it is often ignored in other applications. It is still an open question to how signatures of a signer can be well divided into same segments, as K-L transform needs vectors with the same length. In future work, all kinds of feature

Figure 4.37. Error tradeoff curves of experiments



comparison and the relation between PC and MC need to be analyzed in detail in online signature verification based on PCA and MCA.

SUMMARY

We have described some approaches designed to cope with pattern recognition complication and to find the true invariant for recognition. In this chapter, we introduced some more successful applications using the two important PCA and LDA approaches, such as face recognition, palm identification, gait application, ear biometrics, speaker identification, iris recognition and signature verification.

The eigenspace approach applies the KL transform for feature extraction. It greatly reduces the facial feature dimension, yet maintains reasonable discriminating power. Taking eigenface approach for an example, it transforms face images into a small set of characteristic feature images, which are the principal components of the initial training set of face images. Recognition is performed by projecting a new image into the subspace spanned by the eigenfaces ("face space") and then classifying the face by comparing its position in face space with the positions of known individuals. Automatically learning and later recognizing new faces is practical within this framework. Recognition under reasonably varying conditions is achieved by training on a limited number of characteristic views (e.g., a "straight-on" view, 45º view and profile view). The approach has advantages over other face recognition schemes in its speed and simplicity, learning capacity and relative insensitivity to small or gradual changes in the face image.

The Fisher's approach, though some variants of the algorithm work on feature extraction as well, further reduces the eigenspace by the FLD. For example, fisherface is a face recognition algorithm insensitive to large variation in lighting direction and facial expression. Taking a pattern classification approach, we consider each pixel in an image as a coordinate in a high-dimensional space. We take advantage of the observation that the images of a particular face, under varying illumination but fixed pose, lie in a 3D linear subspace of the high-dimensional image space — if the face is a Lambertian surface without shadowing. However, since faces are not truly Lambertian surfaces and do indeed produce self-shadowing, images will deviate from this linear subspace. The fisherface method is based on FLD and produces well-separated classes in a low-dimensional subspace, even under severe variation in lighting and facial expressions. It linearly projects the image into a subspace in a manner that discounts those regions of the face with large deviation. Extensive experimental results demonstrate that the proposed "fisherface" method has error rates lower than those of the eigenface technique for tests on the Harvard and Yale face databases.

REFERENCES

Adini, Y., Moses, Y., & Ullman, S. (1997). Face recognition: The problem of compensating for changes in illumination direction. *IEEE Transactions on Pattern Analysis and Machine Intelligence, 19*(7), 721-732.

Adler, F. (1965). *Physiology of the eye: Clinical application* (4th ed.). London: The C.V. Mosby Company.

Atal, B. S. (1976). Automatic recognition of speakers from their voices. *Proceedings of the IEEE, 64*, 460-475.

Bamber, D. (2001). Prisoners to appeal as unique "earprint" evidence is discredited. *Telegraph Newspaper (UK)*.

Belhumeur, P. N., Hespanha, J. P., & Kriegman, D. J. (1997). Eigenfaces vs. fisherfaces: Recognition using class specific linear projection. *IEEE Transactions on Pattern Analysis and Machine Intelligence, 19*(7), 711-720.

Belhumeur, P. N., & Kriegman, D. J. (1996). What is the set of images of an object under all possible lighting conditions? *IEEE Proceedings of the Conference on Computer Vision and Pattern Recognition*.

BenAbdelkader, C., Culter, R., & Davis, L. (2002). Stride and cadence as a biometric in automatic person identification and verification. *Proceedings of the International Conference on Automatic Face and Gesture Recognition* (pp. 284-294).

BenAbdelkader, C., Culter, R., Nanda, H., & Davis, L. (2001). EigenGait: Motion-based recognition of people using image self-similarity. *Proceedings of the International Conference on Audio- and Video-Based Biometric Person Authentication* (pp. 284-294).

Bengio, S., Mariethoz, J., & Maroel, S. (2001). *Evaluation of biometric technology on XM2VTS* (IDIAP Research Report 01-21). Martigny, Switzerland: Dalle Molle Institute for Perceptual Artificial Intelligence.

Bertillon, A. (1885). La couleur de l'iris. *Review of Science, 36*(3),65-73.

Beymer, D. (1994). Face recognition under varying pose. *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition* (pp. 756-761).

Bobick, A., & Johnson, A. (2001). Gait recognition using static, activity-specific parameters. *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*.

Boles, W., & Boashash, B. (1998). A human identification technique using images of the iris and wavelet transform. *IEEE Transactions on Signal Processing*, *46*(4), 1185-1188.

Bromba GmbH. (2003). *Bioidentification frequently asked questions*. Retrieved from www.bromba.com/faq/biofaqe.htm

Brunelli, R., & Poggio, T. (1993). Face recognition: Features vs. templates. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, *15*(10), 1042-1053.

Burge, M., & Burger, W. (1998). Ear biometrics. In A. Jain, R. Bolle, & S. Pankanti (Eds.), *BIOMETRICS: Personal identification in a networked society* (pp. 273-286). Kluwer Academic.

Burge, M., & Burger, W. (2000). Ear biometrics in computer vision. *Proceedings of the 15th International Conference of Pattern Recognition, ICPR* (pp. 826-830).

Campbell, J. P. (1997). Speaker recognition: A tutorial. *Proceedings of the IEEE, 85*(9), 1437-1462.

Chang, K., Bowyer, K. W., Sarkar, S., & Victor, B. (2003). Comparison and combination of ear and face images in appearance-based biometrics. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, *25*(9), 1160-1165.

Chen, Q., Wu, H., & Yachida, M. (1995). Face detection by fuzzy pattern matching. *Proceedings of the International Conference on Computer Vision* (pp. 591-596).

Collins, R., Gross, R., & Shi, J. (2002). Silhouette-based human identification from body shape and gait. *Proceedings of the International Conference on Automatic Face and Gesture Recognition*.

Craw, I., Tock, D., & Bennet, A. (1992). Finding face features. *Proceeding of the European Conference on Computer Vision* (pp. 92-86).

Cui, Y., Swets, D., & Weng, J. (1995). Learning-based hand sign recognition using SHOSLIF-M. *Proceedings of the International Conference on Computer Vision* (pp. 631-636).

Daugman, J. (1993). High confidence visual recognition of persons by a test of statistical independence. *IEEE Transactions on Pattern Analysis and Machine Intelligence, 15*(11),1148-1161.

Daugman, J. (1994). Biometric personal identif4cation system based on iris analysis. *U.S. Patent No. 5291560*.

Daugman, J. (2001). Statistical richness of visual phase information: update on recognizing persons by iris patterns. *International Journal of Computer Vision, 45*(1), 25-38.

Daugman, J. (2003). Demodulation by complex-valued wavelets for stochastic pattern recognition. *International Journal Wavelets, Multi-Resolution Information Processing, 1*(1), 1-17.

Davision, H. (1962). *The eye*. London: Academic.

Doddington, G. R. (1985). Speaker recognition: Identifying people by their voices. In *Proceedings of the IEEE*, *73*(11), 1651-1664.

Duda, R. O., & Hart, P. E. (1973). *Pattern classification and scene snalysis*. New York: Wiley.

Duda, R. O., Hart, P. E.,& Stork, D. G. (2000). *Pattern classification* (2nd ed.). New York: John Wiley.

Duda, R. O., Hart, P. E., & Stork, D. G. (2001). *Pattern classification*. New York: John Wiley & Sons.

Duta, N., Jain, A. K., & Mardia, K. V. (2001). Matching of palmprint. *Pattern Recognition Letters*, *23*(4), 477-485.

Duta, N., Jain, A. K., & Mardia, K. V. (2002). Matching of palmprint. *Pattern Recognition Letters, 23*, 477-485.

Feraud, R. (1997). PCA: Neural networks and estimation for face detection. *The NATO Advanced Study Institute to Application.* Scotland: Stirling.

Fisher, R. A. (1936). The use of multiple measures in taxonomic problems. *Annals of Eugenics, 7*, 179-188.

Flanagan, J. (1972). *Speech analysis synthesis and perception* (2nd ed.). New York/Berlin: Springer-Verlag.

Flom, L., & Safir, A. (1987). Iris recognition system. *U.S. Patent No. 4641394*.

Forensic Evidence News. (2000). *Ear identification.*

Forsyth, M. E. (1995). *Hidden Markov models for automatic speaker verification.* PhD thesis. University of Edinburgh.

Fukunaga, K. (1991). *Introduction to statistical pattern recognition* (2nd ed.). New York: Academic Press.

Furui, S. (1991). Speaker-dependent-feature extraction, recognition and processing techniques. *Speech Communication*, *10*, 505-520.

Furui, S. (1997). Recent advances in speaker recognition. *Lecture Notes in Computer Science 1206, Proceedings of Audio and Video Biometric Person Authentication AVBPA'97, First International Conference* (pp. 237-252).

Georphiades, S., Belhumeur, P. N., & Kriegman, D. J. (2001). From few to many: Illumination cone model for face recognition under variable lighting and pose. *IEEE Transactions on Pattern Analysis and Machine Intelligence, 23*(6), 643-660.

Gnanadesikan, R., & Kettenring, J. R. (1989). Discriminant analysis and clustering. *Statistical Science*, *4*(1), 34-69.

Hallinan, P. (1994). A low-dimensional representation of human faces for arbitrary lighting condition. *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition* (pp. 995-999).

Hallinan, P. (1995). A *deformable model for face recognition under arbitrary lighting conditions* (PhD thesis). Cambridge, MA: Harvard University.

Haritaoglu, I., Harwood, D., & Davis, L. (2000). W4: Real-time surveillance of people and their activities. *IEEE Trans. Pattern Analysis and Machine Intelligence*, *22*(8), 809-830.

Harmon, L. D. (1973). The recognition of faces. *Scientific American, 29*, 71-82.

Higgins, A., Bahler, L., & Porter, J. (1991). Speaker verification using randomized phrase prompting. *Digital Signal Processing*, *1*(2), 89-106.

Hoogstrate, A. J., Van den Heuvel, H., & Huyben, E. (2000). *Ear identification based on surveillance camera's images*. Retrieved October 7, 2003, from http://www.forensic-evidence.com/site/ID/IDearCamera.html

Huang, P., Harris, C., & Nixon, M. (1999). Human gait recognition in canonical space using temporal templates. *IEEE Proceedings of the Vision Image and Signal Processing Conference*, *146*(2), 93-100.

Hurley, D. J., Nixon, M. S., & Carter, J. N. (2000a). Automated ear recognition by force field transformations. *Proceedings of the IEE Colloquium: Visual Biometrics*.

Hurley, D. J., Nixon, M. S., & Carter, J. N. (2000b). A new force field transform for ear and face recognition. *Proceedings of the IEEE 2000 International Conference on Image Processin ICIP* (pp. 25-28).

Iannarelli, A. (1989). *Ear identification* (Forensic Identification Series). Fremont, CA: Paramont.

Jain, A. K., et al. (1998). *Biometrics*. Klumer Academic.

Jain, A. K., Griess, F. D., & Connell, S. D. (2002). On-line signature verification. *Pattern Recognition*, 2963-2972.

Johnson, A., & Bobick, A. (2001). A multiview method for gait recognition using static body parameters. *Proceedings of the International Conference on Audio- and Video-Based Biometric Person Authentication* (pp. 301-311).

Johnson, R. (1991). *Can iris patterns be used to identify people*. Los Alamos: Los Alamos National Laboratory, Chemical and Laser Sciences Division, LA-12331-PR.

Kajarekar, S., Malayath, N., & Hermansky, H. (1999). *Analysis of speaker and channel variability in speech.* Workshop on Automatic Speech Recognition and Understanding, Keystone, CO.

Kuhn, R., Junqua, J-C., Nguyen, P., & Niedzielski, N. (2000). Rapid speaker adaptation in eigenvoice space. *IEEE Transactions on Speech Audio Processing, 8*(6), 695-707.

Kuhn, R., Nguyen, P., Junqua, J-C., Boman, R., Niedzielski, N., Fincke, S., et al. (1999). Fast speaker adaptation using *A Priori* knowledge. *The 1999 IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP-99)* (Vol. 2, pp. 749-752). Phoenix, AZ.

Kuhn, R., Nguyen, P., Junqua, J-C., Goldwasser, L., Niedzielski, N., Fincke, S., et al. (1998). Eigenvoices for speaker adaptation. *The 5th International Conference on Spoken Language Processing (ICSLP-98)* (Vol. 5, pp. 1771-1774). Sydney, Australia.

Kuno, Y., Watanabe, T., Shimosakoda, Y., & Nakagawa, S. (1996). Automated detection of human for visual surveillance system. *Proceedings of the International Conference on Pattern Recognition* (pp. 865-869).

Lammi, H-K. (n.d.). *Ear biometrics.* Lappeenranta: Lappeenranta University of Technology, Department of Information Technology, Laboratory of Information Processing.

Lee, L., & Grimson, W. (2002). Gait analysis for recognition and classification. *Proceedings of the International Conference on Automatic Face and Gesture Recognition* (pp. 155-162).

Legetter, C. J., & Woodland, P. C. (1995). Maximum likelihood linear regression for speaker adaptation of continuous density Hidden Markov Models. *Computer Speech and Language*, *9*, 171-185.

Li, B., Wang, K., & Zhang, D. (2004). On-line signature verification based on PCA and MCA. *The First International Conference on Biometric Authentication (ICBA 2004),* LNCS-3072 (pp. 540-546).

Li, W., & Zhang, D. (2002). Palmprint identification by Fourier transform. *International Journal of Pattern Recognition and Artificial Intelligence, 16*(4), 417-432.

Li, W., Zhang, D., & Xu, Z. (2002). Palmprint identification by Fourier transform. *International Journal of Pattern Recognition and Artificial Intelligence, 16*(4), 417-432.

Liao, S., & Pawlak, M. (1996). On image analysis by moments. *IEEE Trans. Pattern Analysis and Machine Intelligence, 18*(3), 254-266.

Lim, S., Lee, K., Byeon, O., & Kim, T. (2001). Effcient iris recognition through improvement of feature vector and classifier. *ETRI Journal, 23*(2), 61-70.

Lin, S-H. (2000). An introduction to face recognition technology. *Informing Science Special Issue on Multimedia Informing Technologies, Part II, 3*(1).

Linguistic Data Consortium. (1994). *YOHO speaker verification.* Retrieved from http://morph.ldc.upenn.edu/Catalog/

Linguistic Data Consortium. (1996). *TIMIT acoustic-phonetic continuous speech corpus*. Retrieved from http://morph.ldc.upenn.edu/Catalog/

Little, J., & Boyd, J. (1998). Recognizing people by their gait: The shape of motion. *Videre: Journal of Computer Vision Research*, *1*(2), 2-32.

Liu, C., & Wechsler, H. (2002). Gabor feature based classification using the enhanced Fisher linear discriminant model for face recognition. *IEEE Transactions on Image Processing, 11*(4), 467-476.

Liu, K., Cheng, Y., & Yang, J. (1993). Algebraic feature extraction for image recognition based on an optimal discriminant criterion. *Pattern Recognition, 26*(6), 903-911.

Loncaric, L. (1998). A survey of shape analysis techniques. *Pattern Recognition, 31*(8), 983-1001.

Lu, G., Zhang, D., & Wang, K. (2003). Palmprint recognition using eigenpalms features. *Pattern Recognition Letters, 24*, 1463-1467

Ma, L., Tan, T. N., Wang, Y. H., & Zhang, D. X. (2004). Local intensity variation analysis for iris recognition. *Pattern Recognition, 37*, 1287-1298.

Ma, L., Wang, Y., & Tan, T. (2002a). Iris recognition based on multi-channel gabor filtering. In *Proceedings of the Fifth Asian Conference on Computer Vision* (Vol. 1, pp. 279-283).

Ma, L., Wang, Y., & Tan, T. (2002b). Iris recognition using circular symmetric filters. *Proceedings of the 16th International Conference on Pattern Recognition* (Vol. II, pp. 414-417).

Mammone, R., Zhang, X., & Ramachandran, R. (1996). Robust speaker recognition – A featurebased approach. *IEEE Signal Processing Magazine, 13*(5), 58-71.

Mansfield, T., Kelly, G., Chandler, D., & Kane, J. (2001). *Biometric product testing final report, issue 1.0.* Middlesex: National Physical Laboratory of UK.

Martens, R., & Claesen, L. (1997). Dynamic programming optimization for on-line signature verification. *Proceedings of 4th ICDAR '97* (pp. 653-656).

Moghaddam, B., Wahid, W., & Pentland, A. (1998). Beyond eigenfaces: Probabilistic matching for face recognition. *3rd Face and Gesture,* 30-35.

Moreno, B., Sánchez, Á.,& Vélez, J. F. (1999). On the use of outer ear images for personal identification in security applications. *IEEE 33rd Annual International Carnahan Conference on Security Technology* (pp. 469-476).

Morgan, J. (1999). *Court holds earprint identification not generally accepted in scientific community*. State vs. David Wayne Kunze. Retrieved September 9, 2003, from http://www.forensic-evidence.com/site/ID/ID-Kunze.html

Munich, M. E., & Perona, P. (1999). Continuous dynamic time warping for translation-invariant curve alignment with applications to signature verification. *Proceedings of the Seventh IEEE International Conference on Computer Vision* (pp. 108-115).

Murase, H., & Nayar, S. (1995). Visual learning and recognition of 3-D objects from appearance. *International Journal of Computer Vision, 14*, 5-24.

Murase, H., & Sakai, R. (1996). Moving object recognition in eigenspace representation: Gait analysis and lip reading. *Pattern Recognition Letters, 17*, 155-162.

Nalwa, V. S. (1997). Automatic on-line signature verification. *Proceedings of the IEEE, 85*(2), 215-239.

Nguyen, P., Wellekens, C., & Junqua, J. C. (1999). Maximum likelihood eigenspace and MLLR for speech recognition in noisy environments. *Eurospeech-99, 6*, 2519-2522.

O'Shaughnessy, D. (1987). *Speech communication, human and machine*. Reading, MA: Addison-Wesley.

Peng, H., & Zhang, D. (1997). Dual eigenspace method for human face recognition. *IEE Electronics Letters, 33*(4), 283-284.

Pentland, A., Moghaddam, B., Starner. (1994). View-based and modular eigenspaces for face recognition. *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition* (pp. 84-91).

Phillips, J., Moon, H., Rizvi, S., & Rause, P. (2000). The FERET evaluation methodology for face recognition algorithms. *IEEE Transactions on Pattern Analysis and Machine Intelligence, 22*(10), 1090-1104.

Prokop, R., & Reeves, A. (1992). A survey of moment-based techniques for unoccluded object representation and recognition. *CVGIP: Graphical models Image Process., 54*, 438-460.

Pun, K. H., & Moon, Y. S. (2004). Recent advances in ear biometrics. *Proceedings of the Sixth IEEE International Conference on Automatic Face and Gesture Recognition (FGR'04)*.

Reynolds, D. A. (1995). Speaker identification and verification using Gaussian mixture speaker models. *Speech Communication*, *17*, 177-192.

Reynolds, D., & Rose, R. (1995). Robust text-independent speaker identification using Gaussian mixture speaker models. *IEEE Transactions on Speech and Audio Processing*, *3*(1), 72-83.

Romdhani, S., Gong, S., & Psarrou, A. (1999). A multi-view nonlinear active shape model using kernel PCA. In T. Pridmore & D. Elliman (Eds.), *Proceedings of the 10th British Machine Vision Conference (BMVC99)* (pp. 483-492). London: BMVA Press.

Rosenberg, A. (1976). Automatic speaker verification: A review. *Proceedings of the IEEE*, *64*(4), 475-487.

Rosenberg, E., & Soong, F. K. (1992). Recent research in automatic speaker recognition. In S. Furui & M.M. Sondhi (Eds.), *Advances in speech signal processing* (pp. 701-738). New York: Marcel Dekker.

Shen, J. (1997). Orthogonal Gaussian-Hermite moments for image characterization. In *Proceedings of SPIE, Intelligent Robots and Computer Vision XVI: Algorithms, Techniques, Active Vision, and Materials Handling* (pp. 224-233).

Shen, J., Shen, W., & Shen, D. (2000). On geometric and orthogonal moments. *International Journal of Pattern Recognition and Artificial Intelligence, 14*(7), 875-894.

Shu, W., Rong, G., Bain, Z., & Zhang, D. (2001). Automatic palmprint verification. *International Journal Image Graphics, 1*(1), 135-152.

Shu, W., & Zhang, D. (1998). Automated personal identification by palmprint. *Optical Engineering*, *37*(8), 2359-2362.

Siedlarz, J. (1994). Iris: more detailed than a fingerprint. *IEEE Spectrum, 31,* 27.

Sirovitdh, L., & Kirby, M. (1987). Low-dimensional procedure for the characterization of human faces. *Journal of the Optical Society of America, A2*, 519-524.

Sukkar, R. A., Gandhi, M. B., & Setlur, A. R. (2000). Speaker verification using mixture decomposition discrimination. *IEEE Transactions on Speech and Audio Processing*, *8*(3), 292-299.

Sutherland, A., & Jack, M. (1988). Speaker verification. In M. Jack & J. Laver (Eds.), *Aspects of speech technology* (pp. 185-215). Edinburgh: Edinburgh University Press.

Swets, D., & Weng, J. (1996). Using discriminant eigenfeatures for image retrieval. *IEEE Transactions on Pattern Analysis and Machine Intelligence, 18*(8), 831-836.

Thyes, O., Kuhn, R., Nguyen, P, & Junqua, J-C. (n.d.). Speaker identification and verification using eigenvoices. Santa Barbara: Panasonic Technologies.

Turk, M., & Pentland, A. (1991a). Face recognition using eigenfaces. *IEEE,* 586-591.

Turk, M., & Pentland, A. (1991b). Eigenfaces for recognition. *Journal of Cognitive Neuroscience, 3*(1).

Vega, I., & Sarkar, S. (2002). Experiments on gait analysis by exploiting nonstationarity in the distribution of feature relationships. *Proceedings of the International Conference on Pattern Recognition*.

Victor, B., Bowyer, K., & Sarkar, S. (2002). An evaluation of face and ear biometrics. *Proceedings of International Conference on Pattern Recognition* (pp. 429-432).

Wang, L., Ning, H., Tan, T., & Hu, W. (2004). Fusion of static and dynamic body biometrics for gait recognition. *IEEE Trans. on Circuits and Systems for Video Technology*, *14*(2), 149-159.

Wang, L., Tan, T., Ning, H., & Hu, W. (2003). Silhouette analysis-based gait recognition for human identification. *IEEE Trans. on Pattern analysis and machine Intelligence*, *25*(2), 1505-1519

Wildes, R. (1997). Iris recognition: An emerging biometric technology. *Proceedings of the IEEE* (Vol. 85, pp. 1348-1363).

Wildes, R., Asmuth, J., Green, G., Hsu, S., Kolczynski, R., Matey, J., & McBride, S. (1996). A machine-vision system for iris recognition. *Machine Vision and Applications, 9*, 1-8.

Wildes, R., Asmuth, J., Hsu, A., Kolczynski, R., Matey, J., & McBride, S. (1996). Automated, noninvasive iris recognition system and method. *US Patent No. 5572596*.

Winter, D. (1990). *The biomechanical and motor control of human movement* (2nd ed.). New York: John Wiley & Sons.

Wu, X., Zhang, D., & Wang, K. (2003). fisherpalms based palmprint recognition. *Pattern Recognition Letters, 24*, 2829-2838.

Yan, P., & Bowyer, K. W. (n.d.). *2D and 3D ear recognition.* Department of Computer Science and Engineering, University of Notre Dame.

Yang, Y., & Levine, M. (1992). The background primal sketch: An approach for tracking moving objects. *Machine Vision and Applications*, *5*, 17-34.

Yong, J., & Jian, L. (1999). On-line handwriting signature verification based on elastic matching of 3D curve. *Journal of Huazhong University of Science and Technology*, *7*(5), 14-16.

You, J., Li, W., & Zhang, D. (2002). Hierarchical palmprint identification via multiple feature extraction. *Pattern Recognition*, *35*(4), 847-859.

Yuela, P. C., Dai, D. Q., & Feng, G. C. (1998). Wavelet-based PCA for human face recognition. *IEEE Southwest Symposium on Image Analysis and Interpretation,* 223-228.

Zhang, D. (2000). *Automated biometrics – Technologies and systems*. Kluwer Academic Publishers.

Zhang, D., & Shu, W. (1999). Two novel characteristics in palmprint verification: Datum point invariance and line feature matching. *Pattern Recognition, 32*, 691-702.

Zhao, W., Chellappa, R., & Phillips, P. (1999). *Subspace linear discriminant analysis for face recognition (Tech. Report CAR-TR-914)*. University of Maryland, Center for Automation Research.

Zhu, Y., & Tan, T. (2000). Biometric personal identification based on handwriting. *Pattern Recognition*, *2*, 797-800.

Zhu, Y., Tan, T., & Wang, Y. (2000). Biometric personal identi4cation based on iris patterns. In *Proceedings of the 15th International Conference on Pattern Recognition* (Vol. 2, pp. 805-808).

Section II

Improved BID Technologies

Chapter V

Statistical Uncorrelation Analysis

ABSTRACT

This chapter shows a special LDA approach called optimal discrimination vectors (ODV), which requires that every discrimination vector satisfy the Fisher criterion. After introduction, we first give some basic definitions. Then, uncorrelated optimal discrimination vectors (UODV) are proposed. Next, we introduce an improved UODV approach, and offer some experiments and analysis. Finally, we summarize some useful conclusions.

INTRODUCTION

ODV is a special LDA approach that requires that every discrimination vector satisfy the Fisher criterion. Various literature discuss ODV. Foley and Sammon present a set of optimal discrimination vectors for two-class problems, which requires the discrimination vectors to satisfy the orthogonality constraint (Foley & Sammon, 1975). Foley's approach is called the Foley-Sammon ODV (FSODV). Okada and Tomita propose an optimal orthonormal system for discrimination analysis (Okada & Tomita, 1985).

Duchene et al. propose orthogonal discrimination analysis in a transformed space (Duchene & Leclercq, 1988). Liu, Cheng and Yang propose more comprehensive solutions for the ODV set (Liu, Cheng, & Yang, 1993).

While all of the above ODV approaches employ the orthogonality constraint, Jin, Yang, Hu, Tang and Lou recently proposed an UODV (Jin, Yang, Hu, & Lou, 1993) approach and a related theorem (Jin, Yang, Tang, & Hu, 2001). UODV uses the constraint of statistical uncorrelation. The experimental results show that UODV produces better outcomes than FSODV on the same hand-written data, where the only difference lies in their respective constraints. On the other hand, Yang, Yang, and Zhang (2002) prove that the uncorrelation constraint is theoretically superior to the orthogonality constraint. However, some disadvantages still exist in Jin's approach. First, in order to guarantee that S_w is nonsingular, it uses the between-class correlation matrix, $\sum_b = \sum_{i=1}^{c} m_i m_i^T$, as the production matrix of the KL transform, where m_i is the average value of the ith class samples. It is not a TPCA method that uses S_t as the production matrix. Therefore, it cannot reflect the total scatter of the whole sample set. Second, its theorem is merely suitable for a specific situation, where the non-zero discrimination values of the Fisher criterion are unequal mutually, implying that it cannot be applicable to other situations.

BASIC DEFINITION

Suppose that X is an N-dimensional sample set, and $w_1, w_2, \ldots w_c$ are C known pattern classes of X. Let m_i and $P_i (i = 1, 2, \ldots, C)$ be the mean vector and a priori probability of class w_i. Let m be the mean vector of X. The between-class scatter matrix, S_b, the within-class scatter matrix, S_w and the total scatter matrix, S_t, are defined as Equations 3.43, 3.37 and 3.41.

The Fisher criterion is expressed by the maximum value of the following function as Equation 3.31. And here, we change the symbol **w** to φ to explain the following problems simply.

The first step is to perform TPCA; that is, to take S_t as the production matrix of the K-L transform. Suppose that the rank of S_t is r_t. We get r_t eigenvectors corresponding to the non-zero eigenvalues of S_t, which form the transform matrix W_{TPCA}. Thus, any N-dimensional sample from X can be transformed into an r_t-dimensional vector. The reason we choose TPCA transform is TPCA has a favorable property; namely, the statistical uncorrelation. Suppose that there are two different discrimination vectors φ_1 and φ_2 $(\varphi_1 \neq \varphi_2)$ The statistical uncorrelation in Jin, Yang, Hu and Lou (2001) is defined as:

$$\varphi_1^T S_t \varphi_2 = 0 \tag{5.1}$$

Let $W_{TPCA} = [w_1\ w_2\ \ldots w_n]$. According to the definition of W_{TPCA}, it is obvious that:

$$w_j^T S_t w_i = 0, \quad j \neq i, \quad 1 \leq (i, j) \leq n \tag{5.2}$$

Obviously, TPCA can satisfy the statistical uncorrelation.

UNCORRELATED OPTIMAL DISCRIMINATION VECTORS

Fisher Vector

As mentioned above, the Fisher criterion function is defined as Equation 3.31 in Chapter III.

The vector φ_1 corresponding to maximum of $F(\varphi)$ is the Fisher optimal discriminant direction; that is, Fisher's vector. It means that the projected set of samples on the direction φ_1 has the minimal within-class scatter and the maximal between-class scatter in the one-dimensional subspace spanned by φ_1. Fisher's vector φ_1 is the eigenvector corresponding to maximum eigenvalue of the following eigenequation:

$$S_b \varphi_1 = \lambda S_w \varphi_1 \tag{5.3}$$

Foley-Sammon Discriminant Vectors

Let φ_1 be Fisher's vector. Suppose r directions $\varphi_1, \varphi_2, \ldots, \varphi_r$ ($j \geq 1$) are obtained. We can obtain the $(r+1)$th direction φ_{r+1}, which maximizes the Fisher criterion function $F(\varphi)$ with the following orthogonality constraints:

$$\varphi_{r+1}^T \varphi_i = 0 \quad (i = 1,2,\ldots,r) \tag{5.4}$$

Based on the optimal discriminant vectors $\varphi_1, \varphi_2, \ldots, \varphi_k$, we can define the following linear transform from $\Re^k$.

$$Y = \begin{bmatrix} y_1 \\ y_2 \\ \vdots \\ y_k \end{bmatrix} = \begin{bmatrix} \varphi_1^T \\ \varphi_2^T \\ \vdots \\ \varphi_k^T \end{bmatrix} X \tag{5.5}$$

Equation 5.5 with Equation 5.2 is called as Foley-Sammon discriminant transformation. It is easy to obtain the following theorem.

Theorem 5.1. Any two features y_i and y_j ($j \neq i$) in Foley-Sammon discriminant vectors are statistically correlated.

Proof. It is easy to obtain the following equation:

$$E\left[(y_i - Ey_i)(y_j - Ey_j)\right] = \varphi_j^T S_t \varphi_i \tag{5.6}$$

With the orthogonality constrain Equation 5.7; we have inequality; in general:

$$\varphi_j^T S_t \varphi_i \neq 0 \quad (j \neq i) \tag{5.7}$$

Therefore, we cannot obtain the following equation in general:

$$E[(y_i - Ey_i)(y_j - Ey_j)] = 0 \tag{5.8}$$

Uncorrelated Discriminant Vectors

Let $\varphi_1 = \xi_1$ be the Fisher vector. Suppose that j vectors $\varphi_1, \varphi_2, \ldots, \varphi_r (r \geq 1)$ are obtained. In order to obtain uncorrelated discriminant features, we can calculate the $(r+1)$-th vector φ_{r+1}, which maximizes the Fisher criterion function $F(\varphi)$ with the following conjugate orthogonality constraints:

$$\varphi_{r+1}^T S_t \varphi_i = 0 \quad (i = 1,2,\ldots,r) \tag{5.9}$$

Equation 5.5 with Equation 5.9 is called as uncorrelated discriminant transformation, as we have the following theorem.

Theorem 5.2. Any two features y_i and y_j $(j \neq i)$ of the uncorrelated discriminant vectors are statistically uncorrelated.

Proof. It is obvious that we have the following equation:

$$E[(y_i - Ey_i)(y_j - Ey_j)] = \varphi_j^T S_t \varphi_i \equiv 0 \tag{5.10}$$

These vectors $\{\varphi_j\}$ are called the uncorrelated optimal discrimination vectors (UODVs), since for any $i \neq j$, $\varphi_i^T X$ and $\varphi_j^T X$ are uncorrelated.

We can compute the $(r+1)$th uncorrelated discriminant direction φ_{r+1} according to the next section.

A Theorem on UODV

In this section, we present a theorem on UODV and give some discussions.

Theorem 5.3. For C-class problems, suppose that the between-class covariance matrix S_b has rank $(C-1)$ and the within-class covariance matrix S_w is nonsingular. Let the $(C-1)$ non-zero eigenvalues of $S_w^{-1} S_b$ be represented and ordered from the largest to the smallest as:

$$\lambda_1 \geq \lambda_r \geq \ldots \geq \lambda_{C-1} > 0 \tag{5.11}$$

and suppose:

$$\lambda_i \neq \lambda_j \quad (i \neq j) \tag{5.12}$$

For $r \leq C-1$, regardless of the direction of eigenvectors, the rth UODV φ_r is the rth eigenvector ϕ_r of $S_w^{-1}S_b$ corresponding to the rth largest nonzero eigenvale λ_r, i.e.:

$$\varphi_r = \phi_r \quad (r = 1, 2, \ldots C-1) \tag{5.13}$$

For $r > C-1$, the rth UODV φ_r has the Fisher criterion value of zero, i.e.:

$$F(\varphi_r) = 0 \quad (r > C-1) \tag{5.14}$$

According to Equation 5.14, for $r > C-1$, the rth UODV φ_r cannot supply any more discriminant information on the meaning of the Fisher criterion function. Thus, the number of effective UODVs can be said to be $(C-1)$ for C-class problems. Therefore, UODV can be said to be equivalent to CODV based on Equation 5.13, and the Fisher criterion function can be said to be equivalent to the Fisher criterion Equation 3.31 with the conjugate orthogonality Equation 5.9.

It is always advantageous to know what the best features for classification are. The Bayes error is an accepted criterion to evaluate feature sets. Since the Bayes classifier for C-class problems compares a posteriori probabilities, $q_1(X)$, $q_2(X)$, . . . , $q_C(X)$, and classifies the unknown sample X to the class whose a posteriori probability is the largest, these c functions carry sufficient information to set up the Bayes classifier. Furthermore, since $\sum_{i=1}^{C} q_i(X)=1$, only $(C-1)$ of these c functions are linearly independent. Thus, Fukunaga (1990) called $\{q_1(X), q_2(X), \ldots, q_C(X)\}$ the ideal feature set for classification.

In practice, the a posteriori probability density functions are hard to obtain. The Bayes error is too complex to extract features for classification and has little practical utility. Fisher criterion functions and Equation 3.31 are much simpler. UODV have much more practical utility.

IMPROVED UODV APPROACH

Approach Description

In the non-zero subspace of S_t, an equivalent variant of Fisher criterion is employed (Fukunaga, 1990).

$$F(\varphi) = \frac{\varphi^T S_b \varphi}{\varphi^T S_t \varphi} \tag{5.15}$$

The first discriminant vector φ_1, which is the eigenvector corresponding to the maximum eigenvalue of $S_t^{-1}S_b$, can be easily obtained. Then, according to the following theorem, we calculate the $(j+1)$-th optimal discrimination vector, $\varphi_{(j+1)}$ $(j \geq 1)$, which maximizes Equation 5.15 and simultaneously satisfies the following constraints for the statistical uncorrelation as Equation 5.9.

Theorem 5.4. $\varphi_{(j+1)}$ *is the eigenvector corresponding to the maximum eigenvalue of the following equation:*

$$PS_b\varphi = \lambda S_t\varphi \tag{5.16}$$

where:

$$P = I - S_t D^T (DS_t D^T)^{-1} D \tag{5.17}$$

$$D = [\varphi_1\ \varphi_2 \cdots \varphi_j]^T \text{ and } I = diag(1,1,\cdots,1) \tag{5.18}$$

Proof. Note that φ_j has been normalized:

$$\varphi_j^T \varphi_j = 1 \tag{5.19}$$

Let $\varphi_{(j+1)}$ satisfy the following equation:

$$\varphi_{j+1}^T S_t \varphi_{j+1} = c \tag{5.20}$$

Use the Lagrange multiplier method to transform Equation 5.1, including all the constraints:

$$L(\varphi_{j+1}) = \varphi_{j+1}^T S_b \varphi_{j+1} - \lambda(\varphi_{j+1}^T S_t \varphi_{j+1} - c) - \sum_{i=1}^{j} \mu_i \varphi_{j+1}^T S_t \varphi_i \tag{5.21}$$

Let the partial derivatives $\dfrac{\partial L(\varphi_{j+1})}{\partial \varphi_{j+1}}$ be equal to zero:

$$2S_b\varphi_{j+1} - 2\lambda S_t\varphi_{j+1} - \sum_{i=1}^{j} \mu_i S_t \varphi_i = 0 \tag{5.22}$$

Multiplying Equation 5.22 by φ_k^T ($k = 1, 2, \cdots, j$), we obtain a set of j equations:

$$2\varphi_k^T S_b \varphi_{j+1} - \sum_{i=1}^{j} \mu_i \varphi_k^T S_t \varphi_i = 0 \text{ , } (k = 1, 2, \ldots, j) \tag{5.23}$$

that is:

$$2\begin{bmatrix} \varphi_1^T \\ \varphi_2^T \\ \vdots \\ \varphi_j^T \end{bmatrix} S_b \varphi_{j+1} - \begin{bmatrix} \varphi_1^T \\ \varphi_2^T \\ \vdots \\ \varphi_j^T \end{bmatrix} S_t \begin{bmatrix} \varphi_1^T \\ \varphi_2^T \\ \vdots \\ \varphi_j^T \end{bmatrix}^T \begin{bmatrix} \mu_1 \\ \mu_2 \\ \vdots \\ \mu_j \end{bmatrix} = 0 \tag{5.24}$$

Let:

$$\mu = [\mu_1 \ \mu_2 \cdots \mu_j]^T \tag{5.25}$$

Using Equations 5.18 and 5.25, Equation 5.24 can be represented as:

$$DS_t D^T \mu = 2DS_b \varphi_{j+1} \tag{5.26}$$

Therefore, we obtain:

$$\mu = 2(DS_t D^T)^{-1} DS_b \varphi_{j+1} \tag{5.27}$$

Equation 5.22 can be written in the following form:

$$2S_b \varphi_{j+1} - 2\lambda S_t \varphi_{j+1} - S_t D^T \mu = 0 \tag{5.28}$$

Substitute Equation 5.27 into Equation 5.28, so that:

$$2S_b \varphi_{j+1} - 2\lambda S_t \varphi_{j+1} - S_t D^T \left[2\left(DS_t D^T\right)^{-1} DS_b \varphi_{j+1}\right] = 0 \tag{5.29}$$

i.e.:

$$\left[I - S_t D^T \left(DS_t D^T\right)^{-1} D\right] S_b \varphi_{j+1} = \lambda S_t \varphi_{j+1} \tag{5.30}$$

Thus, we can obtain Equation 5.16-5.17.

It is noted that the small sample-size problem does not exist in our algorithm. We first use TPCA to generate the non-zero subspace of S_t, and then obtain the optimal discrimination vectors by using Equation 5.1 to express the Fisher criterion. Obviously, our algorithm can effectively obtain the discrimination vectors satisfying the Fisher criterion, even when S_w is singular. Accordingly, it is a simple and complete solution for the small sample-size problem.

Generalized UODV Theorem

Referring to Jin et al. (2001), we present a new and generalized theorem for UODV.

Theorem 5.5. Suppose that S_t and S_b are $n \times n$ square matrix, S_b has rank r, and the non-zero eigenvalues of $S_t^{-1}S_b$ are represented in descending order as $\lambda_1 \geq \lambda_2 \geq \ldots \geq \lambda_r > 0$ (and $\lambda_{r+1} = \lambda_{r+2} = \ldots = \lambda_n = 0$). Let F denote the function in Equation 5.16 and the k^{th} eigenvector ϕ_k of $S_t^{-1}S_b$ correspond to the k^{th} λ_k $(1 \leq k \leq n)$. We have the following conclusions:

First:

$$F(\varphi_k) = F(\phi_k) = \lambda_k \tag{5.31}$$

Second, $\varphi_1 = \phi_1$*, and for* $2 \le k \le n$*, if* $\lambda_k \neq \lambda_{k-1}$*, then* $\varphi_k = \phi_k$.

Proof. From the definition of ϕ_k, we have $S_b\phi_k = \lambda_k S_t \phi_k$ $(1 \le n)$.

- **Step 1.** Proof for $k = 1$.
  Due to the definition of j_1, that is $S_b j_1 = l\, S_t j_1$, it is clear that:

$$\lambda = F(\varphi_1) = F(\phi_1) = \lambda_1 \text{ and } \varphi_1 = \phi_1 \tag{5.32}$$

- **Step 2.** Prove $F(\varphi_k) = F(\phi_k) = \lambda_k$ for $2 \le k \le (r+1)$.
  According to Theorem 1, we have:

$$\left[I - S_t D^T \left(DS_t D^T\right)^{-1} D\right] S_b \varphi_k = \lambda S_t \varphi_k \quad \left(D = [\varphi_1\ \varphi_2\ \cdots\ \varphi_{k-1}]^T\right) \tag{5.33}$$

  Due to the uncorrelation constraints, it is obvious that $DS_t j_k = 0$. From the proven results, we have:

$$F(\varphi_i) = F(\phi_i) = \lambda_i \quad (1 \le i \le (k-1)) \tag{5.34}$$

  that is, $S_b\varphi_i = \lambda_i S_t \varphi_i$ and $S_b\phi_i = \lambda_i S_t \phi_i$. So, we obtain the following equation:

$$DS_b\varphi_k = \begin{bmatrix} \varphi_1^T S_b \varphi_k \\ \varphi_2^T S_b \varphi_k \\ \vdots \\ \varphi_{k-1}^T S_b \varphi_k \end{bmatrix} = \begin{bmatrix} (S_b\varphi_1)^T \varphi_k \\ (S_b\varphi_2)^T \varphi_k \\ \vdots \\ (S_b\varphi_{k-1})^T \varphi_k \end{bmatrix} = \begin{bmatrix} (S_t\varphi_1)^T \varphi_k / \lambda_1 \\ (S_t\varphi_2)^T \varphi_k / \lambda_2 \\ \vdots \\ (S_t\varphi_{k-1})^T \varphi_k / \lambda_{k-1} \end{bmatrix} \tag{5.35}$$

$$= \left[\frac{1}{\lambda_1}, \frac{1}{\lambda_2}, \ldots, \frac{1}{\lambda_{k-1}}\right]^T \begin{bmatrix} \varphi_1^T S_t \varphi_k \\ \varphi_2^T S_t \varphi_k \\ \vdots \\ \varphi_{k-1}^T S_t \varphi_k \end{bmatrix} = 0$$

- Substituting Equation 5.34 into Equation 5.33, we have $S_b j_k = l\, S_t j_k$. According to $S_b f_k = l_k S_t f_k$ and Equation 5.33, both l and l_k should be the k^{th} largest eigenvalue of $S_t^{-1} S_b$, that is $l = l_k$. Hence, we have:

$$F(j_k) = F(f_k) = l_k \tag{5.36}$$

- It is noted that $F(j_k) = F(f_k) = l_k = 0$ when $k = r + 1$.

- **Step 3.** Prove $F(j_k) = F(f_k) = l_k$ for $(r + 2) \le k \le n$.
  Let $D = [D_1\ D_2]$, where $D_1 = [j_1 j_2 \ldots j_r]^T$ and $D_2 = [j_{r+1} j_{r+2} \ldots j_{k-1}]^T$. We use D_1 to replace D in Equation 5.33 and obtain:

$$D_1 S_b \varphi_k = 0 \tag{5.37}$$

Due to $F(j_i) = F(f_i) = l_i = 0$ $((r + 1) \le i \le (k - 1))$, that is $S_b j_i = l_i S_t j_i = 0$, we have:

$$D_2 S_b \varphi_k = \begin{bmatrix} \varphi_{r+1}^T S_b \varphi_k \\ \varphi_{r+2}^T S_b \varphi_k \\ \vdots \\ \varphi_{k-1}^T S_b \varphi_k \end{bmatrix} = \begin{bmatrix} (S_b \varphi_{r+1})^T \varphi_k \\ (S_b \varphi_{r+2})^T \varphi_k \\ \vdots \\ (S_b \varphi_{k-1})^T \varphi_k \end{bmatrix} = 0 \tag{5.38}$$

Therefore:

$$DS_b \varphi_k = [D_1 S_b \varphi_k \ \ D_2 S_b \varphi_k] = 0 \tag{5.39}$$

Substitute Equation 5.39 into Equation 5.33, we have:

$$S_b \varphi_k = \lambda\ S_t \varphi_k \tag{5.40}$$

According to $F(j_k) \le F(j_{k-1})$ and the proved result $F(j_{k-1}) = F(f_{k-1}) = l_{k-1} = 0$, we have:

$$0 \le \lambda = F(\varphi_k) \le F(\varphi_{k-1}) = \lambda_{k-1} = 0 \tag{5.41}$$

Consequently, we obtain the equation $F(j_k) = l = 0 = l_k = F(f_k)$.

- **Step 4.** Prove that for $2 \le k \le n$, if $l_k \neq l_{k-1}$, then $j_k = f_k$.
  If $2 \le k \le r$, then for $1 \le i \le (k - 1)$, in terms of the equations $S_b j_i = l_i S_t j_i$ and $S_b f_k = l_k S_t f_k$, we have:

$$\varphi_i^T S_t \phi_k = \frac{1}{\lambda_i} (\varphi_i^T S_b) \phi_k = \frac{1}{\lambda_i} \varphi_i^T (S_b \phi_k) = \frac{1}{\lambda_i} \varphi_i^T (\lambda_k S_t \phi_k) = \frac{\lambda_k}{\lambda_i} \varphi_i^T S_t \phi_k \tag{5.42}$$

Since $\lambda_k \neq \lambda_{k-1}$, i.e., $\lambda_1 \ge \lambda_2 \ge \cdots \ge \lambda_{k-1} > \lambda_k > 0$ and $\lambda_i \neq \lambda_k$, it is clear that:

$$\varphi_i^T S_t \phi_k = 0 \tag{5.43}$$

If $k = (r + 1)$, then $l_k = 0$. Substituting l_k into Equation 5.42, we can also obtain Equation 5.43.

If $(r + 2) \leq k \leq n$, then $l_{k-1} = l_k = 0$; we cannot obtain Equation 5.43.

Therefore, if $l_k \neq l_{k-1}$, f_k satisfies Equation 5.43. So, f_k has the same Fisher value l_k with j_k. According to the definition of our improved UODV algorithm, if the discrimination vector corresponding to l_k satisfies the uncorrelation constraint, it should be unique. Consequently, we obtain:

$$j_k = f_k \tag{5.44}$$

From Theorem 5.5, we obtain the following two corollaries:

Corollary 5.1. If the rank of $S_t^{-1}S_b$ is r, then we can only use the first r optimal discrimination vectors in our algorithm, instead of all the vectors.

Proof. Due to Equation 5.1, except for the first r optimal discrimination vectors, the Fisher discrimination values of the remained discrimination vectors are all equal to zero. That is, only the first r vectors carry the effective Fisher discrimination information. Therefore, we can use them to represent all the vectors.

Corollary 5.1 indicates that we do not need to calculate all the vectors. This will save some computational time, especially when we use matrix data, where $r << n$.

Corollary 5.2. If the rank of $S_t^{-1}S_b$ is r, and the r non-zero eigenvalues of $S_t^{-1}S_b$ are mutually unequal — that is, they can be represented in the descending order:

$$\lambda_1 > \lambda_2 > \cdots > \lambda_r > 0\,,\ \lambda_{r+1} = \lambda_{r+2} = \cdots = \lambda_n = 0$$

— then the optimal discrimination vectors of our algorithm can be represented by the first r normalized eigenvectors of $S_t^{-1}S_b$ corresponding to the r non-zero eigenvalues.

Proof. Since:

$$\lambda_i \neq \lambda_j\,, 1 \leq \{i, j\} \leq r \tag{5.45}$$

According to Theorem 5.5, we have:

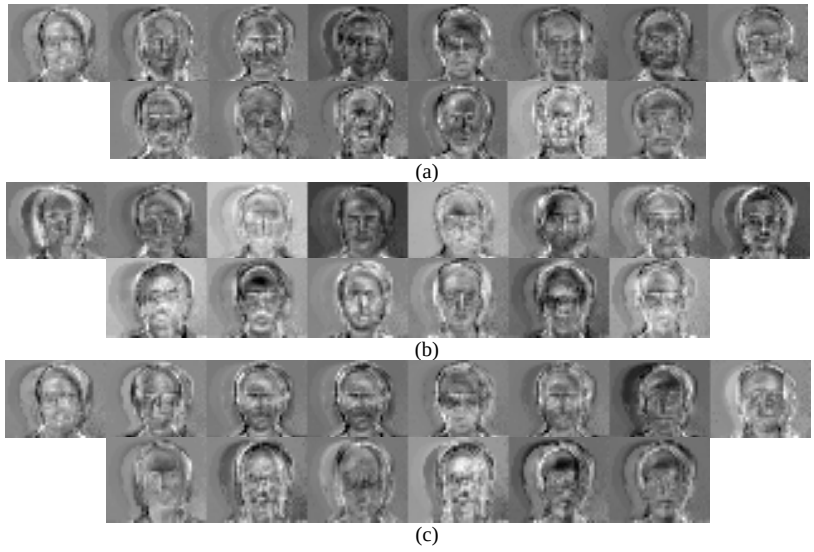
$$\varphi_k = \phi_k\,, \quad 2 \leq k \leq r \tag{5.46}$$

and:

$$\varphi_1 = \phi_1 \tag{5.47}$$

Thus, the first r optimal discrimination vectors of our algorithm are respectively equal to the first r normalized eigenvectors of $S_t^{-1}S_b$. According to Corollary 1 of Theorem

Figure 5.1. Illustrations of extracted 14 discrimination vectors



(a) Our algorithm; (b) Jin's approach; (c) fisherface method

5.5, we can use the first r normalized eigenvectors of $S_t^{-1}S_b$ to represent all the optimal discrimination vectors of our algorithm.

Corollary 5.2 shows that when the Fisher discrimination values satisfy the above condition and we use Equation 5.1 to represent the Fisher criterion, the popular fisherface method can obtain the same discrimination vectors as our algorithm. In other words, the discrimination vectors generated from the fisherface method can possess the statistical uncorrelation. However, while the Fisher discrimination values do not satisfy the above condition in Corollary 5.2, the discrimination vectors generated from the fisherface method cannot possess the statistical uncorrelation. This will influence the classification effect for its extracted discrimination features. Consequently, Theorem 5.5 reveals the essential relationship between UODV and the fisherface method. It also shows why UODV is theoretically superior to the latter.

The discrimination vectors extracted by our algorithm, Jin's approach and the fisherface method on the 2D image data are illustrated in Figure 5.1(a-c), respectively. Notice that the number of discrimination vectors is 14, which is equal to the rank of $S_t^{-1}S_b$.

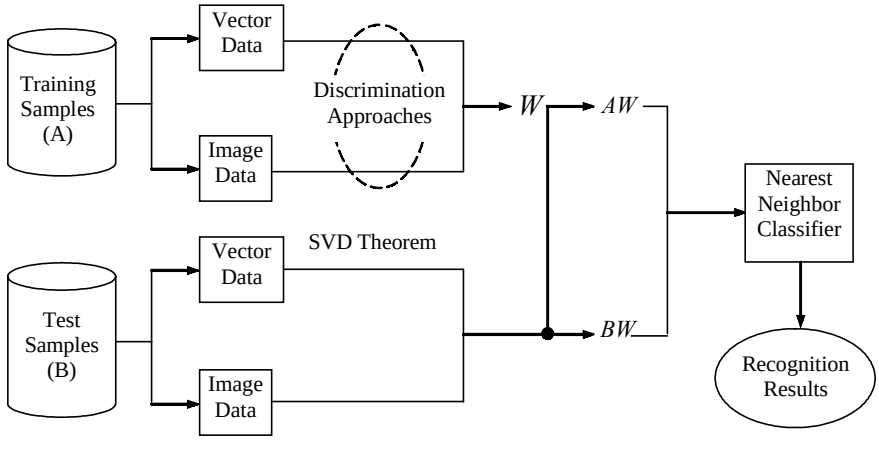
EXPERIMENTS AND ANALYSIS

To verify the effectiveness of our algorithm, we use both 1D and 2D data. The 1D data is taken from the Elena databases composed by the feature vectors (Woods, Kegelmeyer, & Bowyer, 1997). The 2D data is the Yale facial image database (Belhumeur, Hespanha, & Kriegman, 1997). We compare our algorithm with Jin's approach and the

fisherface method. At first, we provide the common implementation requirements for all of these algorithms. Then, the experimental results are given. Last, we present a synthetic evaluation of these results. The requirements are listed below:

1. When we calculate the principal components of the total scatter matrix, S_t, of the facial image databases, we should use the SVD theorem in algebraic theory (Jin, Yang, Hu, & Lou, 2001). This is necessary because of the high-dimension quality of the facial image. Please refer to Jin, Yang, Hu, and Lou (2001), which provided a detailed description of the related solution.
2. To guarantee the validity of the comparison, we use all of the principal components in TPCA when comparing the approaches. With respect to the fisherface method, we use Equation 5.15 to replace Equation 3.31. Thus, the small sample-size problem can be avoided. Moreover, for all the approaches, we only extract the discrimination vectors corresponding to the non-zero Fisher discrimination values.
3. To compute recognition rate, arbitrary M samples per class are taken as training samples and the rest are used for testing. Generally, the maximum value of M is about half of the sample number per class. The nearest-neighbor classifier is employed to perform the classification for the extracted discrimination features. To reduce the variation of the recognition result, every experiment for a discrimination approach is repeated 10 times and the mean value is regarded as the final recognition rate. The compared approaches are programmed in the MATLAB language. Figure 5.2 shows the flowchart of the recognition process for all the approaches. In the illustrations, the abbreviations "Ours," "Jin's" and "fisherface" represent our improved algorithm, Jin's approach and the fisherface method, respectively.

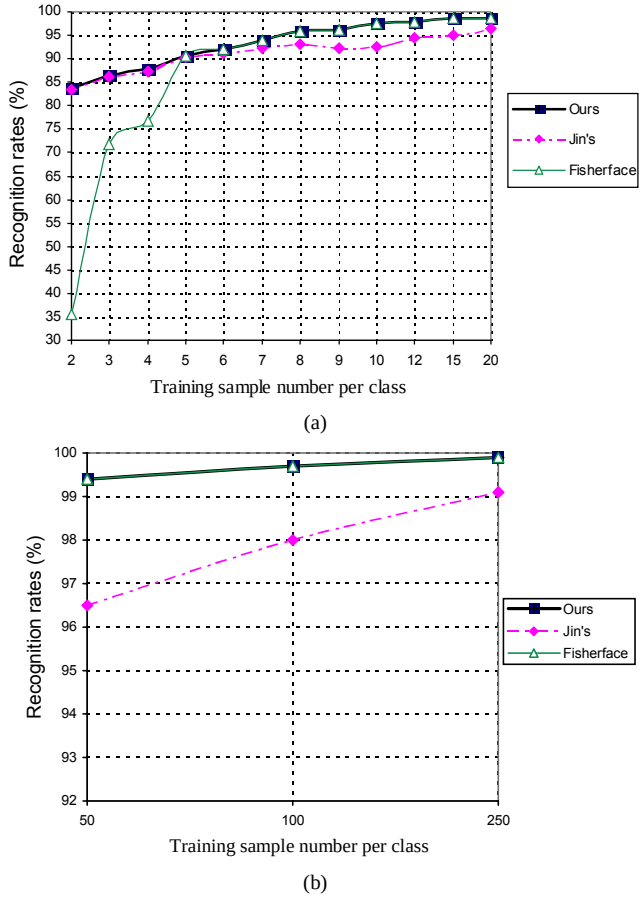
Figure 5.2. Flowchart of the recognition process for all of the compared approaches



Experiments on 1D Data

The well-known Elena databases are divided into two groups (Woods, Kegelmeyer, & Bowyer, 1997). The first group contains the ARTIFICIAL databases, for which the theoretical Bayes error can be computed. The second group contains the REAL databases, which have been collected from existing real-world applications. In the experiments, we use the second group, which includes four databases referred to as Texture, Satimage, Iris and Phoneme. Texture contains 5,500 samples, with 40 features, in 11 classes of 500 instances. Satimage contains 6,435 samples, with 36 features, in six classes with a different number of instances. Classical Iris contains 150 samples, with four features, in three classes of 50 instances. Phoneme contains 5,404 samples, with five features, in two classes with a different number of instances. Here, we select the Texture database because it has sufficient samples and more pattern categories.

Figure 5.3. A comparison of recognition rates of the Texture database for all the approaches



(a) Cases of fewer training samples; (b) cases of more training samples

Table 5.1. Non-zero Fisher discrimination values using the Texture database

		Number of training samples						
		2	3	4	5	10	50	250
Rank of S_t		21	32	37	37	37	37	37
Rank of S_w		11	22	33	37	37	37	37
All non-zero eigenvalues for $S_t^{-1}S_b$	1	1.0000	1.0000	1.0000	0.9996	0.9896	0.9830	0.9814
	2	1.0000	1.0000	1.0000	0.9976	0.9777	0.9680	0.9616
	3	1.0000	1.0000	1.0000	0.9913	0.9418	0.9144	0.9099
	4	1.0000	1.0000	1.0000	0.9856	0.9195	0.8763	0.8806
	5	1.0000	1.0000	0.9931	0.9627	0.8943	0.8477	0.8503
	6	1.0000	1.0000	0.9852	0.9513	0.8684	0.7661	0.7130
	7	1.0000	1.0000	0.9797	0.9332	0.8546	0.7177	0.7105
	8	1.0000	1.0000	0.9578	0.8677	0.7150	0.6228	0.6181
	9	1.0000	1.0000	0.9284	0.8510	0.6867	0.4748	0.4493
	10	1.0000	1.0000	0.8837	0.8119	0.6114	0.4407	0.4209

M samples in each class are taken as the training samples, where $2 \le M \le 250$. Figure 5.3 (a-b) show recognition rates for all the approaches in the cases of fewer training samples and more training samples, respectively. Our algorithm obtains the highest rates in almost all the cases, except the cases that our algorithm obtains the same results as the fisherface method when $2 \le M \le 4$.

Table 5.1 shows the non-zero Fisher discrimination values and the ranks of S_t and S_w in some examples, where the first M samples per class are selected to perform the training. When $M \ge 5$, all the non-zero eigenvalues of $S_t^{-1}S_b$ are mutually unequal. They are less than 1.0, and vary from 0.9996 to 0.4209. According to Theorem 5.5, the fisherface method should obtain the same linear discrimination transform as our algorithm. That is, the fisherface method should obtain the same recognition results as our algorithm in this situation. Figure 5.3 proves this conclusion. It shows that when $M \ge 5$, theclassification results are the same. Conversely, when $2 \le M \le 5$; that is, in the cases of very small training samples, most of the non-zero eigenvalues of $S_t^{-1}S_b$ are mutually equal to 1.0. Only when $M = 4$, from the 5th to the 10th, eigenvalues are less than 1.0, which vary from 0.9931 to 0.8837. According to theorem 5.5, the discrimination vectors of the fisherface method cannot possess the statistical uncorrelation. In other words, the fisherface method should not have the same recognition results as our algorithm in this situation, which is proven in Figure 5.3a. It shows that the recognition rates of the fisherface method is much worse than our algorithm, where $2 \le M \le 4$. Especially when $M = 2$, the rate of the fisherface method is 35.5%, and that of our algorithm is 83.6%. Besides, from Table 5.1, when $2 \le M \le 4$, the rank of S_w is $N - c$, where c is the number of classes, N is the total number of training samples, and $N=M*c=M*11$. And, when $2 \le M \le 3$, the rank of S_t is equal to $N - 1$. When $M \ge 5$, all the ranks of S_w and S_t are equal to 37. Note that the dimension of samples is equal to 40.

Experiments on 2D Data

An experiment of the fisherface method uses the Yale facial image database (Belhumeur, Hespanha, & Kriegman, 1997), which contains images with major variations,

Figure 5.4. A comparison of recognition rates of the Yale database for all approaches

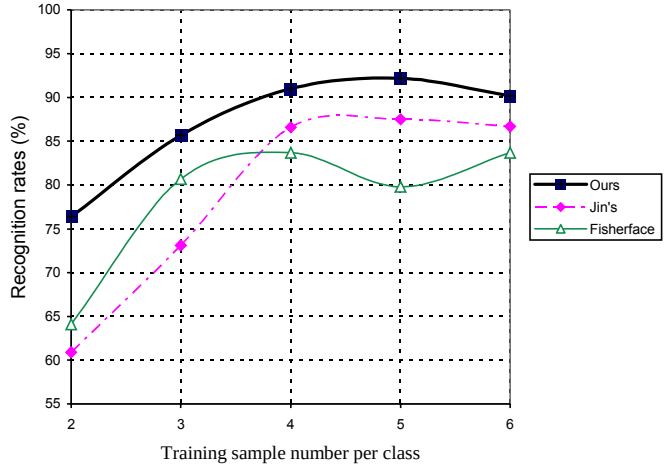


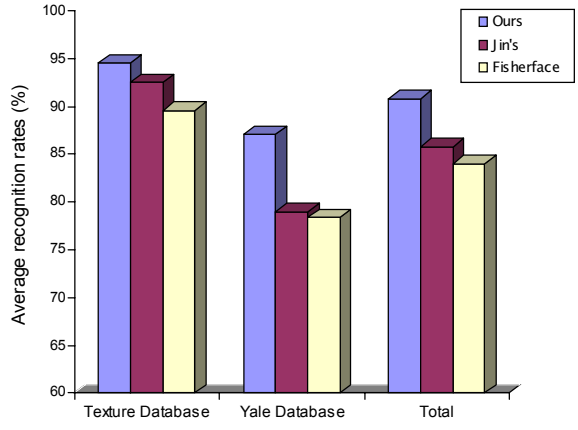
Table 5.2. Non-zero Fisher discrimination values using the Yale database

	Number of training samples				
	2	3	4	5	6
Rank of S_t	29	44	59	74	81
Rank of S_w	15	30	45	60	67
All of the nonzero eigenvalues for $S_t^{-1}S_b$	1.0000	1.0000	1.0000	1.0000	1.0000

such as changes in illumination, subjects wearing eyeglasses and different facial expressions. It involves 165 frontal facial images, with 15 individuals of 11 images. The size of each image is 243×320, with 256 gray levels per pixel. To reduce the computational cost and simultaneously guarantee sufficient resolution, we compress every image to an image size of 60×80. We use the full facial image. M samples in each class are taken as the training samples, where $2 \le M \le 6$. The highest recognition rate is achieved by our algorithm in all the cases, as shown in Figure 5.4.

Table 5.2 shows the non-zero Fisher discrimination values and the ranks of S_t and S_w in some examples, where the first M samples per class are selected to perform the training. All of those eigenvalues are equal to 1.0. According to Theorem 5.5, the fisherface method cannot obtain the same discrimination transform as our algorithm. This is proved by Figure 5.4, which shows that our algorithm performs much better than the fisherface method. Especially when $M = 2$, the recognition rate of the fisherface method is 79.8% and that of our algorithm is 92.2%. In addition, when $2 \le M \le 5$, the rank of S_w is equal to $N - c$ and the rank of S_t is equal to $N - 1$, where $N = M * c = M * 15$. When $M = 6$, we obtain that $N = 6 * 15 = 90$ and $N - c = 90 - 15 = 75$. However, in this case, the rank of S_w is 67, which is less than $N - c = 75$, and the rank of S_t is 81, which is also less

Figure 5.5. Average recognition rates of the two databases and the total average rates for all the compared approaches



than N - 1 = 89. Therefore, if we use the original form of the fisherface method — that is, use the first largest $N - c$ nonzero eigenvalues of S_t — we cannot ensure that S_w is nonsingular. This demonstrates the theoretical fact that the original form of the fisherface method cannot completely solve the small sample-size problem.

SUMMARY

For all the approaches, Figure 5.5 shows their average recognition rates and total average rates in above two databases, respectively. In the Texture database, the improvement of average recognition rates for our algorithm over Jin's approach is 2% (=94.5%-92.5%). The rates for our algorithm over the fisherface method is 4.9% (=94.5%-89.6%). In the Yale database, the improvement of average recognition rates for our algorithm over Jin's approach is 8.1% (=87.1%-79%). The rates for our algorithm over the fisherface method is 8.7% (=87.1%-78.4%). In the above two databases, the total improvement of average recognition rates for our algorithm over Jin's approach is 5% (=90.8%-85.8%). The rates for our algorithm over the fisherface method is 6.8% (=90.8%-84.0%). From all of the experimental results, our algorithm obtains best recognition results on both 1D and 2D data, regardless of the number of training samples.

REFERENCES

Belhumeur, P. N., Hespanha, J. P., & Kriegman, D. J. (1997). Eigenfaces vs. fisherface: Recognition using class specific linear projection. *IEEE Transactions on Pattern Analysis and Machine Intelligence, 19*(7), 711-720.

Duchene, J., & Leclercq, S. (1988). An optimal transformation for discriminant and principal component analysis. *IEEE Transactions on Pattern Analysis and Machine Intelligence, 10*(6), 978-983.

Foley, D. H., & Sammon, J. W. (1975). An optimal set of discrimination vectors. *IEEE Transactions on Computers, 24*(3), 281-289.

Fukunaga, K. (1990). *Introduction to statistical pattern recognition*. New York: Academic Press.

Jin, Z., Yang, J., Hu, Z., & Lou, Z. (2001). Face recognition based on the uncorrelated discrimination transformation. *Pattern Recognition, 34*(7), 1405-1416.

Jin, Z., Yang, J., Tang, Z., & Hu, Z. (2001). A theorem on the uncorrelated optimal discrimination vectors. *Pattern Recognition, 34*(10), 2041-2047.

Liu, K., Cheng, Y. Q., & Yang, J. Y. (1993). Algebraic feature extraction for image recognition based on an optimal discrimination criterion. *Pattern Recognition, 26*(6), 903-911.

Liu, K., Cheng, Y. Q., Yang, J. Y., & Liu, X. (1992). An efficient algorithm for Foley-Sammon optimal set of discrimination vectors by algebraic method. *International Journal of Pattern Recognition and Artificial Intelligencce, 6*(5), 817-829.

Okada, T., & Tomita, S. (1985). An optimal orthonormal system for discriminant analysis. *Pattern Recognition, 18*(2), 139-144.

Woods, K., Kegelmeyer, W. P., & Bowyer, K. (1997). Combination of multiple classifiers using local accuracy estimates. *IEEE Transactions on Pattern Analysis and Machine Intelligence, 19*(4), 405-410.

Yang, J., Yang, J., & Zhang, D. (2002). What's wrong with Fisher criterion?. *Pattern Recognition*, *35*(11), 2665-2668.

Chapter VI

Solutions of LDA for Small Sample Size Problems

ABSTRACT

This chapter shows the solutions of LDA for small sample-size (SSS) problems. We first give an overview on the existing LDA regularization techniques. Then, a unified framework for LDA and a combined LDA algorithm for SSS problem are described. Finally, we provide the experimental results and some conclusions.

INTRODUCTION

It is well known that Fisher LDA has been successfully applied in many practical problems in the area of pattern recognition. However, when LDA is used for solving SSS problems, like face identification, the difficulty that we always encounter is that the within-class scatter matrix is singular. This is due to the high-dimensional characteristic of a face image. For example, face images with a resolution of 100×100 will result in a 10,000-dimensional image vector space, within which the size of the within-class scatter matrix is as high as 10,000×10,000. In real-world problems, it is difficult or impractical to obtain enough samples to make the within-class scatter matrix nonsingular. In this singular case, the classical LDA algorithm becomes infeasible.

So, it is necessary to develop a feasible algorithm for LDA for the high-dimensional and SSS case. Generally, there are two popular strategies for LDA in such cases. One

strategy is transform-based; that is, before LDA is used for feature extraction, another procedure is first applied to reduce the dimension of the original feature space. The other strategy is algorithm-based; that is, to find an algorithm for LDA that can deal with the singular case directly.

The typical transform-based methods include fisherfaces (Belhumeur, Hespanha, & Kriegman, 1997), EFM (Liu & Wechsler, 2000, 2001), uncorrelated LDA (Jin, Yang, Hu, et al., 2001; Yang, Yang, & Jin, 2001) and so on. These methods can also be subdivided into two categories. In the first category, such as fisherfaces, EFM and the discriminant eigenfeatures technique (Swets & Weng, 1996), PCA is first used for dimensional reduction. Then, LDA is performed in the PCA-transformed space. Since the dimension of the PCA-transformed space is usually much lower than the original feature space, the within-class scatter matrix is certain to be nonsingular. So, the classical LDA algorithm becomes applicable. This type of approach is generally known as PCA plus LDA. In the second category of approaches, like uncorrelated LDA and the method adopted in Yang, Yang, and Jin (2001), another K-L transform technique is used instead of PCA for dimensional reduction. Although the methods mentioned above can avoid the difficulty of singularity successfully, they are approximate because some potential discriminatory information contained in some small principal components is lost in the PCA or K-L transform step. In addition, the theoretical foundation of the above methods is not clear yet, by far. For instance, why select PCA (or K-L transform) for dimensional reduction beforehand? Is any important discriminatory information lost in the PCA process because the criterion of PCA is not identical to that of LDA? These essential problems remain unsolved.

Some typical algorithm-based methods were developed by Hong and Yang (1991), Liu and Yang (1992), Guo, Huang, and Yang (1999), Guo, Shu, and Yang (2001), and Chen, Liao, and Ko (2000). Hong and Yang's method (1991) of avoiding singularity is to perturb the singular within-class scatter matrix into a nonsingular one. The methods of Liu and Wechsler (2001), Guo, Huang, and Yang (1999), and Guo, Shu, and Yang (2001), are based on a mapping technique that transforms the singular problem into a nonsingular one. Their idea is good and the developed theory provides a solid foundation for solving this difficult problem. It can be considered that Chen, Liao, and Ko's method is a special case of Guo, Huang and Yang's approach (Chen, Liao, & Ko, 2000; Guo, Huang, & Yang, 1999). Chen, Liao, and Ko (2000) merely emphasize the discriminatory information within the null space of the within-class scatter matrix and overlook the discriminatory information outside of it. Instead, Guo, Huang, and Yang (1999) and Guo, Shu, and Yang (2001) take those two aspects of discriminatory information into account at the same time. However, the methods mentioned above have a common disadvantage; that is, the algorithms have to run in a high-dimensional, original feature space. So, these methods are all very computationally expensive in the high-dimensional case. Differing from the above LDA methods, a novel direct LDA (DLDA) approach was proposed recently by Yu and Yang (2001). Although DLDA was claimed to be an exact algorithm of LDA in the singular case, in fact, a part of the important discriminatory information is still lost by this method, as demonstrated by the experiments in this chapter.

In this chapter, our focus is on an LDA algorithm for the high-dimensional and SSS case. We attempt to give a theoretically optimal, exact and more efficient LDA algorithm that can overcome the weaknesses of the previous methods. Towards achieving this

goal, a theoretical framework for LDA is first built. In this framework, two powerful mapping techniques — compression mapping and isomorphic mapping — are introduced. Compression mapping is used to project the high-dimensional, original feature space into a reduced-dimensional space. Subsequently, an isomorphic mapping is employed to transform the reduced-dimensional space into a Euclidean space of the same dimension. Finally, the optimal discriminant vectors of LDA only need to be determined in this low-dimensional Euclidean space. It can be proven that during the two stages of the mapping process, no discriminatory information is lost with respect to the Fisher criterion. More importantly, on the basis of the developed theory, we reveal the essence of LDA in the singular case; that is, all positive principal components of PCA are first used to reduce the dimension of the original feature space to *m* (the rank of the total scatter matrix). Next, LDA is performed in the transformed space. This strategy is called the *complete PCA plus LDA*. Note that the complete PCA plus LDA strategy is different from the traditional PCA plus LDA (Belhumeur, Hespanha, & Kriegman, 1997; Swets & Weng, 1996; Liu & Wechsler, 2000, 2001) in which c–1 (c is the number of classes) smallest principal components are thrown away during the PCA step.

Based on the complete PCA plus LDA strategy, an efficient combined LDA algorithm for the singular case is presented. The algorithm is capable of deriving all discriminatory information, including information within the null space of the within-class scatter matrix and information outside of it, which are both powerful and important for classification in SSS problems. What is more, this algorithm only needs to run in the low-dimensional, PCA transformed space rather than in the high-dimensional, original feature space (like methods of Guo, Huang, & Yang, 1999; Guo, Shu, & Yang, 2001; or Chen, Liao, & Ko, 2000).

The remainder of this chapter is organized as follows: Next, some fundamentals of LDA are given. Then, a theoretical framework for LDA in the singular and high-dimensional case is developed, and the essence of LDA in such a case is finally revealed. A combined LDA algorithm is proposed and we also compare the proposed combined LDA algorithm with the previous LDA algorithms in detail. Then, the combined LDA is tested on the ORL and NUST face databases. Finally, a summary is given.

OVERVIEW OF EXISTING LDA REGULARIZATION TECHNIQUES

Suppose that there are c known pattern classes and N training samples in total, and the original feature space is n-dimensional. The between-class scatter matrix , the within-class scatter matrix and the total scatter matrix , are defined as follows:

$$S_b = \sum_{i=1}^{c} P(\omega_i)(m_i - m_0)(m_i - m_0)^T \tag{6.1}$$

$$S_w = \sum_{i=1}^{c} P(\omega_i) E\left\{(X - m_i)(X - m_i)^T \mid \omega_i\right\} \tag{6.2}$$

$$S_t = S_b + S_w = E\left\{(X - m_0)(X - m_0)^T\right\} \tag{6.3}$$

where X denotes an n-dimensional sample, $P(\omega_i)$ is the prior probability of class i, $m_i = E\{X|\omega_i\}$ is the mean vector of the samples from class i, and $m_0 = E\{X\} = \sum_{i=1}^{c} P(\omega_i)m_i$ is the mean vector over all classes.

From Equation 6.1 to Equation 6.3, we know that S_w, S_b, and S_t are all semi-positive definite matrices. The classical Fisher criterion function can be defined as follows:

$$J(X) = \frac{X^T S_b X}{X^T S_w X} \tag{6.4}$$

where j is an n-dimensional non-zero column vector.

In the nonsingular case, the within-class scatter matrix S_w is positive definite. That means, for any non-zero vector X, we have $X^T S_w X > 0$. The vector j maximizing the function $J(\varphi)$ is called Fisher optimal projection direction. Its physical meaning is that the ratio of the between-class scatter against the within-class scatter is maximized after the projection of pattern samples onto j. In fact, j is selected as the generalized eigenvector of and S_w corresponding to the maximal eigenvalue. But, in many practical problems, a single projection axe is not enough; thus, a set of projection axes (also called discriminant vectors) are required. Generally, these discriminant vectors are selected as the eigenvectors $u_1, u_2, \ldots, u_d$ of and S_w corresponding to the d ($d \le c - 1$) largest generalized eigenvalues; that is, $S_b u_j = \lambda_j S_w u_j$, where $\lambda_1 \ge \lambda_2 \ge \ldots \lambda_d$. These are the projection axes of the classical LDA.

However, when S_w is singular, the problem of finding a set of optimal discriminant vectors becomes more complicated and difficult. In the following sections, we discuss this singularity problem of LDA in detail.

A UNIFIED FRAMEWORK FOR LDA

Theoretical Framework for LDA in Singular Case

Two Kinds of Discriminant Vectors and the Extended Fisher Criterion

In the singular case, S_w is a semi-positive definite matrix but it is not positive definite, so, for any X derived from the null space of S_w, we have $X^T S_w X = 0$. Its physical meaning is that after the projection of the pattern samples onto X, the within-class scatter is zero; that is, all projected samples within the same class are concentrated into the point of the class mean (which is just as we expected). If, at the same time, X satisfies $X^T S_b X > 0$ (i.e., the class mean point is separate after projection onto X), then the ratio of the between-class scatter against the within-class scatter is $J(X) = +\infty$. So, X must be an effective discriminant vector with respect to the Fisher criterion. Consequently, in the singular case, the projection vectors satisfying $X^T S_w X = 0$ and $X^T S_b X > 0$ contain very important

discriminatory information. Besides this, the projection vectors satisfying $X^T S_w X > 0$ and $X^T S_b X > 0$ also contain some useful discriminatory information. In other words, in the singular case, there exists two categories of effective discriminatory information; that is, information within the null space of S_w and information outside of it.

Now, the problem is how to select the two categories of projection axes that contain the two categories of discriminatory information. First of all, a suitable criterion is required. Naturally, for the second category of projection vectors, which satisfy $X^T S_w X > 0$ and $X^T S_b X > 0$, the classical Fisher criterion can still be used. However, for the first category of projection vectors, which satisfy $X^T S_w X = 0$ and $X^T S_b X > 0$, the classical Fisher criterion is not applicable, because the corresponding criterion value J(X) = + ∞. In this case, an alternative criterion, defined in Equation 6.5, is usually used to replace the classical Fisher criterion.

$$J_t(X) = \frac{X^T S_b X}{X^T S_t X} \tag{6.5}$$

However, for two arbitrary projection vectors, ξ_1 and ξ_2, in the first category, the equality $J_t(\xi_1) = J_t(\xi_2) = 1$ always holds. This means that $J_t(X)$ is unable to justify which one is better. So, although $J_t(X)$ is an extension of $J(X)$, it is not the best one. Fortunately, another criterion, shown in Equation 6.6 and suggested by Guo (1999) and Chen (2000), can overcome the drawback of the criterion in Equation 6.5.

$$J_b(X) = X^T S_b X \quad (||X||=1) \tag{6.6}$$

For the first category of projection vectors, since its corresponding within-class scatter is zero, it is reasonable to measure its discriminatory ability by its corresponding between-class scatter. As the within-class scatter is invariable, the between-class scatter is larger and the projected samples are more separable. So, in this case, it is reasonable to employ the criterion in Equation 6.6 to derive the first category of discriminant vectors.

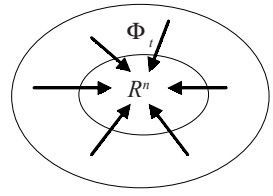
In conclusion, there exist two categories of discriminant vectors that contain discriminatory information in the singular case. The discriminant vectors of Category I satisfy $X^T S_w X = 0$ and $X^T S_b X > 0$, and those of Category II satisfy $X^T S_w X > 0$ and $X^T S_b X > 0$. The extended version of the Fisher criterion, which includes the between-class scatter criterion in Equation 6.6 and the classical Fisher criterion, is used to derive these two categories of discriminant vectors.

For convenience, the extended Fisher criterion is still called the Fisher criterion in the following sections.

Compression Mapping Principle

Now, the second problem is where to find the two categories of optimal discriminant vectors based on the Fisher criterion. Naturally, they can be found in R^n using the method adopted by Chen, Liao, and Ko (2000) and Guo, Huang, and Yang (1999). However, these approaches are too difficult and computationally expensive for high-dimensional problems, such as face recognition. Fortunately, we can prove that the two categories of discriminant vectors can be derived from a much lower-dimensional subspace of R^n, according to the following theory.

Figure 6.1. Illustration of the compression mapping from R^n to Φ_t



Let $\beta_1, \beta_2, \ldots, \beta_n$ be the n orthonormal eigenvectors of S_t. Intuitively, the original feature space R^n = **span** $\{\beta_1, \beta_2, \ldots, \beta_n\}$.

Definition 6.1. Define the subspace Φ_t = **span** $\{\beta_1, \beta_2, \ldots, \beta_m\}$, and its orthogonal complement can be denoted by $\Phi_t^{\perp}$ = **span** $\{\beta_{m+1}, \ldots, \beta_n\}$, where m = **rank** S_t, and $\beta_1, \ldots, \beta_m$ are the corresponding non-zero eigenvalues of S_t.

It is easy to verify that $\Phi_t^{\perp}$ is the null space of S_t using the following lemma.

Lemma 6.1 (Liu, 1992)**.** Suppose that A is a non-negative definite matrix and X is an n-dimensional vector, then $X^T A X = 0$ if and only if $AX = 0$.

Lemma 6.2. If S_t is singular, $X^T S_t X = 0$ if and only if $X^T S_w X = 0$ and $X^T S_b X = 0$.

Since S_w and S_b are non-negative definite and $S_t = S_b + S_w$, it is easy to get the above lemma.

Since R^n = **span** $\{\beta_1, \beta_2, \ldots, \beta_n\}$ for any arbitrary $\varphi \in R^n$, φ can be denoted by:

$$\varphi = \lambda_1\beta_1 + \cdots + \lambda_m\beta_m + \lambda_{m+1}\beta_{m+1} + \cdots + \lambda_n\beta_n$$

Let $X = \lambda_1\beta_1 + \ldots + \lambda_m\beta_m$ and $\xi = \lambda_{m+1}\beta_{m+1} + \ldots + \lambda_n\beta_n$, then, from the definition of Φ_t and $\Phi_t^{\perp}$, φ can be denoted by $\varphi = X + \xi$, where $X \in \Phi_t$, and $\xi \in \Phi_t^{\perp}$.

Definition 6.2. For any arbitrary $\varphi \in R^n$, φ is denoted by $\varphi = X + \xi$, where $X \in \Phi_t$, and $\xi \in \Phi_t^{\perp}$. A mapping L: $R^n \rightarrow \Phi_t$ is defined by:

$$\varphi = X + \xi \rightarrow X \tag{6.7}$$

It is easy to verify that L is a linear transformation from R^n to its subspace Φ_t. This mapping is named *the compression mapping*. The compression mapping from R^n to Φ_t is illustrated in Figure 6.1.

Theorem 6.1 (the Compression Mapping Principle). The compression mapping, $L: \varphi = X + \xi \rightarrow X$, satisfies the following properties with respect to the Fisher criterion:

$$J_b(\varphi) = J_b(X) \text{ and } J(\varphi) = J(X)$$

Proof: Since $\xi \in \Phi_t^{\perp}$, from the definition of $\Phi_t^{\perp}$, it follows that $\xi^T S_t \xi = 0$.

From Lemma 2, we have $\xi^T S_b \xi = 0$, which leads to $S_b \xi = 0$ using Lemma 1.

Hence:

$$\varphi^T S_b \varphi = \xi^T S_b \xi + 2X^T S_b \xi + X^T S_b X = X^T S_b X$$

Similarly:

$$\varphi^T S_w \varphi = X^T S_w X$$

So, $J_b(\varphi) = J_b(X)$ and $J(\varphi) = J(X)$.

According to Theorem 6.1, we can conclude that the two categories of discriminant vectors can be derived from Φ_t without any loss of effective discriminatory information with respect to the Fisher criterion.

Isomorphic Mapping Principle

From Definition 6.1, we know that $\dim \Phi_t = m$ (i.e., the rank of S_t). From linear algebra theory, Φ_t is isomorphic to m-dimensional Euclidean space R^m. The corresponding *isomorphic mapping* is:

$$X = PY, \text{ where } P = (\beta_1, \beta_2, \ldots, \beta_m) \ , Y \in R^m \tag{6.8}$$

which is a one-to-one mapping from R^m onto Φ_t.

From the isomorphic mapping $X = PY$, the criterion function $J(X)$ and $J_b(X)$, respectively, become:

$$J(X) = \frac{Y^T (P^T S_b P) Y}{Y^T (P^T S_w P) Y}$$

and $J_b(X) = Y^T (P^T S_b P) Y$.

Now, let us define the following functions:

$$\tilde{J}(Y) = \frac{Y^T \tilde{S}_b Y}{Y^T \tilde{S}_w Y} \tag{6.9}$$

$$\text{and } \tilde{J}_b(Y) = Y^T \tilde{S}_b Y \tag{6.10}$$

where $\tilde{S}_b = P^T S_b P$, and $\tilde{S}_w = P^T S_w P$.

It is easy to prove that both $\tilde{S}_b$ and $\tilde{S}_t$ are $m \times m$ semi-positive definite matrices. That means $\tilde{J}(Y)$ can act as a criterion like $J(X)$, and that $\tilde{J}_b(Y)$ can act as a criterion like $J_b(X)$.

It is easy to verify that the isomorphic mapping has the following property.

Theorem 6.2 (the isomorphic mapping principle). Suppose that Ω_1 *and* Ω_2 *are two* m*-dimensional vector spaces. If* $X = PY$ *is an isomorphic mapping from* Ω_1 *onto* Ω_2*, then* $X^* = PY^*$ *is the extremum point of* $J(X)$ $(or\ J_b(X))$ *if and only if* Y^* *is the extremum point of* $\tilde{J}(Y)$ $(or\ \tilde{J}_b(Y))$.

From the isomorphic mapping principle, it is easy to draw the following conclusions.

Corollary 6.1. If $Y_1, \ldots, Y_d$ are the optimal discriminant vector of Category I with respect to criterion $\tilde{J}_b(Y)$, then $X_1 = PY_1, \ldots, X_d = PY_d$ are the required optimal discriminant vectors of Category I with respect to criterion $J_b(X)$.

Corollary 6.2. If $Y_1, \ldots, Y_d$ are the optimal discriminant vector of Category II with respect to criterion $\tilde{J}(Y)$, then $X_1 = PY_1, \ldots, X_d = PY_d$ are the required optimal discriminant vectors of Category II with respect to criterion $J(X)$.

According to the isomorphic mapping principle and its two corollaries, the problem of finding the optimal discriminant vectors in subspace Φ_t is transformed into a similar problem in its isomorphic space R^m. Generally, $m = N - 1$, where N is the number of training samples. In high-dimensional and SSS problems, such as face recognition, since the number of training samples is always much less than the dimension of the image vector (i.e., $m << n$), the proposed idea of finding the optimal discriminant vectors is superior to many previous methods in terms of its computational complexity.

Essence of LDA in SSS Cases

The optimal discriminant vectors obtained can be used to form the following linear discriminant transform for feature extraction:

$$Z = W^T X \tag{6.11}$$

where:

$$W^T = (X_1, X_2, \ldots, X_d)^T = (PY_1, PY_2, \ldots, PY_d)^T = (Y_1, Y_2, \ldots, Y_d)^T P^T$$

The transformation in Equation 6.11 can be divided into two items:

$$Y = P^T X, \text{ where } P = (\beta_1, \beta_2, \ldots, \beta_m) \tag{6.12}$$

and:

$$Z = V^T Y, \text{ where V} = (Y_1, Y_2, \ldots, Y_d) \tag{6.13}$$

Since the column vectors of P are eigenvectors corresponding to the non-zero eigenvectors of S_t, the transformation in Equation 6.12 is exactly the PCA that transforms R^n into R^m. In the PCA-transformed space R^m, it is easy to obtain the total scatter matrix $\tilde{S}_t$ as:

$$\begin{aligned}\tilde{S}_t &= E\{(Y-\bar{Y})(Y-\bar{Y})^T\} \\ &= E\{P^T(X-\bar{X})(X-\bar{X})^T P\} \\ &= P^T E\{(X-\bar{X})(X-\bar{X})^T\}P \\ &= P^T S_t P\end{aligned}$$

Similarly, the within-class scatter matrix is $\tilde{S}_w = P^T S^w P$ and the between-class scatter matrix is $\tilde{S}_b = P^T S_b P$. Thus, the criteria $\tilde{J}(Y)$ and $\tilde{J}_b(Y)$ are exactly the extended Fisher criterion in the PCA-transformed space, and $Y_1, Y_2, \ldots, Y_d$ are the corresponding Fisher optimal discriminant vectors. Naturally, the transformation in Equation 6.13 is the linear discriminant transform in the PCA-transformed space.

Now, the essence of LDA in the singular case is revealed. PCA is first used to reduce the dimension of image space to m (i.e., the rank of the total scatter matrix). Next, LDA is performed in the transformed space. This strategy is called the *complete PCA plus LDA*. Note that our strategy is different from the *traditional PCA plus LDA* (Belhumeur, Hespanha, & Kriegman, 1997; Swets & Weng, 1996; Liu & Wechsler, 2000, 2001) in which c-1 (c is the number of classes) smallest principal components are thrown away during the PCA step.

A COMBINED LDA ALGORITHM FOR SSS PROBLEM

Since the fisherfaces and EFM methods are both based on the traditional PCA plus LDA strategy, they are imperfect because some potential and valuable discriminatory information may be lost during the PCA step. In this section, we propose a combined LDA algorithm capable of deriving all discriminatory information. This algorithm is based on the complete PCA plus LDA strategy; that is, in the PCA step we use all of the positive principal components and transform the image space into R^m, where m is the rank of S_t.

Strategy of Finding Two Categories of Optimal Discriminant Vectors

Now, the key problem is how to find the two categories of optimal discriminant vectors in the PCA transformed space R^m. First of all, let us consider where to derive them.

Let $\alpha_1, \ldots, \alpha_m$ be the orthonormal eigenvectors of $\tilde{S}_w$, and the first q eigenvectors are corresponding to the non-zero eigenvalues, where q = **rank** S_w.

Intuitively, R^m = span $\{\alpha_1, \ldots, \alpha_m\}$.

Definition 6.3. Define $\tilde{\Phi}_w = \text{span}\{\alpha_1, \ldots, \alpha_q\}$ and $\tilde{\Phi}_w^{\perp} = \text{span}\{\alpha_{q+1}, \ldots, \alpha_m\}$.

Obviously, $\tilde{\Phi}_w$ is a subspace of R^m, and $\tilde{\Phi}_w^{\perp}$, the null space matrix of $\tilde{S}_w$, is a complementary space of $\tilde{\Phi}_w$. Thus, the transformed space R^m can be divided into two subspaces: the null space of S_w and its orthogonal complement (i.e., $R^m = \tilde{\Phi}_w \oplus \tilde{\Phi}_w^{\perp}$).

From the definition of $\tilde{\Phi}_w^{\perp}$ and Lemma 6.1, it is easy to obtain the following:

Proposition 6.1. In space R^m, for an arbitrary vector $X \neq 0$, $X^T \tilde{S}_w X = 0$ if and only if $X \in \tilde{\Phi}_w^{\perp}$.

Proposition 6.2. For an arbitrary non-zero vector $X \in \tilde{\Phi}_w^{\perp}$, the inequality $X^T \tilde{S}_b X > 0$ always holds.

Proof: Since $\tilde{S}_t = P^T S_t P$ is a positive-definite matrix, for an arbitrary non-zero vector $X \in R^m$, we have $X^T \tilde{S}_w X > 0$.

For any arbitrary non-zero vector $X \in \tilde{\Phi}_w^{\perp}$, from Proposition 1 we have $X^T \tilde{S}_w X = 0$. Since $\tilde{S}_t = \tilde{S}_b + \tilde{S}_w$, therefore $X^T \tilde{S}_b X = X^T \tilde{S}_t X - X^T \tilde{S}_w X > 0$.

From Propositions 6.1 and 6.2, we can draw the conclusion that the first category of optimal discriminant vectors in R^m must be derived from the subspace $\tilde{\Phi}_w^{\perp}$. Conversely, the second category of optimal discriminant vectors in R^m can be derived from the subspace $\tilde{\Phi}_w$, which is the orthogonal complement of $\tilde{\Phi}_w^{\perp}$, since any arbitrary non-zero vector $X \in \tilde{\Phi}_w$ satisfies $X^T \tilde{S}_w X > 0$.

The idea of isomorphic mapping introduced in the earlier section can still be used to derive the first category of optimal discriminant vectors from $\tilde{\Phi}_w^{\perp}$ and the second category of optimal discriminant vectors from $\tilde{\Phi}_w$.

In the first step, we intend to derive the first category of optimal discriminant vectors, $\phi_1, \ldots, \phi_l$ ($l = \dim \tilde{\Phi}_w^{\perp}$), from the subspace $\tilde{\Phi}_w^{\perp}$, which maximize the criterion $\tilde{J}_b(Y)$ and are subject to the orthogonal constraints. That is, $\phi_1, \ldots, \phi_l$ are determined by the following model (Yang, Yang, & Jin, 2001):

$$\text{Model I (1)} \begin{cases} \{\phi_1, \cdots, \phi_l\} = \arg \max\limits_{Y \in \tilde{\Phi}_w^{\perp}} \tilde{J}_b(Y) \\ \phi_i^T \phi_j = 0, \quad i \neq j, \ i, j = 1, \cdots, l \end{cases} \tag{6.14}$$

To solve this model, we form the following isomorphic mapping:

$$Y = P_1 Z \tag{6.15}$$

where P1 = ($\alpha_{q+1}, \ldots, \alpha_m$), and $\alpha_{q+1}, \ldots, \alpha_m$ form the basis of $\tilde{\Phi}_w^{\perp}$.

The mapping transforms Model I (Belhumeur, Hespanha, & Kriegman, 1997) into:

$$\text{Model I (2)} \quad \begin{cases} \{u_1, \cdots, u_l\} = \arg\max\limits_{Z \in R^l} \bar{J}_b(Z) \\ u_i^T u_j = 0, \quad i \neq j, \ i, j = 1, \cdots, l \end{cases} \tag{6.16}$$

where $\bar{J}_b(Z) = Z^T \bar{S}_b Z$, $\bar{S}_b = P_1^T \tilde{S}_b P_1$, and R^l is the l-dimensional Euclidean space (l = **dim** $\tilde{\Phi}_w^{\perp}$), which is isomorphic to $\tilde{\Phi}_w^{\perp}$. It is easy to verify that $\bar{S}_b$ is a positive definite matrix in R^l.

Since the objective function $\bar{J}_b(Z)$ is equivalent to a Rayleigh quotient function, $\bar{J}_R(Z) = \dfrac{Z^T \bar{S}_b Z}{Z^T Z}$, and from its extremum property (Lancaster & Tismenetsky, 1985), it follows that the optimal solution, $u_1, \ldots, u_l$, of Model I (2) is exactly the orthonormal eigenvectors of $\bar{S}_b$. Correspondingly, from the *isomorphic mapping principle* (Theorem 2), the optimal discriminant vectors determined by Model I (1) are $\phi_j = P_1 u_j, j = 1, \ldots, l$.

In the second step, we try to obtain the second category of optimal discriminant vectors, $\varphi_1, \ldots, \varphi_k$, from $\tilde{\Phi}_w$. In fact, they can be determined by the following model (Yang & Yang, 2001):

$$\text{Model II (1)} \quad \begin{cases} \{\varphi_1, \cdots, \varphi_k\} = \arg\max\limits_{Y \in \tilde{\Phi}_w} J(Y) \\ \varphi_i^T \tilde{S}_t \varphi_j = 0, \quad i \neq j, \ i, j = 1, \cdots, k \end{cases} \tag{6.17}$$

In a similar way, we form the following isomorphic mapping:

$$Y = P_2 Z \tag{6.18}$$

where $P_2 = (\alpha_1, \ldots, \alpha_q)$, and $\alpha_1, \ldots, \alpha_q$ are the basis of $\tilde{\Phi}_w$.

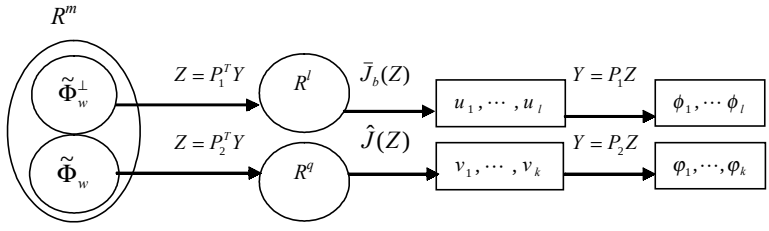
After the mapping, Model II (1) becomes:

$$\text{Model II (2)} \quad \begin{cases} \{v_1, \cdots, v_k\} = \arg\max\limits_{Y \in R^q} \hat{J}(Z) \\ v_i^T \hat{S}_t v_j = 0, \quad i \neq j, \ i, j = 1, \cdots, k \end{cases} \tag{6.19}$$

where $\hat{J}(Z) = \dfrac{Z^T \hat{S}_b Z}{Z^T \hat{S}_w Z}$, $\hat{S}_b = P_2^T \tilde{S}_b P_2$, $\hat{S}_w = P_2^T \tilde{S}_w P_2$, $\hat{S}_t = P_2^T \tilde{S}_t P_2$, and R^q is the q-dimensional Euclidean space (q = **dim** $\tilde{\Phi}_w$), which is isomorphic to $\tilde{\Phi}_w$.

It is easy to verify that $\hat{S}_b$ is semi-positive definite and $\hat{S}_w$ is positive definite (i.e., it must be nonsingular) in R^q. Thus, $\hat{J}(Z)$ is a generalized Rayleigh quotient, and from its extremum property (Lancaster & Tismenetsky, 1985), the optimal solutions, $v_1, \ldots, v_k$,

Figure 6.2. Illustration of the process of finding the two categories of optimal discriminant vectors



of Model II (2) can be selected as the $\hat{S}_t$-orthonormal eigenvectors associated with the first k largest positive eigenvalues of $\hat{S}_b Z = \lambda \hat{S}_w Z$ (Jin, Yang, Hu, et al., 2001). So, from the *isomorphic mapping principle*, $\varphi_j = P_2 v_j$ ($j = 1, \ldots, k$) are the optimal discriminant vectors derived from $\tilde{\Phi}_w$.

The above process of finding the two categories of optimal discriminant vectors is illustrated in Figure 6.2.

By the way, an interesting question is: How many optimal discriminant vectors are there in each category?

In fact, from the theory of linear algebra, it is easy to prove the following proposition.

Proposition 6.3. Rank $\tilde{S}_w$ = rank S_w and rank $\tilde{S}_b$ = rank S_b.

Generally, in SSS problems, rank $\tilde{S}_b = c - 1$ and rank $S_w = N - c$, where N is the total number of training samples and c is number of classes. So, in the m -dimensional ($m = N$ - 1) PCA-transformed space R^m, the dimension of the subspace $\tilde{\Phi}_w^{\perp}$ is l = dim $\tilde{\Phi}_w^{\perp}$ = m-rank $\tilde{S}_w = c - 1$. Since $\bar{S}_b = P_1^T S_b P_1$ is positive definite in $\tilde{\Phi}_w^{\perp}$'s isomorphic space R^l (i.e., rank $\bar{S}_b = l$), the total number of the first category of optimal discriminant vectors is $c - 1$.

In addition, we know that the second category of optimal discriminant vectors is determined by the eigenvectors of $\hat{S}_b Z = \lambda \hat{S}_w Z$ corresponding to the positive eigenvalues. The total number of these positive values is k = rank $\hat{S}_b$. Since $\hat{S}_b = P_2^T \tilde{S}_b P_2$ and rank $\hat{S}_b \le$ rank $\tilde{S}_b = c - 1$, therefore, the total number of optimal discriminant vectors in the second category is at most $c - 1$.

Properties of the Two Categories of Optimal Discriminant Vectors

The two categories of optimal discriminant vectors have some interesting properties.

First, the optimal discriminant vectors in Category I are orthogonal and $\tilde{S}_t$-orthogonal, and those of Category II are $\tilde{S}_t$-orthogonal. More specifically, the first category of optimal discriminant vectors, $\phi_1, \ldots, \phi_l$, satisfies:

$$\phi_i^T \phi_j = u_i^T (P_1^T P_1) u_j = u_i^T u_j = 0, \quad i \neq j,\ i, j = 1, \ldots, l \tag{6.20}$$

and $\phi_i^T \tilde{S}_t \phi_j = u_i^T (P_1^T \tilde{S}_t P_1) u_j = u_i^T (P_1^T \tilde{S}_w P_1) u_j + u_i^T (P_1^T \tilde{S}_b P_1) u_j$

$$= u_i^T (P_1^T \tilde{S}_b P_1) u_j = u_i^T \bar{S}_b u_j = 0, \quad i \neq j,\ i, j = 1, \ldots, l \tag{6.21}$$

The second category of optimal discriminant vectors satisfies:

$$\varphi_i^T \tilde{S}_t \varphi_j = v_i^T (P_2^T \tilde{S}_t P_2) v_j = v_i^T \hat{S}_t v_j = 0 \quad i \neq j,\ i, j = 1, \ldots, k \tag{6.22}$$

Equations 6.21 and 6.22 imply that each category of optimal discriminant vectors has the desirable property that after the projection of the pattern vector onto the discriminant vectors, the components of the transformed pattern vector are uncorrelated (Jin, Yang, Hu, et al., 2001; Jin, Yang, Tang, et al., 2001).

Second, from the *isomorphic mapping principle*, the orthogonal optimal discriminant vectors, $Y_1, \ldots, Y_l$ derived from $\tilde{\Phi}_w^{\perp}$, are extremum points of the criterion function $\tilde{J}_b(Y)$, while the $\tilde{S}_t$-orthogonal discriminant vectors, $\varphi_1, \ldots, \varphi_k$ derived from $\tilde{\Phi}_w$, are the extremum points of the criterion function $\tilde{J}(Y)$. In a word, all of the optimal discriminant vectors are extremum points of the corresponding criterion functions.

The two properties described above are also the reasons we chose the orthogonal constraints in Model I (1) and the $\tilde{S}_t$-orthogonal constraints in Model II (1).

Combined LDA Algorithm (CLDA)

The detailed algorithm is described as follows:

- **Step 1.** Perform PCA. Construct the total scatter matrix in the original sample space. Work out its m (m = **rank** S_t) orthonormal eigenvectors $b_1, \ldots, b_m$ corresponding to the positive eigenvalues using the technique suggested in Turk and Pentland (1991). Let $P = (b_1, b_2, \ldots, b\,m)$, $Y = P^T X$ transform the original sample space into an m-dimensional space.
- **Step 2.** In the PCA-transformed space R^m, work out all of the orthonormal eigenvectors $a_1, \ldots, a_m$ of the within-class scatter matrix $\tilde{S}_w$, and suppose that the first q eigenvectors are corresponding to positive eigenvalues.
- **Step 3.** Let $P_1 = (a_{q+1}, \ldots, a_m)$ and $\bar{S}_b = P_1^T \tilde{S}_b P_1$. Work out the orthonormal eigenvectors $u_1, \ldots, u_l$ of $\bar{S}_b$. Then, the optimal discriminant vectors of Category I are $f_j = P_1 u_j$, $j = 1, \ldots,$ l. Generally, $l = c - 1$, where c is the number of classes.

- **Step 4.** Let $P_2 = (a_1, \ldots, a_q)$, $\hat{S}_b = P_2^T \tilde{S}_b P_2$, and $\hat{S}_w = P_2^T \tilde{S}_w P_2$. Work out the k generalized eigenvectors $v_1, \ldots, v_k$ of $\hat{S}_b$ and $\hat{S}_w$ corresponding to the first k largest eigenvalues. Then, the optimal discriminant vectors of Category II are $j_j = P_2 v_j, j = 1, \ldots, k$. Generally, $k = \text{rank } \hat{S}_b \leq c - 1$.

The two categories of optimal discriminant vectors obtained are used for feature extraction. After the projection of the samples onto the first category of optimal discriminant vectors, we get the discriminant features of Category I. After the projection onto the second category of optimal discriminant vectors, we obtain the discriminant features of Category II.

Generally, these two categories of discriminant features are complementary to each other. So, in practice, to improve the recognition performance, we usually combine them. A simple and practical combination method is based on using all of the discriminant features of Category I and a few of most discriminatory features of Category II. More specifically, suppose $z_1^1, \ldots, z_l^1$ are the discriminant features of Category I, and $z_1^2, \ldots, z_k^2$ are discriminant features of Category II; then, we can use all features of Category I and the first t features of Category II to form the combined feature vector as $(z_1^1, \cdots, z_l^1, z_1^2, \cdots, z_t^2)^T$.

Specially, when there exists a unique training sample in each class, the within-class scatter matrix S_w is a zero matrix, which leads to a zero within-class scatter matrix $\tilde{S}_w$ in the PCA-transformed space R^m. That is, for any non-zero vector $X \in R^m$, $X^T \tilde{S}_w X = 0$ always holds. And, from Proposition 6.2, we have $X^T \tilde{S}_b X > 0$. Therefore, $\tilde{S}_b$ is positive definite in R^m. So, in this case, there is no second category of discriminant vectors, and the first category of optimal discriminant vectors is the orthonormal eigenvectors of $\tilde{S}_b$.

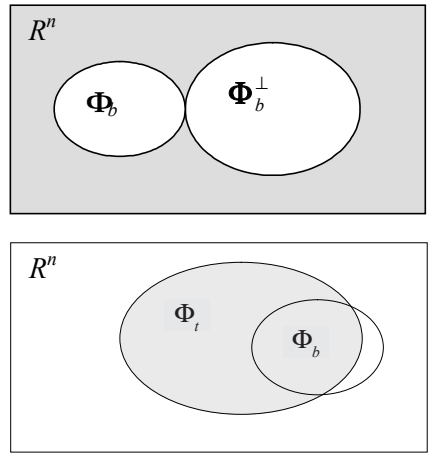
Comparison to Existing LDA Methods

Comparing with Traditional PCA+LDA Methods

In the traditional PCA plus LDA approaches, such as fisherfaces (Belhumeur, Hespanha, & Kriegman, 1997) and EFM (Liu & Wechsler, 1985, 2001), some small principal components are thrown away during the PCA step for the sake of ensuring that the between-class scatter matrix in the transformed space is nonsingular. However, these small principal components may contain very important discriminatory information with respect to the Fisher criterion. So, the traditional PCA plus LDA methods are not approximate. In addition, what is the theoretical basis for selecting the PCA for dimensional reduction? No author has answered this question.

In comparison, in the combined LDA methods, although the PCA is still used for dimensional reduction in the first step, we use all of the positive principal components rather than throw away some small items. More importantly, the procedure is not based on experience but on theoretical derivation. We have proven that there is no discriminatory information loss with respect to the Fisher criterion in this process.

Figure 6.3. (a) Illustration of the vector set $R^n - (\Phi_b \cup \Phi_b^{\perp})$, possibly containing discriminatory information; (b) Illustration of the relationship between Φ_t and Φ_b



Comparing with Direct LDA

The recently proposed DLDA (Yu & Yang, 2001) is claimed to be an exact algorithm of LDA in the singular case. However, it is not exact; the reason being that partial discriminatory information, whether of the first or the second category, is lost in DLDA. The total number of discriminant features derived by DLDA, in both categories, is at most $c - 1$, whereas using combined LDA (CLDA) we can obtain double the amount of discriminant features, and the number of features in each category can reach $c - 1$, in general. The experiments in the following section demonstrate that the two categories of discriminant features obtained by the CLDA are both very effective.

What discriminatory information is lost by the DLDA? Actually, the DLDA algorithm selects discriminatory information from the non-null space Φ_b (its definition is similar to that of Φ_t) of the between-class scatter matrix S_b. Although the null space of S_b, denoted by $\Phi_b^{\perp}$, contains no useful discriminatory information, important information may exist within R^n and outside of the subspaces Φ_b and $\Phi_b^{\perp}$. Figure 6.3a illustrates the vector set, $R^n - (\Phi_b \cup \Phi_b^{\perp})$, that may contain useful discriminatory information. In contrast, the CLDA algorithm selects discriminatory information from the non-null space Φ_t of the total scatter matrix, and it has been proven that Φ_t contains all discriminatory information with respect to the Fisher criterion. Figure 6.3b illustrates the relationship between the non-null spaces Φ_t and Φ_b. So, the discriminatory information within Φ_t and outside of Φ_b is thrown away by DLDA.

Essentially, DLDA is equivalent to the LDA algorithm suggested in Yang (2001a). That is to say, DLDA can also be divided into two steps. In the first step, the K-L transform is used, in which the between-class scatter matrix S_b (rather than the total scatter matrix) acts as a generation matrix to reduce the dimension of the original feature space to $c - 1$. In the second step, classical LDA is employed for feature extraction in the K-L transformed space.

Comparison to Other LDA Methods

Chen's method (Chen, Liao, & Ko, 2000) merely emphasizes the discriminatory information within the null space of the within-class scatter matrix and overlooks the discriminatory information outside of it. That is, Chen's method can only obtain the first category of discriminatory information and discards all discriminatory information that is in the second category. Although Guo (Guo, Huang, & Yang, 1999) and Liu (Liu & Yang, 1992) took these two categories of discriminatory information into account at the same time, their algorithms are too complicated. Besides this, in their methods, the discriminant vectors are subject to orthogonal constraints. In fact, the conjugate orthogonal constraints are more suitable with respect to the classical Fisher criterion (Jin, Yang, Hu, et al., 2001; Jin, Yang, Tang, et al., 2001).

What is more, all of the above methods suffer from the common disadvantage that the algorithms must run in the original feature space. In the high-dimensional case, the algorithms become too time consuming and almost infeasible. Conversely, the combined LDA only needs to run in the low-dimensional PCA-transformed space.

EXPERIMENTS AND ANALYSIS

Experiment Using the ORL Database

We perform experiments using the ORL database (www.cam-orl.co.uk), which contains a set of face images taken at the Olivetti Research Laboratory in Cambridge, United Kingdom. There are 40 distinct individuals in this database, and each individual has 10 views. There are variations in facial expression (open/closed eyes, smiling/non-smiling) and facial details (glasses/no glasses). All of the images were taken against a dark homogeneous background with the subjects in an upright, frontal position, with tolerance for some tilting and rotation of up to about 20 degrees. There are some variations in the scale of the image up to about 10%. The size of each image is 92×112 pixels. Ten images of one person, taken from the ORL database, are shown in Figure 6.4.

Figure 6.4. Ten images of one person in the ORL face database

Experiment One

The first experiment on the ORL database is designed to test the discriminatory ability of each category of discriminant features and their combination with the affect of varying the number of training samples per class.

For this goal, we use the first k (k varying from one to five) images of each person for training and the remaining samples are used for testing. In each case, we use the combined LDA algorithm to find both categories of optimal discriminant vectors, if they exist. More specifically, taking $k = 5$ as an example, the detailed calculating process is as follows. In this case, the total number of training samples is 200, and the rank of the total scatter matrix is 199. Work out its 199 orthonormal eigenvectors corresponding to positive eigenvalues, and exploit these eigenvectors to form the feature extractor and

Figure 6.5. Illustration of the two categories of discriminatory information of combined LDA when the number of training samples per class varies from 1 to 5

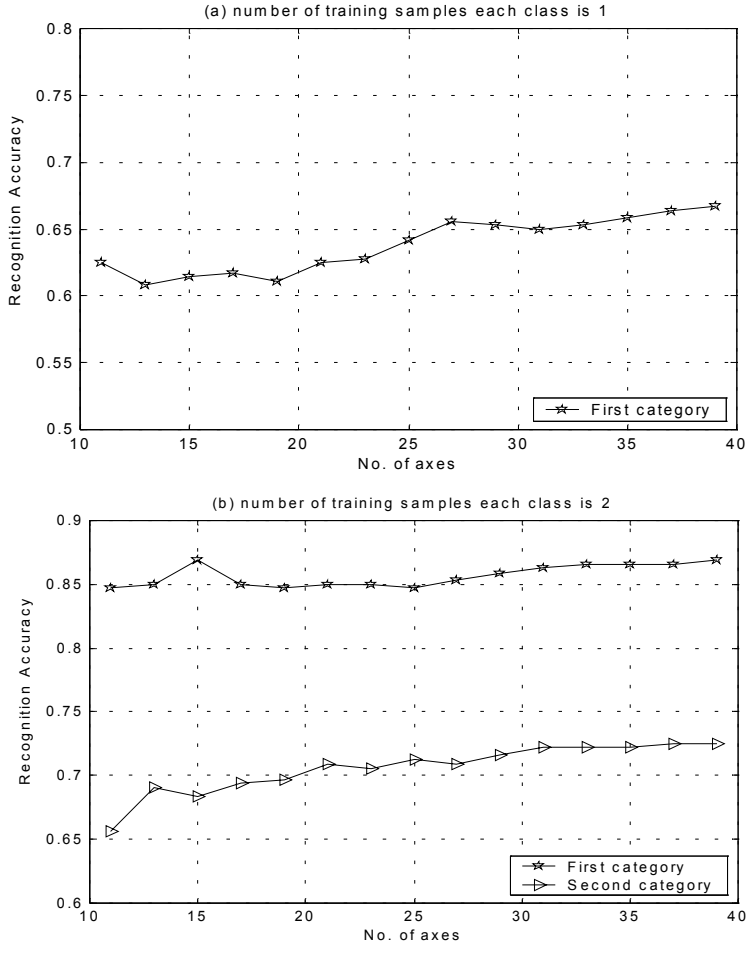
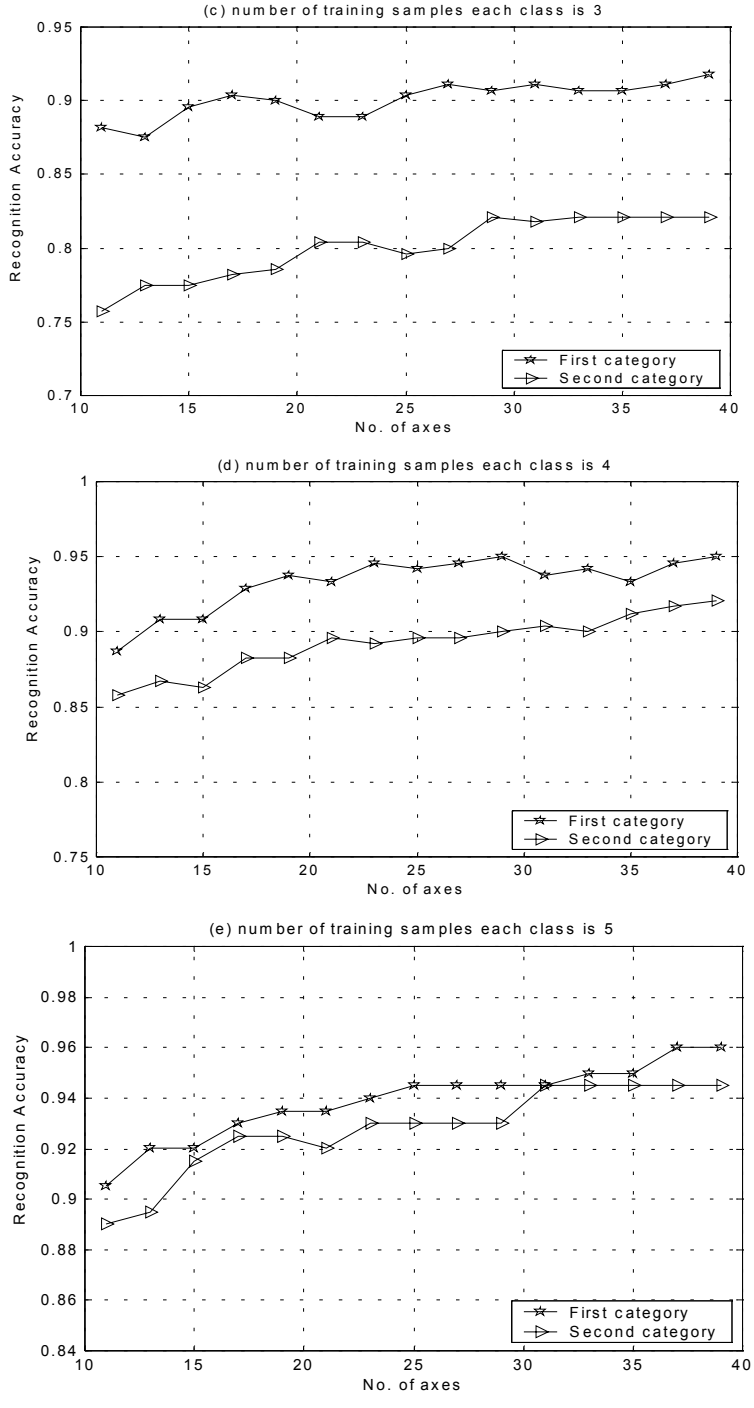


Figure 6.5. cont.



transform the original (92×112=10,304) 10,304-dimensional image vector into a 199-dimensional space (Y-space). Then, in the Y-space, since the rank of $\tilde{S}_w$ is 160, the dimension of the null space $\tilde{\Phi}_w^{\perp}$is 39. From Step 3 of the combined LDA algorithm, we derive the 39 optimal discriminant vectors of Category I. Similarly, we use Step 4 of the combined LDA algorithm to obtain the 39 optimal discriminant vectors of Category II. Note that, when the number of training samples is only one, it is sufficient to find the 39 optimal discriminant vectors of Category I, since there exist no discriminant vectors of Category II.

Finally, the two categories of optimal discriminant vectors obtained are used to transform the Y-space into a 39-dimensional discriminant space (Z-space). In this space, a common minimum distance classifier is adopted for classification. If $\| x - m_j \|_2 = \min_i \| x - m_i \|_2$, then, $x \in \omega_j$, where m_i is the mean vector of class i. The number of discriminant features selected varies from 1 to 39, and the corresponding recognition rates are illustrated in Figure 6.5.

Figure 6.5a shows that there exists no discriminatory information of Category II when there is only one training sample per class. As the number of training samples per class varies from one to two, comparing Figure 6.5b to 6.5a, the discriminatory information in Category I is significantly enhanced, and at the same time the discriminatory information in Category II has an effect. As the number of training samples per class increases beyond two, from Figure 6.5b to 6.5e, we see that the discriminatory information in Category II is significantly increased step by step. At the same time, although the discriminatory information in Category I is increasing as well, its rate of increase gradually slows down. When the number of training samples per class is five, the discriminatory information in Category II is almost as strong as that of Category I.

From Figure 6.5, we can draw some important conclusions. First, the two categories of discriminatory information are both important for classification in SSS problems. Second, when the number of training samples per class is very small (it seems, less than three), the discriminatory information in Category I is more important and plays a dominative role in recognition. As the number of training samples per class increases, the discriminatory information in Category II becomes more and more significant and should not be ignored.

However, the traditional PCA plus LDA algorithms (Belhumeur, Hespanha, & Kriegman, 1997; Swets & Weng, 1996; Liu & Wechsler, 2000, 2001) discard the first category of discriminatory information. Conversely, the null space in the LDA algorithm (Chen, Liao, & Ko, 2000) ignores the second category of discriminatory information.

Now, we turn our attention to the specific recognition accuracy. Table 6.1 shows the recognition rates based on the two categories of features and their combination with the number of training samples per class, varying from two to five. This table indicates that the two categories of discriminatory information are both effective. What is more, after they are combined, a more desirable recognition result is achieved. These results demonstrate that the two categories of discriminatory information are indeed complementary. More specifically, the first category of discriminatory information is not enough to achieve the maximal recognition accuracy. Since the null-space LDA algorithm (Chen, Liao, & Ko, 2000) only utilizes this first category of information, its performance, like the data shown in column of Table 6.1, is not the best. Similarly, the second category of

Table 6.1. Recognition rates based on the features of Category I, Category II and their combined features

Training Number	Category I 39 features	Category II 39 features	Combined features	
			Accuracy	Integrated form
2	86.9%	72.5%	87.8%	39(I)+ 7(II)
3	91.8%	82.1%	92.5%	39(I)+13(II)
4	95.0%	92.1%	95.4%	39(I)+16(II)
5	96.0%	94.5%	97.0%	39(I)+ 3(II)

Note: In the table, 39(I)+ 7(II) means using 39 features of Category I and the first 7 features of Category II

discriminatory information is not sufficient for recognition either, so the popular PCA plus LDA algorithms do not perform perfectly, as expected.

Experiment Two

The second experiment is designed to compare the performance of the proposed combined LDA with the direct LDA, which is claimed able to utilize the two categories of discriminatory information, as well.

The first k (k varying from one to five) images of each person are used for training and the remaining samples are used for testing. In each case, the direct LDA algorithm is used to derive the two categories of discriminant features. Based on each category of discriminant features, the recognition accuracy achieved, corresponding to the number of training samples, is illustrated in Figure 6.6. The specific recognition rates, corresponding to the two categories of features and their combination, are listed in Table 6.2.

Figure 6.6a shows that there exists no discriminatory information of Category II when the number of training samples in each class is only one. Comparing it with Figure 6.5a, we can see that the recognition accuracy of DLDA is much less that of the combined LDA. This is because the combined LDA algorithm can maximize the between-class scatter when the within-class scatter is zero, whereas DLDA cannot. Maximizing the

Table 6.2. Total number of each category of discriminant features of DLDA and the recognition rates based on the features of Category I, Category II and their combined features

Training Number	Category I		Category II		Combined features	
	Num	Accuracy	Num	Accuracy	Accuracy	Integrated form
2	11	67.5%	27	70.6%	84.7%	11(I)+ 26(II)
3	1	11.4%	38	86.1%	87.9%	1(I)+34(II)
4	0	0	39	90.0%	90.0%	38(II)
5	0	0	39	93.0%	93.0%	31(II)

Figure 6.6. Illustration of the two categories of discriminatory information of DLDA when the number of training samples per class varies from 1 to 5

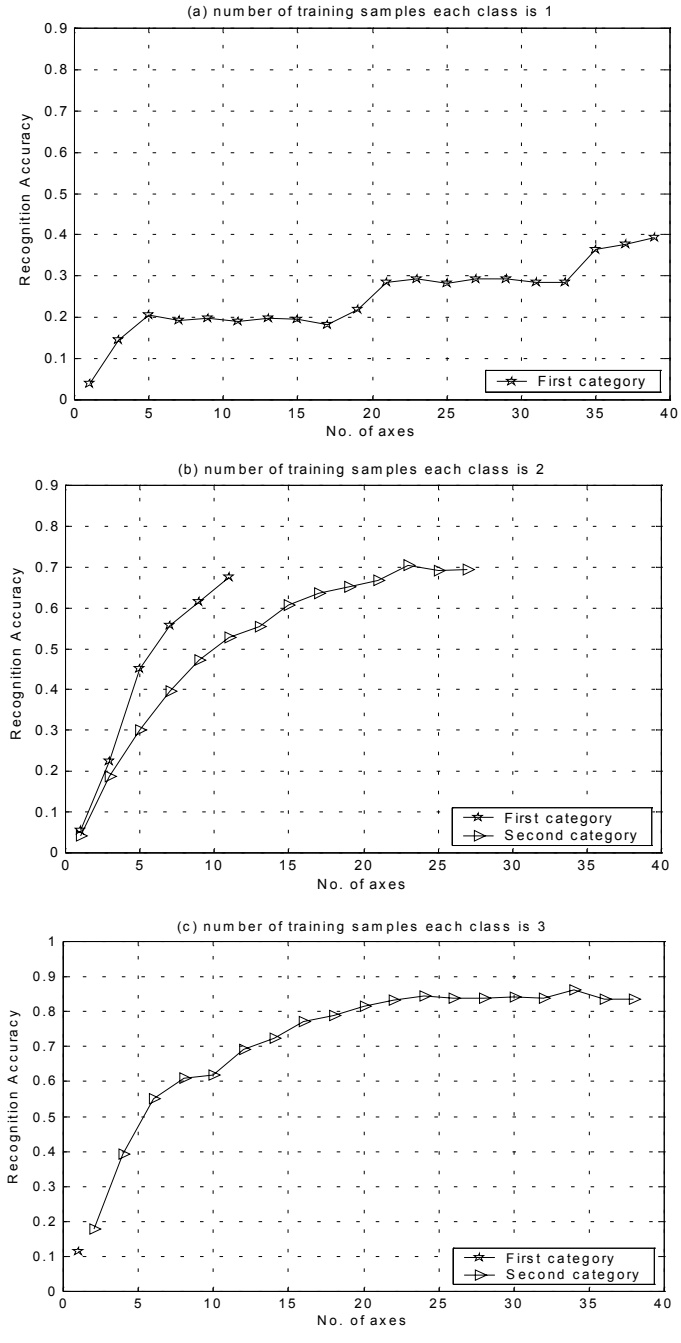
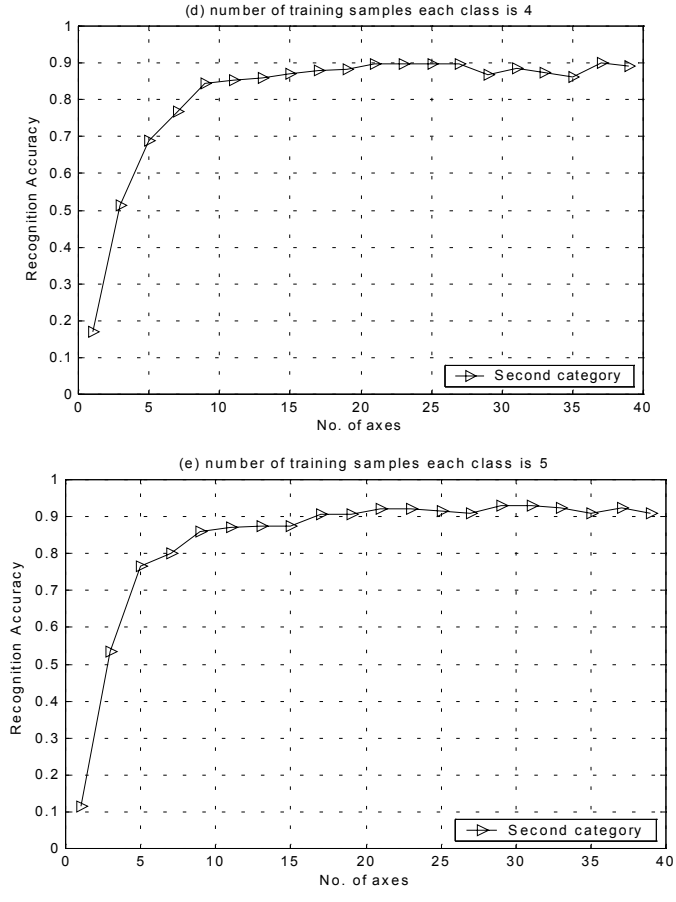


Figure 6.6. ***cont.***



between-class scatter significantly enhances the generalization capability of the classifier.

Comparing Figure 6.6 and Table 6.2 with Figure 6.5 and Table 6.1, it is easy to see that the two categories of discriminant information derived by DLDA are incomplete. When the number of training samples in each class is two, there exist 11 discriminant features in Category I and 28 discriminant features in Category II. When the number of training samples is three, there only exists one discriminant feature in Category I. When the number of training samples is more than three, there contains no features of Category I at all. Besides this, as far as the discriminatory power is concerned, it is obvious that DLDA is not as powerful as CLDA. The recognition accuracy of DLDA, whether based on the two categories of discriminant features or their combination, is much less that that of CLDA.

Experiment Three

The third experiment is aimed to compare the performance of the proposed CLDA with PCA-based method and the traditional LDA-based methods. Generally, before traditional LDA is used for feature extraction in the high-dimensional case, PCA is always applied for dimensional reduction. Specifically, PCA is first used to reduce the dimension of the original feature space to m, and then LDA is performed in the m-dimensional transformed space. This is the well-known PCA plus LDA strategy. Fisherfaces and EFM are both based on this strategy. Their differences are as follows: in fisherfaces, the number of principal components m is selected as N-c; whereas, in EFM, m is determined by the relative magnitude of the eigenvalues' spectra. Eigenfaces is the most well-known PCA-based method.

In this experiment, the first k (k varying from one to five) images of each person were used for training and the remaining samples were used for testing. In each case, we use the eigenfaces, fisherfaces, EFM and PCA plus LDA method (m is selected freely) for feature extraction. In the transformed space, a minimum distance classifier is employed. Recognition accuracy is listed in Table 6.3. When the number of training samples per class is five and the number of selected features varies from 5 to 45, the corresponding recognition accuracy of the above methods, using a minimum distance classifier and a nearest-neighbor classifier, is illustrated in Figure 6.7. Besides this, the corresponding CPU times consumed for the whole process of training and testing are listed in Table 6.4.

Table 6.3. Comparison of the performance of eigenfaces, fisherfaces, EFM, PCA plus LDA and CLDA with a minimum distance classifier

Training Number	Eigenfaces	Fisherfaces		PCA+LDA		EFM		Combined LDA
		Accuracy	*m*	Accuracy	*m*	Accuracy	*m*	
2	84.1%	82.5%	40	81.3%	35	85.0%	30	87.8%
3	84.6%	87.5%	80	88.6%	60	89.6%	45	92.5%
4	86.7%	88.7%	120	92.1%	80	92.5%	48	95.4%
5	89.5%	88.5%	160	94.0%	100	94.0%	50	97.0%

Table 6.4. Total CPU times (s) for the whole process of training and testing when the number of training samples per class is five

Classifier	Eigenfaces 45 features	PCA+LDA (m=50) 39 features	PCA+LDA (m=160) 39 features	Combined LDA 42 features
Minimum distance	373.69	375.20	379.32	383.82
Nearest neighbor	377.24	379.56	383.37	387.61

Figure 6.7. Comparison of the performance of eigenfaces, PCA+LDA methods and CLDA method under (a) the minimum distance classifier and (b) the nearest-neighbor classifier

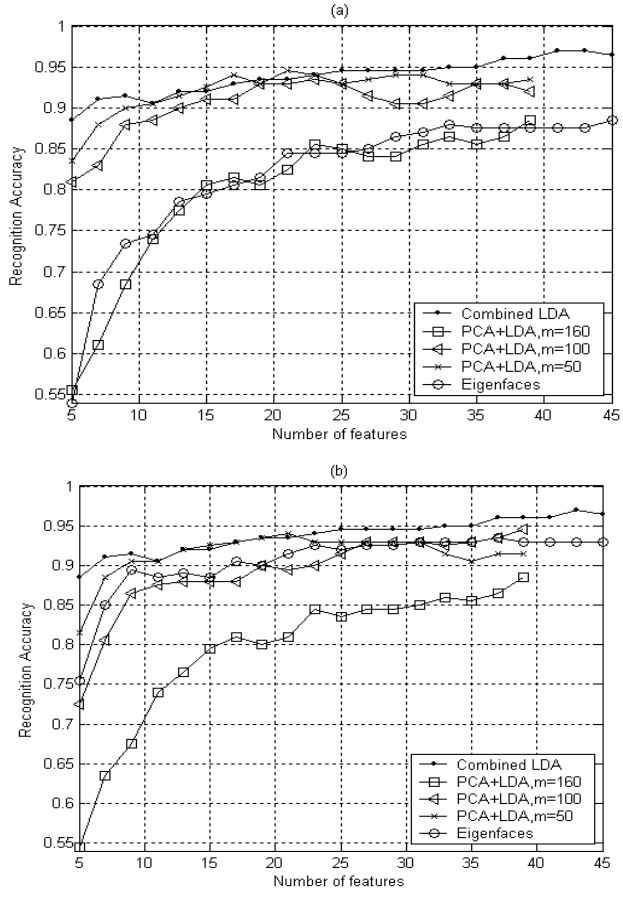


Table 6.3 shows that the recognition performance of CLDA is always the best. When the number of training samples is five, the recognition accuracy of CLDA reaches 97%, which is an increase of 3% compared to EFM, and the improvement is even greater compared to eigenfaces and fisherfaces methods.

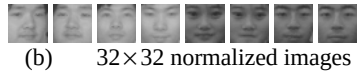
Figure 6.7a and 6.7b show that the performance of CLDA is very robust under two distinct classifiers. Although the recognition rate of EFM (i.e., PCA+LDA, m=50) is a little better than CLDA locally, it can be seen that CLDA outperforms EFM from a global perspective. Table 6.4 also indicates that CLDA is almost as fast as the other methods.

Experiment Using the NUST603 Database

The NUST603 database contains a set of face images taken at Nanjing University of Science and Technology. There are 10 images from each of the 96 distinct subjects.

Figure 6.8. Some examples of images in the NUST 603 database

(a) 128×128 original images



(b) 32×32 normalized images

All images were taken against moderately complex backgrounds and the lighting conditions were not controlled. The images of the subjects are in an upright, frontal position, with tolerance for some tilting and rotation. The images are all grayscale, with a resolution of 128×128 pixels. Some of the original images are shown in Figure 6.8a. In the experiment, we first crop the pure facial portion from the complex backgrounds using the location algorithm suggested in Jin (1999). Then, each image was modified to an image size of 32×32 resolution. Note that in the normalization process, we make no attempt to eliminate the influence of the lighting conditions. Some examples of the normalized images are shown in Figure 6.8b.

Similar to the process used in the experiment with the ORL database, the first k (k varying from one to five) images of each person are used for training and the remaining samples are used for testing. In each case, we first use the CLDA algorithm to obtain the two categories of optimal discriminant features, if they exist. In fact, there only exist 95 optimal discriminant features in Category I when the number of training samples is one. As the number of training samples is more than one, the total number of discriminant features in each category is 95. Based on these two categories of features and their combination, which includes all discriminant features of Category I and the first k (k varies from 1 to 25) features of Category II, using a common minimum distance classifier, the recognition accuracy is illustrated in Figure 6.9.

We then employ DLDA to obtain the two categories of features and their combined features, whose number is 95 in total. The corresponding recognition rates, using a minimum-distance classifier, are listed and compared with those of CLDA in Table 6.5. Next, we use the PCA plus LDA methods (where, in the PCA step, the number of principal components m is selected as 60, $c-1=95$ and 150) for feature extraction. The corresponding error rates, using a minimum-distance classifier, are shown in Table 6.6. For comparison, the performance of the uncorrelated LDA (Jin, Yang, Hu, et al., 2001) on the NUST database is listed in Table 6.6, as well.

The results in Figure 6.9 further demonstrate the conclusion that we drew on the ORL database. Once again, we see that the two categories of discriminatory information

Figure 6.9. Illustration of the recognition accuracy based on two categories of discriminatory features and their combination, while the number of training samples varies

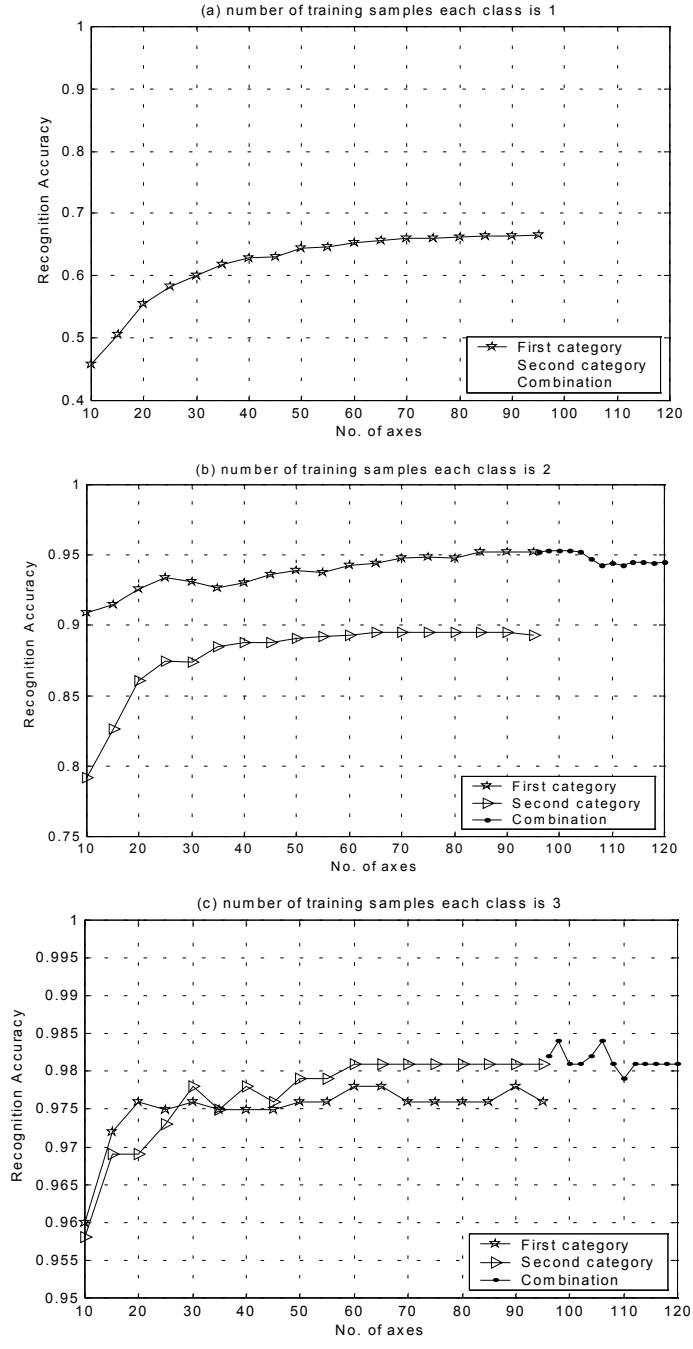
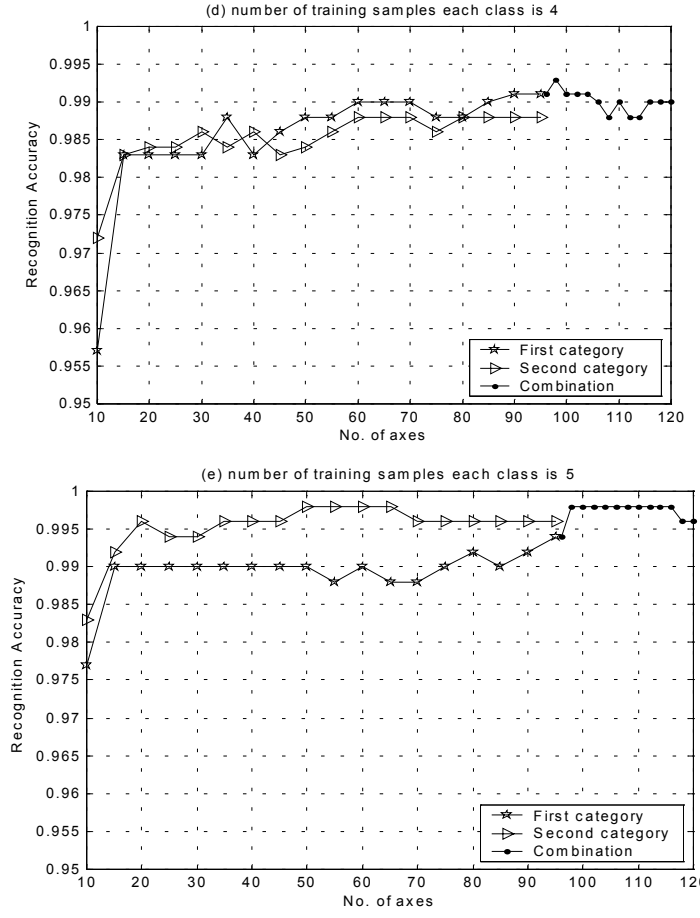


Figure 6.9. cont.



are both important for recognition in the SSS case. More specifically, the discriminatory information in Category I is more important and plays a dominant role when the number of training samples per class is small, and the discriminatory information in Category II becomes more and more significant as the number of training samples increases. What is more, in this experiment, we see that the discriminatory power of the features in Category II increase faster than that seen in the experiment using the ORL database. When the number of training samples is greater than two, the discriminatory power of the features in Category II is as powerful, or more powerful, than that shown by Category I. This may be due to the large amount of training samples that result from the increase in the number of classes. In this experiment, since the number of classes is 96, despite there being only three samples per class for training, the total number of training samples reaches 288. Whereas, in the ORL experiments, the total number of training samples is only 200, even when five samples per class are used for training.

Table 6.5. Comparison of the performance of DLDA and CLDA on the NUST database

Training Number	Category I		Category II		Combination	
	DLDA	CLDA	DLDA	CLDA	DLDA	CLDA
1	32.3% (95)	66.6%			32.3%	66.6%
2	81.3% (23)	95.2%	80.6% (72)	89.5%	89.8%	95.3%
3	95.5% (15)	97.8%	87.6% (80)	98.1%	97.9%	98.4%
4	96.0% (11)	99.1%	94.4% (84)	98.8%	99.0%	99.3%
5	16.7% (1)	99.4%	99.0% (94)	99.8%	99.6%	99.8%

Table 6.6. Comparison of the error rates of PCA plus LDA, uncorrelated LDA and CLDA on the NUST database

Training Number	PCA+LDA (m=60)	PCA+LDA (m=95)	PCA+LDA (m=150)	Uncorrelated LDA	Combined LDA
2	$\frac{60}{768}=7.8\%$	$\frac{79}{768}=10.3\%$		$\frac{84}{768}=10.9\%$	$\frac{36}{768}=4.7\%$
3	$\frac{17}{672}=2.5\%$	$\frac{11}{672}=1.6\%$	$\frac{15}{672}=2.2\%$	$\frac{13}{672}=1.9\%$	$\frac{11}{672}=1.6\%$
4	$\frac{8}{576}=1.4\%$	$\frac{7}{576}=1.2\%$	$\frac{7}{576}=1.2\%$	$\frac{7}{576}=1.2\%$	$\frac{4}{576}=0.7\%$
5	$\frac{3}{480}=0.6\%$	$\frac{2}{480}=0.4\%$	$\frac{3}{480}=0.6\%$	$\frac{2}{480}=0.4\%$	$\frac{1}{480}=0.2\%$

In addition, from Figure 6.9 we can also see the effectiveness of the combined features, which include all the features in Category I and the first t (t varies from 1 to 15) features in Category II. The recognition accuracy is significantly increased after combining the two categories of features, when the number of training samples per class is greater than two.

Table 6.5 shows that CLDA is more effective than DLDA, whether it is based on using each category of discriminant features or the combined features, especially when the number of training samples is less than three. Table 6.6 indicates that CLDA also outperforms PCA plus LDA methods and uncorrelated LDA. The recognition error of CLDA is only one, and recognition accuracy is up to 99.8% when the number of training samples per class is five.

Experimental Conclusions and Analysis

Based on the results from the two experiments using different databases, we can draw the following conclusions:

First, CLDA outperforms the traditional PCA plus LDA approaches, especially when the number of training samples is small. This is because when the number of training samples is very small, the first category of discriminatory information is more important and plays a dominant role in classification, whereas the traditional PCA plus LDA approaches discard this first category of discriminatory information. In the extreme case, when the number of training samples per class is only one, the second category of discriminatory information does not exist. Thus, the traditional PCA plus LDA algorithms cannot work at all. However, when the number of training samples becomes larger, it seems that the performance of PCA plus LDA is nearly as good as that of CLDA. This is due to the second category of discriminatory information that becomes more significant as the number of training samples increases. In another extreme case, when the number of training samples is large enough and the within-class scatter matrix becomes nonsingular, there only exists the second category of discriminatory information. In this case, the first category of discriminatory information disappears, and CLDA is equivalent to the classical LDA.

Second, CLDA is more effective than DLDA. This is due to the following reasons. CLDA is capable of deriving all discriminatory information in both Category I and Category II, whereas DLDA is only able to obtain a part of that information. What is more, the first category of discriminatory information derived by DLDA is not as strong as that derived by CLDA. This is because the CLDA algorithm can maximize the between-class scatter, but the DLDA cannot when the within-class scatter is zero. Maximizing the between-class scatter significantly enhances the generalization capability of the classifier. Especially when the number of training samples is very small, the performance of CLDA is much better than that of DLDA, since the first category of discriminatory information plays a dominant role in classification.

SUMMARY

Fisher LDA has been widely applied in many areas of pattern recognition, but in the high-dimensional and SSS case, three fundamental problems remain to be solved. The first problem is associated with the popular PCA plus LDA strategy. Namely, why can LDA be performed in the PCA-transformed space? Second is: What discriminatory information is optimal with respect to the Fisher criterion and most effective for classification? sm is still not clear. In this chapter, one of our contributions is to provide a theoretical foundation for the PCA plus LDA strategy. The essence of LDA in the singular case is revealed by theoretical derivation, that is, PCA plus LDA. So, PCA plus LDA is not only an effective means that is verified by practice, but also a reasonable strategy in theory (Yang & Yang, 2003). After all, the traditional PCA plus LDA approaches, like fisherfaces and EFM, are all approximate, since some small principal components are thrown away during the PCA step. Instead, in this chapter, we propose the *complete PCA plus LDA* strategy, which is an exact LDA approach.

Concerning the second question, we emphasize that there exist two categories of optimal discriminatory information for LDA in the singular case, provided that the number of training samples per class is greater than one. One category of information is within the null space of the within-class scatter matrix, while the other is within its orthogonal complementary space. Using our algorithm, we can find at most $c - 1$

discriminant features containing the first category of discriminatory information, and at most $c - 1$ discriminant features containing the second category of discriminatory information. That is, in the singular case, the total number of discriminant features in both categories can reach $2\times(c - 1)$. This characteristic of LDA is surprising and exciting. We know that, in the normal (nonsingular) case, there exist at most $c - 1$ discriminant features, whereas, in the singular case, the discriminatory information increases rather than decreases! And, the two categories of discriminatory information are both effective and important for recognition in SSS problems, as demonstrated by our experiments.

With regard to the third question, we give a very efficient algorithm, called *CLDA*, to find the two categories of discriminatory information. Differing from the previous exact LDA algorithms, CLDA is based on a two-stage PCA plus LDA strategy. It only needs to run in m-dimensional transformed space rather than in the high-dimensional original feature space. That is, the computational complexity of CLDA is the same as the traditional PCA plus LDA approaches, such as fisherfaces.

In this chapter, we try to combine the two categories of discriminatory information (features) in a simple way; that is, using all features of Category I and a few features of Category II to form the resulting feature vector. Although the experimental results demonstrate that this simple combination can improve the recognition accuracy to some degree, it is obvious that a majority of the discriminatory information in Category II is not exploited. So, the question of how to make optimal use of the two categories of discriminatory information is still an interesting problem that deserves further investigation in the future.

REFERENCES

Belhumeur, P. N., Hespanha, J. P., & Kriegman, D. J. (1997). Eigenfaces vs. fisherfaces: Recognition using class specific linear projection. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, *19*(7), 711-720.

Chen, L. F., Liao, H. Y., & Ko, M. T. (2000). A new LDA-based face recognition system which can solve the SSSproblem. *Pattern Recognition*, *33*(10), 1713-1726.

Guo, Y. F., Huang, X. W., & Yang, J. Y. (1999). A new algorithm for calculating Fisher optimal discriminant vectors and face recognition. *Chinese Journal of Image and Graphics*, *4*(2), 95-98 (in Chinese).

Guo, Y. F., Shu, T. T., & Yang, J. Y. (2001). Feature extraction method based on the generalized Fisher discriminant criterion and face recognition. *Pattern Analysis & Application*, *4*(1), 61-66.

Hong, Z-Q., & Yang, J. Y. (1991). Optimal discriminant plane for a small number of samples and design method of classifier on the plane. *Pattern Recognition*, *24*(4), 317-324.

Jin, Z. (1999, June). *Research on feature extraction of face images and feature dimensionality.* Ph.D. dissertation. Nanjing University of Science and Technology.

Jin, Z., Yang, J. Y., Hu, Z., & Lou, Z. (2001). Face recognition based on uncorrelated discriminant transformation. *Pattern Recognition*, *33*(7), 1405-1467.

Jin, Z.,Yang, J. Y., Tang, Z., & Hu, Z. (2001). A theorem on uncorrelated optimal discriminant vectors. *Pattern Recognition*, *33*(10), 2041-2047.

Lancaster, P., & Tismenetsky, M. (1985). *The theory of matrices* (2nd ed.). Orlando, FL: Academic Press.

Liu, C-J., & Wechsler, H. (2000). Robust coding schemes for indexing and retrieval from large face databases. *IEEE Transactions on Image Processing*, *9*(1), 132-137.

Liu, C-J., & Wechsler, H. (2001). A shape- and texture-based enhanced Fisher classifier for face recognition. *IEEE Transactions on Image Processing*, *10*(4), 598-608.

Liu, K., & Yang, J. Y. (1992). An efficient algorithm for Foley-Sammon optimal set of discriminant vectors by algebraic method. *International Journal of Pattern Recognition and Artificial Intelligence*, *6*(5), 817-829.

Swets, D. L., & Weng, J. (1996). Using discriminant eigenfeatures for image retrieval. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, *18*(8), 831-836.

Turk, M., & Pentland, A. (1991). Eigenfaces for recognition. *Journal of Cognitive Neuroscience*, *3*(1), 71-86.

Yang, J., & Yang, J. Y. (2001). Optimal FLD algorithm for facial feature extraction. In *SPIE Proceedings of Intelligent Robots and Computer Vision XX: Algorithms, Techniques, and Active Vision*, 4572 (pp. 438-444).

Yang, J., & Yang, J. Y. (2003). Why can LDA be performed in PCA transformed space? *Pattern Recognition*, *36*(2), 563-566.

Yang, J., Yang, J. Y., & Jin, Z. (2001). A feature extraction approach using optimal discriminant transform and image recognition. *Journal of Computer Research and Development*, *38*(11), 1331-1336 (in Chinese).

Yu, H., & Yang, J. (2001). A DLDA algorithm for high-dimensional data – with application to face recognition. *Pattern Recognition*, *34*(10), 2067-2070.

Chapter VII

An Improved LDA Approach

ABSTRACT

This chapter gives an improved LDA (ILDA) approach. After a short review and comparison of major linear discrimination methods, including the eigenface method, fisherface method, DLDA and UODV, we introduce definitions and notations. Then, the approach description of ILDA is presented. Next, we show some experimental results. Finally, we summarize some useful conclusions.

INTRODUCTION

In this section, we first give a brief review of some important linear discrimination methods that we have mentioned in earlier chapters. In the field of pattern recognition, and especially in image recognition, image data are always high dimensional and require considerable computing time for classification. The LDA technique we showed in Chapter III is thus important in extracting effective discriminative features and reducing dimensionality, and costs little computing time. It has been shown in many applications of image recognition that LDA can satisfy these requirements (Swets & Weng, 1996; Loog, Duin, & Haeb-Umbach, 2001; Vailaya, Zhang, Yang, Liu, & Jain, 2002; Nishino, Sato, & Ikeuchi, 2001). So far, many linear discrimination methods have been proposed for use in image recognition. Two of the most well-known are the eigenface and fisherface methods.

Based on PCA (see Chapter II) (JainDuin & Mao, 2000), the eigenface method (see Chapter IV) (Turk & Pentland, 1991) uses the total covariance (or scatter) matrix S_t, as the production matrix to perform the KL transform. It cannot, however, make full use of pattern separability information like the Fisher criterion, and its recognition effect is not ideal when the size of the sample set is large (Martinez & Kak, 2001; Belhumeur, Hespanha, & Kriegman, 1997).

The famous fisherface method (see Chapter IV) (Belhumeur, Hespanha, & Kriegman, 1997) combines PCA and the Fisher criterion (Fisher, 1936) to extract the information that discriminates between the classes of a sample set. It is a most representative method of LDA. Nevertheless, Martinez and Kak (2001) demonstrated that when the training data set is small, the eigenface method outperforms the fisherface method. Should the latter be outperformed by the former? This provoked a variety of explanations. Liu and Wechsler (2000) thought that it might have been because the fisherface method uses all the principal components, but the components with the small eigenvalues correspond to high-frequency components and usually encode noise, leading to recognition results that are less than ideal. In line with this theory, they presented two EFMs (Liu & Wechsler, 2000) and an enhanced Fisher classifier (Liu & Wechsler, 2000) for face recognition. Their experiential explanation lacks sufficient theoretical demonstration; however, an EFM does not provide an automatic strategy for selecting the components.

Chen, Liao, Ko, Lin, and Yu (2000) proved that the null space of the within-class scatter matrix S_w contains the most discriminative information when a small sample size problem takes place. Their method is also inadequate, however, as it does not use any information outside the null space. Yu and Yang (2001) propose a DLDA approach to solve this problem. It simultaneously diagonalizes both the between-class scatter matrix S_b (or S_t) and S_w. Let $W^T S_w W = D_w$, and let $W^T S_b W = I$ or $W^T S_t W = I$. According to the theory, DLDA should discard some of the eigenvectors of D_w that correspond to the higher eigenvalues, and keep the remainders, especially those eigenvectors that correspond to the zero eigenvalues. This approach, however, has a number of limitations. First, it does not demonstrate how to select its eigenvectors. Second, the related demonstration is rather difficult. Third, in the application of DLDA, there is a contradiction between the theory and the experiment. The theory requires that the eigenvectors of D_w corresponding to the higher eigenvalues be discarded, but the experiment obtains the improved recognition results by employing all of the eigenvectors of D_w.

ODV (see Chapter V) is a special kind of LDA method that has been applied to a wide range of applications in pattern classification (Cheng, Zhuang, & Yang, 1992; Liu, Cheng, Yang, & Liu, 1992; Liu, Cheng, & Yang, 1993). It requires that every discrimination vector satisfy the Fisher criterion and the obtained Fisher discrimination vectors are necessary to satisfy the orthogonality constraint (Foley & Sammon, 1975); but as a result, its solution is more complicated than other LDA methods. Jin, Yang, Hu, and Lou (2001) proposed a UODV method (see Chapter V) that used the constraint of statistical uncorrelation. UODV produces better results than ODV on the same handwritten data, where the only difference lies in their respective constraints (Jin, Yang, Tang, & Hu, 2001). Jing, Zhang, and Jin (2003a, 2003b) subsequently presented a more rational UODV method and generalized theorem for UODV.

Many others methods have been proposed. Zhang, Peng, Zhou, and Pal (2002) presented a face recognition system based on hybrid neural and dual eigenfaces

methods. Jing et al. put forward a classifier combination method for face recognition (Jing, Zhang, & Yang, 2003). In Malina (2001) and Cooke (2002), several new discrimination principles based on the Fisher criterion were proposed. Yang used KPCA for facial feature extraction and recognition (Yang, 2002), while Bartlett, Movellan, and Sejnowski (2002) applied ICA in face recognition. However, Yang showed that both ICA and KPCA need much more computing time than PCA. In addition, when the Euclidean distance is used, there is no significant difference in the classification performance of PCA and ICA (Bartlett, Movellan, & Sejnowski, 2002). Yang and Yang (2002) presented an IMGPCA method for face recognition, which is a variant form of PCA. In this chapter, we do not analyze and compare these extended discrimination methods (Jing, Zhang, & Yang, 2003; Zhang, Peng, Zhou, & Pal, 2002; Malina, 2001; Cooke, 2002; Yang, 2002; Bartlett, Movellan, & Sejnowski, 2002; Yang & Yang, 2002), because they do not use the original Fisher criterion or the basic form of the PCA transform. And we confine ourselves to a comparison of major linear discrimination methods, including the eigenface method, fisherface method, DLDA and UODV.

The linear discrimination technique should be improved in three ways:

1. Discrimination vectors should be selected. Not all discrimination vectors are useful in pattern classification. Thus, vectors with the larger Fisher discrimination values should be chosen, since they possess more between-class than within-class scatter information.
2. Discrimination vectors should be made to satisfy the statistical uncorrelation, a favorable classification property. Although UODV satisfies this requirement, it also uses more computing time than the fisherface method, since it respectively calculates every discrimination vector satisfying the constraint of uncorrelation. Our improvement should provide a measure that satisfies the requirement while saving a maximum of computing time. Therefore, this improvement will take advantage of both the fisherface method and UODV. In other words, it is theoretically superior to UODV presented in Jing, Zhang and Jin (2003a, 2003b).
3. An automatic strategy for selecting principal components should be established. This would effectively improve classification performance and further reduce feature dimension. Jing, Zhang, and Yao (2003) presented an elementary method for selecting the components. In this chapter, we will perform a deep theoretical analysis and then provide a more logical selecting strategy.

We will now propose an ILDA approach that synthesizes the foregoing suggestions (Jing, Zhang, & Tang, 2004).

DEFINITIONS AND NOTATIONS

In this section, we first briefly review two representative forms of the fisherface method. Generally, the image data is a 2D matrix ($A \times B$), which can be transformed into a vector with H dimension, where $H = A \times B$. Thus, we can obtain an H-dimensional sample set X from the image database. Assuming there are c known pattern classes and N training samples in X, the original form of the fisherface method is to maximize the following function (Belhumeur, Hespanha, & Kriegman, 1997):

$$F(W_{opt}) = \frac{\left| W_{fld}^{T} W_{pca}^{T} S_b W_{pca} W_{fld} \right|}{\left| W_{fld}^{T} W_{pca}^{T} S_w W_{pca} W_{fld} \right|}, \quad W_{opt} = W_{pca} W_{fld} \tag{7.1}$$

To avoid the complication of a singular S_w, the fisherface method discards the smallest c principal components. This is because the rank of S_w is at most $N - c$ (Belhumeur, Hespanha, & Kriegman, 1997). Nevertheless, when the rank of S_w is less than $N - c$, this method is incapable of completely ensuring that S_w is nonsingular in theory (Cheng, Zhuang, & Yang, 1992). In other words, it cannot completely overcome the SSS problem (Chen, Liao, Ko, Lin, & Yu, 2000). Here, an equivalent form of the fisherface method is used:

$$F(W_{opt}) = \frac{\left| W_{fld}^{T} W_{pca}^{T} S_b W_{pca} W_{fld} \right|}{\left| W_{fld}^{T} W_{pca}^{T} S_t W_{pca} W_{fld} \right|}, \quad W_{opt} = W_{pca} W_{fld} \tag{7.2}$$

In Fukunaga (1990) and Liu, Cheng, Yang, and Liu (1992), the equivalence of Equations 7.1 and 7.2 has been proven. When S_w is non-singular, the same linear discrimination transform can be obtained from these two equations. However, when S_w is singular (the SSS problem arises), Equation 7.2 can perform the linear discrimination transform, whereas Equation 7.1 cannot. Consequently, Equation 7.2 is also a complete solution of the SSS problem. Note that the following proposed improvements and ILDA approach are based on Equation 7.2, and that when we compare the classification performance of different methods, we still use Equation 7.1 to represent the original fisherface method.

APPROACH DESCRIPTION

We present three improvements in LDA: improvements in the selection of discrimination vectors, in their statistical uncorrelation and in the selection of principal components.

Improving the Selection of Discrimination Vectors

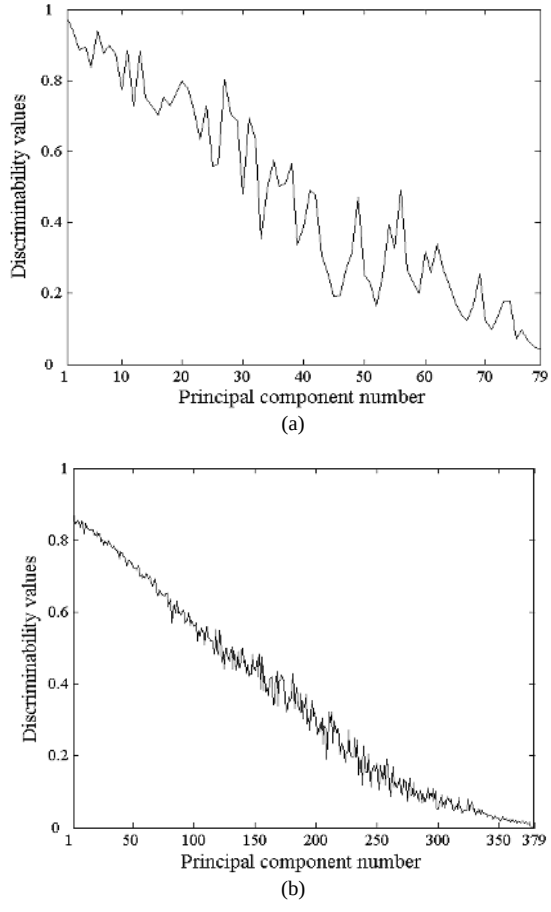
In Equation 7.2, to simplify the expression, we use S_b to represent $W_{pca}^{T} S_b W_{pca}$ and S_t to represent $W_{pca}^{T} S_t W_{pca}$. Suppose that $W_{opt} = [\phi_1, \phi_2, \ldots, \phi_r]$, where r is the number of discrimination vectors. For $\forall \phi_i$ ($i = 1, \ldots, r$), we have:

$$\phi_i^T S_t \phi_i = \phi_i^T S_b \phi_i + \phi_i^T S_w \phi_i \tag{7.3}$$

If $\phi_i^T S_b \phi_i > \phi_i^T S_w \phi_i$, then:

$$F(\phi_i) = \frac{\phi_i^T S_b \phi_i}{\phi_i^T S_t \phi_i} > 0.5 \tag{7.4}$$

Figure 7.1. Fisher discriminative values of the principal components obtained from (a) ORL face database and (b) palmprint database



In this situation, according to the Fisher criterion, there is more between-class separable information than within-class scatter information. So, we choose those discrimination vectors whose Fisher discrimination values are more than 0.5, and discard the others. This improvement allows efficient linear discrimination information to be kept and non-useful information to be discarded. Such a selection of the effective discrimination vectors is important to the recognition effect, especially where the number of vectors is larger, which often happens when the number of pattern classes is large. The experiment will demonstrate the importance of this.

Improving the Statistical Uncorrelation of Discrimination Vectors

Earlier we observed that the statistical uncorrelation of discrimination vectors is a favorable property, useful in pattern classification (Jin, Yang, Hu, & Lou, 2001; Jin, Yang,

Tang, & Hu, 2001; Jing, Zhang, & Jin, 2003a, 2003b). The unique difference between the fisherface method and Jing's UODV method (Jing, Zhang, & Jin, 2003b) is that the discrimination vectors obtained from UODV satisfy the constraint of statistical uncorrelation. It is a simple matter to prove that the eigenface method (Turk & Pentland, 1991) satisfies the statistical uncorrelation. This characteristic of the eigenface method provides an explanation for its relative insensitivity to different training data sets, compared with the fisherface method (Martinez & Kak, 2001). Now, we introduce a corollary provided in Jing, Zhang, and Jin (2003b):

Lemma 7.1 (Jing, Zhang, & Jin, 2003b)**.** Suppose that the discrimination vectors obtained from UODV (refer to Jing's method) are $(\varphi_1, \varphi_2, \ldots, \varphi_r)$, where r is the rank of $S_t^{-1}S_b$, and the non-zero eigenvalues of $S_t^{-1}S_b$ are represented in descending order as $\lambda_1 \geq \lambda_2 \geq \ldots \geq \lambda_r > 0$, and the k^{th} eigenvector ϕ_k of $S_t^{-1}S_b$ corresponds to $\lambda_k (1 \leq k \leq r)$. If $(\lambda_1, \lambda_2, \ldots, \lambda_r)$ are mutually unequal, that is:

$$\lambda_1 > \lambda_2 > \ldots > \lambda_r > 0 \tag{7.5}$$

then φ_k can be represented by ϕ_k.

Lemma 7.1 shows that when the non-zero Fisher discrimination values are mutually unequal, the discrimination vectors generated from the fisherface method can satisfy the statistical uncorrelation. That is, in this situation, the fisherface method and UODV obtain identical discrimination vectors with non-zero discrimination values. Therefore, Lemma 7.1 reveals the essential relationship between these two methods.

Although UODV satisfies the statistical uncorrelation completely, it requires more computational time than the fisherface method. Furthermore, it is not necessary to use UODV if the non-zero Fisher discrimination values are mutually unequal, because the fisherface method can take the place of UODV. In the application of the fisherface method, we find that only a small number of the Fisher values are equal respectively, and the others are unequal mutually. How, then, can computational time be reduced, while simultaneously guaranteeing the statistical uncorrelation for the discrimination approach? Here, we propose an improvement on the fisherface method. Using the assumption in Lemma 7.1, our measure is:

- **Step 1.** Use the fisherface method to obtain the discrimination vectors $(f_1, f_2, \ldots, f_r)$. If the corresponding Fisher values $(l_1, l_2, \ldots, l_r)$ are unequal mutually, over; else, go to the next step.
- **Step 2.** For $2 \leq k \leq r$, if $l_k \neq l_{k-1}$, then keep f_k, else replace f_k by j_k from UODV.

Obviously, the proposal not only satisfies the statistical uncorrelation, it reduces computing time. This will be further demonstrated by our experiments.

Improving the Selection of Principal Components

Assume that W_{pca} in Equations 7.1 and 7.2 is represented by p eigenvectors (principal components) of S_t with non-zero eigenvalues; that is, $W_{pca} = (\beta_1, \beta_2, \ldots, \beta_p)$. The Fisher discriminability of a principal component $\beta_i (1 \leq i \leq p)$ is evaluated as follows:

$$J_i = \frac{\beta_i^T S_b \beta_i}{\beta_i^T S_t \beta_i} \quad (1 \le i \le p) \tag{7.6}$$

Obviously, this quantitative evaluation is rational because it is in accordance with the Fisher criterion. Figure 7.1 shows the Fisher discriminative values of the principal components obtained from: (a) the ORL face database and (b) the palmprint database, where $p = 79$ and $p = 379$, respectively.

From Figure 7.1, two experimental rules can be obtained:

Rule 1. There is no completely direct proportional relationship between the discriminability value of a component and its eigenvalue;

Rule 2. Components with smaller eigenvalues generally have weaker discriminability values.

Rule 1 indicates that the selection method in EFM (Liu & Wechsler, 2000), which uses the components with the larger eigenvalues, is not completely reasonable, while Rule 2 provides a quantitative explanation for why we can select the components with the larger eigenvalues for EFM. This is significant in Figure 7.1b, where the number of components (the training sample set) is large. We will give an automatic and more reasonable strategy for selecting the components than using EFM. The following theorem demonstrates that the total discriminability of LDA equals the sum of the discriminability of each component:

Theorem 7.1. Let tr represent the trace of the matrix. We have:

$$tr\left(\left(W_{pca}^T S_t W_{pca}\right)^{-1}\left(W_{pca}^T S_b W_{pca}\right)\right) = \sum_{i=1}^{p} J_i \tag{7.7}$$

Proof. $W_{pca}^T S_t W_{pca}$ is a diagonalized matrix, that is:

$$W_{pca}^T S_t W_{pca} = \begin{bmatrix} \beta_1^T S_t \beta_1 & 0 & \cdots & 0 \\ 0 & \beta_2^T S_t \beta_2 & \ddots & \vdots \\ \vdots & \ddots & \ddots & 0 \\ 0 & \cdots & 0 & \beta_p^T S_t \beta_p \end{bmatrix} \tag{7.8}$$

and:

$$W_{pca}^T S_b W_{pca} = \begin{bmatrix} \beta_1^T S_b \beta_1 & \cdots & \beta_1^T S_b \beta_p \\ \vdots & \ddots & \vdots \\ \beta_p^T S_b \beta_1 & \cdots & \beta_p^T S_b \beta_p \end{bmatrix} \tag{7.9}$$

So, we have:

$$\left(W_{pca}^T S_t W_{pca}\right)^{-1}\left(W_{pca}^T S_b W_{pca}\right)=\begin{bmatrix} \beta_1^T S_b \beta_1 / \beta_1^T S_t \beta_1 & \cdots & \beta_1^T S_b \beta_p / \beta_1^T S_t \beta_1 \\ \vdots & \ddots & \vdots \\ \beta_p^T S_b \beta_1 / \beta_1^T S_t \beta_1 & \cdots & \beta_p^T S_b \beta_p / \beta_p^T S_t \beta_p \end{bmatrix} \quad (7.10)$$

Due to both Equations 7.6 and 7.10, we obtain Equation 7.7.

Theorem 7.1 implies that to obtain the maximal total Fisher discriminability, we should use all of the components. Nevertheless, some experiments in previous works (Liu & Wechsler, 2000, 2001) have shown that the ideal recognition results may be obtainable by discarding those components with the smaller values. Here, we also provide some experimental results. We use the fisherface method but do a little change on it; that is, not discarding the smallest *c* principal components and using all the components.

Table 7.1 indicates a comparison of recognition rates of the fisherface method and a changed fisherface method using all the components on the ORL face database and the palmprint database, where the first two, three and four samples per class are respectively taken as the training ones. We observe that the results of the changed fisherface method are quite bad. However, the total Fisher discriminability obtained from this changed method is maximal according to Theorem 7.1. Thus, we have to face a contradiction between satisfying the maximal total discriminability and choosing as the discrimination vectors those with favorable characteristics. To solve this contradiction, it may be possible to make a tradeoff; that is, the fundamental Fisher discriminability should be kept and some of components with the smaller Fisher values should be discarded. The following is our strategy:

- **Step 1.** In accordance with Rule 2, discard the smallest *c* components, as in the fisherface method. This helps to reduce computing time.
- **Step 2.** Compute the Fisher discrimination values J_i of the remainder components according to Equation 6, then, rank them in descending order and calculate the sum of their Fisher discriminability values J_{all}.
- **Step 3.** Select the components with the first largest J_i values until a threshold T is satisfied, where T is the ratio of the sum of their values to J_{all}.

Table 7.1. A comparison of recognition rates of the fisherface method and a changed fisherface method using all the components

Different data		ORL face database			Palmprint database		
Number of training samples per class		2	3	4	2	3	4
Recognition rates (%)	Fisherface method [8]	80.94	86.43	88.33	81.35	89.11	90.44
	A changed Fisherface method	47.5	49.28	53.33	48.83	55.75	56.05

Figure 7.2a shows a flowchart of this strategy. In accordance with our tradeoff strategy, we think that the value of T should not be too small or too large. We theoretically estimate that the value range of T might be around 0.5. The following experimental results on face and palmprint databases will show that the value range [0.4, 0.8] is appropriate for *T*, where the variance of the recognition rates is rather small. And in our experiments, T will be set as 0.6.

ILDA Approach

The ILDA approach, which synthesizes our three suggested improvements on LDA, can be described in the following four steps:

- **Step 1.** Select the appropriate principal components according to the strategy defined earlier and perform the fisherface method using its equivalent form expressed by Equation 7.2.
- **Step 2.** From the discrimination vectors obtained, select those whose Fisher discrimination values are more than 0.5.
- **Step 3.** Use the measure defined earlier to make the selected vectors satisfy the statistical uncorrelation. Thus, the generated vectors construct the final linear discrimination transform W.
- **Step 4.** For each sample x in X, extract the linear discrimination feature z:

$$y = x * W \tag{7.11}$$

This obtains a new sample set Y with the linear transformed features corresponding to X. Use the nearest-neighbor classifier to classify Y. Here, the distance between two arbitrary samples, y_1 and y_2, is defined by:

Figure 7.2. Flowcharts of: (a) selecting the principal components and (b) ILDA

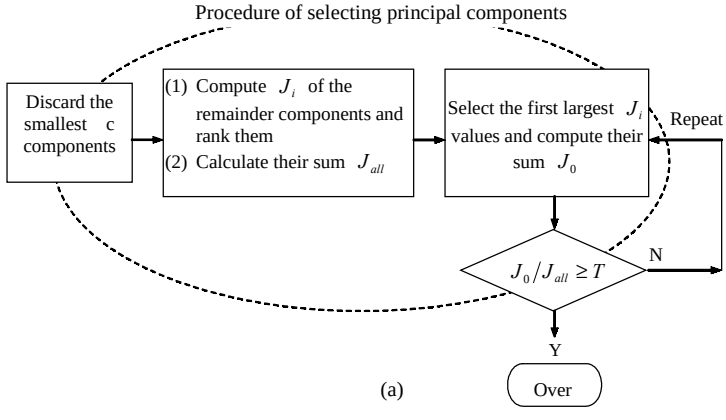
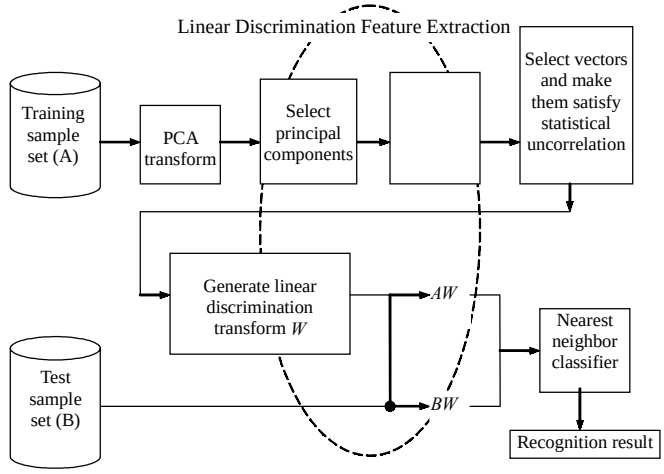


Figure 7.2. cont.



(b)

$$d(y_1, y_2) = \| y_1 - y_2 \|_2 \tag{7.12}$$

where $\| \ \|_2$ denotes the Euclidean distance.

Figure 7.2 (b) shows a flowchart of ILDA.

EXPERIMENTAL RESULTS

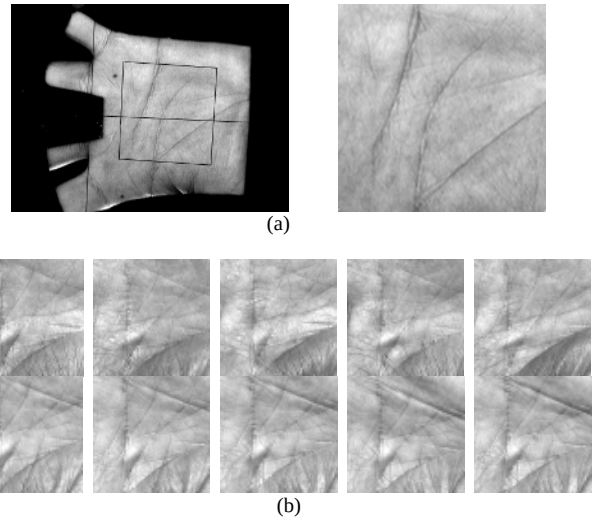
In this section, we first conduct the experiments on the three improvements to LDA. We then compare the experimental results of ILDA and other linear discrimination methods: eigenface, fisherface, DLDA and UODV, using different image data, including a face database and palmprint database. The experiments are implemented on a Pentium 1.4G computer and programmed using the MATLAB language. We do not compare the test time for every method, because it is quite little (less than 1 second) when we test an image sample using any method in the experiments.

Introduction of Databases

We use the ORL facial image database mentioned in Chapter VI (see Figure 6.4). For reasons such as its accommodation of low-resolution imaging, ability to operate on low-cost capture devices and the ease with which the palm can be segmented, palmprint recognition has become an important complement to personal identification. In Lu, Zhang, and Wang (2003), a Gabor-based method is applied to the online palmprint identification. In this chapter, we use the LDA technique to perform off-line palmprint recognition. Two other palmprint recognition methods, eigenpalm and fisherpalm, are

presented in Lu, Zhang, and Wang (2003) and Wu, Zhang, and Wang (2003), respectively. These two methods are very similar to the eigenface (Turk & Pentland, 1991) and the fisherface (Belhumeur, Hespanha, & Kriegman, 1997) methods, so we do not specially compare the eigenpalm and the fisherpalm methods in the following experiments of palmprint images. We collected palmprint images from 190 individuals using our self-designed capture device. The subjects mainly consisted of student and staff volunteers from the Hong Kong Polytechnic University. Of the subjects in this database, 130 persons are male, approximately 87% of the subjects are younger than 30 years old, about 10% are aged between 30 and 50, and about 3% are older than 50. The palmprint images were collected on two separate occasions, at an interval of around two months. After finishing the first collection, we slightly changed the light source and adjusted the focus of the CCD camera so that the images collected on the first and second occasions might be regarded as being captured by two different palmprint devices. On each occasion, the subjects were asked to each provide eight palmprint images for the right hand. Thus, each person provides 16 images and our database contains a total of 3,040 images from 190 different palms. The size of all the original palmprint images is 384×284 pixels with 75dpi resolution. Using the preprocessing approach in Yang (2002), the sub-images with a fixed size (128×128) are extracted from the original images. To reduce the computational cost, each sub-image is compressed to 64×64. We use these sub-images to represent the original palmprint images and to conduct our experiments. Figure 7.3a displays the demo of a sub-image acquired from a palm. Figure 7.3b shows 10 image samples of one person captured at different time. The first five were collected on the first occasion and the second five on the next occasion, the major changes being in illumination and position,

Figure 7.3. Palmprint image data



(a) Demo of a sub-image acquired from a palm; (b) 10 image samples from one person in the palmprint database

including shift and rotation. Similar to the kinds of changes encountered in facial expressions, the image may also be slightly affected by the way the hand is posed, shrunk or stretched.

In the following experiments, the first two samples of every person in each database are used as training samples and the remainder as test samples. Thus, the ORL database provides 80 training samples and 320 test samples. The palmprint database provides 380 training samples and 2,660 test samples. Generally, it is more difficult to classify patterns when there are fewer training samples. This is also illustrated in Table 7.1, where the recognition rates of the fisherface methods are worst when the training sample number per class is two. The experiments take up that challenge and seek to verify the effectiveness of the proposed approach using fewer training samples.

Experiments on the Improvement of Discrimination Vectors Selection

We test the proposed improvement of discrimination vectors selection on the fisherface method. Table 7.2 shows the fisher discriminative values that are obtained, ranged from 0 to 1.

Table 7.3 shows a comparison of the classification performance of the proposed improvement and the fisherface method. The ORL database recognition rate improves 1.25%, while that of the palmprint database improves 4.97%. This improvement can further reduce the dimension of discriminative features. There is little difference in the training time of the fisherface method and the proposed improvement.

Experiments on the Improvement of Statistical UODV

We also test the proposed improvement to the statistical uncorrelation of discrimination vectors on the fisherface method. Table 7.3 also provides a comparison of the classification performance of this improvement, the fisherface method and UODV. The recognition rates of UODV and the improvement are the same, but on the ORL database this improvement is 53.45% faster than UODV, and on the palmprint database it is 43.47% faster. The reason for this, as can be seen in Table 7.2, is that only a small number of Fisher discriminative values are equal, respectively. In other words, most discrimination vectors obtained from the fisherface method are statistically uncorrelated, so there is no need to calculate each discrimination vector using UODV. On the other hand, it is necessary to require the vectors to satisfy this favorable property, since, compared with the fisherface method, our proposed approach can improve recognition rates by 0.31% on the ORL database, and by 7.03% on the palmprint database.

Experiments on the Improvement of Principal Components Selection

We test the proposed improvement to principal components selection on the fisherface method. Table 7.3 also provides a comparison of the classification performance of this improvement and the fisherface method. On the ORL database the improvement increases training time by 7% and on the palmprint database by 11.32%, but improves recognition rates by 5.31% and 9.55%, respectively. The proposed improvement can also greatly reduce the dimension of discriminative features.

Table 7.2. An illustration of Fisher discriminative values obtained using the fisherface method

Different data	Fisher discriminative values
ORL face database	Number of discrimination vectors: 39
	1.0000 1.0000 0.9997 0.9981 0.9973 0.9962 0.9950 0.9932 0.9917 0.9885 0.9855 0.9845 0.9806 0.9736 0.9663 0.9616 0.9555 0.9411 0.9356 0.9151 0.9033 0.8884 0.8517 0.8249 0.8003 0.7353 0.7081 0.6930 0.6493 0.5515 0.4088 0.3226 0.2821 0.2046 0.0493 0.0268 0.0238 0.0081 0.0027
Palmprint database	Number of discrimination vectors: 189
	1.0000 1.0000 1.0000 1.0000 1.0000 0.9999 0.9999 0.9999 0.9998 0.9998 0.9998 0.9997 0.9997 0.9996 0.9996 0.9995 0.9995 0.9994 0.9993 0.9993 0.9992 0.9991 0.9990 0.9989 0.9987 0.9986 0.9985 0.9983 0.9983 0.9982 0.9982 0.9979 0.9976 0.9976 0.9974 0.9971 0.9970 0.9968 0.9967 0.9965 0.9962 0.9960 0.9959 0.9952 0.9948 0.9947 0.9945 0.9943 0.9941 0.9937 0.9932 0.9930 0.9928 0.9922 0.9917 0.9912 0.9910 0.9908 0.9903 0.9900 0.9897 0.9892 0.9888 0.9883 0.9878 0.9870 0.9869 0.9862 0.9858 0.9846 0.9843 0.9836 0.9833 0.9825 0.9822 0.9816 0.9800 0.9795 0.9792 0.9787 0.9783 0.9767 0.9759 0.9752 0.9743 0.9731 0.9723 0.9718 0.9703 0.9701 0.9686 0.9679 0.9656 0.9646 0.9635 0.9621 0.9613 0.9605 0.9591 0.9557 0.9551 0.9535 0.9521 0.9507 0.9486 0.9481 0.9439 0.9436 0.9390 0.9384 0.9371 0.9331 0.9318 0.9313 0.9273 0.9225 0.9194 0.9186 0.9147 0.9118 0.9112 0.9088 0.9069 0.9050 0.9036 0.8889 0.8845 0.8821 0.8771 0.8747 0.8709 0.8659 0.8607 0.8507 0.8488 0.8424 0.8340 0.8280 0.8220 0.8157 0.8070 0.8007 0.7959 0.7825 0.7751 0.7639 0.7626 0.7434 0.7378 0.7284 0.7060 0.6944 0.6613 0.6462 0.6372 0.6193 0.6121 0.5663 0.5436 0.5061 0.4753 0.4668 0.4343 0.3730 0.3652 0.3024 0.2900 0.2273 0.2014 0.1955 0.1758 0.1541 0.1270 0.1159 0.0858 0.0741 0.0683 0.0591 0.0485 0.0329 0.0243 0.0205 0.0184 0.0107 0.0090 0.0049 0.0026 0.0004 0.0001

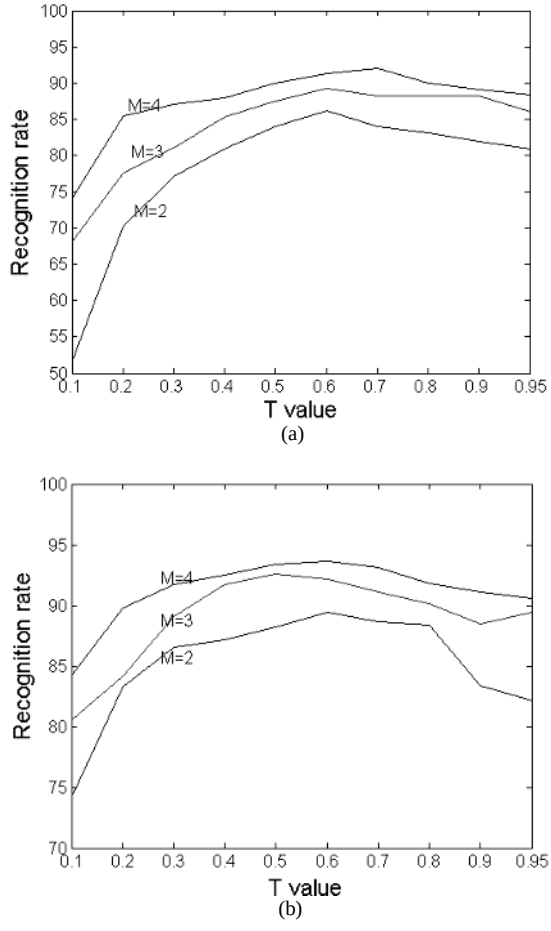
Table 7.3. A comparison of classification performance of three improvements on LDA, the Fisherface method and UODV

Classification performance	Different databases	Discrimination methods				
		Improve-ment 1	Improve-ment 2	Improve-ment 3	Fisherface (Belhumeur, Hespanha & Kriegman, 1997)	UODV (Jing, Zhang & Yang, 2003)
Recognition rates (%)	ORL	82.19	81.25	86.25	80.94	81.25
	Palmprint	86.32	88.38	90.9	81.35	88.38
Extracted feature dimension	ORL	30	39	21	39	39
	Palmprint	160	189	100	189	189
Training time (second)	ORL	14.55	14.7	15.28	14.28	31.58
	Palmprint	36.81	39.06	40.11	36.03	69.1

Figure 7.4 illustrates the recognition rates of this improvement with different image data: (a) ORL face database and (b) palmprint database while the value of T is varied, where $2 \le M \le 4$ (assuming that M is the number of training samples per class). We find that the effective value ranges of the rates for ORL and palmprint databases are [0.4, 0.9] and [0.3, 0.8], respectively. Hence, an appropriate range for both data is [0.4, 0.8].

Table 7.4 shows an analysis of the mean values and the variances of the recognition rates where the value range of T is [0.4, 0.8]. The variances are much smaller than the mean values. In other words, in this range, the recognition effect of our approach is rather robust. Figure 7.4 and Table 7.4 also demonstrate the former theoretical estimation in subsection 2.4, that is, the value range of T might be around 0.5. In the experiments, T is set as 0.6.

Figure 7.4. The recognition rates of the third improvement with different image data



(a) ORL face database and (b) palmprint database, while the value of T is varied

Table 7.4. An analysis of the mean values and the variances of the recognition rates in the third improvement when the value range of T is [0.4, 0.8]

Different data	ORL face database			Palmprint database		
	Number of training samples per class			Number of training samples per class		
	2	3	4	2	3	4
Mean recognition rates (%)	83.69	87.71	90.25	88.37	91.55	92.89
Variance	1.70	1.31	1.41	0.75	0.88	0.65
Total mean recognition rate (%)	87.22			90.94		
Average variance	1.47			0.76		

Experiments on All of the Improvements

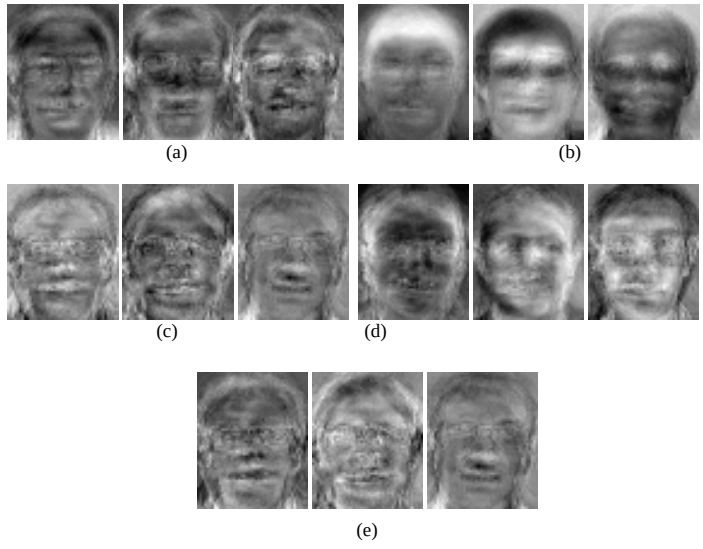
ILDA synthesizes all the above improvements on LDA. Figure 7.5 displays the demo images of discrimination vectors obtained from different methods on the ORL database.

Table 7.5 shows a comparison of the classification performance of ILDA and other methods. Using the ORL face database, the improvements in ILDA's recognition rates over eigenface, fisherface, DLDA and UODV are 5.31%, 6.25%, 4.69% and 5.94%, respectively. Using the palmprint database, the improvements in ILDA's recognition rates over eigenface, fisherface, DLDA and UODV are, again respectively, 18.3%, 12.18%, 19.43% and 5.15%. In addition, compared with fisherface, DLDA and UODV (which uses the second least number of features), ILDA remarkably reduces the feature dimension by 51.28% and 50.26%, respectively, for the ORL database and palmprint database. ILDA is much faster than UODV and its training time is rather close to those of eigenface, fisherface and DLDA. On the ORL database it is 50.29% faster than UODV, and on the palmprint database it is 39.28% faster. Compared to the fisherface method, it only adds training time of 9.94% and 16.46%, respectively, for ORL database and palmprint database.

Table 7.5. A comparison of classification performance of ILDA and other linear discrimination methods

Classification Performance	Different databases	Discrimination methods				
		ILDA	Eigenface[6]	Fisherface[8]	DLDA[13]	UODV[22]
Recognition rates (%)	ORL	87.19	81.88	80.94	82.5	81.25
	Palmprint	93.53	75.23	81.35	74.1	88.38
Extracted feature Dimension	ORL	19	79	39	39	39
	Palmprint	92	379	189	189	189
Training time (second)	ORL	15.7	13.03	14.28	13.01	31.58
	Palmprint	41.96	32	36.03	37.54	69.1

Figure 7.5. Demo images of the discrimination vectors obtained from different methods on the ORL database



(a) ILDA, (b) eigenface, (c) fisherface, (d) DLDA and (e) UODV

SUMMARY

ILDA effectively synthesizes three useful improvements on the current linear discrimination technique: It improves the selection of discrimination vectors, adds a measure so the discrimination vectors satisfy the statistical uncorrelation using less computing time and provides a strategy to select the principal components. We verify ILDA on different image databases. The experimental results demonstrate that it classifies better than major linear discrimination methods. Compared with the most representative LDA method, the fisherface method, ILDA improves the recognition rates up to 12.18% and reduces the feature dimension by up to 51.28%, while adding only up to 16.46% to training time of the fisherface method. Consequently, we conclude that ILDA is an effective linear discrimination approach.

REFERENCES

Bartlett, M. S., Movellan, J. R., & Sejnowski, T. J. (2002). Face recognition by independent component analysis. *IEEE Transactions on Neural Networks, 13*(6), 1450-1464.

Belhumeur, P. N., Hespanha, J. P., & Kriegman, D. J. (1997). Eigenfaces vs. fisherface: Recognition using class specific linear projection. *IEEE Transactions on Pattern Analysis and Machine Intelligence, 19*(7), 711-720.

Chen, L., Liao, H. M., Ko, M., Lin, J., & Yu, G. (2000). A new LDA-based face recognition system which can solve the small sample size problem. *Pattern Recognition, 33*(10), 1713-1726.

Cheng, Y. Q., Zhuang, Y. M., & Yang, J. Y. (1992). Optimal Fisher discrimination analysis using the rank decomposition. *Pattern Recognition, 25*(1), 101-111.

Cooke, T. (2002). Two variations on Fisher's linear discrimination for pattern recognition. *IEEE Transactions on Pattern Analysis and Machine Intelligence, 24*(2), 268-273.

Fisher, R. A. (1936). The use of multiple measurements in taxonomic problems. *Annals of Eugenics, 7*, 178-188.

Foley, D. H., & Sammon, J. W. (1975). An optimal set of discrimination vectors. *IEEE Transactions on Computers, 24*(3), 281-289.

Fukunaga, K. (1990). *Introduction to statistical pattern recognition*. New York: Academic Press.

Jain, A. K., Duin, R. P. W., & Mao, J. (2000). Statitical pattern recognition: A review. *IEEE Transactions on Pattern Analysis and Machine Intelligence, 22*(1), 4-37.

Jin, Z., Yang, J., Hu, Z., & Lou, Z. (2001). Face recognition based on the uncorrelated discrimination transformation. *Pattern Recognition, 34*(7), 1405-1416.

Jin, Z., Yang, J., Tang, Z., & Hu, Z. (2001). A theorem on the uncorrelated optimal discrimination vectors. *Pattern Recognition, 34*(10), 2041-2047.

Jing, X. Y., Zhang, D., & Jin, Z. (2003a). Improvements on the uncorrelated optimal discriminant vectors. *Pattern Recognition, 36*(8), 1921-1923.

Jing, X. Y., Zhang, D., & Jin, Z. (2003b). UODV: Improved algorithm and generalized theory. *Pattern Recognition, 36*(11), 2593-2602.

Jing, X. Y., Zhang, D., & Tang, Y. (2004). An improved LDA approach. *IEEE Transactions on Systems, Man and Cybernetics, Part B, 34*(5), 1942-1951.

Jing, X. Y., Zhang, D., & Yang, J. Y. (2003). Face recognition based on a group decision-making combination approach. *Pattern Recognition, 36*(7), 1675-1678.

Jing, X. Y., Zhang, D., & Yao, Y. F. (2003). Improvements on the linear discrimination technique with application to face recognition. *Pattern Recognition Letters, 24*(15), 2695-2701.

Liu, C., & Wechsler, H. (2000). Robust coding scheme for indexing and retrieval from large face databases. *IEEE Transactions on Image Processing, 9*(1), 132-137.

Liu, C., & Wechsler, H. (2001). A shape- and texture-based enhanced Fisher classifier for face recognition. *IEEE Transactions on Image Processing, 10*(4), 598-608.

Liu, K., Cheng, Y. Q., & Yang, J. Y. (1993). Algebraic feature extraction for image recognition based on an optimal discrimination criterion. *Pattern Recognition, 26*(6), 903-911.

Liu, K., Cheng, Y. Q., Yang, J. Y., & Liu, X. (1992). An efficient algorithm for Foley-Sammon optimal set of discrimination vectors by algebraic method. *International Journal of Pattern Recognition and Artificial Intelligence, 6*(5), 817-829.

Loog, M., Duin, R. P. W. & Haeb-Umbach, R. (2001). Multiclass linear dimension reduction by weighted pairwise Fisher criteria. *IEEE Transactions on Pattern Analysis and Machine Intelligence, 23*(7), 762-766.

Lu, G., Zhang, D., & Wang, K. (2003). Palmprint recognition using eigenpalms features. *Pattern Recognition Letters, 24*(9-10), 1463-1467.

Malina, W. (2001). Two-parameter Fisher criterion. *IEEE Transactions on Systems, Man, and Cybernetics, Part B, 31*(4), 629-636.

Martinez, A. M., & Kak, A. C. (2001). PCA versus LDA. *IEEE Transactions on Pattern Analysis and Machine Intelligence, 23*(2), 228-233.

Nishino, K., Sato ,Y., & Ikeuchi, K. (2001). Eigen-texture method: appearance compression and synthesis based on a 3D model. *IEEE Transactions on Pattern Analysis and Machine Intelligence, 23*(11), 1257-1265.

Swets, D. L., & Weng, J. J. (1996). Using discrimination eigenfeatures for image retrieval. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, *18*(8), 831-836.

Turk, M., & Pentland, A. (1991). Eigenfaces for recognition. *Journal of Cognitive Neuroscience, 3*(1), 71-86.

Vailaya, A., Zhang, H., Yang, C., Liu, F., & Jain, A.K. (2002).Automatic image orientation detection. *IEEE Transactions on Image Processing, 11*(7), 746-755.

Wu, X., Zhang, D., & Wang, K. (2003). Fisherpalms based on palmprint recognition. *Pattern Recognition Letters, 24*(15), 2829-2938.

Yang, J., & Yang, J. Y. (2002). From image vector to matrix: A straightforward image projection technique – IMPCA vs. PCA. *Pattern Recognition, 35*(9), 1997-1999.

Yang, M. H. (n.d.). Kernel eigenfaces vs. kernel fisherfaces: Face recognition using kernel methods. *Proceedings of the Fifth IEEE International Conference on Automatic Face and Gesture Recognition (RGR'02),* Washington, DC (pp. 215-220).

Yu, H., & Yang, J. (2001). A direct LDA algorithm for high-dimensional data with application to face recognition. *Pattern Recognition, 34*(12), 2067-2070.

Zhang, D., Kong, W. K., You, J., & Wong, M. (2003). On-line palmprint identification. *IEEE Transactions on Pattern Analysis and Machine Intelligence, 25*(9), 1041-1050.

Zhang, D., Peng, H., Zhou, J., & Pal, S. K. (2002). A novel face recognition system using hybrid neural and dual eighefaces methods. *IEEE Transactions on Systems, Man, and Cybernetics, Part A, 32*(6), 787-793.

Chapter VIII

Discriminant DCT Feature Extraction

ABSTRACT

This chapter provides a feature extraction approach that combines the discrete cosine transform (DCT) with LDA. The DCT-based frequency-domain analysis technique is introduced first. Then, we describe the presented discriminant DCT approach and analyze its theoretical properties. Finally, we offer detailed experimental results and a chapter summary.

INTRODUCTION

Frequency-domain analysis is a commonly used image processing and recognition technique. During the past years, some work has been done to extract the frequency-domain features for image recognition. Li, Zhang, and Xu (2002) extract Fourier range and angle features to identify the palmprint image. Lai, Yuen, and Feng (2001) use holistic Fourier invariant features to recognize the facial image. Another spectral feature generated from SVD is used by some researchers (Chellappa, 1995). However, Tian, Tan, Wang and Fang (2003) indicate that this feature does not contain adequate information for face

recognition. In Hafed and Levine (2001), they extract DCT feature for face recognition. They point out that DCT obtains the near-optimal performance of K-L transform in facial information compression. And the performance of DCT is superior to those of discrete Fourier transform (FT) and other conventional transforms. By manually selecting the frequency bands of DCT, their recognition method achieves similar recognition effect as the eigenface method (Turk & Pentland, 1991) which is based on K-L transform. Nevertheless, their method cannot provide a rational band selection rule or stategy. And it cannot outperform the classical eigenface method.

To enhance the image classification information and improve the recognition effect, we propose a new image recognition approach in this section (Jing & Zhang, 2004), which combines DCT with the linear discrimination technique. It first uses a 2D separability judgment that can facilitate the selection of useful DCT frequency bands for image recognition, because not all the bands are useful in classification. It will then extract the linear discriminative features by an improved fisherface method and perform the classification by the nearest-neighbor classifier. We will perform the detailed analysis of the theoretical advantages of our approach. The rest of this section is organized as follows: First, we provide the description of our approach. Then, we show its theoretical analysis. Next, the experimental results on different image data and some conclusions are given.

APPROACH DEFINITION AND DESCRIPTION

In this section, we present a 2D separability judgment and introduce the whole recognition procedure of our approach.

Select DCT Frequency Bands by Using a 2D Separability Judgment

Suppose that image training and test sample sets are X_1 and X_2, respectively; each gray image matrix is sized $M \times N$ and expressed by $f(x,y)$, where $1 \le x \le M$, $1 \le y \le N$ and $M \ge N$. Assume there are c known pattern classes (w_1, w_2, . . . , w_c) in X, where $P_i (i = 1, 2, \ldots, c)$ denotes the a priori probability of class w_i. Perform a 2DDCT on each image (Hafed & Leveine, 2001) by:

$$F(u,v) = \frac{1}{\sqrt{MN}} \alpha(u)\alpha(v) \sum_{x=1}^{M} \sum_{y=1}^{N} f(x,y) \cos\left[\frac{(2x+1)u\pi}{2M}\right] \cos\left[\frac{(2y+1)v\pi}{2N}\right] \quad (8.1)$$

where $F(u,v)$ is sized $M \times N$, and $\alpha(\bullet)$ is defined as follows:

$$\alpha(w) = \begin{cases} \frac{1}{\sqrt{2}}, & w = 1 \\ 1, & otherwise \end{cases} \quad (8.2)$$

Figure 8.1. Demo of a facial image and its DCT transformed image

Figure 8.2. Illustration of expression ways of DCT frequency bands

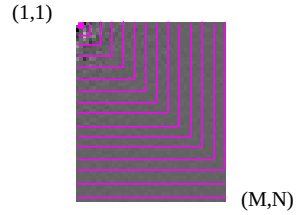


Figure 8.1 represents: (a) a facial image and (b) its transformed image. From Figure 8.1b, most information or energy of image is concentrated in the left-up corner; that is, in the low-frequency bands. Here, we provide a 2D expression for different bands of the transformed image. A half-square ring $Ring(k)$ is used to represent the k^{th} frequency band. Different DCT frequency bands with the above expression ways are illustrated in Figure 8.2.

When $k \le N$, the three vertexes of $Ring(k)$ are $(1, k)$, $(k, 1)$ and (k, k), respectively. When $N < k \le M$, $Ring(k)$ is represented by only one side whose two vertexes are (k, 1) and (k, N) , respectively. So, the k^{th} frequency band denotes:

$$F(u,v) \in Ring(k), \quad 1 \le k \le M \tag{8.3}$$

If we select the k^{th} frequency band, then we keep the original values of $F(u,v)$, otherwise set the values of $F(u,v)$ to change to zero. Which principle should we follow to select the appropriate bands? Here, we propose a 2D separability judgment to evaluate the separability of the frequency bands and select them:

1. Use the k^{th} frequency band:

$$F(u,v) = \begin{cases} Original\ values, & if\ F(u,v) \in Ring(k) \\ 0, & if\ F(u,v) \notin Ring(k) \end{cases} \tag{8.4}$$

Thus, for the images in X_1, we obtain the corresponding band-pass filtered images $F(u,v)$, which construct a new 2D sample set Y_k. Obviously, Y_k and X_1 have the

same numbers of classes, the number of samples and the priori probabilities. Assuming that A_i (i = 1, 2, . . . , c) denotes a mathematic expected value of class w_i in Y_k and A denotes the total expected value of Y_k:

$$A_i = E(Y_k(w_i)) \text{ and } A = E(Y_k) \tag{8.5}$$

Here, A_i and A are 2D matrices whose dimensions are corresponding to u and v in Equation 8.3. For Y_k, the between-class scatter matrix S_b, the within-class scatter matrix S_w and the total scatter matrix S_t are defined as:

$$S_b = \sum_{i=1}^{c} P_i \left[(A_i - A)(A_i - A)^T \right] \tag{8.6}$$

$$S_w = \sum_{i=1}^{c} P_i E\left[(Y_k - A_i)(Y_k - A_i)^T \right] \tag{8.7}$$

$$S_t = E\left[(Y_k - A)(Y_k - A)^T \right] = S_b + S_w \tag{8.8}$$

2. We evaluate the separability of Y_k, $J(Y_k)$, using the following judgment:

$$J(Y_k) \quad \frac{tr(S_b)}{tr(S_w)} \tag{8.9}$$

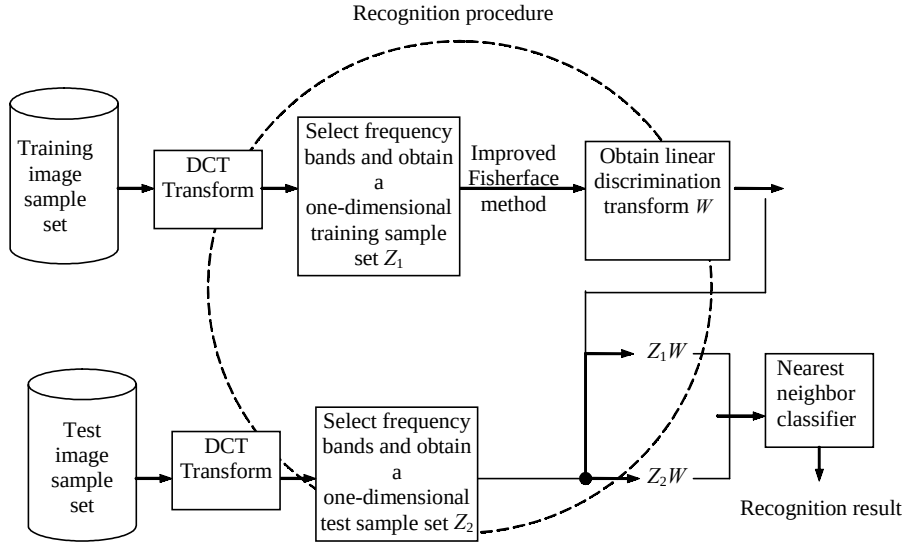
where $tr(\,)$ represents the trace of the matrix. For all the frequency bands ($1 \le k \le M$), we select the bands by:

$$J(Y_k) > T_1 \tag{8.10}$$

When T_1 = 1, that is $tr(S_b) > tr(S_w)$. There is more between-class separable information than within-class scatter information for Y_k according to the Fisher criterion. In other words, the corresponding selected frequency band has good linear separability. Hence, the theoretical value of T_1 should be 1.0. However, its experimental value might not be completely consistent with its theoretical value. In the experiments, we tune the experimental value of T_1 according to different data. The data with fewer samples often has fewer frequency bands whose separability values are more than 1. So, T_1 is set at less than 1.0 in order to use the bands with comparatively higher separability values as much as possible. The data with more samples often has more frequency bands whose separability values are more than 1. So, T_1 is set more than 1.0 in order to select the most effective bands from many candidates.

We obtain a 2D training sample set Y with all the selected bands. Note that Y_k is corresponding to only one selected band; that is, the k^{th} frequency band, but Y is corresponding to all selected bands. It should have favorable total separability value

Figure 8.3. Illustration of the image recognition procedure of our approach



We first select the appropriate frequency bands for the training sample set, then an improved fisherface method is proposed to extract the image discrimination features and the nearest-neighbor classifier is applied to the feature classification.

$J(Y)$, which can be similarly computed by Equation 8.9. The experiments will show that $J(Y)$ obtained after band selection is greater than that obtained before selection. Notice that if we only use one frequency band with the maximum of $J(Y_k)$, it is difficult to guarantee that the selected band has good generalization capability in classification, because the number of training image samples is always very limited. Therefore, for image recognition, a range of frequency bands should be selected.

Recognition Procedure

- **Step 1:** Use the measure introduced earlier to select the appropriate frequency bands. If the k^{th} frequency band is selected, then all $F(u,v)$ values belonging to this band are kept and represented by a feature vector. Then, we link all the feature vectors to a vector. In other words, each sample is represented by a feature vector. Thus, we obtain a 1D training sample set Z_1 corresponding to X_1 and Y. We can also acquire a 1D test sample set Z_2 corresponding to X_2. For Z_1, compute its S_b, S_w and S_t.
- **Step 2:** Perform the following improvements on the original fisherface method:
  1. Calculate the discriminant transform W_{opt}:

$$W_{opt} = W_{pca} W_{fld} \tag{8.11}$$

where W_{pca} and W_{fld} represent principal component analysis and Fisher linear discrimination analysis (Yu & Yang, 2001). W_{pca} is constructed by selecting principal components of S_t. We use a simple selection measure of principal components for W_{pca}. If the total number of components is less than $2*c$, where c is the number of classes, then we keep all the components; otherwise, we discard the smallest c ones like the original fisherface method. This is an experimental measure. We find that after selecting frequency bands, the dimension of obtained feature vector is small, and the number of generated principal components is often lower than $2*c$. In such a situation, if we discarded the smallest c components, then the number of remaining ones is lower than c and the recognition effect is often not ideal. Therefore, this measure is suitable for our proposed approach, which involves the method of selecting components in the original fisherface method.

2. Select the achieved discrimination vectors in the following way. Suppose that $W_{opt} = (\varphi_1, \varphi_2, \ldots, \varphi_M)$, where M is the number of vectors. The Fisher discrimination value of φ_i $(1 \le i \le M)$ is defined as follows:

$$F(\varphi_i) = \frac{\varphi_i^T \left(W_{pca}^T S_b W_{pca}\right) \varphi_i}{\varphi_i^T \left(W_{pca}^T S_w W_{pca}\right) \varphi_i} \tag{8.12}$$

Select φ_i if $F(\varphi_i) > T_2$ and obtain the final discrimination transform matrix W. Similarly to T_1, the theoretical value of T_2 should be 1.0. However, its experimental value might not be completely consistent with its theoretical value. In the experiments, T_2 is set no more than 1. The reason is that extracting discrimination vectors in this step is after the selection of frequency bands in Step 1. In other words, we have carried out one selection procedure for using effective bands by setting T_1. Thus, for the generated discrimination vectors whose separability values are less than 1, they might have useful discrimination information for the classification. We need to make use of as many vectors as possible. Our experimental results will show that T_2 is set no more than 1 for all data.

- **Step 3:** For each sample z_1 in Z_1 and z_2 in Z_2, extract linear discrimination feature l_1 and l_2:

$$l_1 = z_1 W \text{ and } l_2 = z_2 W \tag{8.13}$$

Then, use the nearest-neighbor classifier for classification. Here, the distance d between training sample l_1 and test sample l_2 is defined by:

$$d(l_1, l_2) = \| l_1 - l_2 \|_2 \tag{8.14}$$

where $\| \ \|_2$ denotes the Euclidean distance.

Theoretical Analysis

In this section, we analyze the theoretical advantages of our approach.

Favorable Properties of DCT

The K-L transform is an optimal transform for removing statistical correlation. Of the discrete transforms, DCT best approaches the K-L transform. In other words, DCT has strong ability of removing correlation and compressing images. This is also illustrated by Hafed and Levine in face recognition (2001). They directly extracted DCT feature from facial images and achieved similar classification effect as the eigenface method, which is based on K-L transform. Besides, DCT can be realized by fast Fourier transform (FFT), while there is no fast realization algorithm for K-L transform. Therefore, our apporach sufficiently utilizes these favorable properties of DCT.

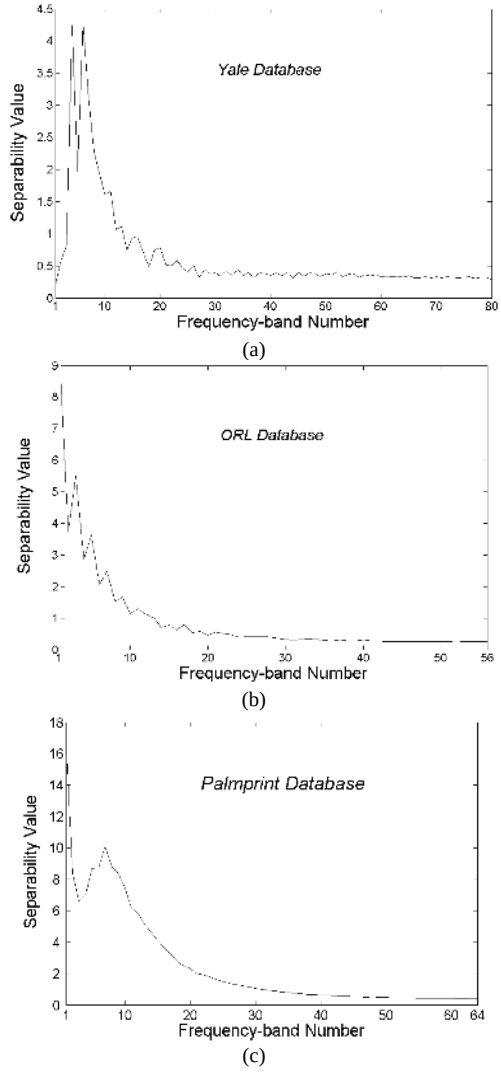
Precise Frequency Band Selection

Another advantage of our approach is that it can precisely select appropriate frequency bands with favorable linear separability. Figure 4 provides a demo of separability values of all bands for various image data: (a) Yale face database, (b) ORL face database and (c) palmprint database, where the numbers of training samples per class for all data are five. From Figure 4, an experiential rule can be obtained: The lower-frequency bands generally have larger separability values, and there is no completely direct proportional relationship between the separability of a band and the band's level.

The discriminant waveletface method (Chen & Wu, 2002) extracts the third-level low-frequency sub-image of the original image by using wavelet transform. According to the obtained experimental rule, three disadvantages exist for this method:

1. It cannot theoretically determine which level of sub-image is most appropriate for extracting linear discrimination features.
2. Not all information in the low-frequency sub-image is useful. Figure 8.4a provides an effective illustration. The separability values of the first two frequency bands are smaller (less than 1). Corresponding to these bands, the related information of the sub-image should be removed, since it is useless to pattern classification.
3. The useful discriminant information of other sub-images may be discarded. Table 8.1 shows the separability values of different sub-images of wavelet decomposition calculated by the image separability judgment, where the types of sub-images include low frequency, horizontal edge, vertical edge and diagonal edge, and the levels of sub-images are from one to four. This table also displays the recognition rates of different sub-images by using the discriminant waveletface method, where the nearest-neighbor classifier is adopted. For the edge sub-images of the third and fourth levels, most of their separability values are more than 1.0. Moreover, the recognition effects of the fourth-level edge sub-images demonstrate that some useful discriminant information in them should not be discarded. Besides, from the fourth-level low-frequency sub-images, we can obtain better recognition rate (95%) than that from the third-level low-frequency sub-images (94.5%). This also illustrates the first disadvantage of the discriminant waveletface method.

Figure 8.4. Demo of separability values of all frequency bands for different image data



(a) Yale face database, (b) ORL face database and (c) palmprint database

Operation Facility in Frequency-Domain

The third advantage is the operation facility of our approach. It can select the bands directly in the frequency-domain, since the transformed results of DCT can be expressed by a real number. However, if our approach is based on FT, we cannot directly select the bands in the frequency domain. This is because the transformed results of FT are expressed by complex numbers. If we wish to evaluate the linear separability of frequency bands of FT, we must conduct inverse FT for the interesting bands. In other words, we must evaluate the separability of interesting bands in the space-domain of image.

Table 8.1. Separability values of different sub-images by wavelet decomposition and the corresponding recognition rates by using the discriminant waveletface method on ORL database

Classification performance	Different levels	Image size	Sub-images			
			Low-frequency	Horizontal edge	Vertical edge	Diagonal edge
Separability values	1	46*56	2.2430	0.4429	0.3649	0.2560
	2	23*28	2.6391	0.6837	0.5591	0.3145
	3	12*14	3.2727	1.2091	1.0479	0.4860
	4	6*7	4.2262	2.3663	1.5121	1.0043
Recognition rates (%)	1	46*56	N/A	N/A	N/A	N/A
	2	23*28	N/A	N/A	N/A	N/A
	3	12*14	94.5[14]	N/A	N/A	N/A
	4	6*7	95	84.5	71	58

Obviously, this will increase the computational cost. Hence, our DCT-based approach can save the computing time than the potential FT-based method.

In the experiments, we have demonstrated that after selecting the frequency bands using our approach, the same total separability value can be achieved from the DCT frequency-domain images and the space-domain images generated using inverse DCT.

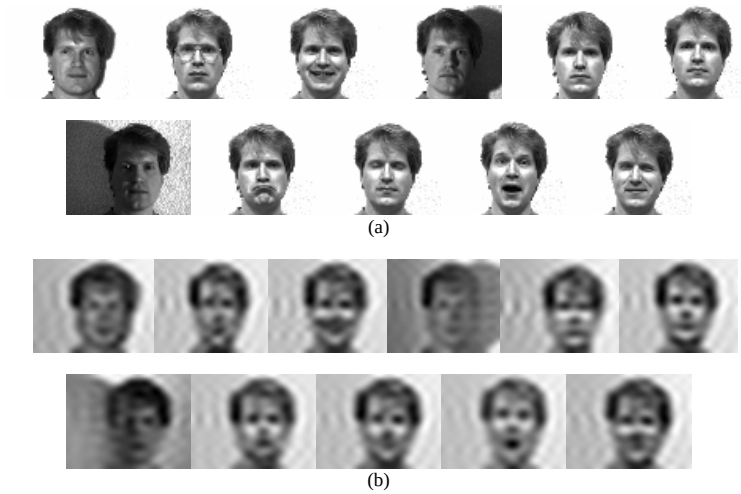
EXPERIMENTS AND ANALYSIS

This section will compare the experimental results of our approach with four conventional linear discrimination methods: eigenface, fisherface, DLDA and discriminant waveletface. All methods adopt the same classifier as our approach. The experiments are implemented on a Pentium 1.4GHz computer (256MB RAM) and programmed in the MATLAB language (v. 6.5).

Experiments with the Yale Face Database

The Yale face database (http://cvc.yale.edu) contains images with major variations, including changes in illumination, subjects wearing eyeglasses and different facial expressions. This database involves 165 frontal facial images, with 11 images of 15 individuals. The size of each image is 243×320, with 256 gray levels. To decrease computing time and simultaneously guarantee sufficient resolution, each image is scaled to an image size of 60×80. We use the full facial image without manually cutting out the background, which is different from the fisherface method. Figure 8.5a shows 11 sample images for one person. Figure 8.5b shows the corresponding images processed by frequency band selection. We take the first five images of each person as the training samples and the remainder as the test samples. So, the numbers of training and test samples are 75 and 90. The related parameters in our approach for Yale database can be seen in Table 8.2. The experimental values of T_1 and T_2 are set as 0.8 and 1.0, respectively. Not all of the low-frequency bands are selected. The first two and the 14th bands are discarded, as they are useless to pattern discrimination. After band selection, the total separability value of the training sample set is improved by 0.415 (=1.955-1.540). The total

Figure 8.5. Demo images from the Yale database



(a) Original facial images and (b) the processed images after band selection

number of principal components is 74, which is more than $2*c$ (=2*15=30). According to the improved fisherface method, the smallest 15 components are discarded. Hence, the first 59 components are used for achieving the discrimination transform. And, 14 discrimination vectors are obtained. Note that the total number of components is equal to the rank of S_t defined in Equation 8.8. S_t is used to solve W_{pca}.

Table 8.2. Implement procedure of our approach for different image data

Implement procedure of our approach			Image data		
			Yale face database	ORL face database	Palmprint database
Parameter setting	T_1		0.8	2.0	2.0
	T_2		1.0	0.6	0.5
Important experimental results	Numbers of the selected frequency bands		3-13, 15-16	1-7	1-20
	Total separability of training set	Before band selection	1.540	1.994	5.014
		After band selection	1.955	3.914	8.458
	Feature vector dimension of selected bands		225	49	400
	Total number of principal components		74	49	210
	Number of classes		15	40	190
	Number of used principal components		59	49	210
	Extracted feature dimension		14	29	181

Table 8.3. Comparison of classification performance using the Yale database

Methods	Our approach	Eigenface	Fisherface	DLDA[7]	Discriminant waveletface[8]
Recognition rates (%)	97.78	91.11	80	87.78	85.56
Extracted feature dimension	14	74	14	14	14
Training time (second)	14.8	14.3	14.9	13.1	15.2

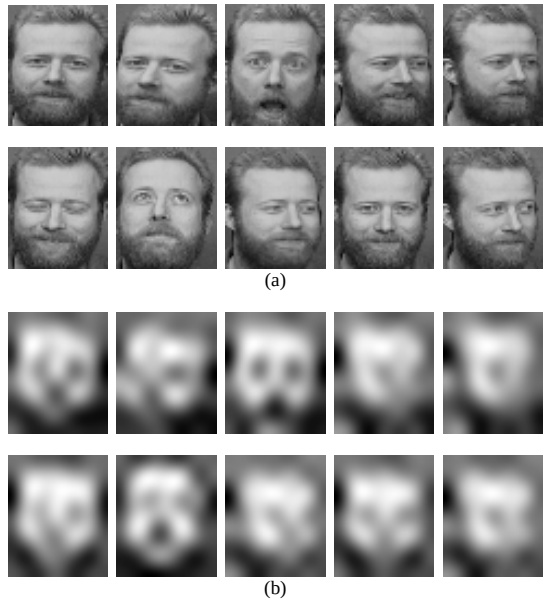
A comparison of the classification performance of all the methods is provided in Table 8.3. Our approach obtains the highest recognition rate. The maximum improvements in the recognition rate of our approach over those of eigenface, fisherface, DLDA and discriminant waveletface are 6.67%, 17.78%, 10% and 12.22%, respectively. Notice that for the discriminant waveletface method, we take the fourth-level low-frequency sub-images of the initial 60×80 images; that is, the sub-image size is 4×5. With respect to the sub-images of the first to the third levels, we cannot obtain the solution of discrimination transform, since the within-class scatter matrix S_w in this method is singular. Our approach extracts the discriminative features with the same low dimension as other methods except for the eigenface method. There is little difference in the training time of all methods.

Besides, there are three methods also using Yale face database to perform the experiments. Jing et al. present some improvements on LDA and a generalized UODV discrimination method, respectively (Jing, Zhang, & Yao, 2003; Jing, Zhang, & Jin, 2003). These two methods take the first five images of each person as the training samples, like our approach. The acquired recognition rates are 89.01% and 92.22%, which are less than the recognition result acquired by our approach; that is, 97.78%. Dai et al. present a regularized discriminant analysis method for face recognition (Dai & Yuen, 2003). Using the Yale database, they obtained a mean recognition rate 97.5% on arbitrarily selecting five images of each person as the training samples four times. It cannot compare with the recognition rate acquired by our approach, because we use the first five images as the training samples.

Experiments with the ORL Face Database

The ORL database (www.cam-orl.co.uk) contains images varied in facial expressions, facial details, facial poses and in scale. The database contains 400 facial images: 10 images of 40 individuals. The size of each image is 92×112, with 256 gray levels. Each image is scaled to 46×56. Figure 8.6a shows 10 sample images for one person. Figure 8.6b shows the corresponding processed images by frequency band selection. We use the first five images of each person as the training samples and the remainder as the test samples. In other words, there is an equal number (200) of training and test samples. The related parameters in our approach for the ORL database can also be seen in Table 8.2. The experimental values of T_1 and T_2 are set as 2.0 and 0.6, respectively. Only a small part of the lowest-frequency bands are selected, which are the first seven bands. The total

Figure 8.6. Demo images from the ORL database



(a) Original facial images and (b) the processed images after band selection

separability value of the training sample set is remarkably improved by 1.92 (=3.914-1.994) after band selection. The total number of principal components is 49, which is less than 2**c* (=2*40=80). So, we do not discard any components. In the end, 29 discrimination vectors are obtained.

A comparison of the classification performance of all the methods is provided in Table 8.4. Our approach obtains the highest recognition rate and the lowest feature dimension. The maximum improvements in the recognition rate of our approach over those of eigenface, fisherface, DLDA and discriminant waveletface are 7.5%, 15%, 8.5% and 3%, respectively. Compared with fisherface, DLDA and discriminant waveletface (which uses the second least number of features), our approach reduces the feature dimension by 25.64%. There is little difference in the training time of all methods.

Some other methods also use the ORL database. Ko et al. present a N-division output coding method for face recognition (Ko & Byun, 2003) and Yang et al. put forward

Table 8.4. Comparison of classification performance using the ORL database

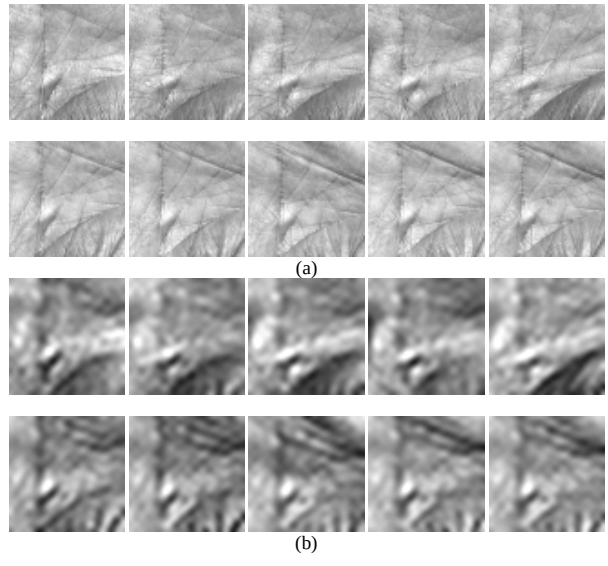
Methods	Our approach	eigenface	fisherface	DLDA[7]	Discriminant waveletface[8]
Recognition rates (%)	97.5	90	82.5	89	94.5[8]
Extracted feature dimension	29	199	39	39	39
Training time (second)	24.9	23.7	26.4	22.1	28.5

an image PCA method (Yang & Yang, 2002). These two methods take the first five images of each person as the training samples, like our approach. The acquired recognition rates are 93.5% and 95.5%, which are less than the recognition result acquired by our approach; that is, 97.5%. Dai et al. present a regularized discriminant analysis method for face recognition (Dai & Yuen, 2003). Using the ORL database, they obtained a mean recognition rate 95.25% on arbitrarily selecting five images of each person as the training samples four times. It cannot compare with the recognition rate acquired by our approach, because we use the first five images as the training samples.

Experiments with the Palmprint Database

For reasons such as its accommodation of low-resolution imaging, ability to operate on low-cost capture devices and the ease with which the palm can be segmented, palmprint recognition has become an important complement to personal identification. Wu et al. use the fisherpalm method in palmprint recognition (Wu, Zhang, & Wang, 2003), which is very similar to the fisherface method (Yu & Yang, 2001). We collected palmprint images from 190 individuals using our self-designed capture device. The subjects mainly consisted of student and staff volunteers from the Hong Kong Polytechnic University. Of the subjects in this database, 130 persons are male, approximately 87% of the subjects are younger than 30 years old, about 10% are aged between 30 and 50, and about 3% are older than 50. The palmprint images were collected on two separate occasions, at an interval of about 2 months. After finishing the first collection, we slightly changed the light source and adjusted the focus of the CCD camera so that the images collected on the first and second occasions might be regarded as being captured by two different

Figure 8.7. Demo images from the palmprint database



(a) Original palmprint images, and (b) the processed images after band selection

Table 8.5. Comparison of classification performance using the palmprint database

Methods	Our approach	eigenface	fisherface	DLDA[7]	Discriminant waveletface[8]
Recognition rates (%)	98.13	71.34	90.91	71	94.97
Extracted feature dimension	181	949	189	189	64
Training time (second)	196.6	204.3	323.9	161.9	172.7

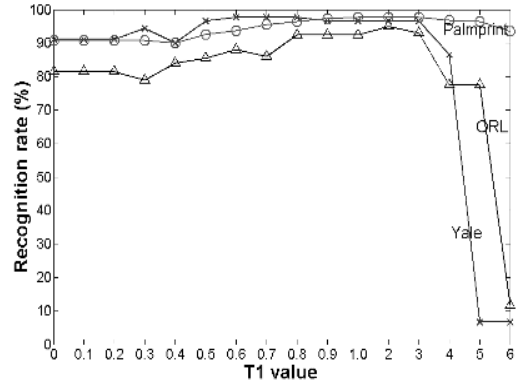
palmprint devices. On each occasion, the subjects were asked to each provide eight palmprint images for the right hand. Thus, each person provides 16 images and our database contains a total of 3,040 images from 190 different palms. The size of all the original palmprint images is 384×284 pixels with 75dpi resolution. Using the preprocessing approach in Zhang, Kong, You, and Wong (2003), the sub-images with a fixed size (128×128) are extracted from the original images. In order to reduce the computational cost, each sub-image is scaled to 64×64. We use these sub-images to represent the original palmprint images and to conduct our experiments. Figure 8.7a shows 10 image samples of one person captured at different time. The first five were collected first collections and second five on the next occasion, the major changes being in illumination and position, including shift and rotation. Similar to the kinds of changes encountered in facial expressions, the image may also be slightly affected by the way the hand is posed, shrunk or stretched. Figure 8.7b shows the corresponding processed images by frequency band selection. We also use the first five images of each person as the training samples and the remainder as the test samples. So, the numbers of training and test samples are 950 and 2,090. The related parameters in our approach for the palmprint database can be seen in Table 8.2. The experimental values of T_1 and T_2 are set as 2.0 and 0.5, respectively. The first 20 low-frequency bands are selected. After band selection, the total separability value of the training sample set is remarkably increased by 3.444 (=8.458-5.014). The total number of principal components is 210, which is also less than 2**c* (=2*190=380). So, we do not discard any components. And 181 discrimination vectors are obtained.

A comparison of the classification performance of all the methods is provided in Table 8.5. Our approach obtains the highest recognition rate. The maximum improvements in the recognition rate of our approach over those of eigenface, fisherface, DLDA and discriminant waveletface are 26.79%, 7.22%, 27.13% and 3.16%, respectively. The second-least number of features is acquired in our approach. We think that our approach makes a trade-off between obtaining the high recognition rate and reducing the dimension of feature space. It takes the third-least training time in all methods, and there is no significant difference in the time of the fastest method (DLDA) and our approach.

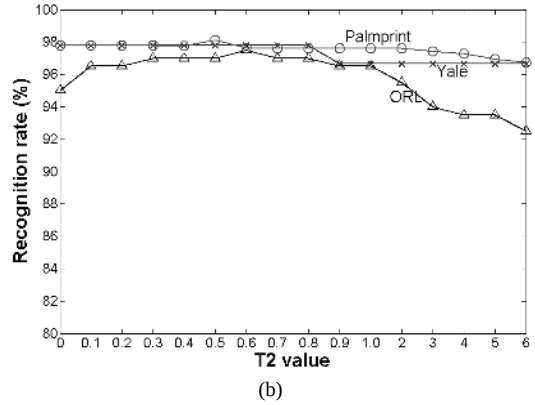
Analysis of Threshold Setting

We perform some analysis for setting the experimental values of T_1 and T_2. Figure 8.8 illustrates the recognition rates of three image databases, while the values of T_1 and T_2 are varied. From Figure 8.8a we find that with respect to T_1, the appropriate value ranges for Yale, ORL and palmprint databases are [0.5, 3.0], [0.8, 3.0] and [0.7, 5.0], respectively.

Figure 8.8. The recognition rates of our approach with different image data while (a) the value of T_1 is varied, and (b) the value of T_2 is varied



(a)



(b)

Table 8.6. An analysis of the mean values and the variances of the recognition rates in our approach when the value ranges of T_1 and T_2 are [0.8, 3.0] and [0.0, 2.0], respectively

Different thresholds	T_1			T_2		
	Different databases			Different databases		
	Yale	ORL	Palmprint	Yale	ORL	Palmprint
Mean recognition rates (%)	96.89	93.10	97.43	97.50	96.58	97.73
Variance	0.50	1.08	0.55	0.50	0.70	0.15

That is, in each range, all the recognition rates are near to the maximal rate. Hence, an appropriate range for both data is [0.8, 3.0]. From Figure 8.8b we find that with respect to T_2, the appropriate value ranges for Yale, ORL and palmprint databases are [0.0, 6.0], [0.0, 2.0] and [0.0, 3.0], respectively. Hence, an appropriate range for both data is [0.0, 2.0].

Table 8.6 shows an analysis of the mean values and the variances of the recognition rates where the value ranges of T_1 and T_2 are [0.8, 3.0] and [0.0, 2.0]. The variances are much smaller than the mean values, especially for T_2. That is, in these ranges the recognition effect of our approach is robust.

SUMMARY

A novel face and palmprint recognition approach based on DCT and linear discrimination technique is developed in this chapter. A 2Dseparability judgment is used to select appropriate DCT frequency bands with favorable linear separability. And, an improved fisherface method is then applied to extract linear discriminative features from the selected bands. The detailed analysis shows the theoretical advantages of our approach over other frequency-domain transform techniques and state-of-the-art linear discrimination methods. The practicality of our approach as an image recognition approach is well evidenced in the experimental results, where different image data — including two face databases and a palmprint database — are used. Our approach can significantly improve image recognition effect. In contrast with four conventional discrimination methods (eigenface, fisherface, DLDA and discriminant waveletface), it improves the average recognition rates of all data by 13.65%, 13.33%, 15.21% and 6.13%, respectively. Besides, this approach can reduce the dimension of feature space and cost little computing time.

REFERENCES

Chellappa, R., Wilson, C., & Sirohey, S. (1995). Human and machine recognition of faces: A survey. *Proceedings of the IEEE*, *83*(5), 705-740.

Chien, J. T., & Wu, C. C. (2002). Discriminant waveletfaces and nearest feature classifiers for face recognition. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, *24*(12), 1644-1649.

Dai, D. Q., & Yuen, P. C. (2003). Regularized discriminant analysis and its application to face recognition. *Pattern Recognition, 36*(3), 845-847.

Hafed, Z. M., & Levine, M. D. (2001). Face recognition using the discrete cosine transform. *International Journal Computer Vision*, *43*(3), 167-188.

Jing, X. Y., & Zhang, D. (2004). A face and palmprint recognition approach based on discriminant DCT feature extraction. *IEEE Transactions on Systems, Man, and Cybernetics, Part B, 34*(6), 2405-2415.

Jing, X. Y., Zhang, D., & Jin, Z. (2003). UODV: Improved algorithm and generalized theory. *Pattern Recognition, 36*(11), 2593-2602.

Jing, X. Y., Zhang, D., & Yao, Y. F. (2003). Improvements on the linear discrimination technique with application to face recognition. *Pattern Recognition Letters, 24*(15), 2695-2701.

Ko, J., & Byun, H. (2003). N-division output coding method applied to face recognition. *Pattern Recognition Letters, 24*(16), 3115-3123.

Lai, J. H., Yuen, P. C., & Feng, G.C. (2001). Face recognition using holistic Fourier invariant features. *Pattern Recognition*, *34*(1), 95-109.

Li, W., Zhang, D., & Xu, Z. (2002). Palmprint identification by Fourier transform. *International Journal Pattern Recognition and Artificial Intelligence*, *16*(4), 417-432.

Tian, Y., Tan, T. N., Wang, Y. H., & Fang, Y. C. (2003) Do singular values contain adequate information for face recognition? *Pattern Recognition*, *36*(3), 649-655.

Turk, M., & Pentland, A. (1991). eigenfaces for recognition. *International Journal Cognitive Neuroscience*, *3*(1), 71-86.

Wu, X., Zhang, D., & Wang, K. (2003). Fisherpalms based on palmprint recognition. *Pattern Recognition Letters, 24*(15), 2829-2938.

Yang, J., & Yang, J. Y. (2002). From image vector to matrix: A straightforward image projection technique: IMPCA vs. PCA. *Pattern Recognition, 35*(9), 1997-1999.

Yu, H., & Yang, J. (2001). A direct LDA algorithm for high-dimensional data with application to face recognition. *Pattern Recognition*, *34*(12), 2067-2070.

Zhang, D., Kong, W. K., You, J., & Wong, M. (2003). On-line palmprint identification. *IEEE Transactions on Pattern Analysis and Machine Intelligence, 25*(9), 1041-1050.

Chapter IX

Other Typical BID Improvements

ABSTRACT

In this chapter, we discuss some other typical BID improvements, including dual eigenspaces method (DEM) and post-processing on LDA-based method for automated face recognition. After the introduction, we describe DEM. Then, post-processing on LDA-based method is defined. Finally, we offer some brief conclusions.

INTRODUCTION

So far, there have been four BID technologies proposed in Part II, including improved UODV, CLDA, ILDA and discriminant DCT feature extraction. As other typical BID improvements, this chapter presents two effective schemes called DEM and post-processing on LDA-based method for automated face recognition.

Based on K-L transform, the dual eigenspaces are constructed by extracting algebraic features of training samples and applying them to face identification with a two-layer minimum distance classifier. Experimental results show that DEM is significantly better than the traditional eigenfaces method (TEM).

PCA- (see Chapter II) and LDA- (see Chapter III) based methods are state-of-art approaches to facial feature extraction. Recently, pre-processing approaches have been used to further improve recognition performance, but few investigations have been made into the use of post-processing techniques. Later in this chapter, we intend to explore the feasibility and effectiveness of the post-processing technique on LDA's discrimi-

nant vectors. In this chapter, we also propose a Gaussian filtering approach to post-process the discriminant vectors. The results of our experiments demonstrate that the post-processing technique can be used to improve recognition performance.

DUAL EIGENSPACES METHOD

Introduction to TEM

Automated face recognition is mainly applied to individual identification systems, such as criminal discrimination, authentication of ID cards and many security facilities (Chellappa, Wilson, & Sirohey, 1995). During the last 30 years, numerous algorithms based on geometrical features of face images have been developed. But they met great difficulty in accurately determining both positions and shapes of facial organs. Another sort of algorithms is to use algebraic features extracted by various orthogonal transforms. TEM uses principal components of an ensemble of face images and then completes the recognition procedure in an orthonormal "face space" (Turk & Pentland, 1991). However, its recognition rate is largely reduced when head posture, lighting conditions or facial expressions vary (Moghaddam & Pentland, 1994; Belhumeur, Hespanha, & Kriegman, 1997). To solve this problem, this chapter provides DEM to further analyze the features distribution in the "face space" and use coarse-to-fine matching strategy for face recognition. It is shown that this method is superior to TEM in recognition rate.

Algebraic Features Extraction

As the most optimal orthonormal expansion for image compression, K-L transform can also be used to feature extraction and pattern recognition (Oja, 1983). In TEM, the generating matrix of K-L transform is a total scatter matrix in Chapter III, and in order to achieve higher computational simplicity without loss of accuracy, a between-class scatter matrix is adopted as the generating matrix as mentioned in that chapter, too. And:

$$\mathbf{S}_b = \frac{1}{P} XX^T \tag{9.1}$$

where $X = [(m_1 - m), \ldots, (m_P - m)]$; m_i is the average image of the person's $_{i\text{th}}$ training samples; and P is the number of people in the training set.

It is evident that the eigenvectors of $\mathbf{S}_b$ can span an algebraic eigenspace and provide optimal approximation for those training samples in the sense of mean-square error. Given a face image, it can be projected onto these eigenvectors and represented in terms of a weight vector regarded as its algebraic features.

However, determining the eigenvectors of the matrix, $\mathbf{S}_b \in \Re^{N^2 \times N^2}$, is an intractable task. It can be solved by using SVD theorem (Oja, 1983).

First, the following matrix is formalized as:

$$R = \frac{1}{P} X^T X \in \Re^{P \times P} \tag{9.2}$$

Obviously, it is much easier to calculate both eigenvalues, $A = diag[\lambda_1, \ldots, \lambda_{P-1}]$, and orthonormal eigenvectors, $V = [v_1, \ldots, v_{P-1}]$, in this lower-dimensional matrix.

Then, the eigenvectors of $\mathbf{S}_b$ can be derived by SVD theorem:

$$U = XVA^{-\frac{1}{2}} \tag{9.3}$$

where $U = [u_1, \ldots, u_{P-1}]$ denotes the basis vectors, which span an algebraic subspace called unitary eigenspace of the training set.

Finally, we can obtain the following result:

$$C = U^T X = A^{\frac{1}{2}} V^T \tag{9.4}$$

where $C = [c_1, \ldots, c_P]$ is referred to the standard feature vectors of each person.

In TEM, face recognition is performed only in above unitary eigenspace. However, some eigenvectors might act primarily as "noise" for identification because they mainly capture the variations due to illumination and facial expressions. It results in the reduction in recognition rate of TEM. To further characterize the variations among each person's face and analyze different distributions of their weight vectors in the unitary eigenspace, our method is to construct new eigenspaces for each person by carrying out another K-L transform. For the i_{th} person, its generating matrix is selected as a within-class scatter matrix of all the weight vectors of its training samples:

$$W_i = \frac{1}{M_i} \sum_{j=1}^{M_i} (y_i^{(j)} - c_i)(y_i^{(j)} - c_i)^T, \qquad i = 1, \ldots, P \tag{9.5}$$

where $y_i^{(j)} = U^T(x_i^{(j)} - m)$ is defined as the weight vector of the i_{th} person's training sample $x_i^{(j)}$; and M_i is the number of person's images in the training set.

Note that the eigenvectors of each W_i are easily obtained. Here, those MCs are chosen to span each person's individual eigenspace, denoted by $\tilde{U}_i$ ($i = 1, \ldots, P$). In cooperation with the unitary eigenspace, the construction of dual eigenspaces has been completed.

Face Recognition Phase

A two-layer classifier is built in this phase. In the top layer, a common minimum-distance classifier is used in the unitary eigenspace. For a given input face image, f, its weight vector can be derived with a simple inner product operation:

$$y = U^T(f - m) \tag{9.6}$$

In this way, the coarse classification can be performed by the distance between and each person's standard feature vector, $c_i (i = 1, \cdots, P)$. Then, a few candidates who have the minimum distance are chosen for the finer classification.

In the bottom layer, the weight vector, y, is mapped separately onto each candidates' individual eigenspace to yield coordinate vectors:

$$\tilde{y}_i = \tilde{U}_i^T (y - c_i) \tag{9.7}$$

If $d_j = \min\{d_i : d_i = \|\tilde{y}_i\|\}$, the input image, f, can be recognized as the j_{th} person.

Experimental Results

The scheme shown above has been implemented on a Sun Sparc20 workstation. We first set up a database of about 250 face images. Eighteen Chinese male students had frontal photos taken under controlled conditions, but without any special restrictions to their posture. These images are different in lighting conditions, facial expressions, head orientation and distance to the camera.

In experiments, the number of each person's training samples varies from 2 to 12, while the remaining images constitute a test set. The recognition rates depicted in Figure 9.1 indicate that DEM is obviously better than TEM. For example, when six face images of each person are selected as training samples, there is a dramatic improvement in the recognition rate from 86.36% (TEM) to 94.63% (DEM). In particularly, when 12 images of each person are used as training samples, the recognition rate of DEM can be up to 97.93%. Considering the characteristics of the test images that contain the changes of head posture, facial expressions and illumination directions, it is obvious that our method is effective to these ambiguous images.

POST-PROCESSING ON LDA-BASED METHOD

Introduction

As an important issue in the face recognition system, facial feature extraction can be classified in two categories: geometric or structural methods, and holistic methods (Zhao, Chellappa, Phillips, & Rosenfeld, 2003). So far, holistic methods, which use the whole face region as the input, have been a major facial feature extraction approach and among various holistic methods, the state-of-art approaches are PCA- and LDA-based methods (Turk & Pentland, 1991; Belhumeur, Hespanha, & Kriegman, 1997).

Recently, pre-processing approaches have been introduced to further improve the recognition performance of PCA- and LDA-based methods. 2D-Gabor filters (Liu & Wechsler, 2002), edge detection (Yilmaz & Gokmen, 2001) and wavelet techniques (Chien & Wu, 2002) have been used for facial image pre-processing before the application of PCA or LDA. Most recently, Wu and Zhou (2002) proposed to apply the projection-combined version of the original image for PCA.

Unlike the pre-processing techniques, few works have dealt with the use of post-processing to improve recognition performance. Precious work has shown that LDA's discriminant vectors are very noisy and wiggly (Zhao, Chellappa, & Phillips, 1999). One general approach to address this problem is to add a penalty matrix to the within-class

covariance matrix (Dai & Yuen, 2003). Since the discriminant vector can be mapped into image, in this section we believe that appropriate image post-processing techniques can also be used to address this problem. To validate this view, we propose to use a Gaussian filtering approach to post-process discriminant vectors and carry out a series of experiments to test the effectiveness of the post-processing.

LDA-Based Facial Feature Extraction Methods

LDA is an effective feature extraction approach used in pattern recognition (Fukunaga, 1990). It finds the set of optimal vectors that map features of the original data into a low-dimensional feature space in such a way that the ratio of the between-class scatter to the within-class scatter is maximized. When LDA is applied to facial feature extraction, the recognition performance would be degraded due to the singularity of the within-class scatter matrix $\mathbf{S}_w$. To date, a considerable amount of research has been carried out on this problem.

Although the aim of this chapter is to study the effectiveness of the post-processing, it is not possible to test the effect of the post-processing on all the LDA-based methods. Consequently, we reviewed only two representative approaches, fisherfaces (Belhumeur, Hespanha, & Kriegman, 1997) and D-LDA (Yu & Yang, 2001).

Fisherfaces

As mentioned in Chapter IV, the fisherfaces method is essentially LDA in a PCA subspace. When using fisherfaces, each image is mapped into a high-dimensional vector by concatenating together the rows of the original image. Chapter IV has introduced this method in detail.

D-LDA

D-LDA (Direct-LDA) is another representative LDA-based method that has been widely investigated in facial feature extraction (Yu & Yang, 2001; Lu, Plataniotis, & Venetsanopoulos, 2003). The key idea of the D-LDA method is to find a projection that simultaneously diagonalizes both the between-class scatter matrix $\mathbf{S}_b$ and the within-class scatter matrix $\mathbf{S}_w$. To diagonalize $\mathbf{S}_b$, the D-LDA method first finds the matrix V with constraint,

$$V^T S_b V = \Lambda \tag{9.8}$$

where $V^T V = I$ and Λ is a diagonal matrix sorted in decreasing order. Let Y denote the first m columns of V, and calculate $D_b = Y^T \mathbf{S}_b Y$. Then we calculate $Z = YD_b^{-1/2}$, and diagonalize $Z^T \mathbf{S}_w Z$ by calculating matrix U and diagonal matrix D_w with the constraint,

$$U^T (Z^T \mathbf{S}_w Z) U = D_w \tag{9.9}$$

Finally, the D-LDA projection T_{dlda} is defined as:

$$T_{dlda} = D_w^{-1/2} U^T Z^T \tag{9.10}$$

Post-Processing on Discriminant Vectors

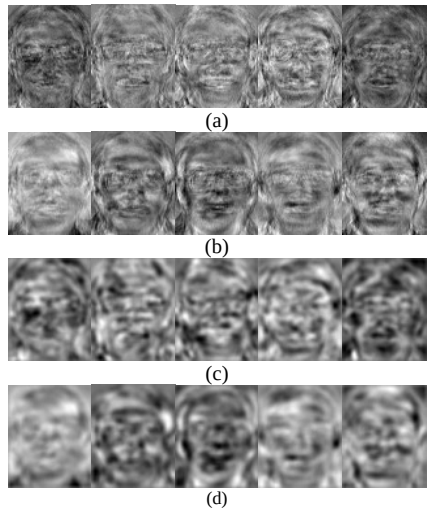
Why Post-Processing

Using the ORL database, we give an intuitional illustration of fisherfaces and D-LDA's discriminant vectors. The ORL database contains 400 facial images with 10 images per individual. Ten images of one person are shown in Figure 9.1. The images in the ORL database vary in sampling time, light conditions, facial expressions, facial details (glasses/no glasses), scale and tilt. Moreover, all the images are taken against a dark homogeneous background, with the person in an upright frontal position. The tolerance for some tilting and rotation is up to about 20°. These gray images are 112×92 (Olivetti, n.d.). In this experiment, we choose the first five images of each individual for training, and thus, obtained a training set consisting of 200 facial images. Then fisherfaces and D-LDA were used to calculate the discriminant vectors.

Figure 9.2a shows a set of fisherfaces' discriminant vectors obtained from the training set and Figure 9.2b shows five discriminant vectors obtained using D-LDA. It

Figure 9.1. Ten images of one person from the ORL database

Figure 9.2. An illustration of different LDA-based methods' discriminant vectors



(a)Fisherfaces, (b) D-LDA, (c) post-processed fisherfaces, (d) post-processed D-LDA

is observed that the discriminant vectors in Figure 9.2 were not smooth. Since a facial image is a 2D smooth surface, it is reasonable to consider that better recognition performance would be obtained by further improving the smoothness of the discriminant vectors using the post-processing techniques.

Post-Processing Algorithm

When the discriminant vectors obtained using fisherfaces or D-LDA were reshaped into images, we observed that the discriminant vectors were not smooth. We expect that this problem can be solved by introducing a post-processing step on discriminant vectors.

We propose to post-process the discriminant vectors using a 2D-Gaussian filter. A Gaussian filter is an ideal filter in the sense that it reduces the magnitude of high spatial frequency in an image and has been widely applied in image smoothing (Pratt, 1991). The 2D-Gaussian filter could be used to blur the discriminant images and remove noise. A 2D Gaussian function is defined as:

$$G(x,y) = \frac{1}{2\pi\sigma^2} e^{-(x^2+y^2)/2\sigma^2} \tag{9.11}$$

where $\sigma > 0$, which is the standard deviation. First, we define a 2D-Gaussian model M according to the standard deviation σ. Once the standard deviation is determined, the window size $[w, w]$ can be determined as $w = 4\sim6\times\sigma$, and the Gaussian model M is defined as the $w\times w$ truncation from the Gaussian kernel $G(x,y)$. Then each discriminant vector v_i is mapped into its corresponding image I_i by de-concatenating it into rows of I_i. The Gaussian filter M is used to smooth discriminant image I_i:

$$I_i'(x,y) = I(x,y) \oplus M(x,y) \tag{9.12}$$

$I_i'(x,y)$ is transformed into a high-dimensional vector v_i' by concatenating its rows together. Finally, we could obtain the post-processed LDA projection $T_{pLDA} = [v_1', v_2', \ldots, v_m]$, where m is the number of discriminant vectors.

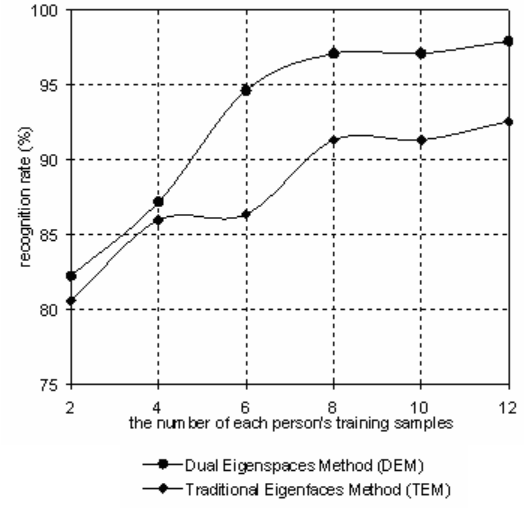
Other image smoothing techniques, such as wavelet and nonlinear diffusion filtering, can also be used to post-process the discriminant vectors. Since the aim of this section is to investigate the feasibility of post-processing in improving recognition performance, we adopt the simple Gaussian filtering method.

Experimental Results and Discussions

In this section, we use two popular face databases, the ORL and the FERET database, to evaluate the effectiveness of the proposed post-processing approach. Since the aim is to evaluate the feature extraction approaches, a simple nearest-neighbor classifier is adopted.

Using the ORL database, we give an intuitional demonstration of the ability of Gaussian filter in smoothing the discriminant vectors. Figure 9.2c and Figure 9.4d show the first five post-processed discriminant vectors of fisherfaces and D-LDA. Compared with Figure 9.2a and Figure 9.2b, we can observe that the proposed method improved the smoothness of discriminant images.

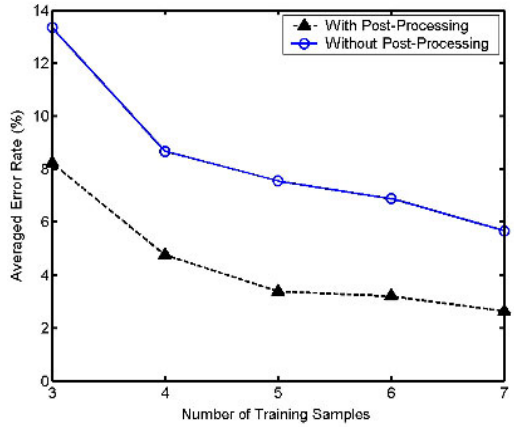
Figure 9.3. Comparision of performances between DEM and TEM



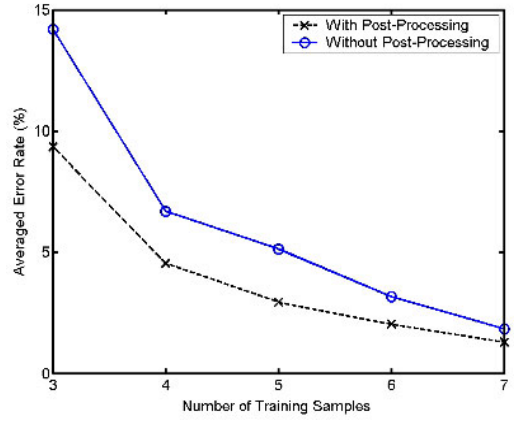
For the ORL database, we randomly select n samples per person for training, resulting in a training set of $40\times n$ images and a testing set of $40\times(10-n)$ images with no overlapping between the two sets. To reduce the variation of recognition results, the averaged error rate (AER) is adopted by calculating the mean error rate over 20 runs. We set the window of the Gaussian filter as $[w, w] = [11, 11]$ and the variance $\sigma = 25$. Figure 9.4a shows the AER obtained using fisherfaces and post-processed fisherfaces with different number of training samples n. Figure 9.4b compares the AER obtained using D-LDA and post-processed D-LDA. It is simple to see that 2D-Gaussian filter can improve the recognition performance. Table 9.1 compares the AER obtained with and without post-processed approaches when the number of training samples is five. The AER of post-processed fisherfaces is 3.38, much less than that obtained by classical fisherfaces. The AER of post-processed D-LDA is 2.93, while the AER obtained by D-LDA is 5.12. We also compared the proposed post-processed LDA with some recently reported results using the ORL database, as listed in Table 9.2 (Dai & Yuen, 2003; Lu, Plataniotis, & Venetsanopoulos, 2003; Liu, Wang, Li, & Tan, 2004; Zheng, Zhao, & Zhao, 2004; Zheng, Zou, & Zhao, 2004). What is to be noted is that all the error rates are obtained with the number of training samples $n = 5$. It can be observed that post-processed LDA-based method is very effective and competitive in facial feature extraction.

The FERET face image database is the second database used to test the post-processing method (Phillips, Moon, Rizvi, & Rauss, 2000). We choose a subset of the FERET database consisting of 1,400 images corresponding to 200 individuals (each individual has seven images, including a front image and its variations in facial expression, illumination, $\pm 15\sigma°$ and ±30° pose). The facial portion of each original image was cropped to the size of 80×80 and pre-processed by histogram equalization. In our experiments, we randomly selected three images of each subject for training, resulting in a training set of 600 images and a testing set of 800 images. Figure 9.5 illustrates some cropped images of one person.

Figure 9.4. Comparison of the averaged error rates with and without post-processing for different LDA-based methods



(a)



(b)

(a) Fisherfaces, (b) D-LDA

Table 9.1.AER obtained using the ORL database with and without post-processing

Methods	fisherfaces	D-LDA
Without Post-Processing	7.55	5.12
With Post-Processing	3.38	2.93

Previous work on the FERET database indicates that the dimensionality of PCA subspace has an important effect on fisherfaces' recognition (Liu & Wechsler, 1998). Thus, we investigate the recognition performance of fisherfaces and post-processed fisherfaces with different numbers of PCs. The number of discriminant vectors is set as 20 according to Yang, Yang, and Frangi (2003). The averaged recognition rate is used by

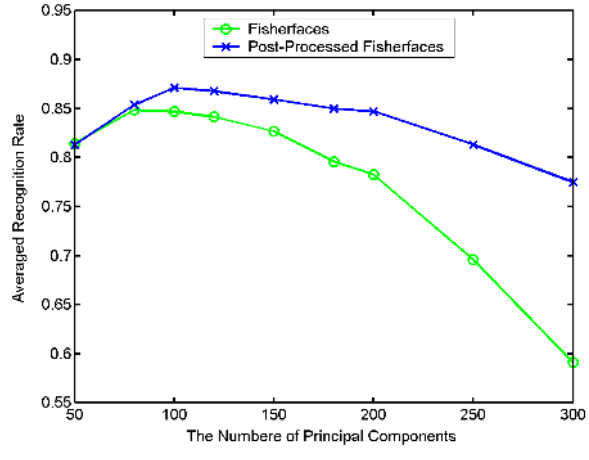
Table 9.2. Other results recently reported on the ORL database

Methods	Error Rate (%)	Year
DF-LDA [12]	4.2	2003
RDA [9]	4.75	2003
NKFDA [15]	4.9	2004
ELDA [16]	4.15	2004
GSLDA [17]	4.02	2004

Figure 9.5. Some cropped images of one person in FERET database

calculating the mean across 10 tests. The window of the Gaussian filter is set as [h, w] = [9, 9] and the variance is set as $\sigma = 1.5$. Figure 9.5 shows the averaged recognition rates obtained using fisherfaces and post-processed fisherfaces with different numbers of PCs. The highest averaged recognition rate of post-processed fisherfaces is 87.12%, and that of fisherfaces is 84.87%. It is observed that post-processing has little improvement on fisherfaces' recognition rate when the number of PCs is less than 80. When the number of PCs is greater than 100, post-processed fisherfaces is distinctly superior to fisherfaces in recognition performance. Besides, the dimensionality of PCA subspace has much less effect on the performance of post-processed fisherfaces, whereas the averaged recognition rate of fisherfaces varied greatly with the number of PCs.

Figure 9.6. Comparison of the averaged recognition rates obtained by fisherfaces and post-processed fisherfaces with different number of PCs



SUMMARY

In this chapter, a novel DEM algorithm is presented and applied to human face recognition at first, where the dual eigenspaces are constructed by extracting the algebraic features of face images. Face recognition is performed by the coarse-to-fine matching strategy with a two-layer minimum-distance classifier. The experimental results show that DEM can offer a superior performance than TEM. It is also demonstrated that DEM has insensitivity to the face posture, expressions and illumination conditions to a certain extent.

Other than the using of pre-processing techniques, this chapter also shows that post-processing can be used to improve the performance of LDA-based methods. In this section, we proposed a 2D-Gaussian filter to post-process discriminant vectors. Experimental results indicate that the post-processing technique can be used to improve LDA's recognition performance. While using the ORL with five training samples per individual, the AER obtained by post-processed fisherfaces is 3.38, and the AER of post-processed D-LDA is 2.93. A large set of faces, a subset of the FERET database consisting of 1,400 images of 200 individuals, is also used to test the post-processing approach, and post-processed fisherfaces can achieve a recognition rate of 87.12% on this subset.

Some problems worthy of further study still remain. Further work should include the automatic determination of the window and variance of the Gaussian filter; the investigation of other possible post-processing techniques, such as wavelets; exploration of the effect of post-processing on other LDA-based methods; and the application of post-processing in other biometrics, such as palmprint and gait biometrics.

REFERENCES

Belhumeur, P. N., Hespanha, J. P., & Kriegman, D. J. (1997). Eigenfaces vs. fisherfaces: Recognition using class specific linear projection. *IEEE Transaction on Pattern Analysis and Machine Intelligence*, 19, 711-720.

Chellappa, R., Wilson, C., & Sirohey, S. (1995). Human and machine recognition of faces: A survey. *Proceedings of the IEEE*, *83*(5), 705-740.

Chien, J. T., & Wu, C. C. (2002). Discriminant waveletfaces and nearest feature classifiers for face recognition. *IEEE Transaction on Pattern Analysis and Machine Intelligence*, *24*, 1644-1649.

Dai, D., & Yuen, P. C. (2003). Regularized discriminant analysis and its application to face recognition. *Pattern Recognition, 36*, 845-847.

Fukunaga, K. (1990). *Introduction to statistical pattern recognition* (2nd ed.). Academic Press.

Liu, C., & Wechsler, H. (1998). Enhanced Fisher linear discriminant models for face recognition. *The 14th International Conference on Pattern Recognition (ICPR'98)*, 1368-1372.

Liu, C., & Wechsler, H. (2002). Gabor feature based classification using the enhanced Fisher linear discriminant model for face recognition. *IEEE Transaction on Image Processing*, *11*, 467-476.

Liu, W., Wang, Y., Li, S. Z., & Tan, T. (2004). Null space approach of Fisher discriminant analysis for face recognition. In D. Maltoni & A. K. Jain (Eds.), *BioAW 2004, Lecture Notes in Computer Science* (pp. 32-44). Springer.

Lu, J., Plataniotis, K. N., & Venetsanopoulos, A.N. (2003). Face recognition using LDA-based algorithms. *IEEE Transaction on Neural Networks*, *14*, 195-200.

Moghaddam, B., & Pentland, A. (1994). Face recognition using view-based and modular eigenspaces. *Proceedings of the SPIE*, *2277* (pp. 12-21).

Oja, E. (1983). *Subspace method of pattern recognition*. UK: Research Studies Press.

ORL Face Database. (2002). AT&T Research Laboratories. The ORL Database of Faces. Cambridge. Retrieved from www.uk.research.att.com/facedatabase.html

Phillips, P. J., Moon, H., Rizvi, S. A., & Rauss, P. J. (2000). The FERET evaluation methodology for face-recognition algorithm. *IEEE Transaction on Pattern Analysis and Machine Intelligence, 22*, 1090-1104.

Pratt, W. K. (1991). *Digital image processing* (second edition). New York: Wiley.

Turk, M., & Pentland, A. (1991). Eigenfaces for recognition. *Journal of Cognitive Neuroscience*, *3*(1), 71-86.

Wu, J., & Zhou, Z-H. (2002). Face recognition with one training image per person. *Pattern Recognition Letters*, *23*,1711-1719.

Yang, J., Yang, J-Y., & Frangi, A. F. (2003). Combined fisherfaces framework. *Image and Vision Computing*, *21*, 1037-1044.

Yilmaz, A., & Gokmen, M. (2001). Eigenhill vs. eigenface and eigen edge. *Pattern Recognition*, *34*, 181-184.

Yu, H., & Yang, J. (2001). A direct LDA algorithm for high-dimensional data with application to face recognition. *Pattern Recognition, 34*, 2067-2070.

Zhao, W., Chellappa, R., & Phillips, P. J. (1999). Subspace linear discriminant analysis for face recognition. *Tech Report CAR-TR-914.* Center for Automation Research, University of Maryland.

Zhao, W., Chellappa, R., Phillips, P. J., & Rosenfeld, A. (2003). Face recognition: A lit Zhao, L., & Zou, C. (2004). An efficient algorithm to solve the small sample size problem for LDA. *Pattern Recognition, 37*, 1077-1079.

Zheng, W., Zou, C., & Zhao, L. (2004). Real-time face recognition using Gram-Schmidt orthogonalization for LDA. *The 17th International Conference on Pattern Recognition (ICPR'04)*, 403-406.

Section III

Advanced BID Technologies

Chapter X

Complete Kernel Fisher Discriminant Analysis

ABSTRACT

This chapter introduces a complete kernel Fisher discriminant analysis (KFD) that is a useful statistical technique applied to biometric application. We first describe theoretical perspective of KPCA. Then, a new KFD algorithm framework, KPCA plus LDA, is given. Afterwards, we discuss the complete KFD algorithm. Finally, the experimental results and chapter summary are given.

INTRODUCTION

Over the last few years, kernel-based learning machines — that is, SVMs, KPCA, and kernel Fisher discriminant analysis (KFD) — have aroused considerable interest in the fields of pattern recognition and machine learning (Müller, Mika, Rätsch, Tsuda, & Schölkopf, 2001). KPCA was originally developed by Schölkopf (Schölkopf, Smola, & Müller, 1998) while KFD was first proposed by Mika (Mika, Rätsch, Weston, Schölkopf, & Müller, 1999). Subsequent research saw the development of a series of KFD algorithms (Baudat & Anouar, 2000; Roth & Steinhage, 2000; Mika, Rätsch, & Weston, 2003; Yang, 2002; Lu, Plataniotis, & Venetsanopoulos, 2003; Xu, Zhang, & Li, 2001; Billings & Lee, 2002; Cawley & Talbot, 2003). The KFD algorithms developed by Mika, Billings, and Cawley (Mika, Rätsch, & Weston, 2003; Billings & Lee, 2002; Cawley & Talbot, 2003) are formulated for two classes, while those of Baudat, Roth, and Yang (Baudat & Anouar, 2000; Roth & Steinhage, 2000; Yang, 2002) are formulated for multiple classes. Because of its ability to extract the most discriminatory non-linear features, KFD has been found to be very effective in many real-world applications.

KFD, however, always encounters the *ill-posed* problem in its real-world applications (Mika, Rätsch, & Weston, 2003; Tikhonov & Arsenin, 1991). A number of regularization techniques that might alleviate this problem have been suggested. Mika (Mika, Rätsch, & Weston, 2003; Mika, Rätsch, & Weston, 1999) used the technique of making the inner product matrix non-singular by adding a scalar matrix. Baudat (Baudat & Anouar, 2000) employed the QR decomposition technique to avoid the singularity by removing the zero eigenvalues. Yang (2002) exploited the PCA plus LDA technique adopted in fisherface (Belhumeur, Hespanha, & Kriegman, 1997) to deal with the problem. Unfortunately, all of these methods discard the discriminant information contained in the null space of the within-class covariance matrix, yet this discriminant information is very effective for the SSS problem (Liu & Yang, 1992; Chen, Liao, & Ko, 2000; Yu & Yang, 2001; Yang & Yang, 2001; Yang & Yang, 2003). Lu (Lu, Plataniotis, & Venetsanopoulos, 2003) has taken this issue into account and presented kernel direct discriminant analysis (KDDA) by generalization of DLDA (Yu & Yang, 2001).

In real-world applications, particularly in image recognition, there are a lot of SSS problems where the number of training samples is less than the dimension of feature vectors. For kernel-based methods, due to the implicit high-dimensional nonlinear mapping determined by kernel, almost all problems are turned into SSS problems in *feature space* (actually, all problems will become SSS problems as long as the dimension of nonlinear mapping is large enough). Actually, KPCA and KFD are inherently in tune with the linear feature extraction techniques like PCA and Fisher LDA for SSS problems. Eigenface (Turk & Pentland, 1991) and fisherface (Belhumeur, Hespanha, & Kriegman, 1997) typically are PCA and LDA techniques for SSS problems. They are essentially carried out in the space spanned by all M training samples by virtue of the SVD technique. Like eigenface and fisherface, KPCA and KFD are also performed in all training samples' spanned space. This inherent similarity makes it possible to improve KFD using the state-of-the-art LDA techniques.

LDA has been well studied and widely applied to SSS problems in recent years. Many LDA algorithms have been proposed. The most famous method is fisherface, which is based on a two-phase framework of PCA plus LDA. The effectiveness of this framework in image recognition has been broadly demonstrated. Recently, the theoretical foundation for this framework has been laid (Yang & Yang, 2003). Besides, many researchers have dedicated to search for more effective discriminant subspaces. A significant result is the finding that there exists crucial discriminative information in the null space of the within-class scatter matrix (Liu & Yang, 1992; Chen, Liao, & Ko, 2000; Yu & Yang, 2001; Yang & Yang, 2001, 2003). In this chapter, we call this kind of discriminative information *irregular* discriminant information, in contrast with *regular* discriminant information outside of the null space.

KFD would be likely to benefit in two ways from the state-of-the-art LDA techniques. One is the adoption of a more concise algorithm framework, and the other is that it would allow the use of *irregular* discriminant information. This chapter seeks to improve KFD in these ways: first of all, by developing a new KFD framework, KPCA plus LDA, based on a rigorous theoretical derivation in Hilbert space. Then, a complete KFD algorithm (CKFD) is proposed based on the framework. Unlike current KLD algorithms, CKFD can take advantage of two kinds of discriminant information: *regular and irregular*. Finally, CKFD was used in face recognition and handwritten numeral recognition. The experimental results are encouraging.

The remainder of this chapter is organized as follows: First, a theoretical perspective of KPCA is given. Then, a two-phase KFD framework, KPCA plus LDA, is developed, and a complete KFD algorithm (CKFD) is proposed. We perform the experiments on the FERET face database, whereby the proposed algorithm is evaluated and compared to other methods. Finally, a conclusion and discussion is offered.

THEORETICAL PERSPECTIVE OF KPCA

For a given nonlinear mapping Φ, the *input data space* $\mathbb{R}^n$ can be mapped into the *feature space H*:

$$\Phi : \mathbb{R}^n \to \mathcal{H} \qquad (10.1)$$
$$x \mapsto \Phi(x)$$

As a result, a pattern in the original *input space* $\mathbb{R}^n$ is mapped into a potentially much higher-dimensional feature vector in the *feature space* $\mathcal{H}$. Since the *feature space H* is possibly infinite-dimensional and the orthogonality needs to be characterized in such a space, it is reasonable to view $\mathcal{H}$ as a Hilbert space. In this chapter, $\mathcal{H}$ is always regarded as a Hilbert space.

An initial motivation of KPCA (or KFD) is to perform PCA (or LDA) in the *feature space* $\mathcal{H}$. However, it is difficult to do so directly because it is computationally very intensive to compute the dot products in a high-dimensional *feature space.* Fortunately, kernel techniques can be introduced to avoid this difficulty. The algorithm can be actually implemented in the *input space* by virtue of kernel tricks. The explicit mapping process is not required at all. Now, let us describe KPCA as follows.

Given a set of M training samples $\mathbf{x}_1, \mathbf{x}_2, \ldots, \mathbf{x}_M$ in $\mathbb{R}^n$, the *covariance operator* on the *feature space* $\mathcal{H}$ can be constructed by:

$$\mathbf{S}_t^{\Phi} = \frac{1}{M}\sum_{j=1}^{M}\left(\Phi(\mathbf{x}_j) - \mathbf{m}_0^{\Phi}\right)\left(\Phi(\mathbf{x}_j) - \mathbf{m}_0^{\Phi}\right)^T \qquad (10.2)$$

where $\mathbf{m}_0^{\Phi} = \frac{1}{M}\sum_{j=1}^{M}\Phi(\mathbf{x}_j)$. In a finite-dimensional Hilbert space, this operator is generally called covariance matrix. The covariance operator satisfies the following properties:

Lemma 10.1. (Yang, Zhang, Yang, Jin, & Frangi, 2005) $\mathbf{S}_t^{\Phi}$ is a (1) bounded operator, (2) compact operator, (3) positive operator and (4) self-adjoint (symmetric) operator on Hilbert space $\mathcal{H}$.

Since every eigenvalue of a positive operator is non-negative in a Hilbert space (Rudin, 1973), from Lemma 1, it follows that all non-zero eigenvalues of $\mathbf{S}_t^{\Phi}$ are positive. It is these positive eigenvalues that are of interest to us. Schölkopf, Smola, and Müller (1998) have suggested the following way to find them.

It is easy to show that every eigenvector of $\mathbf{S}_t^{\Phi}$, β, can be linearly expanded by:

$$\beta = \sum_{i=1}^{M} a_i \Phi(\mathbf{x}_i) \tag{10.3}$$

To obtain the expansion coefficients, let us denote $\mathbf{Q} = [\Phi(\mathbf{x}_1), \ldots, \Phi(\mathbf{x}_M)]$ and form an $M \times M$ Gram matrix $\tilde{\mathbf{R}} = \mathbf{Q}^{\mathrm{T}}\mathbf{Q}$, whose elements can be determined by virtue of kernel tricks:

$$\tilde{\mathbf{R}}_{ij} = \Phi(\mathbf{x}_i)^{\mathrm{T}} \Phi(\mathbf{x}_j) = \left(\Phi(\mathbf{x}_i) \cdot \Phi(\mathbf{x}_j)\right) = \mathrm{k}(\mathbf{x}_i, \mathbf{y}_j) \tag{10.4}$$

Centralize $\tilde{\mathbf{R}}$ by:

$$\mathbf{R} = \tilde{\mathbf{R}} - \mathbf{1}_M \tilde{\mathbf{R}} - \tilde{\mathbf{R}}\, \mathbf{1}_M + \mathbf{1}_M \tilde{\mathbf{R}}\, \mathbf{1}_M ,$$

where the matrix $\mathbf{1}_M = (1/M)_{M \times M}$ (10.5)

Calculate the orthonormal eigenvectors $\gamma_1, \gamma_2, \ldots, \gamma_m$ of $\mathbf{R}$ corresponding to m largest positive eigenvlaues $\lambda_1 \geq \lambda_2 \geq \ldots \geq \lambda_m$. The orthonormal eigenvectors $\boldsymbol{\beta}_1, \boldsymbol{\beta}_2, \ldots, \boldsymbol{\beta}_m$ of $\mathbf{S}_t^{\Phi}$ corresponding to m largest positive eigenvlaues $\lambda_1, \lambda_2, \ldots, \lambda_m$ then are:

$$\beta_j = \frac{1}{\sqrt{\lambda_j}} \mathbf{Q}\, \gamma_j , \quad j = 1, \ldots, m \tag{10.6}$$

After the projection of the mapped sample $\Phi(\mathbf{x})$ onto the eigenvector system $\boldsymbol{\beta}_1, \boldsymbol{\beta}_2, \ldots, \boldsymbol{\beta}_m$, we can obtain the KPCA-transformed feature vector $y = (y_1, y_2, \ldots, y_m)^{\mathrm{T}}$ by:

$$\mathrm{y} = \mathbf{P}^{\mathrm{T}}\, \Phi(\mathbf{x}), \text{ where } \mathbf{P} = (\boldsymbol{\beta}_1, \boldsymbol{\beta}_2, \ldots, \boldsymbol{\beta}_m) \tag{10.7}$$

Specifically, the j_{th} KPCA feature (component) y_j is obtained by:

$$y_j = \beta_j^{\mathrm{T}} \Phi(\mathbf{x}) = \frac{1}{\sqrt{\lambda_j}} \gamma_j^{\mathrm{T}} \mathbf{Q}^{\mathrm{T}} \Phi(\mathbf{x})$$

$$= \frac{1}{\sqrt{\lambda_j}} \gamma_j^{\mathrm{T}} \left[\mathrm{k}(\mathbf{x}_1, \mathbf{x}), \mathrm{k}(\mathbf{x}_2, \mathbf{x}), \cdots, \mathrm{k}(\mathbf{x}_M, \mathbf{x})\right]^{\mathrm{T}} \tag{10.8}$$

A NEW KFD ALGORITHM FRAMEWORK: KPCA PLUS LDA

In this section, we will build a rigorous theoretical framework for KFD. This framework is important because it provides a solid theoretical foundation for our two-phased KFD algorithm that will be presented later. That is, the presented two-phased KFD algorithm is not empirically based but theoretically based.

To provide more theoretical insights into KFD, we would like to examine the problems in a whole Hilbert space rather than in the space spanned by training samples. Here, an infinite-dimensional Hilbert space is preferred because any proposition that holds in an infinite-dimensional Hilbert space will hold in a finite-dimensional Hilbert space (but the reverse might be not true). So, in this section, we will discuss the problems in an infinite-dimensional Hilbert space.

Fundamentals

Suppose there are c known pattern classes. The between-class scatter operator $\mathbf{S}_b^{\Phi}$ and the within-class scatter operator $\mathbf{S}_w^{\Phi}$ in the *feature space* $\mathcal{H}$ are defined below:

$$\mathbf{S}_b^{\Phi} = \frac{1}{M}\sum_{i=1}^{c} l_i \left(\mathbf{m}_i^{\Phi} - \mathbf{m}_0^{\Phi}\right)\left(\mathbf{m}_i^{\Phi} - \mathbf{m}_0^{\Phi}\right)^{\mathrm{T}} \tag{10.9}$$

$$\mathbf{S}_w^{\Phi} = \frac{1}{M}\sum_{i=1}^{c}\sum_{j=1}^{l_i}\left(\Phi(\mathbf{x}_{ij}) - \mathbf{m}_i^{\Phi}\right)\left(\Phi(\mathbf{x}_{ij}) - \mathbf{m}_i^{\Phi}\right)^{\mathrm{T}} \tag{10.10}$$

where $\mathbf{x}_{ij}$ denotes the jth training sample in class i; l_i is the number of training samples in class i; $\mathbf{m}_i^{\Phi}$ is the mean of the mapped training samples in class i; and $\mathbf{m}_0^{\Phi}$ is the mean across all mapped training samples.

From the above definitions, we have $\mathbf{S}_t^{\Phi} = \mathbf{S}_b^{\Phi} + \mathbf{S}_w^{\Phi}$. Following along with the proof of Lemma 1 (Yang, Zhang, Yang, Jin, & Frangi, 2005), it is easy to prove that the two operators satisfy the following properties:

Lemma 10.2. $\mathbf{S}_b^{\Phi}$ and $\mathbf{S}_w^{\Phi}$ are both (1) bounded operators, (2) compact operators, (3) self-adjoint (symmetric) operators and (4) positive operators on Hilbert space $\mathcal{H}$.

Since $\mathbf{S}_b^{\Phi}$ is self-adjoint (symmetric) operator in Hilbert space $\mathcal{H}$, the inner product between $\boldsymbol{\varphi}$ and $\mathbf{S}_b^{\Phi}\boldsymbol{\varphi}$ satisfies $\left\langle \varphi, \mathbf{S}_b^{\Phi}\varphi \right\rangle = \left\langle \mathbf{S}_b^{\Phi}\varphi, \varphi \right\rangle$. So, we can write it as $\left\langle \varphi, \mathbf{S}_b^{\Phi}\varphi \right\rangle \triangleq \varphi^T \mathbf{S}_b^{\Phi}\varphi$. Note that if $\mathbf{S}_b^{\Phi}$ is not self-adjoint, this denotation is meaningless. Since $\mathbf{S}_b^{\Phi}$ is also a positive operator, we have $\varphi^{\mathrm{T}}\mathbf{S}_b^{\Phi}\varphi \geq 0$. Similarly, we have $\left\langle \varphi, \mathbf{S}_{\mathrm{w}}^{\Phi}\varphi \right\rangle = \left\langle \mathbf{S}_{\mathrm{w}}^{\Phi}\varphi, \varphi \right\rangle \triangleq \varphi^{\mathrm{T}}\mathbf{S}_{\mathrm{w}}^{\Phi}\varphi \geq 0$. Thus, in Hilbert space $\mathcal{H}$, the Fisher criterion function can be defined by:

$$J^{\Phi}(\varphi) = \frac{\varphi^{\mathrm{T}}\mathbf{S}_b^{\Phi}\varphi}{\varphi^{\mathrm{T}}\mathbf{S}_w^{\Phi}\varphi}\ ,\ \boldsymbol{\varphi} \neq \mathbf{0} \tag{10.11}$$

If the within-class scatter operator $\mathbf{S}_w^{\Phi}$ is invertible, $\varphi^{\mathrm{T}}\mathbf{S}_w^{\Phi}\varphi > 0$ always holds for every non-zero vector $\boldsymbol{\varphi}$. In such a case, the Fisher criterion can be directly employed to extract a set of optimal discriminant vectors (projection axes) using the standard LDA algorithm. Its physical meaning is that after the projection of samples onto these axes, the ratio of the between-class scatter to the within-class scatter is maximized.

However, in a high-dimensional (even infinite-dimensional) *feature space H* , it is almost impossible to make $\mathbf{S}_w^{\Phi}$ invertible because of the limited amount of training samples in real-world applications. That is, there always exist vectors satisfying $\varphi^{\mathrm{T}}\mathbf{S}_w^{\Phi}\varphi = 0$ (actually, these vectors are from the null space of $\mathbf{S}_w^{\Phi}$). These vectors turn out to be very effective if they satisfy $\varphi^{\mathrm{T}}\mathbf{S}_b^{\Phi}\varphi > 0$ at the same time (Chen, Liao, & Ko, 2000; Yang & Yang, 2001, 2003). This is because the positive between-class scatter makes the data become well separable when the within-class scatter is zero. In such a case, the Fisher criterion degenerates into the following between-class scatter criterion:

$$J_b^{\Phi}(\varphi) = \varphi^{\mathrm{T}}\mathbf{S}_b^{\Phi}\varphi\ ,(\|\varphi\|=1) \tag{10.12}$$

As a special case of the Fisher criterion, the criterion given in Equation 10.12 is very intuitive, since it is reasonable to use the between-class scatter to measure the discriminatory ability of a projection axis when the within-class scatter is zero.

In this chapter, we will use the between-class scatter criterion defined in Equation 10.12 to derive the *irregular discriminant vectors* from $\mathbf{null}(\mathbf{S}_w^{\Phi})$ (i.e., the null space of $\mathbf{S}_w^{\Phi}$) while using the standard Fisher criterion defined in Equation 10.11 to derive the *regular discriminant vectors* from the complementary set $H - \mathbf{null}(\mathbf{S}_w^{\Phi})$.

Strategy for Finding Fisher Optimal Discriminant Vectors in Feature Space

Now, a problem is how to find the two kinds of Fisher optimal discriminant vectors in *feature space* $\mathcal{H}$. Since $\mathcal{H}$ is very large (high- or infinite-dimensional), it is computationally too intensive or even infeasible to calculate the optimal discriminant vectors directly. To avoid this difficulty, the present KFD algorithms all formulate the problem in the space spanned by the mapped training samples. The technique is feasible when the *irregular* case is disregarded, but the problem becomes more complicated when the *irregular* discriminant information is taken into account, since the *irregular* discriminant vectors exist in the null space of $\mathbf{S}_w^{\Phi}$. Because the null space of $\mathbf{S}_w^{\Phi}$ is possibly infinite-dimensional, the existing techniques for dealing with the singularity of LDA (Chen, Liao, & Ko, 2000; Yang & Yang, 2003) are inapplicable, since they are limited to a finite-dimensional space in theory.

In this section, we will examine the problem in an infinite-dimensional Hilbert space and try to find a way to solve it. Our strategy is to reduce the *feasible solution space*

(search space) where two kinds discriminant vectors might hide. It should be stressed that we would not like to lose any effective discriminant information in the process of space reduction. To this end, some theory should be developed first.

*Theorem 10.1 (Hilbert-Schmidt Theorem, Hutson & Pym, 1980). Let **A** be a compact and self-adjoint operator on Hilbert space $\mathcal{H}$. Then its eigenvector system forms an orthonormal basis for $\mathcal{H}$.*

Since $\mathbf{S}_w^{\Phi}$ is compact and self-adjoint, it follows from Theorem 1 that its eigenvector system $\{\boldsymbol{\beta}_i\}$ forms an orthonormal basis for H. Suppose $\boldsymbol{\beta}_1, \ldots, \boldsymbol{\beta}_m$ are eigenvectors corresponding to positive eigenvalues of $\mathbf{S}_t^{\Phi}$, where $m = \text{rank}(\mathbf{S}_t^{\Phi}) = \text{rank}(\mathbf{R})$. Generally, $m = M - 1$, where M is the number of training samples. Let us define the subspace $\Psi_t = \textbf{span}\{\boldsymbol{\beta}_1, \boldsymbol{\beta}_2, \ldots, \boldsymbol{\beta}_m\}$. Suppose its orthogonal complementary space is denoted by $\Psi_t^{\perp}$. Actually, $\Psi_t^{\perp}$ is the null space of $\mathbf{S}_w^{\Phi}$. Since Ψ_t, due to its finite dimensionality, is a closed subspace of $\mathcal{H}$, from the *projection theorem* (Weidmann, 1980), we have:

Corollary 10.1. $\mathcal{H} = \Psi_t \oplus \Psi_t^{\perp}$. That is, for an arbitrary vector $\boldsymbol{\varphi} \in \mathcal{H}$, $\boldsymbol{\varphi}$ can be uniquely represented in the form $\boldsymbol{\varphi} = \boldsymbol{\phi} + \boldsymbol{\zeta}$ with $\boldsymbol{\phi} \in \boldsymbol{\Psi}_t$ and $\boldsymbol{\zeta} \in \Psi_t^{\perp}$.

Now, let us define a mapping L: $H \rightarrow \Psi_t$ by:

$$\boldsymbol{\varphi} = \boldsymbol{\phi} + \boldsymbol{\zeta} \rightarrow \boldsymbol{\phi} \tag{10.13}$$

where $\boldsymbol{\phi}$ is called the orthogonal projection of $\boldsymbol{\varphi}$ onto Ψ_t. It is easy to verify that L is a linear operator from $\mathcal{H}$ onto its subspace Ψ_t.

Theorem 10.2 (Yang, Zhang, Yang, Jin, & Frangi, 2005). Under the mapping L: $\mathcal{H} \rightarrow \Psi_t$ determined by $\boldsymbol{\varphi} = \boldsymbol{\phi} + \boldsymbol{\zeta} \rightarrow \boldsymbol{\phi}$, the Fisher criterion satisfies the following properties:

$$J_b^{\Phi}(\boldsymbol{\varphi}) = J_b^{\Phi}(\boldsymbol{\phi}) \quad \textit{and} \quad \mathrm{J}^{\Phi}(\boldsymbol{\varphi}) = \mathrm{J}^{\Phi}(\boldsymbol{\phi}) \tag{10.14}$$

According to Theorem 2, we can conclude that both kinds of discriminant vectors can be derived from Ψ_t without any loss of effective discriminatory information with respect to the Fisher criterion. Since the new search space Ψ_t is finite-dimensional and much smaller (less dimensional) than $\textbf{null}(\mathbf{S}_w^{\Phi})$ and $\mathcal{H}$ - $\textbf{null}(\mathbf{S}_w^{\Phi})$, it is feasible to derive discriminant vectors from it.

Idea of Calculating Fisher Optimal Discriminant Vectors

In this section, we will offer our idea of calculating Fisher optimal discriminant vectors in the reduced search space Ψ_t. Since the dimension of Ψ_t is m, according to functional analysis theory (Kreyszig, 1978), Ψ_t is isomorphic to m-dimensional Euclidean space $\mathbb{R}^m$. The corresponding *isomorphic mapping* is:

$$\boldsymbol{\varphi} = \mathbf{P}\boldsymbol{\eta}\text{, where } \mathbf{P} = (\boldsymbol{\beta}_1, \boldsymbol{\beta}_2, \ldots, \boldsymbol{\beta}_m),\ \boldsymbol{\eta} \in \tag{10.15}$$

which is a one-to-one mapping from $\mathbb{R}^m$ onto Ψ_t.

Under the isomorphic mapping $\boldsymbol{\varphi} = \mathbf{P}\boldsymbol{\eta}$, the criterion function $J^{\Phi}(\boldsymbol{\varphi})$ and $J_b^{\Phi}(\boldsymbol{\varphi})$ in *feature space* are, respectively, converted into:

$$J^{\Phi}(\varphi) = \frac{\eta^{\mathrm{T}}(\mathbf{P}^{\mathrm{T}}\mathbf{S}_b^{\Phi}\mathbf{P})\eta}{\eta^{\mathrm{T}}(\mathbf{P}^{\mathrm{T}}\mathbf{S}_w^{\Phi}\mathbf{P})\eta} \quad \text{and} \quad J_b^{\Phi}(\varphi) = \eta^{\mathrm{T}}(\mathbf{P}^{\mathrm{T}}\mathbf{S}_b^{\Phi}\mathbf{P})\eta \tag{10.16}$$

Now, based on Equation 10.16, let us define two functions:

$$J(\eta) = \frac{\eta^{\mathrm{T}}\mathbf{S}_b\eta}{\eta^{\mathrm{T}}\mathbf{S}_w\eta}, \ (\eta \neq 0) \text{ and } \ J_b(\eta) = \eta^{\mathrm{T}}\mathbf{S}_b\eta, \ (\|\eta\| = 1) \tag{10.17}$$

where $\mathbf{S}_b = \mathbf{P}^{\mathrm{T}}\mathbf{S}_b^{\Phi}\mathbf{P}$ and $\mathbf{S}_w = \mathbf{P}^{\mathrm{T}}\mathbf{S}_w^{\Phi}\mathbf{P}$.

It is easy to show that $\mathbf{S}_b$ and $\mathbf{S}_w$ are both $m \times m$ semi-positive definite matrices. This means that $J(\boldsymbol{\eta})$ is a generalized Rayleigh quotient (Lancaster & Wechsler, 2001) and $J_b(\boldsymbol{\eta})$ is a Rayleigh quotient in the isomorphic space $\mathbb{R}^m$. Note that $J_b(\boldsymbol{\eta})$ is viewed as a Rayleigh quotient because the formula $^{\mathrm{T}}\mathbf{S}_b$ ($\|$ $\|$ 1) is equivalent to $\frac{^{\mathrm{T}}\mathbf{S}_b}{\mathrm{T}}$.

Under the isomorphic mapping mentioned above, the stationary points (optimal solutions) of the Fisher criterion have the following intuitive property:

Theorem 10.3. *Let* $\boldsymbol{\varphi} = \mathbf{Ph}$ *be an isomorphic mapping from* $\mathbb{R}^m$ *onto* Ψ_t. *Then* $\boldsymbol{\varphi}^* = \mathbf{Ph}^*$ *is the stationary point of* $J^{\Phi}(\boldsymbol{\varphi})$ ($J^{\Phi}(\boldsymbol{\varphi})$) *if and only if* $\boldsymbol{\eta}^*$ *is the stationary point of* $J(\boldsymbol{\eta})$ ($J_b(\boldsymbol{\eta})$).

From Theorem 3, it is easy to draw the following conclusion:

Corollary 10.2. If $\boldsymbol{\eta}_1, \ldots, \boldsymbol{\eta}_d$ is a set of stationary points of the function $J(\boldsymbol{\eta})$ ($J_b(\boldsymbol{\eta})$), then $\boldsymbol{\varphi}_1 = \mathbf{P}\boldsymbol{\eta}_1, \ldots, \boldsymbol{\varphi}_d = \mathbf{P}\boldsymbol{\eta}_d$ is a set of *regular* (*irregular*) optimal discriminant vectors with respect to the Fisher criterion $J^{\Phi}(\boldsymbol{\varphi})$ ($J_b^{\Phi}(\boldsymbol{\varphi})$).

Now, the problem of calculating the optimal discriminant vectors in subspace Ψ_t is transformed into the extremum problem of the (generalized) Rayleigh quotient in the isomorphic space $\mathbb{R}^m$.

A Concise KFD Framework: KPCA Plus LDA

The obtained optimal discriminant vectors are used for feature extraction in *feature space*. Given a sample $\mathbf{x}$ and its mapped image $\Phi(\mathbf{x})$, we can obtain the discriminant feature vector $\mathbf{z}$ by the following transformation:

$$\mathbf{z} = \mathbf{W}^{\mathrm{T}}\Phi(\mathbf{x}) \tag{10.18}$$

where:

$$\mathbf{W}^{\mathrm{T}} = (\boldsymbol{\varphi}_1, \boldsymbol{\varphi}_2, \ldots, \boldsymbol{\varphi}_d)^{\mathrm{T}} = (\mathbf{P}\boldsymbol{\eta}_1, \mathbf{P}\boldsymbol{\eta}_2, \ldots, \mathbf{P}\boldsymbol{\eta}_d)^{\mathrm{T}} = (\boldsymbol{\eta}_1, \boldsymbol{\eta}_2, \ldots, \boldsymbol{\eta}_d)^{\mathrm{T}}\,\mathbf{P}^{\mathrm{T}}$$

The transformation in Equation 10.18 can be decomposed into two transformations:

$$\mathbf{y} = \mathbf{P}^{\mathrm{T}}\Phi(\mathbf{x}), \text{ where } \mathbf{P} = (\boldsymbol{\beta}_1, \boldsymbol{\beta}_2, \ldots, \boldsymbol{\beta}_m) \tag{10.19}$$

$$\text{and } \mathbf{z} = \mathbf{G}^{\mathrm{T}}\mathbf{y}, \text{ where } \mathbf{G} = (\boldsymbol{\eta}_1, \boldsymbol{\eta}_2, \ldots, \boldsymbol{\eta}_d) \tag{10.20}$$

Since $\boldsymbol{\beta}_1, \boldsymbol{\beta}_2, \ldots, \boldsymbol{\beta}_m$ are eigenvectors of $\mathbf{S}_t^{\Phi}$ corresponding to positive eigenvalues, the transformation in Equation 10.19 is exactly KPCA; see Equations 10.7 and 10.8. This transformation transforms the input space $\mathbb{R}^n$ into space $\mathbb{R}^m$.

Now, let us view the issues in the KPCA-transformed space $\mathbb{R}^m$. Looking back at Equation 10.17 and considering the two matrices $\mathbf{S}_b$ and $\mathbf{S}_w$, it is easy to show that they are between-class and within-class scatter matrices in $\mathbb{R}^m$. In fact, we can construct them directly by:

$$\mathbf{S}_b = \frac{1}{M}\sum_{i=1}^{c} l_i (\mathbf{m}_i - \mathbf{m}_0)(\mathbf{m}_i - \mathbf{m}_0)^{\mathrm{T}} \tag{10.21}$$

$$\mathbf{S}_w = \frac{1}{M}\sum_{i=1}^{c}\sum_{j=1}^{l_i} (\mathbf{y}_{ij} - \mathbf{m}_i)(\mathbf{y}_{ij} - \mathbf{m}_i)^{\mathrm{T}} \tag{10.22}$$

where $\mathbf{y}_{ij}$ denotes the j_{th} training sample in class i; l_i is the number of training samples in class i; $\mathbf{m}_i$ is the mean of the training samples in class i; $\mathbf{m}_0$ the mean across all training samples.

Since $\mathbf{S}_b$ and $\mathbf{S}_w$ are between-class and within-class scatter matrices in $\mathbb{R}^m$, the functions $J(\boldsymbol{\eta})$ and $J_b(\boldsymbol{\eta})$ can be viewed as Fisher criterions, and their stationary points $\boldsymbol{\eta}_1, \ldots, \boldsymbol{\eta}_d$ are the associated Fisher optimal discriminant vectors. Correspondingly, the transformation in Equation 10.20 is the FLD in the KPCA-transformed space $\mathbb{R}^m$.

Up to now, the essence of KFD has been revealed. That is, KPCA is first used to reduce (or increase) the dimension of the *input space* to m, where m is the rank of $\mathbf{S}_t$ (i.e., the rank of the centralized Gram matrix **R**). Next, LDA is used for further feature extraction in the KPCA-transformed space $\mathbb{R}^m$.

In summary, a new KFD framework, KPCA plus LDA, is developed in this section. This framework offers us a new insight into the nature of KFD.

COMPLETE KFD ALGORITHM

In this section, we will develop a CKFD algorithm based on the two-phase KFD framework. Two kinds of discriminant information, *regular* and *irregular*, will be derived and fused for classification tasks.

Extraction of Two Kinds of Discriminant Features

Our task is to explore how to perform LDA in the KPCA-transformed space $\mathbb{R}^m$. After all, the standard LDA algorithm remains inapplicable, since the within-class scatter matrix $\mathbf{S}_w$ is still singular in $\mathbb{R}^m$. We would rather take advantage of this *singularity* to extract more discriminant information than avoid it by means of the previous regularization techniques (Mika, Rätsch, Weston, Schölkopf, & Müller, 1999; Baudat & Anouar, 2000; Yang, 2002). Our strategy is to split the space $\mathbb{R}^m$ into two subspaces: the null space and the range space of $\mathbf{S}_w$. We then use the Fisher criterion to derive the *regular* discriminant vectors from the range space and use the between-class scatter criterion to derive the *irregular* discriminant vectors from the null space.

Suppose $\boldsymbol{\alpha}_1, \ldots, \boldsymbol{\alpha}_m$ are the orthonormal eigenvectors of $\mathbf{S}_w$ and assume that the first q ones are corresponding to non-zero eigenvalues, where $q = \text{rank}(\mathbf{S}_w)$. Let us define a subspace $\Theta_w = \text{span}\{\boldsymbol{\alpha}_{q+1}, \ldots, \boldsymbol{\alpha}_m\}$. Its orthogonal complementary space is $\Theta_w^{\perp} = \text{span}\{\boldsymbol{\alpha}_1, \ldots, \boldsymbol{\alpha}_q\}$.

Actually, Θ_w is the null space and $\Theta_w^{\perp}$ is the range space of $\mathbf{S}_w$ and, $\mathbb{R}^m = \boldsymbol{\Theta}_w \oplus \Theta_w^{\perp}$. The dimension of the subspace $\Theta_w^{\perp}$ is q. Generally, $q = M - c = m - c + 1$. The dimension of the subspace Θ_w is $p = m - q$. Generally, $p = c - 1$.

Lemma 10.3 (Yang, Zhang, Yang, Jin, & Frangi, 2005). For every nonzero vector $\boldsymbol{\eta} \in \Theta_w$, the inequality $\boldsymbol{\eta}^{\mathrm{T}} \mathbf{S}_b \boldsymbol{\eta} > 0$ always holds.

Lemma 3 tells us there indeed exists *irregular* discriminant information in the null space of $\mathbf{S}_w$, Θ_w, since the within-class scatter is zero while the between-class scatter is always positive. Thus, the optimal *irregular* discriminant vectors must be derived from this space. On the other hand, since every non-zero vector $\boldsymbol{\eta} \in \Theta_w^{\perp}$ satisfies $\boldsymbol{\eta}^{\mathrm{T}} \mathbf{S}_w \boldsymbol{\eta} > 0$, it is feasible to derive the optimal *regular* discriminant vectors from $\Theta_w^{\perp}$ using the standard Fisher criterion.

The idea of isomorphic mapping discussed in Chapter III can still be used for calculations of the optimal *regular* and *irregular* discriminant vectors.

Let us first consider the calculation of the optimal *regular* discriminant vectors in $\Theta_w^{\perp}$. Since the dimension of $\Theta_w^{\perp}$ is q, $\Theta_w^{\perp}$ is isomorphic to Euclidean space $\mathbb{R}^q$, and the corresponding isomorphic mapping is:

$$\boldsymbol{\eta} = \mathbf{P}_1 \boldsymbol{\xi}, \text{ where } \mathbf{P}_1 = (\boldsymbol{\alpha}_1, \ldots, \boldsymbol{\alpha}_q) \tag{10.23}$$

Under this mapping, the Fisher criterion $J(\boldsymbol{\eta})$ in Equation 10.17 is converted into:

$$\tilde{J}(\boldsymbol{\xi}) = \frac{\boldsymbol{\xi}^{\mathrm{T}} \tilde{\mathbf{S}}_b \boldsymbol{\xi}}{\boldsymbol{\xi}^{\mathrm{T}} \tilde{\mathbf{S}}_w \boldsymbol{\xi}}, \ (\boldsymbol{\xi} \neq 0) \tag{10.24}$$

where $\tilde{\mathbf{S}}_b = \mathbf{P}_1^{\mathrm{T}} \mathbf{S}_b \mathbf{P}_1$ and $\tilde{\mathbf{S}}_w = \mathbf{P}_1^{\mathrm{T}} \mathbf{S}_w \mathbf{P}_1$. It is easy to verify that $\tilde{\mathbf{S}}_b$ is semi-positive definite and $\tilde{\mathbf{S}}_w$ is positive definite (must be invertible) in $\mathbb{R}^q$. Thus, $\tilde{J}(\boldsymbol{\xi})$ is a standard generalized Rayleigh quotient. Its stationary points $\mathbf{u}_1, \ldots, \mathbf{u}_d \ (\mathrm{d} \leq c-1)$ are actually the generalized

eigenvectors of the generalized eigen-equation $\tilde{\mathbf{S}}_b\boldsymbol{\xi} = \lambda\tilde{\mathbf{S}}_w\boldsymbol{\xi}$ corresponding to d largest positive eigenvalues (Lancaster & Tismenetsky, 1985). It is easy to calculate them using the standard LDA algorithm. After working out $\mathbf{u}_1, \ldots, \mathbf{u}_d$, we can obtain $\tilde{\boldsymbol{\eta}}_j = \mathbf{P}_1\mathbf{u}_j (j = 1, \ldots, d)$ using Equation 10.23. From the property of isomorphic mapping, we know $\tilde{\boldsymbol{\eta}}_1, \ldots, \tilde{\boldsymbol{\eta}}_d$ are the optimal *regular* discriminant vectors with respect to $J(\boldsymbol{\eta})$.

In a similar way, we can calculate that the optimal *irregular* discriminant vectors within Θ_w. Θ_w are isomorphic to Euclidean space $\mathbb{R}^p$, and the corresponding isomorphic mapping is:

$$\boldsymbol{\eta} = \mathbf{P}_2\boldsymbol{\xi}, \text{ where } \mathbf{P}_2 = (\boldsymbol{\alpha}_{q+1}, \ldots, \boldsymbol{\alpha}_m) \tag{10.25}$$

Under this mapping, the criterion $J_b(\boldsymbol{\eta})$ in Equation 10.17 is converted into:

$$\hat{J}_b(\boldsymbol{\xi}) = \boldsymbol{\xi}^{\mathrm{T}}\hat{\mathbf{S}}_b\boldsymbol{\xi}, \ (\|\boldsymbol{\xi}\|=1) \tag{10.26}$$

where $\hat{\mathbf{S}}_b = \mathbf{P}_2^{\mathrm{T}}\mathbf{S}_b\mathbf{P}_2$. It is easy to verify that $\hat{\mathbf{S}}_b$ is positive definite in $\mathbb{R}^p$. The stationary points $\mathbf{v}_1, \ldots, \mathbf{v}_d$ ($d \le c$ - 1) of $\hat{J}_b(\boldsymbol{\xi})$ are actually the orthonormal eigenvectors of $\hat{\mathbf{S}}_b$ corresponding to d largest eigenvalues. After working out $\mathbf{v}_1, \ldots, \mathbf{v}_d$, we can obtain $\hat{\boldsymbol{\eta}}_j = \mathbf{P}_2\mathbf{v}_j (j = 1, \ldots, d)$ using Equation 10.25. From the property of isomorphic mapping, we know $\hat{\boldsymbol{\eta}}_1, \ldots, \hat{\boldsymbol{\eta}}_d$ are the optimal *irregular* discriminant vectors with respect to $J_b(\eta)$.

Based on the derived optimal discriminant vectors, the linear discriminant transformation in Equation 10.20 can be performed in $\mathbb{R}^m$. Specifically, after the projection of the sample $\mathbf{y}$ onto the *regular* discriminant vectors $\tilde{\boldsymbol{\eta}}_1, \ldots, \tilde{\boldsymbol{\eta}}_d$, we can obtain the *regular* discriminant feature vector:

$$\mathbf{z}^1 = (\tilde{\boldsymbol{\eta}}_1, \ldots, \tilde{\boldsymbol{\eta}}_d)^{\mathrm{T}}\mathbf{y} = \mathbf{U}^{\mathrm{T}}\mathbf{P}_1^{\mathrm{T}}\mathbf{y} \tag{10.27}$$

where $\mathbf{U} = (\mathbf{u}_1, \ldots, \mathbf{u}_d)$, $\mathbf{P}_1 = (\boldsymbol{\alpha}_1, \ldots, \boldsymbol{\alpha}_q)$.

After the projection of the sample $\mathbf{y}$ onto the *irregular* discriminant vectors $\hat{\boldsymbol{\eta}}_1, \ldots, \hat{\boldsymbol{\eta}}_d$, we can obtain the *irregular* discriminant feature vector:

$$\mathbf{z}^2 = (\hat{\boldsymbol{\eta}}_1, \ldots, \hat{\boldsymbol{\eta}}_d)^{\mathrm{T}}\mathbf{y} = \mathbf{V}^{\mathrm{T}}\mathbf{P}_2^{\mathrm{T}}\mathbf{y} \tag{10.28}$$

where $\mathbf{V} = (\mathbf{v}_1, \ldots, \mathbf{v}_d)$, $\mathbf{P}_2 = (\boldsymbol{\alpha}_{q+1}, \ldots, \boldsymbol{\alpha}_m)$

Fusion of Two Kinds of Discriminant Features for Classification

The minimum distance classifier has been demonstrated to be very effective based on Fisher discriminant features (Liu & Wechsler, 2001; Yang & Yang, 2003). For simplicity, a minimum-distance classifier is employed and the Euclidean measure is

adopted here. The Euclidean distance between sample $\mathbf{z}$ and pattern class k is defined by:

$$g_k(\mathbf{z}) = \| \mathbf{z} - \boldsymbol{\mu}_k \|_2 \tag{10.29}$$

where $\boldsymbol{\mu}_k$ denotes the mean vector of the training samples in class k. The decision rule is: If sample $\mathbf{z}$ satisfies $g_i(\mathbf{z}) = \min_k g_k(\mathbf{z})$, then $\mathbf{z}$ belongs to class i.

Since for any given sample $\mathbf{z}$ we can obtain two d-dimensional discriminant feature vectors, it is possible to fuse them in the decision level. Here, we suggest a simple fusion strategy based on a summed normalized distance. Specifically, let us denote $\mathbf{z} = [\mathbf{z}^1, \mathbf{z}^2]$, where $\mathbf{z}^1$, $\mathbf{z}^2$ are CKFD *regular* and *irregular* discriminant feature vectors of a same pattern. The summed normalized distance between sample $\mathbf{z}$ and the mean vector $\boldsymbol{\mu}_k = [\boldsymbol{\mu}_k^1, \boldsymbol{\mu}_k^2]$ of class k is defined by:

$$\overline{g}_k(\mathbf{z}) = \frac{\| \mathbf{z}^1 - \boldsymbol{\mu}_k^1 \|_2}{\sum_{j=1}^{c} \| \mathbf{z}^1 - \boldsymbol{\mu}_j^1 \|_2} + \frac{\| \mathbf{z}^2 - \boldsymbol{\mu}_k^2 \|_2}{\sum_{j=1}^{c} \| \mathbf{z}^2 - \boldsymbol{\mu}_j^2 \|_2} \tag{10.30}$$

Then, we use the minimum distance decision rule mentioned above for classification.

Complete KFD Algorithm

In summary of the discussion so far, the complete KFD algorithm is given as follows:

CKFD Algorithm

- **Step 1.** Use KPCA to transform the input space $\mathbb{R}^n$ into an m-dimensional space $\mathbb{R}^m$, where m = rank($\mathbf{R}$), $\mathbf{R}$ is the centralized Gram matrix. Pattern $\mathbf{x}$ in $\mathbb{R}^n$ is transformed to be KPCA-based feature vector $\mathbf{y}$ in $\mathbb{R}^m$.
- **Step 2.** In $\mathbb{R}^m$, construct the between-class and within-class scatter matrices $\mathbf{S}_b$ and $\mathbf{S}_w$. Calculate $\mathbf{S}_w$'s orthonormal eigenvectors, $\boldsymbol{\alpha}_1, \ldots, \boldsymbol{\alpha}_m$, assuming the first q (q = rank ($\mathbf{S}_w$)) ones are corresponding to positive eigenvalues.
- **Step 3.** Extract the *regular* discriminant features: Let $\mathbf{P}_1 = (\boldsymbol{\alpha}_1, \ldots, \boldsymbol{\alpha}_q)$. Define $\tilde{\mathbf{S}}_b = \mathbf{P}_1^{\mathrm{T}} \mathbf{S}_b \mathbf{P}_1$ and $\tilde{\mathbf{S}}_w = \mathbf{P}_1^{\mathrm{T}} \mathbf{S}_w \mathbf{P}_1$, and calculate the generalized eigenvectors $\mathbf{u}_1, \ldots, \mathbf{u}_d (d \le c-1)$ of $\tilde{\mathbf{S}}_b \boldsymbol{\xi} = \lambda \tilde{\mathbf{S}}_w \boldsymbol{\xi}$ corresponding to d largest positive eigenvalues. Let $\mathbf{U} = (\mathbf{u}_1, \ldots, \mathbf{u}_d)$. The *regular* discriminant feature vector is $\mathbf{z}^1 = \mathbf{U}^{\mathrm{T}} \mathbf{P}_1^{\mathrm{T}} \mathbf{y}$.
- **Step 4.** Extract the *irregular* discriminant features: Let $\mathbf{P}_2 = (\boldsymbol{\alpha}_{q+1}, \ldots, \boldsymbol{\alpha}_m)$. Define $\hat{\mathbf{S}}_b = \mathbf{P}_2^{\mathrm{T}} \mathbf{S}_b \mathbf{P}_2$ and calculate $\hat{\mathbf{S}}_b$'s orthonormal eigenvectors $\mathbf{v}_1, \ldots, \mathbf{v}_d$ ($d \le c-1$) corresponding to d largest eigenvalues. Let $\mathbf{V} = (\mathbf{v}_1, \ldots, \mathbf{v}_d)$. The *irregular* discriminant feature vector is $\mathbf{z}^2 = \mathbf{V}^{\mathrm{T}} \mathbf{P}_2^{\mathrm{T}} \mathbf{y}$.
- **Step 5.** Fuse the *regular* and *irregular* discriminant features using summed normalized distance for classification.

Concerning the implementation of the CKFD algorithm, a remark should be made. For numerical robustness, in Step 2 of CKFD algorithm, q could be selected as a number that is properly less than the real rank of $\mathbf{S}_w$ in practical applications. Here, we choose q as the number of eigenvalues that are less than $\frac{\lambda_{\max}}{2000}$, where $\lambda_{\max}$ is the maximal eigenvalue of $\mathbf{S}_w$.

Relationship to Other KFD (or LDA) Algorithms

In this section, we will review some other KFD (LDA) methods and explicitly distinguish them from the proposed CKFD. Let us begin with LDA methods. Liu (Liu & Yang, 1992) first claimed that there exist two kinds of discriminant information for LDA in SSS cases, irregular discriminant information (within the null space of *within-class* scatter matrix) and regular discriminant information (beyond the null space). Chen (Chen, Liao, & Ko, 2000) emphasized the irregular information and proposed a more effective way to extract it, but overlooked the regular information. Yu (Yu & Yang, 2001) took two kinds of discriminatory information into account and suggested extracting them within the range space of the *between-class* scatter matrix. Since the dimension of the range space is up to $c-1$, Yu and Yang's (2001) algorithm (DLDA) is computationally more efficient for SSS problems in that the computational complexity is reduced to be $\mathcal{O}(c^3)$.

LDA, however, is sub-optimal, in theory. Although there is no discriminatory information within the null space of the *between-class* scatter matrix, no theory (like Theorem 2) can guarantee that all discriminatory information must exist in the range space, because there is a large space beyond the null and the range space, which may contain crucial discriminant information (see the shadow area in Figure 6.3a in Chapter VI). For two-class problems (such as gender recognition), the weakness of DLDA becomes more noticeable. The range space is only 1-D and spanned by the difference of the two-class mean vectors. This subspace is too small to contain enough discriminant information. Actually, in such a case, the resulting discriminant vector of DLDA is the difference vector itself, which is not optimal with respect to the Fisher criterion, let alone the ability to extract two kinds of discriminant information.

Lu (Lu, Plataniotis, & Venetsanopoulos, 2003) generalized DLDA using the idea of kernels and presented kernel direct discriminant analysis (KDDA). KDDA was demonstrated effective for face recognition but, as a nonlinear version of DLDA, KDDA unavoidably suffers the weakness of DLDA. On the other hand, unlike DLDA, which can significantly reduce computational complexity of LDA (as discussed above), KDDA has the same computational complexity; that is, $\mathcal{O}(M^3)$, with other KFD algorithms (Baudat & Anouar, 2000; Mika, Rätsch, & Weston, 2003; Mika, Rätsch, & Weston, 1999), because KDDA still needs to calculate the eigenvectors of an $M \times M$ Gram matrix.

Like Liu's method, our previous LDA algorithm (Yang & Yang, 2001, 2003) can obtain more than $c-1$ features; that is, all $c-1$ irregular discriminant features plus some regular ones. This algorithm turned out to be more effective than Chen and Yu's methods, which can extract at most $c-1$ features. In addition, our LDA algorithm is more powerful and simpler than Liu's method. The algorithm in literature (Yang, Frangi, & Yang, 2004) can be viewed as a nonlinear generalization of that in Yang and Yang (2003). However, the derivation of the algorithm is based on an assumption that the feature space is assumed to be a finite dimensional space. This assumption is no problem for polynomial

kernels, but unsuitable for other kernels that determine mappings that might lead to an infinite-dimensional feature space.

Compared to our previous idea (Yang, Frangi, & Yang, 2004) and Lu's KDDA, CKFD has two prominent advantages. One is in the theory, and other is in the algorithm itself. The theoretical derivation of the algorithm does not need any assumption. The developed theory in Hilbert space lays a solid foundation for the algorithm. The derived discriminant information is guaranteed not only optimal but also complete (lossless) with respect to the Fisher criterion. The completeness of discriminant information enables CKFD to be used to perform discriminant analysis in "double discriminant subspaces." In each subspace, the number of discriminant features can be up to $c - 1$. This means $2(c - 1)$ features can be obtained in total. This is different from the KFD (or LDA) algorithms discussed above and beyond, which can yield only one discriminant subspace containing at most c - 1 discriminant features. What is more, CKFD provides a new mechanism for decision fusion. This mechanism makes it possible to take advantage of the two kinds of discriminant information.

CKFD has a computational complexity of $\mathcal{O}(M^3)$ (M is the number of training samples), which is the same as the existing KFD algorithms. The reason for this is that the KPCA phase of CKFD is actually carried out in the space spanned by M training samples, so its computational complexity still depends on the operations of solving $M \times M$-sized eigenvalue problems. Despite this, compared to other KFD algorithms, CKFD indeed requires additional computation mainly owing to its space decomposition process performed in the KPCA-transformed space. In such a space, all eigenvectors of $\mathbf{S}_w$ should be calculated.

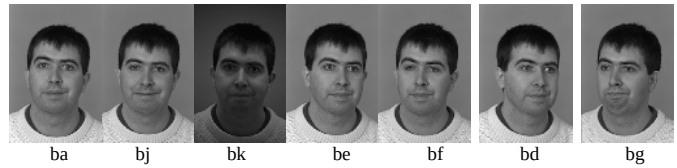
EXPERIMENTS

The FERET face image database is a result of the FERET program, which was sponsored by the Department of Defense through the DARPA Program (Phillips, Moon, Rizvi, & Rauss, 2000). It has become a standard database for testing and evaluating state-of-the-art face recognition algorithms.

Table 10.1. The two-letter strings in image names indicate the kind of imagery

Two-letter mark	Pose Angle (degrees)	Description	Number of Subjects
ba	0	Frontal "b" series	200
bj	0	Alternative expression to "ba"	200
bk	0	Different illumination to "ba"	200
bd	+25	Subject faces to his left which is the photographer's right	200
be	+15		200
bf	-15	Subject faces to his right which is the photographer's left	200
bg	-25		200

Figure 10.1. Images of one person in FERET database



(a) original images

(b) cropped images (after histogram equalization)

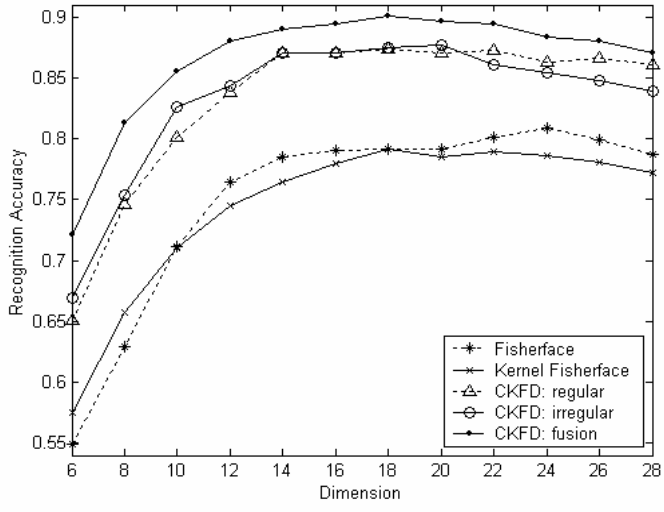
(a) Original images, (b) cropped images corresponding to images in (a)

The proposed algorithm was applied to face recognition and tested on a subset of the FERET database. This subset includes 1,400 images of 200 individuals (each individual has seven images). It is composed of the images whose names are marked with two-character strings: "*ba,*" "*bj,*" "*bk,*" "*be,*" "*bf,*" "*bd*" and "*bg.*" These strings indicate the kind of imagery as shown in Table 10.1. This subset involves variations in facial expression, illumination and pose. In our experiment, the facial portion of each original image was cropped based on the location of eyes, and the cropped image was resized to 80×80 pixels and pre-processed by histogram equalization. Some example images of one person are shown in Figure 10.1.

In our experiments, three images of each subject are randomly chosen for training, while the remaining images are used for testing. Thus, the total number of training samples is 600 and the total number of testing samples is 800. Fisherface (Belhumeur, Hespanha, & Kriegman, 1997), kernel fisherface (Yang, 2002) and the proposed CKFD algorithm are used, respectively, for feature extraction. For fisherface and kernel fisherface, 200 principal components ($l = c = 200$) are chosen in the PCA phase, taking the generalization of LDA into account (Liu & Wechsler, 2001). Yang (2002) has demonstrated that a second- or third-order polynomial kernel suffices to achieve good results for face recognition. So, a second-order polynomial kernel, $k(\mathbf{x}, \mathbf{y}) = (\mathrm{x} \cdot \mathrm{y} + 1)^2$, is first adopted for all kernel-related methods. Concerning the proposed CKFD algorithm, in order to gain more insight into its performance, we test its three different versions: (1) CKFD: *regular*, in which only the *regular* discriminant features are used; (2) CKFD: *irregular*, in which only the *irregular* discriminant features are used; (3) CKFD: *fusion*, in which *regular* and *irregular* discriminant features are both used and fused in the way suggested in Chapter IV. Finally, a minimum-distance classifier is employed for classification for all methods mentioned above. The classification results are illustrated in Figure 10.2.

From Figure 10.2, we can see that (a) the *irregular* discriminant feature of CKFD, which is always discarded by the existing KFD algorithms, is as effective as the *regular*

Figure 10.2. Illustration of the recognition rates of fisherface, kernel fisherface, CKFD: regular, CKFD: irregular, and CKFD: fusion on the first test



ones. Although it seems that CKFD: *regular* begin to outperform CKFD: *irregular* when the dimension is more than 20, the maximal recognition accuracy of the former is higher than that of the latter. (b) After the fusion of two kinds of discriminant information, performance is improved irrespective the variation of dimensions. This fact indicates that the *regular* discriminant features and the *irregular* ones are complimentary for achieving a better result. (c) CKFD: *fusion* (even CKFD: *regular* or CKFD: *irregular*) consistently outperforms fisherface and kernel fisherface. Why can CKFD: *regular* perform better than kernel fisherface? The underlying reason is the CKFD algorithm (Step 3) can achieve an accurate evaluation of the eigenvectors of the within-class scatter matrix while the PCA plus LDA technique adopted in kernel fisherface cannot, due to the loss of the subordinate components in the PCA phase.

Now, a question is: Are the above results with respect to the choice of training set? In other words, if another set of training samples are chosen at random, would we obtain same results? To answer this question, we repeat the experiment 10 times. Each time, the training sample set (containing three samples per class) is selected at random so that the training sample sets are different for 10 tests (Correspondingly, the testing sets are also different). For each method and four different dimensions (16, 18, 20, 22, respectively), the recognition rates across 10 tests are illustrated in Figure 10.3. Note that we chose dimension = 16, 18, 20, 22 because it can be seen from Figure 10.2 that the maximal recognition rates of fisherface, kernel fisherface and CKFD all occur in the interval where the dimension varies from 16 to 22. Also, for each method mentioned above, the average recognition rate and standard deviation standard deviation across 10 tests are listed in Table 10.2. Besides, Table 10.2 also gives the testing results of eigenface (Turk & Pentland, 1991) and kernel eigenface (Yang, 2002) on this database.

As shown in Figure 10.3 and Table 10.2, the *irregular* discriminant features stand comparison with the regular ones with respect to the discriminatory power. Both kinds

Table 10.2. The mean and standard deviation standard deviation of the recognition rates (%) of eigenface, kernel eigenface, fisherface, kernel fisherface, CKFD: regular, CKFD: irregular, and CKFD: fusion across ten tests when the dimension is chosen as 16, 18, 20 and 22

Dimension	eigenface	kernel eigenface	fisherface	kernel fisherface	CKFD: regular	CKFD: irregular	CKFD: fusion
16	13.92 ± 0.91	13.45 ± 0.89	76.54 ± 1.93	75.69 ± 1.61	85.25 ± 1.72	86.42 ± 1.38	88.56 ± 1.23
18	15.45 ± 0.83	14.92 ± 0.83	77.31 ± 1.59	76.32 ± 2.12	85.49 ± 1.75	86.48 ± 1.07	88.75 ± 1.28
20	17.21 ± 0.90	16.48 ± 0.92	77.97 ± 1.35	76.95 ± 1.63	85.53 ± 1.65	86.64 ± 1.04	88.53 ± 1.21
22	18.25 ± 1.08	17.80 ± 1.07	78.02 ± 1.73	77.39 ± 1.62	85.69 ± 1.52	86.26 ± 1.05	88.49 ± 1.14
Average	16.21 ± 0.93	15.66 ± 0.93	77.46 ± 1.65	76.59 ± 1.75	85.49 ± 1.66	86.45 ± 1.14	88.58 ± 1.21

Note: All kernel-based methods here use the second-order polynomial kernels

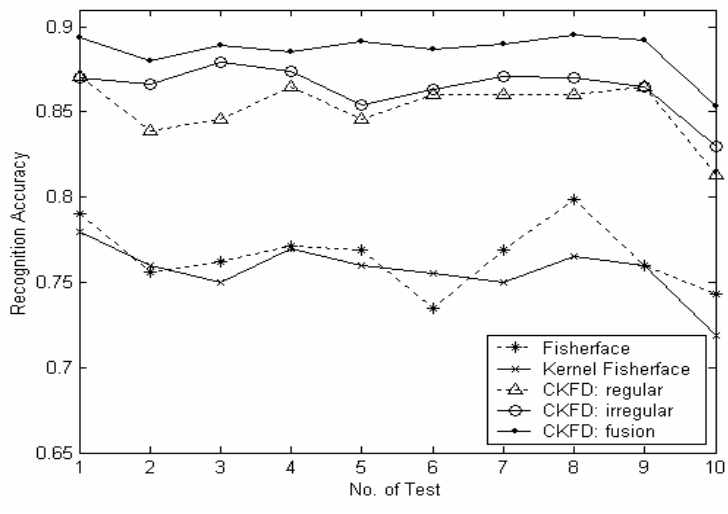
of discriminant features contribute to a better classification performance by virtue of fusion. All of three CKFD versions consistently outperform fisherface and kernel fisherface across 10 trials and four dimensions. These results are consistent with the results from Figure 10.2. That is to say, our experimental results are independent of the choice of training sets and dimensional variations. Table 10.2 also shows that fisherface, kernel fisherface and CKFD are all superior to eigenface and kernel eigenface in terms of recognition accuracy. This indicates that linear or nonlinear discriminant analysis is really helpful for improving the performance of PCA or KPCA for face recognition.

Moreover, from Table 10.2 and Figure 10.3, we can also see the standard deviation of CKFD *fusion* is smaller than those of fisherface and kernel fisherface. The standard deviation of CKFD: *regular* and CKFD: *irregular* are obviously different, the former is almost equal to that of fisherface while the latter is much smaller. Fortunately, after their fusion, the standard deviation of CKFD is satisfying; it is only slightly higher than that of CKFD: *irregular*. Although the standard deviation of eigenface and kernel eigenface are very small, we have no interest in them because their recognition performances are not satisfying.

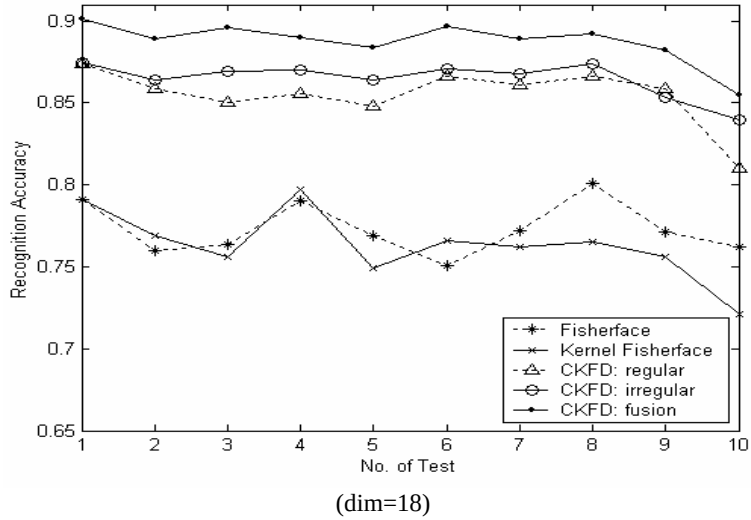
Another question is: Is CKFD statistically significantly better than other methods? To answer this question, let us evaluate the experimental results in Table 10.2 using McNemar's (Devore & Peck, 1997) significance test. McNemar's test is essentially a null hypothesis statistical test based on Bernoulli model. If the resulting p-value is below the desired significance level (for example, 0.02), the null hypothesis is rejected and the performance difference between two algorithms are considered to be statistically significant. By this test, we find that CKFD: *fusion* statistically significantly outperforms eigenface, kernel eigenface, fisherface and kernel fisherface at a significance level $p = 3.15 \times 10^{-9}$. Actually, CKFD: *regular* and CKFD: *irregular* also statistically significantly outperform other methods at a significance level $p = 2.80 \times 10^{-6}$.

A third question is: Do the results depend on the choice of kernels? In other words, if we use another kernel instead of the polynomial kernel, can we obtain similar results? To answer this question, let us try another popular kernel: Gaussian RBF kernel, which is formed by $k(\mathbf{x},\mathbf{y}) = \exp\left(\frac{\|\mathbf{x}-\mathbf{y}\|^2}{\delta}\right)$. In the formula, the width $\boldsymbol{\delta}$ is chosen to be

Figure 10.3. Illustration of the recognition rates of fisherface, kernel fisherface, CKFD: regular, CKFD: irregular, and CKFD: fusion across 10 tests when the dimension (number of features) is chosen as 16, 18, 20, and 22, respectively

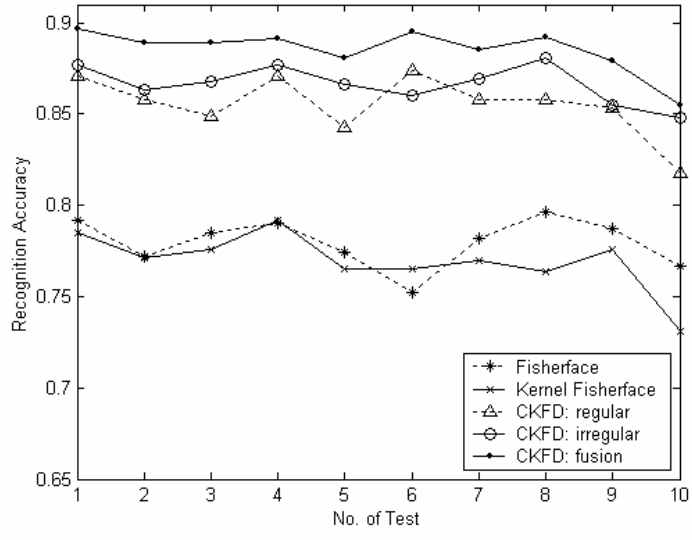


(dim=16)

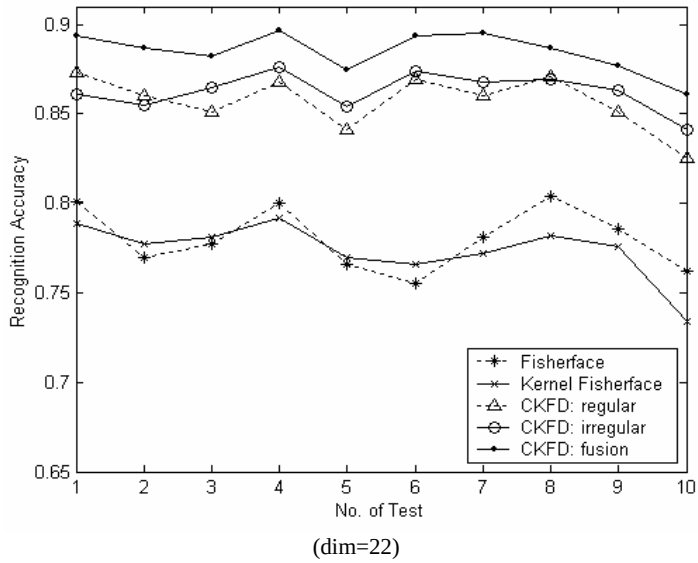


(dim=18)

Figure 10.3. Illustration of the recognition rates of fisherface, kernel fisherface, CKFD: regular, CKFD: irregular, and CKFD: fusion across 10 tests when the dimension (number of features) is chosen as 16, 18, 20, and 22, respectively (cont.)



(dim=20)



(dim=22)

$0.3 \times n$, where n is the dimension of *input space*. This parameter turned out to be optimal for SVMs (Mika, 2003). Here, $\delta = 0.3 \times 80^2 = 1920$. For all kernel-based methods mentioned above and the four chosen dimensions, the experimental results based on Gaussian RBF kernel are listed in Table 10.3. In general, these results accord with those shown in Table

10.2 based on polynomial kernel. CKFD: *fusion* is statistically significantly superior to kernel eigenface and kernel fisherface (significance level $p = 7.13\times10^{-8}$), and the classification performance of CKFD is improved after the fusion of CKFD *regular* and *irregular* features. Compared to Table 10.2, the only difference is that the discriminatory power of CKFD *regular* features is slightly enhanced while that of CKFD *irregular* features is relatively weakened. Despite this, CKFD *irregular* features are still very effective. They remain more powerful than those of kernel fisherface and contribute to better results by involving in fusion.

Comparing Tables 10.2 and 10.3, we find that Gaussian RBF kernel is not very helpful for improving the classification performance. In other words, the second-order polynomial kernel can really compete with Gaussian RBF kernel for face recognition. This is consistent with the previous results in Yang (2002) and Lu, Plataniotis, and Venetsanopoulos (2003).

Finally, to evaluate the computational efficiency of algorithms, we would like to give the average total CPU time of each method involved. The "total CPU time" refers to the CPU time consumed for the whole training process using 600 training samples and the whole testing process using 800 testing samples. The average "total CPU time" of 10 tests when the dimension = 20 is listed in Table 10.4. Table 10.4 shows CKFD (regular, irregular and fusion) algorithms are only slightly slower than kernel fisherface and kernel eigenface, no matter what kernel is adopted. For all kernel-based methods, the consumed CPU time increases double using Gaussian RBF kernel instead of polynomial kernel. Moreover, all kernel-based methods are more time-consuming than linear methods like eigenface and fisherface.

Table 10.3. The mean and standard deviation of recognition rates (%) of kernel eigenface, kernel fisherface, CKFD regular, CKFD irregular, and CKFD fusion across 10 tests when dimensions are 16, 18, 20 and 22

Dimension	kernel eigenface	kernel fisherface	CKFD: regular	CKFD: irregular	CKFD: fusion
16	13.91 ± 0.91	76.49 ± 1.92	85.78 ± 1.99	82.07 ± 1.66	87.53 ± 1.33
18	15.46 ± 0.84	77.29 ± 1.67	86.12 ± 2.17	82.38 ± 1.57	87.66 ± 1.36
20	17.23 ± 0.92	77.96 ± 1.30	86.13 ± 1.81	82.94 ± 0.88	87.94 ± 1.37
22	18.26 ± 1.06	78.08 ± 1.59	85.93 ± 1.68	82.65 ± 1.01	87.58 ± 0.91
Average	16.21 ± 0.93	77.46 ± 1.62	85.99 ± 1.91	82.51 ± 1.28	87.68 ± 1.24

Note: All methods here use Gaussian RBF kernels

SUMMARY

A new KFD framework — KPCA plus LDA — is developed in this chapter. Under this framework, a two-phase KFD algorithm is presented. Actually, based on the developed KFD framework, a series of existing KFD algorithms can be reformulated in alternative ways. In other words, it is easy to give equivalent versions of the previous KFD algorithms. Taking kernel fisherface as an example, we can first use KPCA to reduce the dimension to l (Note that here only l components are used; l is subject to $c \le l \le M - c$, where M is the number of training samples and c is the number of classes), and then perform standard LDA in the KPCA-transformed space. Similarly, we can construct alternative versions for others. These versions make it easier to understand and implement kernel Fisher discriminant, particularly for the new investigator or programmer.

A CKFD is proposed to implement the KPCA plus LDA strategy. This algorithm allows us to perform discriminant analysis in "double discriminant subspaces": regular and irregular. The previous KFD algorithms always emphasize the former and neglect the latter. In fact, the irregular discriminant subspace contains important discriminative information, which is as powerful as the regular discriminant subspace. This has been demonstrated by our experiments. It should be emphasized that for kernel-based discriminant analysis, the two kinds of discriminant information (particularly the irregular one) are widely existent, not limited to the SSS problems like face recognition. The underlying reason is that the implicit nonlinear mapping determined by "kernel" always turns large sample-size problems in observation space into SSS ones in *feature space*. More interestingly, the two discriminant subspaces of CKFD turn out to be mutually complementary for discrimination, despite the fact that each of them can work well independently. The fusion of two kinds of discriminant information can achieve better results.

Specially, for SSS problems, CKFD is exactly in tune with the existing two-phase LDA algorithms based on *PCA plus LDA* framework. Actually, if a linear kernel — that is, $k(\mathbf{x}, \mathbf{y}) = (\mathrm{x} \cdot \mathrm{y})$ — is adopted instead of nonlinear kernels, CKFD would degenerate to be a *PCA plus LDA* algorithm like that in Yang (2003). Therefore, the existing two-phase LDA (*PCA plus LDA*) algorithms can be viewed as a special case of CKFD.

Finally, we have to point out that the computational efficiency of CKFD is a problem deserving further investigation. Actually, all kernel-based methods, including KPCA (Schölkopf, Smola, & Müller, 1998), GDA (Baudat & Anouar, 2000) and KFD (Mika, Rätsch, & Weston, 2003), encounter the same problem. This is because all kernel-based discriminant methods have to solve an $M \times M$-sized eigen-problem (or generalized eigen-problem). When the sample size M is fairly large, it becomes very computationally intensive. Several ways suggested by Mika (Mika, Rätsch, & Weston, 2003) and Burges (Burges & Schölkopf, 1997) can be used to deal with this problem, but the optimal implementation scheme (e.g., developing more efficient numerical algorithm for large scale eigen-problem) is still open.

REFERENCES

Baudat, G., & Anouar, F. (2000). Generalized discriminant analysis using a kernel approach. *Neural Computation*, *12*(10), 2385-2404.

Belhumeur, P. N.,. Hespanha, J. P., & Kriegman, D. J. (1997). Eigenfaces vs. fisherfaces: Recognition using class specific linear projection. *IEEE Transaction on Pattern Analysis and Machine Intelligence, 19*(7), 711-720.

Billings, S. A., & Lee, K. L. (2002). Nonlinear Fisher discriminant analysis using a minimum squared error cost function and the orthogonal least squares algorithm. *Neural Networks*, *15*(2), 263-270.

Burges, C., & Schölkopf, B. (1997). Improving the accuracy and speed of support vector learning machines. In M. Mozer, M. Jordan, & T. Petsche (Eds.), *Advances in neural information processing systems, 9* (pp. 375-381). Cambridge, MA: MIT Press.

Cawley, G. C., & Talbot, N. L. C. (2003). Efficient leave-one-out cross-validation of kernel Fisher discriminant classifiers. *Pattern Recognition*, *36*(11), 2585-2592.

Chen, L. F., Liao, H. Y., & Ko, M. T. (2000). A new LDA-based face recognition system which can solve the small sample size problem. *Pattern Recognition*, *33*(10), 1713-1726.

Devore, J., & Peck, R. (1997). *Statistics: The exploration and analysis of data* (3rd ed.). Pacific Grove, CA: Brooks Cole.

Hutson, V., & Pym, J. S. (1980). *Applications of functional analysis and operator theory*. London: Academic Press.

Kreyszig, E. (1978). *Introductory functional analysis with applications*. New York: John Wiley & Sons.

Lancaster, P., & Tismenetsky, M. (1985). *The theory of matrices* (2nd ed.). Orlando, FL: Academic Press.

Liu, C-J., & Wechsler, H. (2001). A shape- and texture-based enhanced Fisher classifier for face recognition. *IEEE Trans. Image Processing*, *10*(4), 598-608.

Liu, K., & Yang, J-Y. (1992). An efficient algorithm for Foley-Sammon optimal set of discriminant vectors by algebraic metho. *International Journal of Pattern Recognition and Artificial Intelligence*, *6*(5), 817-829.

Lu, J., Plataniotis, K. N., & Venetsanopoulos, A. N. (2003). Face recognition using kernel direct discriminant analysis algorithms. *IEEE Transactions on Neural Networks*, *14*(1), 117-126.

Mika, S., Rätsch, G., Weston, J., Schölkopf, B., Smola, A. J., & Müller, K. R. (2003). Constructing descriptive and discriminative non-linear features: Rayleigh coefficients in kernel feature spaces. *IEEE Transaction on Pattern Analysis and Machine Intelligence*, *25*(5), 623-628.

Mika, S., Rätsch, G., & Weston, J., Schölkopf, B., & Müller, K. R. (1999). Fisher discriminant analysis with kernels. *IEEE International Workshop on Neural Networks for Signal Processing* (Vol. 9, pp. 41-48).

Müller, K-R., Mika, S., Rätsch, G., Tsuda, K., & Schölkopf, B. (2001). An introduction to kernel-based learning algorithms. *IEEE Transactions on Neural Networks*, *12*(2), 181-201.

Phillips, P. J., Moon, H., Rizvi, S. A., & Rauss, P. J. (2000). The FERET evaluation methodology for face-recognition algorithms. *IEEE Transaction on Pattern Analysis and Machine Intelligence*, *22*(10), 1090-1104.

Roth, V., & Steinhage, V. (2000). Nonlinear discriminant analysis using kernel functions. In S. A. Solla, T. K. Leen, & K.-R. Mueller (Eds.), *Advances in neural information processing systems* (Vol. 12, pp. 568-574). Cambridge, MA: MIT Press.

Schölkopf, B., Smola, A., & Müller, K. R. (1998). Nonlinear component analysis as a kernel eigenvalue problem. *Neural Computation*, *10*(5), 1299-1319.

Tikhonov, A. N., & Arsenin, V. Y. (1997). *Solution of ill-posed problems*. New York: Wiley.

Turk, M., & Pentland, A. (1991). Eigenfaces for recognition. *Journal of Cognitive Neuroscience*, *3*(1), 71-86.

Weidmann, J. (1980). *Linear operators in Hilbert spaces*. New York: Springer-Verlag.

Xu, J., Zhang, X., & Li, Y. (2001). Kernel MSE algorithm: A unified framework for KFD. *LS-SVM and KRR, Proceedings of the International Joint Conference on Neural Networks* (pp. 1486-1491).

Yang, J., Frangi, A. F., & Yang, J. Y. (2004). A new kernel Fisher discriminant algorithm with application to face recognition. *Neurocomputing*, *56*, 415-421.

Yang, J., & Yang, J.Y. (2001). Optimal FLD algorithm for facial feature extraction. In *SPIE Proceedings of the Intelligent Robots and Computer Vision XX: Algorithms, techniques, and Active Vision*, *4572* (pp. 438-444).

Yang, J., & Yang, J. Y. (2003). Why can LDA be performed in PCA transformed space? *Pattern Recognition*, *36*(2), 563-566.

Yang, J., Zhang, D., Yang, J-Y., Jin, Z., & Frangi, A. F. (2005). KPCA plus LDA: A complete kernel Fisher discriminant framework for feature extraction and recognition. *IEEE Transaction on Pattern Analysis and Machine Intelligence, 27*(2), 230-244.

Yang, M. H. (2002). Kernel eigenfaces vs. kernel fisherfaces: Face recognition using kernel methods. In *Proceedings of the Fifth IEEE International Conference on Automatic Face and Gesture Recognition (RGR'02,* 215-220.

Yu, H., & Yang, J. (2001). A direct LDA algorithm for high-dimensional data – with application to face recognition. *P.*

Chapter XI

2D Image Matrix-Based Discriminator

ABSTRACT

This chapter presents two straightforward image projection techniques — two-dimensional (2D) image matrix-based principal component analysis (IMPCA, 2DPCA) and 2D image matrix-based Fisher linear discriminant analysis (IMLDA, 2DLDA). After a brief introduction, we first introduce IMPCA. Then IMLDA technology is given. As a result, we summarize some useful conclusions.

INTRODUCTION

The conventional PCA and Fisher LDA are both based on vectors. That is to say, if we use them to deal with the image recognition problem, the first step is to transform original image matrices into same dimensional vectors, and then rely on these vectors to evaluate the covariance matrix and determine the projector. Two typical examples, the famous eigenfaces (Turk & Pentland, 1991a, 1991b) and fisherfaces (Swets & Weng, 1996; Belhumeur, Hespanha, & Kriegman, 1997) both follow this strategy. The drawback of this strategy is obvious. For instance, considering an image of 100×100 resolution, its corresponding vector is 10,000-dimensional. To perform K-L transform or Fisher linear discriminant on basis of such high-dimensional image vectors is a time-consuming process. What's more, the high dimensionality usually leads to singularity of the within-class covariance matrix, which causes trouble for calculation of optimal discriminant vectors (projection axes).

In this chapter, we will develop two straightforward image projection techniques; that is, 2DPCA and *2DLDA,* to overcome the weakness of the conventional PCA and LDA as applied in image recognition. Our main idea is to directly construct three image covariance matrices, including *image between-class*, *image within-class* and *image total scatter matrices*; and then, based on them, perform PCA or Fisher LDA. Since the scale of image covariance matrices is same as that of images and the within-class image covariance matrix is usual nonsingular, thus, the difficulty resulting from high dimensionality and singular case are artfully avoided. We will outspread our idea in the following sections.

2D IMAGE MATRIX-BASED PCA

IMPCA Method

Differing from PCA and KPCA, IMPCA, which is also called 2DPCA (Yang & Yang, 2002; Yang, Zhang, Frangi, & Yang, 2004), is based on 2D matrices rather than 1D vectors. This means that we do not need to transform an image matrix into a vector in advance. Instead, we can construct an *image covariance matrix* directly using the original image matrices, and then use it as a generative matrix to perform principal component analysis.

The *image covariance* (*scatter*) *matrix* of 2DPCA is defined by:

$$\mathbf{G}_t = E(\mathbf{A} - E\mathbf{A})^T(\mathbf{A} - E\mathbf{A}) \tag{11.1}$$

where $\mathbf{A}$ is an $m \times n$ random matrix representing a generic image. Each training image is viewed as a sample generated from the random matrix $\mathbf{A}$. It is easy to verify that $\mathbf{G}_t$ is an $n \times n$ non-negative definite matrix by its construction.

We can evaluate $\mathbf{G}_t$ directly using the training image samples. Suppose that there are a total of M training image samples, the j_{th} training image is denoted by an $m \times n$ matrix $\mathbf{A}_j$ ($j = 1, 2, \ldots, M$), and the mean image of all training samples is denoted by $\bar{\mathbf{A}}$. Then, $\mathbf{G}_t$ can be evaluated by:

$$\mathbf{G}_t = \frac{1}{M}\sum_{j=1}^{M}(\mathbf{A}_j - \bar{\mathbf{A}})^T(\mathbf{A}_j - \bar{\mathbf{A}}) \tag{11.2}$$

The projection axes of 2DPCA, $\mathbf{X}_1, \ldots, \mathbf{X}_d$, are required to maximize the total scatter criterion $J(\mathbf{X}) = \mathbf{X}^T \mathbf{G}_t \mathbf{X}$ and satisfy the orthogonal constraints; that is:

$$\begin{cases} \{\mathbf{X}_1, \cdots, \mathbf{X}_d\} = \arg\max J(\mathbf{X}) \\ \mathbf{X}_i^T \mathbf{X}_j = 0,\ i \neq j,\ i, j = 1, \ldots, d \end{cases} \tag{11.3}$$

Actually, the optimal projection axes, $\mathbf{X}_1, \ldots, \mathbf{X}_d$, can be chosen as the orthonormal eigenvectors of $\mathbf{G}_t$ corresponding to the first d largest eigenvalues.

The optimal projection vectors of 2DPCA, $\mathbf{X}_1, \ldots, \mathbf{X}_d$, are used for image feature extraction. For a given image sample **A**, let:

$$\mathbf{Y}_k = (\mathbf{A} - \bar{\mathbf{A}})\mathbf{X}_k, \; k = 1, 2, \ldots, d \tag{11.4}$$

Then, we obtain a family of projected feature vectors, $\mathbf{Y}_1, \ldots, \mathbf{Y}_d$, which are called the *principal component* (*vectors*) of the image sample **A**. This set of principal component vectors is viewed as a representation of the image sample **A**. Note that each *principal component* of 2DPCA is a *vector*, whereas the *principal component* of PCA is a *scalar*.

The obtained principal component vectors can be combined to form an $m \times d$ matrix $\mathbf{B} = [\mathbf{Y}_1, \ldots, \mathbf{Y}_d]$, which is called the *feature matrix* of the image sample **A**. The classification will rely on the *feature matrices* of images. The similarity measure (distance) between two feature matrices, $\mathbf{B}_i = [\mathbf{Y}_1^{(i)}, \mathbf{Y}_2^{(i)}, \ldots, \mathbf{Y}_d^{(i)}]$ and $\mathbf{B}_j = [\mathbf{Y}_1^{(j)}, \mathbf{Y}_2^{(j)}, \ldots, \mathbf{Y}_d^{(j)}]$, can be given by:

$$d(\mathbf{B}^{(i)}, \mathbf{B}^{(j)}) = \sum_{k=1}^{d} \left\| \mathbf{Y}_k^{(i)} - \mathbf{Y}_k^{(j)} \right\|_2 \tag{11.5}$$

where $\left\| \mathbf{Y}_k^{(i)} - \mathbf{Y}_k^{(j)} \right\|_2$ denotes the Euclidean distance between the two principal component vectors $\mathbf{Y}_k^{(i)}$ and $\mathbf{Y}_k^{(j)}$. That is to say, the summated Euclidean distance is adopted to measure the similarity of two sets of principal component vectors corresponding to two image patterns.

IMPCA-Based Image Reconstruction

Like PCA, 2DPCA allows the reconstruction of the original image pattern by combining its principal component vectors and the corresponding eigenvectors.

Suppose the orthonormal eigenvectors corresponding to the first d largest eigenvectors of the image covariance matrix $\mathbf{G}_t$ are $\mathbf{X}_1, \ldots, \mathbf{X}_d$. After the image samples are projected onto these axes, the resulting principal component vectors are $\mathbf{Y}_k = \mathbf{A}\mathbf{X}_k$ ($k = 1, 2, \ldots, d$). Let $\mathbf{V} = [\mathbf{Y}_1, \ldots, \mathbf{Y}_d]$ and $\mathbf{U} = [\mathbf{X}_1, \ldots, \mathbf{X}_d]$, then:

$$\mathbf{V} = (\mathbf{A} - \bar{\mathbf{A}})\mathbf{U} \tag{11.6}$$

Since $\mathbf{X}_1, \ldots, \mathbf{X}_d$ are orthonormal, from Equation 11.6, it is easy to obtain the reconstructed image of sample **A**:

$$\tilde{\mathbf{A}} = \bar{\mathbf{A}} + \mathbf{V}\mathbf{U}^T = \bar{\mathbf{A}} + \sum_{k=1}^{d} \mathbf{Y}_k \mathbf{X}_k^T \tag{11.7}$$

Let $\tilde{\mathbf{A}}_k = \mathbf{Y}_k \mathbf{X}_k^T$ ($k = 1, 2, \ldots, d$), which is of the same size as image **A**, and represents the *reconstructed sub-image* of **A**. That is, image **A** can be approximately reconstructed by adding up the first d sub-images. In particular, when the selected number of principal component vectors $d = n$ (n is the total number of eigenvectors of $\mathbf{G}_t$), we have $\tilde{\mathbf{A}} = \mathbf{A}$;

that is, the image is completely reconstructed by its principal component vectors without any loss of information. Otherwise, if $d < n$, the reconstructed image $\tilde{\mathbf{A}}$ is an approximation for $\mathbf{A}$.

Relationship to PCA

In the 2DPCA method, we use the image matrices of the training samples to construct the image covariance matrix $\mathbf{G}_t$. In particular, when the training images, a set of $m \times n$ matrices, degenerate into $1 \times n$ row vectors, the image covariance matrix $\mathbf{G}_t$ becomes the covariance matrix of standard PCA. At the same time, the principal component vectors of 2DPCA, obtained from Equation 11.4, degenerate into values that are actually the principal components of PCA.

Consequently, standard PCA is a special case of 2DPCA. In other words, 2DPCA can be viewed as a generalization of standard PCA.

Minimal Mean-Square Error Property of IMPCA

In this section, we will address the question: Why do we choose the eigenvector system $\mathbf{G}_t$ rather than other orthogonal vector system to expand the images? The physical meaning of the 2DPCA-based image expansion (representation) will be revealed in theory; that is, the mean-square approximation error (in the sense of the matrix Frobenius norm) is proven to be minimal when the image patterns are represented by a small number of principal component vectors generated from 2DPCA.

Definition 11.1 (Golub & Loan, 1996). The Frobenius norm of a matrix $\mathbf{A} = [a_{ij}]_{m \times n}$ is defined by:

$$\| \mathbf{A} \|_F = \sqrt{\sum_{i=1}^{m} \sum_{j=1}^{n} a_{ij}^2}$$

Since the space $\mathbb{R}^{m \times n}$ is isomorphic to the space $\mathbb{R}^{mn}$, the above definition of the matrix Frobenius norm is equivalent to the definition of the vector 2-norm.

Lemma 11.1. If $\mathbf{A} \in \mathbb{R}^{m \times n}$, then $\| \mathbf{A} \|_F^2 = \sigma_1^2 + \sigma_2^2 + \cdots + \sigma_r^2$, where $\sigma_1, \sigma_2, \ldots, \sigma_r$ are the non-zero singular values of $\mathbf{A}$, and $r = \mathbf{rank}(\mathbf{A})$.

Theorem 11.1 (SVD theorem; Golub & Loan, 1996). Suppose $\mathbf{A}$ *is a real* m *by* n *matrix and* $r = \mathbf{rank}(\mathbf{A})$*, then, there exist two orthogonal matrices* $\mathbf{U} \in \mathbb{R}^{m \times m}$ *and* $\mathbf{V} \in \mathbb{R}^{n \times n}$*, such that:*

$$\mathbf{U}^T \mathbf{A} \mathbf{V} = \mathrm{diag}(\sigma_1, \sigma_2, \ldots, \sigma_r, 0, \ldots, 0) \in \mathbb{R}^{m \times n} \tag{11.8}$$

where $\sigma_i (i = 1, \ldots, r)$ *are the non-zero singular values of* $\mathbf{A}$*, and* $\sigma_i^2 (i = 1, \ldots, r)$ *are the non-zero eigenvalues of* $\mathbf{A}^T\mathbf{A}$ *and* $\mathbf{A}\mathbf{A}^T$*.*

Lemma 11.2. Suppose matrix $\mathbf{B} \in \mathbb{C}^{n\times n}$ (the complex $n \times n$ space), then the trace of **B** satisfies $\mathbf{tr}(\mathbf{B}) = \lambda_1 + \lambda_2 + \ldots + \lambda_n$, where $\lambda_1, \lambda_2, \ldots, \lambda_n$ are eigenvalues of **B**.

Lemma 11.3. If $\mathbf{A} \in \mathbb{R}^{m\times n}$, then $\| \mathbf{A} \|_F^2 = \mathrm{tr}(\mathbf{A}^T\mathbf{A}) = \mathrm{tr}(\mathbf{A}\mathbf{A}^T)$.

Proof: It follows by Lemma 1 that $\| \mathbf{A} \|_F^2 = \sigma_1^2 + \sigma_2^2 + \ldots + \sigma_r^2$, where $\sigma_1, \sigma_2, \ldots, \sigma_r$ are non-zero singular values of **A**.

Also, it follows by Lemma 2 and Theorem 1 that $\mathbf{tr}(\mathbf{A}^T\mathbf{A}) = \mathbf{tr}(\mathbf{A}\mathbf{A}^T) = \sigma_1^2 + \sigma_2^2 + \ldots + \sigma_r^2$.

So, $\| \mathbf{A} \|_F^2 = (\mathbf{A}^T\mathbf{A}) = \mathbf{tr}(\mathbf{A}\mathbf{A}^T) = \sigma_1^2 + \sigma_2^2 + \ldots + \sigma_r^2$.

Assume that **A** is an $m \times n$ random image matrix. Without loss of generality, the expectation of image samples generated from **A** is supposed to be zero; that is, $E\mathbf{A} = 0$, in the following discussion since it is very easy to centralize image **A** by $\mathbf{A} - E\mathbf{A}$ if $E\mathbf{A} \neq 0$.

Suppose that in $\mathbb{R}^n$, we are given an arbitrary set of vectors $\mathbf{u}_1, \mathbf{u}_2, \ldots, \mathbf{u}_n$ which satisfy:

$$\mathbf{u}_i^T\mathbf{u}_j = \begin{cases} 1 & i = j \\ 0 & i \neq j \end{cases} \tag{11.9}$$

Projecting **A** onto these orthonormal basis vectors $\mathbf{u}_1, \mathbf{u}_2, \ldots, \mathbf{u}_n$, we have:

$$\mathbf{A}\mathbf{u} = \mathbf{v}_j, \quad j = 1, 2, \ldots, n \tag{11.10}$$

Then, the image can be completely recovered by:

$$\mathbf{A} = \sum_{j=1}^{n} \mathbf{v}_j\mathbf{u}_j^T \tag{11.11}$$

If we use the first d components to represent **A**, the reconstructed approximation is:

$$\hat{\mathbf{A}} = \sum_{j=1}^{d} \mathbf{v}_j\mathbf{u}_j^T \tag{11.12}$$

Correspondingly, the reconstruction error image of **A** is:

$$\mathbf{\Delta} = \mathbf{A} - \hat{\mathbf{A}} = \sum_{j=d+1}^{n} \mathbf{v}_j\mathbf{u}_j^T \tag{11.13}$$

And, the reconstruction mean-square error can be characterized by:

$$\varepsilon^2 = E \| \mathbf{\Delta} \|_F^2 = E \| \mathbf{A} - \hat{\mathbf{A}} \|_F^2 \tag{11.14}$$

Theorem 11.2. Suppose $\mathbf{u}_1, \mathbf{u}_2, \ldots, \mathbf{u}_n$ *are the eigenvectors of* $\mathbf{G}_t$ *corresponding to* $\lambda_1, \lambda_2, \ldots, \lambda_n$ *and* $\lambda_1 \geq \lambda_2 \geq \ldots \geq \lambda_n$ *. If we use the first* d *eigenvectors as projection axes and the resulting component vectors* $\mathbf{v}_1, \mathbf{v}_2, \ldots, \mathbf{v}_d$ *to represent* $\mathbf{A}$*, the reconstruction mean-square error can be minimized in the sense of the matrix Frobenius norm, and:*

$$\varepsilon^2 = \sum_{j=d+1}^{n} \lambda_j \tag{11.15}$$

Proof: It follows by Lemma 11.3 that:

$$\begin{aligned}
&\varepsilon^2 = \mathrm{E} \| \mathbf{A} - \hat{\mathbf{A}} \|_F^2 = E\{\mathrm{tr}[(\mathbf{A} - \hat{\mathbf{A}})(\mathbf{A} - \hat{\mathbf{A}})^T]\} \\
&= E\{\mathrm{tr}[(\sum_{j=d+1}^{n} \mathbf{v}_j \mathbf{u}_j^T)(\sum_{j=d+1}^{n} \mathbf{v}_j \mathbf{u}_j^T)^T]\} = E\{\mathrm{tr}[(\sum_{j=d+1}^{n} \mathbf{v}_j \mathbf{u}_j^T)(\sum_{j=d+1}^{n} \mathbf{u}_j \mathbf{v}_j^T)]\} \\
&= E\{\mathrm{tr}(\sum_{j=d+1}^{n} \mathbf{v}_j \mathbf{v}_j^T)\} = E\{\mathrm{tr}[(\mathbf{v}_{d+1}, \ldots, \mathbf{v}_n)(\mathbf{v}_{d+1}, \ldots, \mathbf{v}_n)^T]\} \\
&= E\{\mathrm{tr}[(\mathbf{v}_{d+1}, \ldots, \mathbf{v}_n)^T (\mathbf{v}_{d+1}, \ldots, \mathbf{v}_n)]\} = E\{\mathrm{tr}[(\mathbf{A}\mathbf{u}_{d+1}, \ldots, \mathbf{A}\mathbf{u}_n)^T (\mathbf{A}\mathbf{u}_{d+1}, \ldots, \mathbf{A}\mathbf{u}_n)]\} \\
&= E\{\mathrm{tr}[(\mathbf{u}_{d+1}, \ldots, \mathbf{u}_n)^T \mathbf{A}^T \mathbf{A} (\mathbf{u}_{d+1}, \ldots, \mathbf{u}_n)]\} = E\{\sum_{j=d+1}^{n} \mathbf{u}_j^T (\mathbf{A}^T \mathbf{A}) \mathbf{u}_j\} \\
&= \sum_{j=d+1}^{n} \mathbf{u}_j^T \{E(\mathbf{A}^T \mathbf{A})\} \mathbf{u}_j = \sum_{j=d+1}^{n} \mathbf{u}_j^T \mathbf{G}_t \mathbf{u}_j
\end{aligned} \tag{11.16}$$

To minimize $\sum_{j=d+1}^{n} \mathbf{u}_j^T \mathbf{G}_t \mathbf{u}_j$ under the orthonormal constraint in Equation 11.9, we use the Lagrange multiplier method. Let:

$$L = \sum_{j=d+1}^{n} \mathbf{u}_j^T \mathbf{G}_t \mathbf{u}_j - \lambda_j (\mathbf{u}_j^T \mathbf{u}_j - 1) \tag{11.17}$$

Taking derivative of L with respect to $\mathbf{u}_j$, we have:

$$\frac{\partial L}{\partial \mathbf{u}_j} = (\mathbf{G}_t - \lambda_j \mathbf{I}) \mathbf{u}_j, j = d+1, \ldots, n \tag{11.18}$$

Equating the above derivative to zero, we obtain:

$$\mathbf{G}_t \mathbf{u}_j = \lambda_j \mathbf{u}_j, j = d+1, \ldots, n \tag{11.19}$$

Letting $d = 0$, it follows that $\mathbf{u}_1, \mathbf{u}_2, \ldots, \mathbf{u}_n$ are the eigenvectors of $\mathbf{G}_t$ corresponding to eigenvalues $\lambda_1, \lambda_2, \ldots, \lambda_n$. Suppose the eigenvalues satisfy $\lambda_1 \geq \lambda_2 \geq \ldots \geq \lambda_n$, if we use the first d eigenvectors as projection axes to expand the image $\mathbf{A}$, the reconstruction mean-square error can be minimized and $\varepsilon^2 = \sum_{j=d+1}^{n} \lambda_j$.

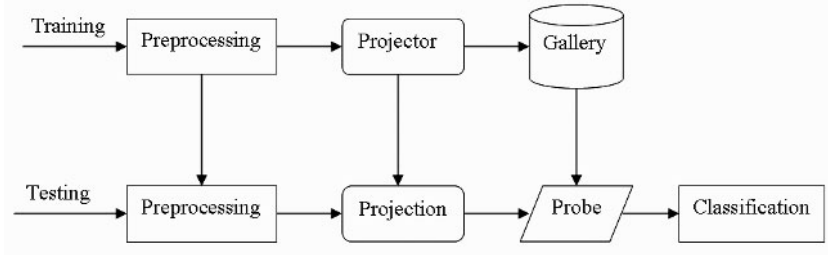
Theorem 11.2 provides a theoretical foundation for the selection of the eigenvector system of $\mathbf{G}_t$ to expand the images. This eigenvector coordinate system yields an optimal image representation framework in the sense of minimal mean-square error. In other words, if we use an alternative set of n-dimensional orthogonal vectors to expand the images, the resulting mean-square error will be larger than (if not equal to) that of 2DPCA.

As we know, PCA can minimize the mean-square approximation error as well when a small number of principal components are used to represent the observations. Now, a question is: Does the minimal mean-square error property of 2DPCA contradict that of PCA? To answer this question, let us first examine the expansion forms of 2DPCA and PCA. 2DPCA aims to find a set of orthonormal projection axes (vectors) in n-dimensional space (n is the number of columns of image matrix) and, image patterns are assumed to be expanded using the form shown in Equation 11.11. Actually, the minimal mean-square error property of 2DPCA-based image representation is with respect to this specific expansion form. More specifically, in $\mathbb{R}^n$, the eigenvectors of $\mathbf{G}_t$ construct an optimal coordinate system, and the expansion coefficients (a small number of principal component vectors) provide the optimal representation for images in the sense of minimal mean-square error, while PCA aims to find a set of orthonormal projection axes in N-dimensional image vector space, and image patterns are assumed to be expanded using the form shown in Equation 11.5. Based on this expansion form, we can say that PCA-based image representation is optimal in the sense of minimal mean-square error. In a word, the minimal mean-square error characteristics of 2DPCA and PCA are based on different expansion forms. They are not contradictory at all.

Comparison of PCA- and 2DPCA-Based Image Recognition Systems

Any statistical pattern recognition system is operated in two modes: training (learning) and testing (classification). For a projection-subspace-based image recognition system, the two modes can be specified as follows: In the training process, the projector (transformation matrix) is obtained by learning from the training sample set, and

Figure 11.1. A sketch of projection-subspace based image recognition system



the given images with class-labels are projected into feature vectors, which represent the known subjects as prototypes to form a *gallery*. In the testing process, a given image of unknown subject is first projected and the resulting feature vector is viewed as a *probe*; then the similarity measure (distance) between the *probe* and any object in the *gallery* is computed (suppose that a nearest-neighbor classifier is used) and followed by a classification decision. Both processes form a general projection-subspace-based image recognition system, which is illustrated in Figure 11.1.

Since it is widely accepted that a KPCA-based system requires more computation and memory due to the additional computation of kernels, for simplicity, we only compare 2DPCA with PCA in this section. The specific comparison involves two key aspects: *computation requirement* and *memory requirement*.

Comparison of Computation Requirement

Now, let us compare the computation requirement involved in PCA- and 2DPCA-based systems. It is natural to resolve the consumed computation into two phases: *training* and *testing*. Also, it is reasonable to use the number of multiplications as a measure to assess the computation involved. In the training phase, the required computation concentrates on two aspects: (a) obtaining the projector by solving an eigen-problem, and (b) projection of images in gallery. First, for PCA, we have to solve an $M \times M$ eigenvalue problem, whose size depends on the number of training samples. Since its computational complexity is $O(M^3)$, we need at least M^3 multiplications to obtain the projector. When the number of training samples becomes large, the computation is considerable. For 2DPCA, the size of eigen-problem is $n \times n$. Since the number of columns, n, is generally much smaller than the number of training samples and keeps invariant with the increase of training samples, less computaion is required by 2DPCA than PCA. Second, in the process of projection of images in gallery, the number of multiplications performed by PCA is $(m \times n) \times d_{PCA}$ while that performed by 2DPCA is $(m \times n) \times d_{2DPCA}$. Generally, the required number of PCA components, d_{PCA}, is much larger than that of 2DPCA for image representation. Therefore, PCA needs more computation than 2DPCA as well in the transformation (projection) process.

In the testing phase, the computation also involves with two aspects: (c) projection of image in probe set, and (d) calculation of the distance (similarity measure). As discussed above, 2DPCA requires less computation than PCA for the projection of images into component features. However, 2DPCA requires more computation than PCA for the calculation of distance between the probe and patterns in gallery. The reason is that the dimension of 2DPCA-transformed features (a set of component vectors), $m \times d_{2DPCA}$, is always larger than that of PCA. As a result, using the summated Euclidean distance shown in Equation 11.5 to calculate similarity measure of two 2DPCA-based feature matrices must be more computationally intensive than using a single Euclidean distance to calculate similarity measure of two PCA-based feature vectors.

Now, let us see the testing phase from a data compression point of view. Since the dimension of 2DPCA transformed features is always larger than that of PCA, it can be said that the *compression rate* of 2DPCA is lower than that of PCA. So, compared to PCA, 2DPCA costs more time for calculation of similarity measure in classification. Nevertheless, this can be compensated for by its *compression speed*; the *compression speed* of 2DPCA is much faster than PCA, since less computation is needed in the projection

Table 11.1. Comparisons of memory and computation requirements of PCA- and 2DPCA-based image recognition systems

Method	Memory Requirements		Computation Requirements	
	Projector	Gallery	Training	Testing
PCA	$(m\times n)\times d_{PCA}$ **Large**	$M_g\times d_{PCA}$ **Small**	a) Solving eigen-problem: M^3, **Large** b) Projection of images in gallery: $M_g\times(m\times n)\times d_{PCA}$, **Large**	c) Projection of probe: $(m\times n)\times d_{PCA}$, **Large** d) Calculation of distance: $M_g\times d_{PCA}$, **Small**
2DPCA	$n\times d_{2DPCA}$ **Small**	$M_g\times d_{2DPCA}\times m$ **Large**	a) Solving eigen-problem: n^3, **Small** b) Projection of images in gallery: $M_g\times(m\times n)\times d_{2DPCA}$, **Small**	c) Projection of probe: $(m\times n)\times d_{2DPCA}$, **Small** d) Calculation of distance: $M_g\times d_{2DPCA}\times m$, **Large**

Note: $d_{PCA} >> d_{2DPCA}$

process. In a word, 2DPCA is still competitive with PCA in the testing phase with respect to computational efficiency.

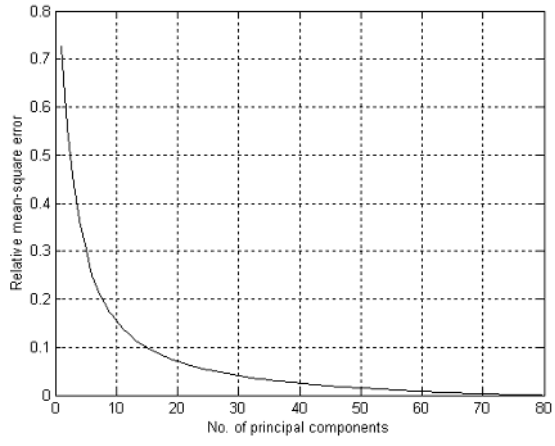
All of the computational requirements (measured by the number of multiplications) by PCA and 2DPCA involved in training and testing phases are listed in Table 11.1. Concerning their comparison, a specific instance will be given in the experiment.

Comparison of Memory Requirement

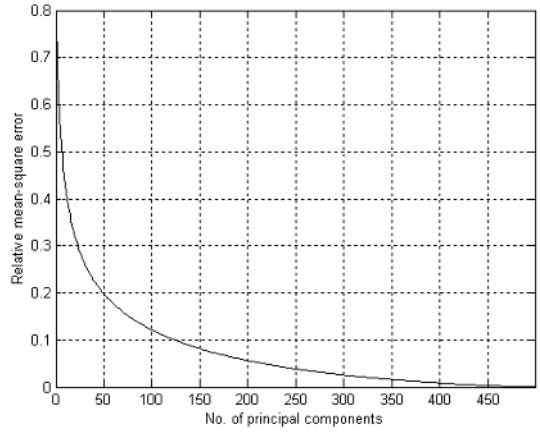
The memory requirement of the system shown in Figure 11.1 mainly depends on the size of projector and the total size of data in gallery. So we try to compare PCA and 2DPCA in terms of these. Here, it should be noted that the size of projector couldn't be neglected since it is rather large for PCA-based image recognition system. This projector of PCA contains d_{PCA} eigen-images (eigenvectors), each of which has the same size with the original image. In comparison, the projector of 2DPCA is much smaller. Its size is only $n \times d_{2DPCA}$, which is much less than a single original image, since $d_{2DPCA} << m$. On the other hand, concerning the total size of data in gallery, the PCA-based system has the advantage over 2DPCA. This is because 2DPCA has a lower compression rate than PCA. The size of projector and data in gallery corresponding to PCA and 2DPCA are listed in Table 11.1.

As far as the total memory requirement is concerned, it can be said that 2DPCA does not require more memory than PCA provided that an image recognition system has a medium-size gallery (the number of subjects); for example, a system containing a few thousand subjects (classes). Concerning this, a specific instance will be given in the experiment section.

Figure 11.2. The curve of relative mean-square error of PCA and 2DPCA



(2DPCA)



(PCA)

Experiments and Analysis

The performance of the proposed 2DPCA method was evaluated using the FERET 1996 standard subset, which was employed originally in the FERET 1996 tests. In this subset, the basic gallery contains 1,196 face images. There are four sets of probe images compared to this gallery: the *fafb* probe set contains 1,195 images of subjects taken at the same time as the gallery images but with different facial expression; the *fafc* probe set contains 194 images of subjects under significantly different lighting conditions; the *Duplicate I* probe set contains 722 images of subjects taken between 1 minute and 1,031 days after the gallery image was taken; the *Duplicate II* probe set is a subset of the

duplicate I set, containing 234 images taken at least 18 months after the gallery images. In our experiments, the face portion of each original image is cropped based on the location of eyes and resized to an image of 80×80 pixels. The resulting image is then preprocessed by a histogram equalization algorithm.

In our first test, the first 500 images are selected in turn from the gallery to form the training sample set. PCA and 2DPCA, respectively, are employed for image representation. Since there are 500 training samples, there exist 499 eigenvectors (eigenfaces) corresponding to non-zero eigenvalues for PCA. For 2DPCA, we can obtain 80 eigenvectors in total because the size of the *image covariance matrix* $\mathbf{G}_t$ is 80×80. For both methods, if we use the first d components to represent the image, the relative mean-square error can be calculated by:

$$\varepsilon_r^2 = \sum_{j=d+1}^{L} \lambda_j \Big/ \sum_{j=1}^{L} \lambda_j$$

where L is the total number of non-zero eigenvalues. The curve of the relative mean-square error corresponding to PCA and 2DPCA are shown in Figure 11.2. From Figure 11.2, we can see that the curve of 2DPCA is similar to that of PCA in form. The relative mean-square error of 2DPCA degrades fast when the number of principal component vectors increases from 1 to 10. After that, the degradation rate diminishes gradually. This fact indicates that the energy of images is concentrated on a small number of principal component vectors. So, it is reasonable to use these component vectors for image representation.

To visualize 2DPCA- and PCA-based image representation, some examples of the reconstructed images based on PCA and on 2DPCA corresponding to one original image are shown in Figure 11.3. For PCA, the component number d varies from 10 to 100, with an interval of 10. For 2DPCA, we adopt Equation 11.7, and d varies from 3 to 12. Based on the given principal component number, the corresponding relative mean-square errors of PCA and 2DPCA are listed in Table 11.2. It is obvious that the relative mean-square error of 2DPCA is always larger than that of PCA. Despite this, it appears that 2DPCA-based reconstructed images are more "like" the original image than PCA-based reconstructed images.

It should be pointed out that the reconstruction mechanisms of PCA and 2DPCA are different. For PCA, the eigenvectors themselves can be exhibited as images, which

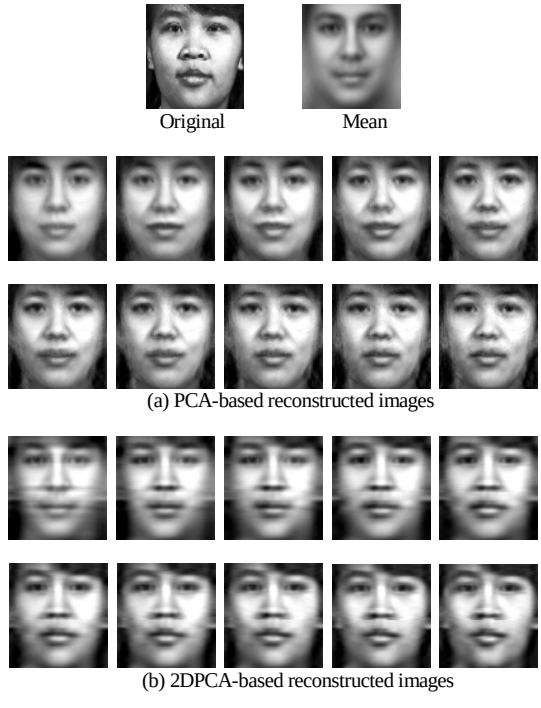
Table 11.2. Relative mean-square error corresponding to the synthetic images in Figure 11.3 using 2DPCA and PCA

PCA	d	10	20	30	40	50	60	70	80	90	100
	ε_r^2	0.429	0.320	0.263	0.226	0.198	0.177	0.159	0.145	0.132	0.121
2DPCA	d	3	4	5	6	7	8	9	10	11	12
	ε_r^2	0.438	0.362	0.302	0.250	0.221	0.193	0.172	0.155	0.139	0.126

are generally referred to as *eigenimages* (also called PCA-based reconstructed sub-images in this chapter). The image is synthesized by a weighted combination of these *eigenimages* and mean image. Differing from PCA, the eigenvectors of 2DPCA are *n*-dimensional-only vectors, which cannot be exhibited as images. But, for a given image **A**, if we combine its principal component vectors and eigenvectors by $\tilde{\mathbf{A}}_k = \mathbf{Y}_k \mathbf{X}_k^T$ ($k = 1, 2, \ldots, d$), a set of images, called 2DPCA-based reconstructed sub-images, are obtained. Then, the image **A** can be synthesized by a summation of these sub-images and mean image. Figure 11.4 shows the first 12 eigenimages of PCA and reconstructed sub-images of 2DPCA corresponding to the original image. Note that to fully exhibit the information (especially that contained in the negative elements) within these sub-images, we perform the following transformation in prior: Every element of the image is subtracted by the minimal element value and then normalized by dividing the maximal value.

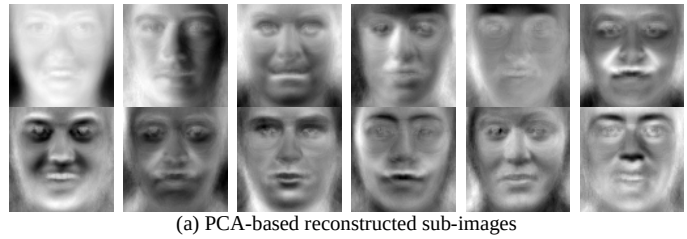
It is apparent that the reconstructed sub-images of PCA and 2DPCA are very different. The sub-images (eigenimages) of PCA are face-like, which represent the global information of images, while the sub-images of 2DPCA are not face-like at all. It seems

Figure 11.3. Examples of PCA- and 2DPCA-based reconstructed images

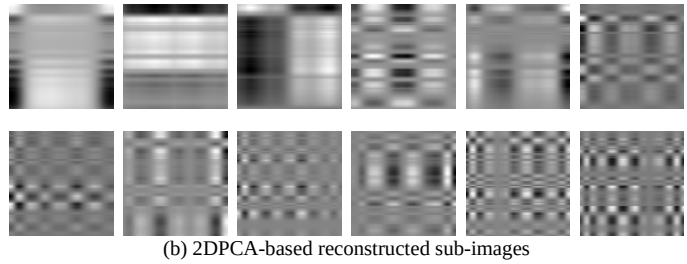


(a) PCA-based reconstructed images using d components, where d varies from 10 to 100 with an interval of 10 (from left to right, top to bottom); (b) 2DPCA-based reconstructed images using d component vectors, where d varies from 3 to 12 (from left to right, top to bottom)

Figure 11.4. Examples of PCA- and 2DPCA-based reconstructed sub-images



(a) PCA-based reconstructed sub-images



(b) 2DPCA-based reconstructed sub-images

(a) PCA-based reconstructed sub-images (eigenfaces) corresponding to 12 largest eigenvalues (from left to right, top to bottom); (b) 2DPCA-based reconstructed sub-images corresponding to 12 largest eigenvalues (from left to right, top to bottom)

that they contain the local details from different levels. Based on the mean image, 2DPCA synthesizes images by modifying the local details step by step with the increase of sub-images. Differently, PCA synthesizes images by combining a set of eigenimages and the mean image. So, in contrast to PCA, 2DPCA-based image reconstruction should own more local characteristics. This gives a reasonable explanation of why 2DPCA-based reconstructed images appear more "like" the original image.

Now, let us compare the discriminatory power of PCA, KPCA and 2DPCA. Note that in the algorithm of KPCA, two popular kernels are involved. One is the second-order polynomial kernel $k(\mathbf{x}, \mathbf{y})=(\mathbf{x}\cdot\mathbf{y}+1)^2$, and the other is Gaussian kernel $k(\mathbf{x},\mathbf{y}) = \exp\left(\| \mathbf{x}-\mathbf{y} \|^2 / \delta\right)$. For the Gaussian kernel, the parameter δ is chosen as $0.3 \times N$, where N is the dimension of *input space*. This parameter selection turned out to be effective in practical application (Schölkopf & Smola, 2002). For PCA and KPCA, 200 principal components are extracted to represent a face (this is consistent with the PCA-based baseline system in Phillips, Moon, Rizvi, and Rauss (2000)); while for 2DPCA, 10 principal component vectors are extracted. Finally, a nearest-neighbor classifier is employed for classification. Note that the Euclidean distance is used to measure PCA and KPCA features, and the summated Euclidean distance defined in Equation 11.5 is used for 2DPCA features. The recognition rate and the total CPU time consumed for training and testing are listed in Tables 11.3 and 11.4.

From Table 11.3, it can be seen that 2DPCA outperforms PCA and KPCA on three probe sets: *fafb* and *Duplicates I and II*. On the *fafc* probe set, PCA and KPCA (using

Table 11.3. Recognition rates (%) of PCA, KPCA and 2DPCA on four Probe sets of FERET 1996 using the first 500 images in Gallery as training set

Method	*fafb* (1195)	*fafc* (194)	*Duplicate I* (722)	*Duplicate II* (234)	Total (2345)
PCA	78.2	18.0	32.1	10.3	52.25
KPCA (P)	76.5	15.5	31.6	9.8	50.97
KPCA (G)	78.2	18.0	32.5	10.3	52.37
2DPCA	80.1	17.5	35.3	12.0	54.33

Note: In the above table, (P) denotes polynomial kernel and (G) denotes Gaussian kernel. The same notations will be used in the following tables and figures.

Gaussian kernel) perform a little better than 2DPCA; the errors of 2DPCA (160 errors) are only one more than that of PCA (159 errors). Taking the four probe sets as a whole testing set, the total recognition rate of 2DPCA is higher than those of PCA and KPCA.

Table 11.4 shows that 2DPCA is much faster than PCA and KPCA for training and slightly faster for testing. Since KPCA requires more computation for calculating the inner products (in form of kernel), it is easy to understand why it is more time consuming than PCA. Now, let us compare PCA and 2DPCA on memory and computation requirements based on the earlier discussion. The comparison results are exhibited in Table 11.5. From this table, we can see that: (1) 2DPCA needs less total memory requirement than PCA. Although the 2DPCA-transformed data in the gallery is larger than the PCA-transformed data, its total memory requirement is still competitive with PCA, since its projector is much smaller. 2) 2DPCA needs less computation requirement for training and for testing. Compared to 2DPCA, PCA requires more than 20 times computation in the training phase. Also, PCA needs more computation than 2DPCA for testing a given probe. The above two aspects are enough to explain why 2DPCA is faster than PCA both for training and testing.

In the above test, the first 500 images in the gallery are selected for training. The experimental results show that 2DPCA is more effective (except one case) than PCA and KPCA. Now, a question is: Are these results with respect to the choice of training set? In other words, if another set of training samples are chosen at random, would we obtain

Table 11.4. The total CPU time(s) for training and testing on FERET 1996 subset (CPU: Pentium IV 1.7GHz, RAM: 1Gb)

Method	Time for Training	Time for Testing			
		fafb (1195)	*fafc* (194)	*Duplicate I* (722)	*Duplicate II* (234)
PCA	384.78	491.76	81.24	299.95	93.90
KPCA (P)	460.87	550.66	93.22	347.86	111.74
KPCA (G)	527.84	705.29	117.26	427.34	136.75
2DPCA	28.76	460.86	78.29	278.16	89.29

similar results? To answer this question, let us run the image recognition system 10 times. Note that each time the training sample set (containing 500 images) is selected *at random* from the gallery, so that the training sample sets are different for 10 tests. The recognition rates corresponding to three methods across 10 tests are illustrated in Figure 11.5. Also, for each method mentioned above, the average recognition rate and standard deviationacross 10 tests are listed in Table 11.6.

Figure 11.5 shows that 2DPCA outperforms PCA and KPCA (using two kinds of kernels) for all tests and all probe sets. These results are consistent with those in Table 11.3 on the whole, except for one case on the probe set *fafc*. Concerning this, the present results should be more convincing than those in Table 11.3, since they are based on more tests and the training sets are chosen at random. So, the result on probe set *fafc* in Table 11.3 can be viewed as an exception. Actually, 2DPCA is superior to PCA and KPCA not only at its recognition rate but also at its robustness. From Table 11.6, we can see that the standard deviation of 2DPCA is much smaller than others' for each probe set. This indicates that the performance of 2DPCA is more insensitive to the variation of training sets.

Table 11.5. Comparisons of memory and computation requirements of PCA- and 2DPCA-based image recognition systems using FERET 1996 subset

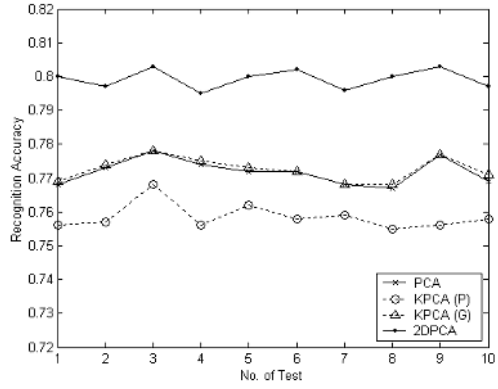
Method	Memory Requirements			Computation Requirements	
	Projector	Gallery	Total	Training	Testing
PCA	$80^2 \times 200$ = **1,280,000**	1196×200 = **239,200**	***1,519,200***	(a) Solving eigen-problem: 500^3 = **125,000,000** (b) Projection of images in gallery: $1196 \times 80^2 \times 200$ = **1,530,880,000** Total = ***1,655,880,000***	(c) Projection of probe: $80^2 \times 200$ = **1,280,000** (d) Calculation of distance: 1196×200 = **239,200** Total = ***1,519,200***
2DPCA	80×10 = **800**	1196×80×10 = **956,800**	***957,600***	(a) Solving eigen-problem: 80^3 = **512,000** (b) Projection of images in gallery: $1196 \times 80^2 \times 10$ = **76,544,000** Total =***77,056,000***	(c) Projection of probe: $80^2 \times 10$ = **64,000** (d) Calculation of distance: 1196×80×10 = **956,800** Total = ***1,020,800***

Table 11.6. Average recognition rates (%) and standard deviation of PCA, KPCA and 2DPCA on four Probe sets of FERET 1996 across 10 random tests

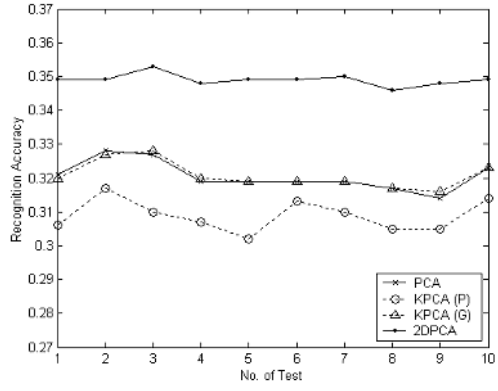
Method (standard deviation)	*fafb* (1195)	*fafc* (194)	*Duplicate I* (722)	*Duplicate II* (234)	Total (2345)
PCA	77.18 ± 0.38	14.84 ± 1.30	32.06 ± 0.43	10.15 ± 0.61	51.442
KPCA (P)	75.85 ± 0.39	12.21 ± 1.61	30.89 ± 0.47	9.50 ± 0.83	50.122
KPCA (G)	77.25 ± 0.36	14.68 ± 1.22	32.08 ± 0.40	10.15 ± 0.61	51.471
2DPCA	79.93 ± 0.29	19.35 ± 0.49	34.90 ± 0.18	11.51 ± 0.21	54.227

Table 11.6 also shows the total recognition rate (four probe sets are viewed as a whole testing set) of 2DPCA is higher than others. Based on this result, can we say that 2DPCA is significantly better than PCA and KPCA? Now, let us evaluate this result using McNemar's (Beveridge, She, Draper, & Givens, 2001; Yambor, Draper, & Beveridgem, 2002) significance test. McNemar's test is an accepted method for performance evaluation of face recognition systems. It is essentially a null hypothesis statistical test based on the Bernoulli model. If the resulting *p*-value is below the desired significance level (for example, 0.05), the null hypothesis is rejected and the performance difference between two algorithms are considered to be statistically significant. By this test, we find that 2DPCA is statistically significantly better than PCA and KPCA at a significance level $p = 0.0293$ (one-tailed).

Figure 11.5. Illustration of the recognition rates of PCA, KPCA and 2DPCA across 10 random tests

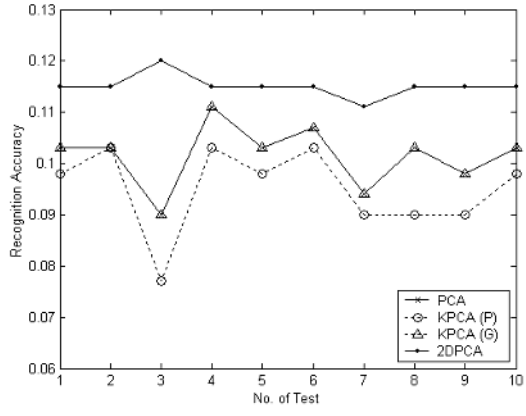


(fafb)

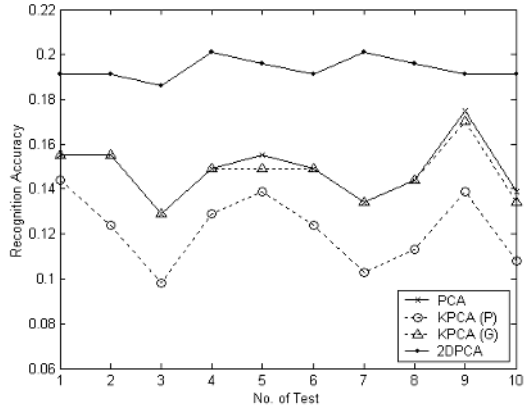


(Duplicate I)

Figure 11.5. Illustration of the recognition rates of PCA, KPCA and 2DPCA across 10 random tests (cont.)



(fafc)



(Duplicate II)

2D IMAGE MATRIX-BASED LDA

Fundamentals

Let X denote an n-dimensional column vector; our idea is to project the image A, an $m \times n$ matrix, onto X by the following linear transformation:

$$Y=AX \tag{11.19}$$

Thus, an m-dimensional projected vector Y is produced, which is called the projected feature vector of image A. Now, the problem is how to find a good image

projection vector X. Intuitively, the maximum between-class scatter and the minimum within-class scatter are expected to being reached after projection. For this sake, we intend to adopt the following criterion (Liu, Cheng, & Yang, 1993):

$$J(X) = \frac{tr(BS_x)}{tr(WS_x)} \tag{11.20}$$

where, BS_x and WS_x denote the between-class scatter matrix and the within-class scatter matrix of the projected feature vectors of the training image samples, and $tr(BS_x)$ denotes the trace of BS_x. Let us now outspread this idea in detail.

Suppose there are L known pattern classes, and a training sample, the j_{th} image in class i is denoted by an $m \times n$ matrix $A_j^{(i)}$, where $i = 1, 2, \ldots, \text{L}$, $j = 1, 2, \ldots, M_i$, M_i denotes the total number of training samples in class i. The mean image of the training samples in class i is:

$$\overline{A}^{(i)} = \frac{1}{M_i} \sum_{j=1}^{M_i} A_j^{(i)} \tag{11.21}$$

The mean image of all training samples is:

$$\overline{A} = \sum_{i=1}^{L} P_i \overline{A}^{(i)} \tag{11.22}$$

where $P_i = (i = 1, 2, \ldots L)$ is the prior probability of the class i.

After the projection of a training image onto X, we get its corresponding feature vector:

$$Y_j^{(i)} = A_j^{(i)} X, \; i = 1, 2, \ldots, \text{L}, \; j = 1, 2, \ldots, M_i \tag{11.23}$$

Suppose the mean vector of projected features in class i and the total mean vector are denoted by $\overline{Y}^{(i)}$ and $\overline{Y}$ respectively; it is easy to get:

$$\overline{Y}^{(i)} = \overline{A}^{(i)} X \tag{11.24}$$

and $$\overline{Y} = \overline{A} X \tag{11.25}$$

Then, the between-class scatter matrix and the within-class scatter matrix of the projected feature vectors can be evaluated by:

$$BS_x = \sum_{i=1}^{L} P_i (\overline{Y}_i - \overline{Y})(\overline{Y}_i - \overline{Y})^T$$

$$= \sum_{i=1}^{L} P_i [(\overline{A}_i - \overline{A})X][(\overline{A}_i - \overline{A})X]^T \tag{11.26}$$

$$WS_x = \sum_{i=1}^{L} P_i \left(\frac{1}{M_i} \sum_{i=1}^{M_i} (Y_j^{(i)} - \bar{Y}^{(i)})(Y_j^{(i)} - \bar{Y}^{(i)})^T \right)$$

$$= \sum_{i=1}^{L} P_i \frac{1}{M_i} \sum_{i=1}^{M_i} [(A_j^{(i)} - \bar{A}^{(i)})X][(A_j^{(i)} - \bar{A}^{(i)})X]^T \quad (11.27)$$

From Equations 11.26 and 11.27, the between-class scatter and the within-class scatter are determined by:

$$tr\,(BS_x) = \sum_{i=1}^{L} P_i[(\bar{A}_i - \bar{A})X]^T[(\bar{A}_i - \bar{A})X]$$

$$= X^T \left(\sum_{i=1}^{L} P_i(\bar{A}_i - \bar{A})^T(\bar{A}_i - \bar{A}) \right) X \quad (11.28)$$

$$tr\,(WS_x) = \sum_{i=1}^{L} P_i \frac{1}{M_i} \sum_{i=1}^{M_i} [(A_j^{(i)} - \bar{A}^{(i)})X]^T[(A_j^{(i)} - \bar{A}^{(i)})X]$$

$$= X^T \left(\sum_{i=1}^{L} P_i \frac{1}{M_i} \sum_{i=1}^{M_i} (A_j^{(i)} - \bar{A}^{(i)})^T(A_j^{(i)} - \bar{A}^{(i)}) \right) X \quad (11.29)$$

Let us define the following matrices:

$$G_b = \sum_{i=1}^{L} P_i(\bar{A}_i - \bar{A})^T(\bar{A}_i - \bar{A}) \quad (11.30)$$

$$G_w = \sum_{i=1}^{L} P_i \frac{1}{M_i} \sum_{i=1}^{M_i} (A_j^{(i)} - \bar{A}^{(i)})^T(A_j^{(i)} - \bar{A}^{(i)}) \quad (11.31)$$

Then $tr\,(BS_x) = X^T G_b X$ (11.32)

and $tr\,(WS_x) = X^T G_W X$ (11.33)

Hence, the criterion in Equation 11.20 can be expressed by:

$$J(X) = \frac{X^T G_b X}{X^T G_w X} \quad (11.34)$$

G_b and G_w are called *image between-class scatter matrix* and *image within-class scatter matrix*. From their definitions, it is easy to verify that they are all $n \times n$ nonnegative

definite matrices. Also, G_w should be positive definite if it is invertible. As a matter of fact, in face recognition problems, G_w is usually invertible unless there is only one training sample in each category. The criterion in Equation 11.34 is called *generalized Fisher criterion.* Liu (Liu, Cheng, & Yang, 1993) has proven that the classical Fisher criterion is a special case of this criterion.

The vector X maximizing the criterion is called generalized Fisher optimal projection direction. Its physical meaning is obvious; that is, after projection of image matrix onto X, the maximal between-class scatter and the minimal within-class scatter are achieved at the same time.

By the way, if we define the *image total scatter matrix* by:

$$G_t = E\{(A - EA)^T (A - EA)\} \tag{11.35}$$

In fact, its evaluation based on the training samples is:

$$G_t = \frac{1}{M}\sum_{i,j}(A_j^{(i)} - \bar{A})^T (A_j^{(i)} - \bar{A}) \tag{11.36}$$

Then, it is easy to verify that $G_t = G_b + G_w$ and the generalized Fisher criterion in Equation 11.34 is equivalent to:

$$J_t(X) = \frac{X^T G_b X}{X^T G_t X} \tag{11.37}$$

Orthogonal IMLDA (O-IMLDA)

The generalized Fisher optimal projection direction can be obtained by calculating the generalized eigenvector corresponding to the largest eigenvalue of the following eigen-equation:

$$G_b \xi = \lambda G_w \xi \tag{11.38}$$

Generally, the single projection axis, even if it is optimal in theory, is not sufficient because much discriminatory information has been lost after image projection onto it alone. Accordingly, Liu (Liu, Cheng, & Yang, 1993) proposed a set of optimal orthogonal discriminant vectors $X_1, \ldots, X_d$ to solve the problem. Liu's idea of finding $X_1, \ldots, X_d$ can be described as follows:

Let X_1 be chosen as generalized Fisher optimal projection direction. Once the projection vectors $X_1, \ldots, X_i$ are determined, the (i - 1)-th projection vector X_{i+1} can be obtained by solving the following optimization problem:

$$\text{Model I} \begin{cases} \max\left(J(X)\right) \\ X_j^T X = 0, j = 1, \ldots, i \\ X \in R^n \end{cases} \tag{11.39}$$

Jin (Jin, Yang, Hu, & Lou, 2001) proposed a lemma that can be modified to solve the problem seen in Model I (11.39).

Lemma 11.4. X_{i+1} is the unit eigenvector corresponding to the largest eigenvalue of the following generalized eigen-equation:

$$B_i G_b X = \lambda G_w X \tag{11.40}$$

where $B_i = I_n - D_i^T (D_i G_w^{-1} D_i^T)^{-1} D_i G_w^{-1}$, $D_i = (X_1, X_2, \ldots, X_i)^T$

Obviously, Liu's optimal image projection vectors $X_1, \ldots, X_d$ satisfy the orthogonal constraints:

$$X_i^T X_j = 0, \quad \forall i \neq j, \quad i, j = 1, \ldots, d \tag{11.41}$$

So, Liu's method is called the orthogonal IMLDA (O-IMLDA).

Uncorrelated IMLDA (U-IMLDA)

Recently, Jin and Yang (Jin, Yang, Hu, & Lou, 2001; Jin, Yang, Tang, & Hu, 2001) presented a set of uncorrelated optimal discriminant vectors and demonstrated that it is more powerful than the set of Foley-Sammon discriminant vectors (Tian, Barbero, Gu, & Lee, 1986). The major difference between these discriminant vectors is that Foley-Sammon discriminant vectors satisfy orthogonal constraints, while Jin's discriminant vectors are subject to the conjugate orthogonal constraints.

Here, we further extend Jin's idea and introduce a new set of uncorrelated optimal image projection vectors (Yang, Yang, Frangi, & Zhang, 2003). The G_t-orthogonal constraints are adopted instead of the orthogonal constraints in Equation 11.41; that is, the uncorrelated optimal image projection vectors $X_1, \ldots, X_d$ are required to satisfy:

$$X_i^T G_t X_j = 0, \quad \forall i \neq j, \quad i, j = 1, \ldots, d \tag{11.42}$$

In fact, they can be derived in this way. X_1 is still chosen as generalized Fisher optimal projection direction; After determining $X_1, \ldots, X_i$, the $(i+1)$-th projection vector X_{i+1} can be obtained by solving the following optimization problem:

$$\text{Model II} \begin{cases} \max\left(J(X)\right) \\ X_j^T G_t X = 0, j = 1, \ldots, i \\ X \in R^n \end{cases} \tag{11.43}$$

To solve this problem, some related theory is first introduced as follows.

Theorem 11.3 (Lancaster & Tismenetsky, 1985). Suppose that G_w is invertible; there exist n eigenvectors $\xi_1, \ldots, \xi_n$ corresponding to eigenvalues $\lambda_1, \ldots, \lambda_n$ of the eigen-equation $G_b \xi = \lambda G_w \xi$, such that:

$$\xi_i^T G_w \xi_j = \begin{cases} 1 & i = j \\ 0 & i \neq j \end{cases} \quad i, j = 1, \ldots, n \tag{11.44}$$

and:

$$\xi_i^T G_b \xi_j = \begin{cases} \lambda_i & i = j \\ 0 & i \neq j \end{cases} \quad i, j = 1, \ldots, n \tag{11.45}$$

Corollary 11.1. The n-dimensional Euclidean space R^n = **span** $\{\xi_1, \ldots, \xi_n\}$.

Since $G_t = G_b + G_w$ by Theorem 11.3, it follows that:

Corollary 11.2. The eigenvectors $\xi_1, \ldots, \xi_n$ of the eigen-equation $G_b\xi = \lambda G_w \xi$ satisfy:

$$\xi_i^T G_t \xi_j = \begin{cases} 1+\lambda_i & i = j \\ 0 & i \neq j \end{cases} \quad i, j = 1, \ldots, n \tag{11.46}$$

Suppose the eigenvalues $\lambda_1, \lambda_2, \ldots, \lambda_n$ of $G_b\xi = \lambda G_w \xi$ satisfy $\lambda_1 \geq \lambda_2 \geq \ldots \geq \lambda_n$; we can draw the following conclusion:

Proposition 11.1. If the first i discriminant vectors have been chosen as $X_1 = \xi_1, \ldots, X_i = \xi_i$, then, the ($i$+1)th optimal discriminant vector X_{i+1} (the solution of Model II) can be selected as ξ_{i+1}.

Proof: By Corollaries 11.1 and 11.2, it follows that the (i+1)-th optimal discriminant vector $X_{i+1} \in$ **span** $\{\xi_{i+1}, \ldots, \xi_n\}$. That is, X_{i+1} can be denoted by $X_{i+1} = c_{i+1}\xi_{i+1} + \ldots + c_n\xi_n$. According to Theorem 11.3, we have:

$$J(X_{k+1}) = \frac{\lambda_{k+1}c_{k+1}^2 + \ldots + \lambda_n c_n^2}{c_{k+1}^2 + \ldots + c_n^2} \leq \lambda_{k+1} \tag{11.47}$$

Since $J(\xi_{k+1})\, J(\xi_{k+1}) = \lambda_{k+1}$, so X_{k+1} can be selected as ξ_{k+1}.

This proposition tells us that the projection vectors of U-IMLDA can be selected as $\xi_1, \ldots, \xi_d$; that is, the G_t-orthogonal eigenvectors corresponding to the first d largest eigenvalues of the generalized eigen-equation $G_b\xi = \lambda G_w \xi$. They can be calculated using the following algorithm.

U-IMLDA Algorithm:

- **Step 1:** Form the image between-class scatter matrix G_b and image within-class scatter matrix G_w according to the definition in Equations 11.30 and 11.31.

- **Step 2:** Work out the pre-whitening transformation matrix W, such that $W^T G_w W = I$.
- **Step 3:** Let $\tilde{G}_b = W^T G_b W$, and calculate the orthonormal eigenvectors $\xi_1, \ldots, \xi_n$ of $\tilde{G}_b$. Suppose the associated eigenvalues satisfy $\lambda_1 \geq \ldots \geq \lambda_n$; then the optimal projection axes of U-IMLDA are $X_1 = W\xi_1, \ldots, X_d = W\xi_d$.
  The optimal image projection vectors $X_1, \ldots, X_d$ are used for feature extraction. Let:

$$Y_k = AX_k, k = 1, 2, \ldots, d \tag{11.48}$$

Then, we get a family of image projected feature vectors $Y_1, \ldots, Y_d$, which are used to form an $N=md$ dimensional projected feature of image A as follows:

$$Y = \begin{bmatrix} Y_1 \\ Y_2 \\ \vdots \\ Y_d \end{bmatrix} = \begin{bmatrix} AX_1 \\ AX_2 \\ \vdots \\ AX_d \end{bmatrix} \tag{11.49}$$

Thus, the image space is transformed into a projected feature space (Y-space).

Correlation Analysis

For 1D random variables ξ and η , we know that their covariance is defined by $E\{(\xi - E\xi)(\eta - E\eta)\}$. Now, let us generalize this concept to the n-dimensional case. Suppose ξ and η are n-dimensional random column vectors; define their covariance as:

$$Cov(\xi, \eta) = E\{(\xi - E\xi)^T (\eta - E\eta)\} \tag{11.50}$$

Note that the covariance of n-dimensional random vectors defined above is still a scalar (not a matrix). Obviously, when ξ and η both degenerate into 1D random variables, their covariance defined in Equation 11.50 is equivalent to $E\{(\xi - E\xi)(\eta - E\eta)\}$.

Accordingly, we can define the correlation coefficient between ξ and η as follows:

$$\rho(\xi, \eta) = \frac{Cov(\boldsymbol{\xi}, \boldsymbol{\eta})}{\sqrt{Cov(\boldsymbol{\xi}, \boldsymbol{\xi}) \cdot Cov(\boldsymbol{\eta}, \boldsymbol{\eta})}} \tag{11.51}$$

By Equation 11.48, $Y_k = AX_k$, $(k = 1, 2, \ldots d)$. Thus, the covariance of two projected feature vectors Y_i and Y_j is:

$$Cov(Y_i, Y_j) = E\{(Y_i - EY_i)^T (Y_i - EY_j)\}$$

$$= E\{[AX_i - E(AX_i)]^T [AX_j - E(AX_j)]\} \tag{11.52}$$

$$= X_i^T \{E[(A - EA)^T (A - EA)]\} X_j$$

It follows from Equation 11.35 that:

$$Cov(Y_i, Y_j) = X_i^T G_t X_j \tag{11.53}$$

So, the correlation coefficients between two projected feature vectors Y_i and Y_j can be evaluated by:

$$\rho(Y_i, Y_j) = \frac{X_i^T G_t X_j}{\sqrt{X_i^T G_t X_i}\sqrt{X_j^T G_t X_j}} \tag{11.54}$$

Since the proposed projection vectors are selected as $\xi_1, \ldots, \xi_d$, which are the G_t-orthogonal eigenvectors of $G_b\xi = \lambda G_w \xi$, by Corollary 2, it is easy to draw the conclusion:

Proposition 11.2. Suppose the U-IMLDA image projection vectors $X_1 = \xi_1, \ldots, X_d = \xi_d$; then their corresponding projected feature vectors $Y_i = AX_i (i = 1, 2, \ldots, d)$ satisfy:

$Cov(Y_i, Y_j) = 0, i \neq j, i, j = 1, \ldots, d$, which means:

$$\rho(Y_i, Y_j) = \begin{cases} 1 & i = j \\ 0 & i \neq j \end{cases} \quad i, j = 1, \ldots, d$$

Proposition 11.2 indicates that the projected feature vectors $Y_1, \ldots, Y_d$ resulting from U-IMLDA are mutually uncorrelated. However, those projected feature vectors extracted by O-IMLDA generally do not satisfy this property (the experimental result shown in Table 11.10 can demonstrate this). The uncorrelated property of U-IMLDA is due to the G_t-orthogonal constraints of image projection vectors, which is why we adopt this kind of constraints instead of Liu's orthogonal constraints.

Experiments and Analysis

The proposed method was first tested on the ORL database that contains a set of face images taken at the Olivetti Research Laboratory in Cambridge, United Kingdom. There are 10 different images for each of 40 individuals. In this experiment, we used the first five images of each person for training and the remaining five for testing. Thus, the total amount of training samples and testing samples are both 200. The number of selected image projection vectors varied from 2 to 10, and the selected image projection vectors of O-IMLDA and U-IMLDA, respectively, are used for feature extraction. Then, in each projected feature space (Y-space), a minimum-distance classifier and nearest-neighbor classifier are respectively employed. The corresponding recognition rates are shown in Table 11.7. Moreover, the eigenfaces (Turk & Pentland, 1991a, 1991b) and Fisherfaces (Belhumeur, Hespanha, & Kriegman, 1997) were used for feature extraction as well, and their maximal recognition accuracy under a nearest-neighbor classifier and the time

Table 11.7. Recognition rates (%) of Liu's O-IMLDA and the proposed U-IMLDA (Algorithm 1) on the ORL database

Projection Vector Number		2	3	4	5	6	7	8	9	10
Minimum Distance	O-IMLDA	83.0	87.0	86.0	87.0	87.0	87.0	87.0	87.0	87.0
	U-IMLDA	87.0	87.5	88.5	89.0	89.0	90.0	90.5	91.0	90.5
Nearest Neighbor	O-IMLDA	88.0	92.0	93.5	93.0	93.0	93.5	93.0	93.0	93.0
	U-IMLDA	93.5	93.5	95.5	95.5	95.5	94.5	95.0	95.5	95.0

consumed for feature extraction and classification of 200 testing images are listed in Table 11.8.

In Table 11.7, it is obvious that the recognition rate of the proposed method U-IMLDA is higher than Liu's method O-IMLDA irrespective of the variation of the projection vector number. And the maximal recognition rate of U-IMLDA can reach 95.5% with a nearest-neighbor classifier.

Table 11.8 shows that the proposed U-IMLDA outperforms the eigenfaces and fisherfaces methods. And, U-IMLDA is as efficient as O-IMLDA and much faster than eigenfaces and fisherfaces with respect to the speed of feature extraction. This is because U-IMLDA is image matrix-based while eigenfaces and fisherfaces are image vector-based. More specifically, when eigenfaces and fisherfaces are used for image feature extraction, they need to convert the 92×112 image matrix into 10,304-dimensional image vector and calculate the eigenvectors of a 200×200 total scatter matrix; whereas U-IMLDA can perform feature extraction directly based on image matrices and only needs to deal with 92×92 image scatter matrices.

Why is U-IMLDA better than O-IMLDA? To answer the question, let us observe the values of the generalized criterion function in Equation 11.30 corresponding to each image projection vector, which are listed in Table 11.9. To our surprise, the value of the generalized Fisher criterion corresponding to each projection vector of O-IMLDA (except for the first one) is much larger than that of U-IMLDA. According to the physical meaning of the generalized Fisher criterion, the larger the ratio is, the more discriminatory the corresponding projection vector should be. It seems as if O-IMLDA should do better, but the fact is not so. Why?

We know that the projected feature vectors $Y_1, \ldots, Y_d$ resulting from the proposed method are mutually uncorrelated. However, those projected feature vectors extracted by O-IMLDA generally do not satisfy this property. Their correlation coefficients can be calculated according to Equation 11.53 and are listed in Table 11.10. Table 11.10 shows there exists considerable correlation between Liu's projected feature vectors. Due to this correlation, when the projected feature vectors are aligned into one feature vector like Equation 11.49, there exists much information redundancy among these features. Accordingly, the effective discriminatory information contained in Liu's projected feature vectors is insufficient despite the ratio of between-class scatter, and within-class scatter

Table 11.8. Comparison of recognition rates (under nearest-neighbor classifier) and CPU time for extraction and classification of the four methods (CPU: PIII 800, Memory: 256M)

Methods	eigenfaces	fisherfaces	O-IMLDA	U-IMLDA
Dimension	37	39	112*7	112*6
Recognition rate	93.5%	88.5%	93.5%	95.5%
Feature extraction time(s)	371.79	378.10	26.52	25.65
Classification time(s)	5.16	5.27	24.30	23.01
Total time (s)	376.95	383.37	50.82	48.66

Table 11.9. Values of generalized Fisher criterion function corresponding to each image projection vectors

$J(X_i)$	X_1	X_2	X_3	X_4	X_5	X_6	X_7	X_8	X_9	X_{10}
O-IMLDA	6.79	6.35	5.89	5.67	5.38	4.89	4.25	3.70	3.58	3.23
U-IMLDA	6.79	5.48	2.18	1.42	1.37	1.08	0.99	0.83	0.68	0.65

is larger after the projection. This is the key reason why Liu's image projection method does not perform as well as expected.

The second experiment is performed on the NUST603 database that contains a set of face images taken at Nanjing University of Science and Technology in 1997. There are 10 different images of 96 subjects. We use the first five images of each subject for training and the other five for testing so that the there are 480 training samples and 480 testing samples in total. The number of selected image projection vectors varies from 2 to 10, and the selected image projection vectors of O-IMLDA and U-IMLDA, respectively, are used for feature extraction. Then, in each image projected feature space (Y-space), a minimum-distance classifier and nearest-neighbor classifier, respectively, are employed. The

Table 11.10. Correlation coefficients between Liu's projected feature vectors

$\rho(Y_i, Y_j)$	Y_1	Y_2	Y_3	Y_4	Y_5	Y_6	Y_7	Y_8	Y_9	Y_{10}
Y_1	1.00	0.98	0.72	0.54	0.43	0.16	0.04	0.46	0.42	0.06
Y_2	0.98	1.00	0.84	0.69	0.59	0.34	0.13	0.32	0.55	0.19
Y_3	0.72	0.84	1.00	0.97	0.93	0.77	0.59	0.14	0.81	0.55
Y_4	0.54	0.69	0.97	1.00	0.99	0.89	0.75	0.35	0.86	0.66
Y_5	0.43	0.59	0.93	0.99	1.00	0.94	0.82	0.46	0.88	0.72
Y_6	0.16	0.34	0.77	0.89	0.94	1.00	0.95	0.69	0.87	0.83
Y_7	0.04	0.13	0.59	0.75	0.82	0.95	1.00	0.85	0.83	0.90
Y_8	0.46	0.32	0.14	0.35	0.46	0.69	0.85	1.00	0.54	0.80
Y_9	0.42	0.55	0.81	0.86	0.88	0.87	0.83	0.54	1.00	0.89
Y_{10}	0.06	0.19	0.55	0.66	0.72	0.83	0.90	0.80	0.89	1.00

Table 11.11. Recognition rates (%) of Liu's O-IMLDA and the proposed U-IMLDA on NUST database

Projection Vector Number		2	3	4	5	6	7	8	9	10
Minimum Distance	O-IMLDA	79.8	81.9	82.1	82.7	83.3	84.4	85.6	86.3	86.3
	U-IMLDA	89.0	90.8	92.1	92.5	92.5	94.2	94.4	94.4	94.4
Nearest Neighbor	O-IMLDA	87.7	88.7	90.0	90.8	91.0	91.9	92.1	92.1	91.5
	U-IMLDA	92.1	95.6	95.4	96.0	96.0	96.5	96.5	96.5	96.0

recognition rates are shown in Table 11.11. U-IMLDA is demonstrated again to be more effective than O-IMLDA in this test.

SUMMARY

In this chapter, two image matrix-based projection-analysis techniques, IMPCA and IMLDA, were developed for image representation. These methods have a series of advantages over conventional PCA and LDA for image feature extraction. First, since IMPCA and IMLDA are both based on the image matrix, they are simpler and more straightforward to use for image feature extraction. Second, IMPCA and IMLDA are superior or comparable to PCA and LDA in terms of recognition accuracy. Third, IMPCA and IMLDA are computationally more efficient than PCA and LDA. They can improve the speed of image feature extraction significantly.

A desirable property of IMPCA-based image representation was revealed; that is, the mean-square error (in the sense of matrix Frobenius norm) between the approximation and the original pattern is minimal when a small number of the principal component vectors are used to represent an image. This property provides a solid theoretical foundation for IMPCA-based image representation and recognition. It should be noted that the minimal mean-square error property of IMPCA depends on the expansion form in Equation 11.9, which is different from that of PCA. That is to say, IMPCA provides an optimal expansion for images in the n-dimensional space, where n is the number of columns of image matrix. In contrast, PCA provides a holistically optimal expansion for images in $(m \times n)$-dimensional image vector space.

The uncorrelated IMLDA was demonstrated as more effective than Liu's orthogonal IMLDA technique. This is because there is considerable correlation between Liu's projected feature vectors. Due to this correlation, when the projected feature vectors are arranged into one feature vector, there exists much information redundancy among these features. Therefore, the effective discriminatory information contained in Liu's projected feature vectors is insufficient despite the ratio of between-class scatter, and within-class scatter is larger after the projection. This is the key reason why Liu's orthogonal IMLDA does not perform as well as the uncorrelated IMLDA. From another viewpoint, we find

that the generalized Fisher criterion in Equation 11.34, like the classical Fisher criterion (Yang, Yang, & Zhang, 2002), is not an absolute criterion for measuring the discriminatory power of projection vectors. In other words, we cannot determine the effectiveness of a set of projection vectors based merely on the corresponding values of the generalized Fisher criterion. The reason is that the correlation between the projected features is also a critical factor deserving serious consideration. So, the generalized Fisher criterion and the statistical correlation should be combined to assess the discriminatory power of a set of projection vectors.

REFERENCES

Belhumeur, P. N., Hespanha, J. P., & Kriegman, D. J. (1997). Eigenfaces vs. fisherfaces: Recognition using class specific linear projection. *IEEE Transaction on Pattern Analysis and Machine Intelligence*, *19*(7), 711-720.

Beveridge, J. R., She, K., Draper, B., & Givens, G.H. (2001). Parametric and nonparametric methods for the statistical evaluation of human ID algorithms. In K. W. Bowyer & P. J. Phillips (Eds.), *Empirical evaluation techniques in computer vision.* IEEE Computer Society Press.

Golub, G. H., & Loan, C. F. (1996). *Matrix computations* (3rd ed.). Baltimore; London: The Johns Hopkins University Press.

Jin, Z., Yang, J., Hu, Z., & Lou, Z. (2001). Face recognition based on the uncorrelated discrimination transformation. *Pattern Recognition, 34*(7), 1405-1416.

Jin, Z., Yang, J., Tang, Z., & Hu, Z. (2001). A theorem on the uncorrelated optimal discrimination vectors. *Pattern Recognition, 34*(10), 2041-2047.

Lancaster, P., & Tismenetsky, M. (1985). *The theory of matrices* (2nd ed.). Orlando, FL: Academic Press.

Liu, K., Cheng, Y. Q., & Yang, J. Y. (1993). Algebraic feature extraction for image recognition based on an optimal discrimination criterion. *Pattern Recognition, 26*(6), 903-911.

Phillips, P. J., Moon, H., Rizvi, S. A., & Rauss, P. A. (2000). The FERET evaluation methodology for face-recognition algorithms. *IEEE Transaction on Pattern Analysis and Machine Intelligence*, *22*(10), 1090-1104.

Schölkopf, B., & Smola, A. (2002). *Learning with kernels*. Cambridge, MA: MIT Press.

Swets, D. L., & Weng, J. (1996). Using discriminant eigenfeatures for image retrieval. *IEEE Transaction on Pattern Analysis and Machine Intelligence*, *18*(8), 831-836.

Tian, Q., Barbero,M, Gu, Z. H., & Lee, S. H. (1986). Image classification by the Foley-Sammon transform. *Optical Engineering*, *25*(7), 834-839.

Turk, M., & Pentland, A. (1991a). Eigenfaces for recognition. *Journal of Cognitive Neuroscience, 3*(1), 71-86.

Turk, M. A., & Pentland, A. P. (1991b). Face recognition using eigenfaces. *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition* (pp. 586-591).

Yambor, W., Draper, B., & Beveridgem, R. (2002). Analyzing PCA-based face recognition algorithms: Ei-genvector selection and distance measures. In H. Christensen & J. Phillips (Eds.), *Empirical evaluation methods in computer vision*. Singapore: World Scientific Press.

Yang, J., & Yang, J. Y. (2002). From image vector to matrix: A straightforward image projection technique – IMPCA vs. PCA. *Pattern Recognition*, *35*(9), 1997-1999.

Yang, J., Yang, J. Y., Frangi, A.F., & Zhang, D. (2003). Uncorrelated projection discriminant analysis and its application to face image feature extraction. *International Journal of Pattern Recognition and Artificial Intelligence*, *17*(8), 1325-1347.

Yang, J., Yang, J. Y., & Zhang, D. (2002). What's wrong with the Fisher criterion? *Pattern Recognition*, *35*(11), 2665-2668.

Yang, J., Zhang, D., Frangi, A. F., & Yang, J. Y. (2004). 2D PCA: A new approach to face representation and recognition. *IEEE Transaction on Pattern Analysis and Machine Intelligence*, *26*(1), 131-137.

Chapter XII

Two-Directional PCA/LDA

ABSTRACT

This chapter introduces a two-directional PCA/LDA approach that is a useful statistical technique applied to biometric authentication. We first describe both bi-directional PCA (BDPCA) and BDPCA plus LDA. Then, some basic models and definitions related to two-directional PCA/LDA approach are given. Next, we discuss two-directional PCA plus LDA. And, finally, the experimental results and chapter summary are given.

INTRODUCTION

BDPCA Method

PCA has been very successful in image recognition. Recent researches on PCA-based methods are mainly concentrated on two issues, feature extraction and classification. In this chapter, we propose BDPCA with assembled matrix distance (AMD) metric to simultaneously deal with these two issues. For feature extraction, we propose a BDPCA approach. BDPCA can be used for image feature extraction by reducing the dimensionality in both column and row directions. For classification, we present an AMD metric to calculate the distance between two feature matrices and then use the nearest-neighbor and nearest feature line classifiers for image recognition. The results of our experiments show that BDPCA with AMD metric is very effective in image recognition.

PCA-based approaches have been very successful in image representation and recognition. In 1987, Sirovich and Kirby used PCA to represent human faces (Sirovich

& Kirby, 1987; Kirby & Sirovich, 1990). Subsequently, Turk and Pentland proposed a PCA-based face recognition method, eigenfaces (Turk & Pentland, 1991). PCA has now been widely investigated and successfully applied to other image recognition tasks (Lu, Zhang, & Wang, 2003; Wu, Zhang, & Wang, 2003; Huber, Ramoser, Mayer, & Penz, 2005).

Despite the great success of PCA, some issues remain that deserve further investigation. First, we have showed in this section that PCA is prone to be over-fitted to the training set because of the high dimensionality and SSS problem. Although no researchers directly pointed out the over-fitting problem, some PCA-based approaches, such as $(PC)^2A$ (Wu & Zhou, 2002; Chen, Zhang, & Zhou, 2004) 2DPCA (Yang & Yang, 2002; Yang, Zhang, Frangi, & Yang, 2004; Chen & Zhu, 2004) and modular PCA (Gottumukkal & Asari, 2004), had been proposed to address this problem. But $(PC)^2A$ just alleviates the over-fitting problem by blurring the original image with an intrinsic low-dimensional image, and both 2DPCA and modular PCA obtain a much higher feature dimensionality than classical PCA (Yang & Yang, 2002). Thus, further work is needed to solve the over-fitting problem and avoid the high-feature dimensionality problem of 2DPCA and modular PCA.

Second, there some work needs to be investigated in the design of classifiers based on the PCA feature. One general classifier is nearest-neighbor (NN) classifier using the Euclidean distance measure. Other distance measures, such as angle-based distance and Mahalanobis distance measures, had been studied to further improve recognition performance (Navarrete & Ruiz-del-Solar, 2001; Moon & Phillips, 1998; Yambor, Draper, & Beveridge, 2002; Perlikbakas, 2004). Recently, nearest feature line (NFL) classifier is introduced to eliminate the performance deterioration of NN caused by the reduction of prototypes (Li & Lu, 1999). Most recently, nearest feature space (NFS) and other variants or extensions of the NFL classifier had been investigated in Chien and Wu (2002), Ryu and Oh (2002), Wang and Zhang (2004) and Zheng, Zhao, and Zou (2004). Yet, even though previous studies of NN have shown that distance measures greatly affect the recognition performance, with reference to the NFL classifier, distance measures have been little investigated. Actually, other distance measures may produce better recognition performance for the NFL classifier.

In this chapter, we tried to simultaneously investigate these two issues. First, we propose a BDPCA method to circumvent the over-fitting problem. Besides, BDPCA can also avoid the high-feature dimensionality problem of 2DPCA and modular PCA. Second, we present an AMD metric to calculate the distance between two feature matrices and apply the proposed distance metric into the implementation of NN and NFL classifiers.

To test the efficiency of BDPCA with AMD metric, experiments were carried out using the ORL face database and PolyU palmprint database. Experimental results show that the proposed method is very effective and competitive compared with other image recognition approaches, and the AMD measure can be used to further improve the performance of the NN and NFL classifiers.

BDPCA Plus LDA Method

Appearance-based methods, especially LDA, have been very successful in facial feature extraction, but the recognition performance of LDA is often degraded by the so-called SSS problem. One popular solution to the SSS problem is PCA+LDA (fisherfaces), but LDA in other low-dimensional subspaces may be more effective. In this section, we

proposed a novel fast-feature extraction technique, BDPCA plus LDA (BDPCA+LDA), which performs LDA in the BDPCA subspace. Three famous face databases, ORL, UMIST and FERET, are employed to evaluate BDPCA+LDA. Experimental results show that BDPCA+LDA needs less computation and memory requirements, and has higher recognition accuracy than PCA+LDA.

Face recognition has been an important issue in computer vision and pattern recognition over the last several decades (Chellappa, Wilson, & Sirohey, 1995; Zhao, Chellappa, Phillips, & Rosenfeld, 2003). While humans recognize faces easily, automated face recognition remains a great challenge in computer-based automated recognition research. One difficulty in face recognition is how to handle the variations in expression, pose and illumination with only limited training samples.

Facial feature extraction methods are of two types: geometric and holistic. Geometric (or structure-based) methods extract local features, such as the locations and local statistics of the eyes, nose, mouth and so forth. Holistic methods extract a holistic representation of the whole face region. Since the correct feature detection and good measure techniques are required for geometric or structure based approaches, this section considers only holistic methods.

PCA (see Chapter II), independent component analysis (ICA) and LDA (see Chapter III) are three main holistic approaches for facial feature extraction. PCA is one of the most important facial feature extraction approaches. In 1987, Sirovich and Kirby first used PCA to represent facial images (Hyvarinen, 2001; Bartlett, Movellan, & Sejnowski, 2002). Subsequently, Turk, and Pentland (1991) applied PCA to face recognition and presented the well-known eigenfaces method. Since then, PCA has been widely studied and has become one of most successful facial feature extraction approaches. Recently, other PCA-based approaches, such as 2DPCA, have been proposed for facial feature extraction (Liu & Wechsler, 2003; Yuen & Lai, 2002; Fukunaga, 1990; Deerwester, Dumais, Furnas, Landauer, & Harshman, 1990).

As an extension and generalization of PCA, ICA (Hyvarinen, 2001) has been used by Bartlett to extract features for face recognition (Bartlett, Movellan, & Sejnowski, 2002). Later, Draper found that distance measure has an important effect on the recognition accuracy of ICA (Draper, Baek, Bartlett, & Beveridge, 2003). Other investigations on ICA-based facial feature extraction technique can be seen in Baeka and Kimb (2004) and Chen, Liao, Ko, Lin, and Yu (2000).

It is usually believed that LDA outperforms PCA in classification because PCA emphasizes only the optimal low-dimensional representation and has no direct relation to classification performance (Fukunaga, 1990). LDA finds the set of optimal projection vectors that map the original data into a low-dimensional feature space, with the restriction that the ratio of the between-class scatter $\mathbf{S}_b$ to the within-class scatter $\mathbf{S}_w$ is maximized. When applied to face recognition, LDA seriously suffers from the SSS problem caused by the limited number of high-dimensional training samples (Chen, Liao, Ko, Lin, & Yu, 2000). To date, many approaches have been proposed to handle this problem.

One of the most successful approaches recently developed for solving the SSS problem is subspace LDA. Subspace LDA first uses a dimensionality reduction technique to map the original data to a low-dimensional subspace, and then LDA is performed in the subspace. So far, researchers have applied PCA, latent semantic indexing (LSI)

Figure 12.1. Ten images of one individual in the ORL face database

(Deerwester, Dumais, Furnas, Landauer, & Harshman, 1990) and partial least squares (PLS) (Garthwaite, 1994) as pre-processors for dimensionality reduction (Jing, Tang, & Zhang, 2005; Etemad & Chellappa, 1996; Ray & Driver, 1970; Habibi & Wintz, 1971). Among all the subspace methods, over the past few years, the PCA plus LDA (PCA+LDA) approach has received significant attention. In Belhumeur, Hespanha and Kriegman's famous fisherfaces, PCA is first applied to eliminate the singularness of $\mathbf{S}_w$, and then LDA can be performed in the PCA subspace (Belhumeur, Hespanha, & Kriegman, 1997). However, the discarded null space of $\mathbf{S}_w$ may also contain some important discriminant information, causing the performance deterioration of fisherfaces. To solve this problem, a class of direct LDA (DLDA) method is proposed (Yu & Yang, 2001), and Yang proposed a complete PCA+LDA method ,which simultaneously considered the discriminant information both outside and within the within-class scatter matrix (Yang & Yang, 2003).

In this chapter, we propose a fast subspace LDA technique, BDPCA+LDA. BDPCA is a natural result of classical PCA and assumes that the transform kernel of PCA is separable (Phillips, 2001; Liu, Wang, Li, & Tan, 2004). The separation of the PCA kernel at least has three main advantages: less training time, less feature extraction time and a lower memory requirement. BDPCA is also a generalization of Yang's 2DPCA (Yang & Yang, 2002; Yang, Zhang, Frangi, & Yang, 2004). To further evaluate the efficiency of BDPCA+LDA, experiments were carried out using three popular face databases: ORL, UMIST and FERET. Experimental results show that BDPCA+LDA is superior to the PCA+LDA framework in recognition accuracy.

BASIC MODELS AND DEFINITIONS

Classical PCA's Over-Fitting Problem

When applied to image recognition, classical PCA is apt to be over-fitted to the training set due to the SSS problem. As a statistical method, classical PCA's statistical meaning is problematic when the number of samples is small and the sample's dimensionality is high. To validate this perspective, we carried out a series of experiments using the ORL face database (Karhunan & Joutsensalo, 1995).

The ORL database contains 400 facial images with 10 images per individual. The 10 images of one person are shown in Figure 12.1. The images vary in sampling time, light

conditions, facial expressions, facial details (glasses/no glasses), scale and tilt. All the images are taken against a dark homogeneous background, with the person in an upright frontal position, with tolerance for some tilting and rotation of up to about 20°. The size of these gray images is 112×92 (Olivetti, n.d.).

Here we use the normalized mean-square error (MSE) to evaluate the over-fitting problem of classical PCA. One statistical characteristic of PCA is that the MSE between random vector x and its subspace projection is minimal (Karhunan & Joutsensalo, 1995). Thus, the difference of MSE on the training set and the testing set can be used to investigate the over-fitting problem. If PCA is over-fitted to the training set, the MSE on the training set would be much lower than that on the testing set.

Given the first L principal components, we can obtain the projection matrix $WL = [\Psi_1, \Psi_2, \ldots, \Psi_L]$. Then a vector x can be transformed into PCA subspace by:

$$y = W_L^T (x - \bar{x}) \tag{12.1}$$

and the reconstructed vector $\tilde{x}$ can be represented as:

$$\tilde{x} = \bar{x} + W_L y = \bar{x} + W_L W_L^T (x - \bar{x}) \tag{12.2}$$

The normalized MSE on the training set MSE_L^{train} is defined as:

$$MSE_L^{train} = \sum_{i=1}^{N_1} \left\| x_i^{train} - \tilde{x}_i^{train} \right\|^2 \Big/ \sum_{i=1}^{N_1} \left\| x_i^{train} - \overline{x^{train}} \right\|^2 \tag{12.3}$$

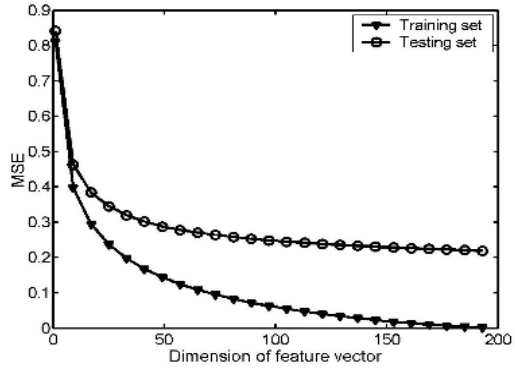
where N_1 is the number of training samples, x_i^{train} is the ith training sample, $\tilde{x}_i^{train}$ is the reconstructed vector of x_i^{train} and $\overline{x^{train}}$ is the mean vector of all training samples. Similarly, we can calculate the normalized MSE on the testing set as MSE_L^{test}:

$$MSE_L^{test} = \sum_{i=1}^{N_2} \left\| x_i^{test} - \tilde{x}_i^{test} \right\|^2 \Big/ \sum_{i=1}^{N_2} \left\| x_i^{test} - \overline{x^{test}} \right\|^2 \tag{12.4}$$

where N_2 is the number of testing samples, x_i^{test} is the ith testing sample, $\tilde{x}_i^{test}$ is the reconstructed vector of x_i^{test} and $\overline{x^{test}}$ is the mean vector of all testing samples.

We select the first five images per individual for training to obtain a training set of 200 samples and a testing set of 200 samples with no overlap between the two sets. Then we calculate the normalized MSE on training set and the testing set for given W_L, as shown in Figure 12.2. It can be observed that when the number of principal components is small,

Figure 12.2. The PCA's normalized MSE on the training set and the testing set as the function of feature dimension



the difference of MSE_L^{train} and MSE_L^{test} is about 2.5%. The normalized MSE on the testing set MSE_L^{test} would be much lower than MSE_L^{train} with the increase of the number of principal components L. The difference is up to 12.5% when the number of principal components is 40. The great difference between the normalized MSE_L^{train} and MSE_L^{test} indicates that classical PCA is inclined to be over-fitted to the training set.

Previous Work in Solving PCA's Over-Fitting Problem

Although no researchers directly pointed out classical PCA's over-fitting problem as yet, some PCA-based methods been proposed to alleviate the over-fitting problem. On improvement methodology, three representative approaches are: (1) $(PC)^2A$ (Wu & Zhou, 2002; Chen, Zhang, & Zhou, 2004); (2) IMPCA or 2DPCA (Yang & Yang, 2002; Yang, Zhang, Frangi, & Yang, 2004; Chen & Zhu, 2004); and (3) Modular PCA (Gottumukkal & Asari, 2004; Toyhar & Acan, 2004; Chen, Liu, & Zhou, 2004).

$(PC)^2A$

$(PC)^2A$ adopted an image pre-processing plus PCA mechanism (Wu & Zhou, 2002). Given an $m \times n$ image $I(x,y)$, its vertical and horizontal integral projections are defined as:

$$V_p(x) = \frac{1}{n}\sum_{y=1}^{n} I(x,y) \tag{12.5}$$

$$H_p(y) = \frac{1}{m}\sum_{x=1}^{m} I(x,y) \tag{12.6}$$

Then we define the projection map $M_p(x,y)$:

$$M_p(x,y) = V_p(x)H_p(y)/\bar{I} \tag{12.7}$$

where $\bar{I}$ is the average intensity of the image. Then we can obtain $I_\alpha(x,y)$, the projection-combined version of $I(x,y)$ with combination parameter α:

$$I_\alpha(x,y) = (1 - \alpha)\, I(x,y) + \alpha M_p(x,y) \tag{12.8}$$

Finally, classical PCA is performed on the projection-combined version of $I(x,y)$.

Since the projection map $M_p(x,y)$ is generated by the vertical and horizontal integral projection $V_p(x)$ and $H_p(x)$, the intrinsic dimensionality of $M_p(x,y)$ should be less than $(m+n-1)$ (Wu & Zhou, 2002). Thus the projection-combined version of $I(x,y)$ is a blurred version of the original image by the low-dimensional $M_p(x, y)$. The employment of $I_\alpha(x,y)$ in PCA can relax the high dimensionality problem. We considered that this is the intrinsic reason of $(PC)^2A$'s better recognition performance.

2DPCA

Yang's 2DPCA actually is a row PCA which regards an $m \times n$ image matrix as an m-set of $1 \times n$ row vectors. Given training set $\{X_1, X_2, \ldots, X_N\}$, N is the number of the samples. The image total scatter matrix G_t in (Yang & Yang, 2002; Yang, Zhang, Frangi, & Yang, 2004) is defined as:

$$G_t = \frac{1}{N}\sum_{i=1}^{N}(X_i - \overline{X})^T (X_i - \overline{X}) \tag{12.9}$$

where X_i denotes the ith training image and $\overline{X}$ denotes the mean image of all training images. By representing X_i as an m-set of $1 \times n$ row vectors:

$$X_i = \begin{bmatrix} x_i^1 \\ x_i^2 \\ \vdots \\ x_i^m \end{bmatrix} \tag{12.10}$$

the image total scatter matrix G_t can be rewritten as follows:

$$G_t = \frac{1}{N}\sum_{i=1}^{N}\sum_{j=1}^{m}(x_i^j - \overline{x^j})^T (x_i^j - \overline{x^j}) \tag{12.11}$$

where x_i^j denotes the jth row of X_i, and $\overline{x_i^j}$ denotes the jth row of mean matrix $\overline{X}$. We can define the row total scatter matrix S_t^{row}, the scatter matrix of all the row vectors in the training set:

$$S_t^{row} = \frac{1}{Nm}\sum_{i=1}^{N}\sum_{j=1}^{m}(x_i^j - \bar{x})^T (x_i^j - \bar{x}) \tag{12.12}$$

where $\bar{x}$ is the mean vector of all row vectors in $\overline{X}$. Comparing Equation 12.11 with Equation 12.12, the computing of G_t and S_t^{row} is almost the same, except the substitution of $\bar{x}$ for $\overline{x_i^j}$ and an addition of constant 1/*m*. The constant 1/m has no effect on the calculation of row eigenvectors, and makes the eigenvalues of S_t^{row} more meaningful. So we argue that 2DPCA actually is a variant of row PCA.

2DPCA regards an image as m $1 \times n$ row vectors and performs PCA on all row vectors in the training set. So, the actual vector dimensionality for 2DPCA is n and the actual sample size is *mN*. Thus, the high-dimensionality and SSS problems are solved. However, 2DPCA suffers from the disadvantage that its feature dimensionality is still high (one typical dimension in Yang and Yang (2002) is 8´112). Yang proposed a 2DPCA + PCA method to solve this problem. But when a 2DPCA + PCA strategy is adopted, we must face the high-dimensionality and SSS problems once again.

Modular PCA

In modular PCA, an image is divided into n1 smaller sub-images and PCA is performed on all these sub-images (Gottumukkal & Asari, 2004). Given an *m´n* image I, these sub-images can be represented mathematically as:

$$I_{ij}(k,l) = I\left(\frac{m}{\sqrt{n_1}}(i-1)+k, \frac{n}{\sqrt{n_1}}(j-1)+l\right) \tag{12.13}$$

where I_{ij} denotes the vertical *i*th and horizontal *j*th sub-image, *i, j* varies from 1 to $\sqrt{n_1}$, *k* varies from 1 to $\frac{m}{\sqrt{n_1}}$ and l varies from 1 to $\frac{n}{\sqrt{n_1}}$. Then all sub-images are applied in the modular PCA approach.

Since modular PCA divides an image into some sub-images, the actual vector dimensionality in modular PCA will be much lower than that in classical PCA. Moreover, the actual samples used in modular PCA are more than that used in classical PCA. Thus, modular PCA can be utilized to solve the over-fitting problem caused by the high dimensionality and SSS.

In our opinion, modular PCA still has some problems. One is how to determine the number of sub-images. For example, Toygar and Acan had proposed another method that divides the facial image into five horizontal sub-images (Toygar & Acan, 2004) and Chen, Liu, and Zhou proposed 5×5 and 5×3 partitions of the original images (Chen, Liu, & Zhou, 2004). Another is that the feature dimensionality will increase with the increasing of the number of sub-images.

As stated previously, $(PC)^2A$ can be used to alleviate the over-fitting problem by blurring the original image with a intrinsic low-dimensional image. 2DPCA and modular PCA can solve the over-fitting problems by both reducing the dimensionality and increasing the training samples. Actually, 2DPCA can be regarded as a special implementation of modular PCA, where each row of the original image is regarded as a sub-image. However, the over-fitting problem is just relaxed when $(PC)^2A$ is adopted, while the feature dimensionality will increase when using the 2DPCA or modular PCA approaches.

BDPCA with Assembled Matrix Distance Metric

Bi-Directional PCA

When classical PCA is used for feature extraction, the original image X should be mapped into its high-dimensional vector representation x; then the feature vector y is computed by:

$$y = W_{pca}^T x \tag{12.14}$$

where W_{pca} is PCA projection. Unlike classical PCA, BDPCA directly extracts a feature matrix Y from the original image matrix X by:

$$Y = W_{col}^T X W_{row} \tag{12.15}$$

where W_{col} is the column projection matrix and W_{row} is the row projection matrix. With W_{col} and W_{row}, BDPCA can be used for image feature extraction by reducing the dimensionality in both column and row directions.

Next, we present our method to calculate W_{col} and W_{row}. Given a training set $\{X_1, X_2, \ldots, X_d\}$, N is the number of the samples, and the size of each image is $m \times n$. By representing the ith image matrix X_i as an m-set of $1 \times n$ row vectors:

$$X_i = \begin{bmatrix} x_i^1 \\ x_i^2 \\ \vdots \\ x_i^m \end{bmatrix} \tag{12.16}$$

the row total scatter matrix S_t^{row} can be obtained by:

$$S_t^{row} = \frac{1}{Nm} \sum_{i=1}^{N} (X_i - \bar{X})^T (X_i - \bar{X}) \tag{12.17}$$

where x_i^j denotes the jth row of X_i, and $\overline{x_i^j}$ denotes the jth row of mean matrix $\overline{X}$. We choose the row eigenvectors corresponding to the first k_{row} largest eigenvalues of S_t^{row} to construct the row projection matrix W_{row}. By treating an image matrix X_i as an n-set of $m \times 1$ column vectors:

$$X_i = [x_i^1 \quad x_i^1 \quad \ldots \quad x_i^n] \tag{12.18}$$

we obtain the column total scatter matrix S_t^{col}:

$$S_t^{col} = \frac{1}{Nn} \sum_{i=1}^{N} (X_i - \overline{X})(X_i - \overline{X})^T \tag{12.19}$$

Then we choose the column eigenvectors corresponding to the first k_{col} largest eigenvalues of S_t^{col} to construct the column projection matrix $W_{col.}$. Finally, we use:

$$Y = W_{col}{}^T X W_{row} \tag{12.20}$$

to extract feature matrix Y from image X.

Actually, BDPCA is a generalization of Yang's 2DPCA, and 2DPCA can be regarded as a special BDPCA with $W_{col} = I_m$ where I_m denotes an m-by-m identity matrix (Yang, Zhang, Frangi, & Yang, 2004).

Assembled Matrix Distance Metric

We propose an AMD metric to calculate the distance between two feature matrices. Unlike a PCA-based approach, which produces a feature vector, BDPCA produces a feature matrix. So we present an assembled matrix distance metric to measure the distance between feature matrices.

First, we briefly reviewed some other matrix measures. Give two feature matrices $A = (a_{ij})_{k_{col} \times k_{row}}$ and $B = (b_{ij})_{k_{col} \times k_{row}}$, the Frobenius distance is defined as:

$$d_F(A,B) = \left(\sum_{i=1}^{k_{col}} \sum_{j=1}^{k_{row}} (a_{ij} - b_{ij})^2 \right)^{1/2} \tag{12.21}$$

Yang proposed another matrix distance (Yang distance) in Yang, Zhang, Frangi, and Yang (2004):

$$d_Y(A,B) = \sum_{j=1}^{k_{row}} \left(\sum_{i=1}^{k_{col}} (a_{ij} - b_{ij})^2 \right)^{1/2} \tag{12.22}$$

Here we define the assembled matrix distance $d_{AMD}(A,B)$ as follows:

$$d_{AMD}(A,B) = \left(\sum_{j=1}^{k_{row}} \left(\sum_{i=1}^{k_{col}} (a_{ij} - b_{ij})^{p_1} \right)^{p_2/p_1} \right)^{1/p_2}, (p_1, p_2 > 0) \tag{12.23}$$

Definition 12.1 (Karhunan & Joutsensalo, 1995)**.** A *vector norm* on $\Re^n$ is a function $f: \Re^n \to \Re$ with the following properties:

$$f(x) \geq 0, \qquad x \in \Re^n (f(x) = 0 \Leftrightarrow x = 0) \tag{12.24}$$

$$f(x+y) \leq f(x) + f(y), \qquad x, y \in \Re^n \tag{12.25}$$

$$f(\alpha x) \le |\alpha| f(x), \qquad \alpha \in \Re, x \in \Re^n \tag{12.26}$$

Definition 12.2 (Karhunan & Joutsensalo, 1995)**.** A *matrix norm* on $\Re^{k_{col} \times k_{row}}$ is a function $f : \Re^{k_{col} \times k_{row}} \to \Re$ with the following properties:

$$f(A) \ge 0, \qquad A \in \Re^{k_{col} \times k_{row}} \; (f(A) = 0 \Leftrightarrow A = 0) \tag{12.27}$$

$$f(A + B) \le f(A) + f(B), \qquad A, B \in \Re^{k_{col} \times k_{row}} \tag{12.28}$$

$$f(\alpha A) \le |\alpha| f(A), \qquad \alpha \in \Re, A \in \Re^{k_{col} \times k_{row}} \tag{12.29}$$

Definition 12.3 (Karhunan & Joutsensalo, 1995)**.** The Frobenius norm of a matrix $A = [a_{ij}]_{k_{col} \times k_{row}}$ is defined by:

$$\|A\|_F = \sqrt{\sum_{i=1}^{k_{col}} \sum_{j=1}^{k_{row}} a_{ij}^2}.$$

From **Definition 12.3**, it is simple to see that the Frobenius distance is a metric derived from the Frobenius matrix norm. Actually, both the Frobenius and the Yang distance measures are matrix metrics, and we will prove this in the next.

Theorem 12.1 (Karhunan & Joutsensalo, 1995). Function $\|x\|_p = (\sum_i |x_i|^p)^{1/p}$ *is a vector norm.*

Theorem 12.2. Function $\|A\|_{AMD} = \left(\sum_{j=1}^{k_{row}} \left(\sum_{i=1}^{k_{col}} (|a_{ij}|)^{p_1} \right)^{p_2/p_1} \right)^{1/p_2}$ *is a matrix norm.*

Proof: It can be easily shown that:

$$\|A\|_{AMD} \ge 0$$

$$\|A\|_{AMD} = 0 \Leftrightarrow A = 0$$

$$\|\alpha A\|_{AMD} = |\alpha| \|A\|_{AMD}$$

Now, we prove $\|A+B\|_{AMD} \le \|A\|_{AMD} + \|B\|_{AMD}$. From Theorem 12.1:

$$\|A+B\|_{AMD} = \left(\sum_{j=1}^{k_{row}} \left(\sum_{i=1}^{k_{col}} (a_{ij} + b_{ij})^{p_1} \right)^{p_2/p_1} \right)^{1/p_2}$$

$$\le (\sum_{j=1}^{k_{row}} (\|a^{(j)}\|_{p_1} + \|b^{(j)}\|_{p_1})^{p_2})^{1/p_2}$$

From Theorem 12.1, the function $g(a) = \sum_{j=1}^{k_{row}} (\|a^{(j)}\|_2)^{p_2})^{1/p_2}$, $(a = [\|a^{(1)}\|_{p_1}, \ldots, \|a^{(k_{row})}\|_{p_1}]^T)$ is a vector norm. Let $b = [\|b^{(1)}\|_{p_1}, \ldots, \|b^{(k_{row})}\|_{p_1}]^T$:

$$(\sum_{j=1}^{k_{row}} (\|a^{(j)}\|_{p_1} + \|b^{(j)}\|_{p_1})^{p_2})^{1/p_2} = g(a+b)$$

$$\le g(a) + g(b)$$

$$= (\sum_{j=1}^{k_{row}} (\|a^{(j)}\|_{p_1})^{p_2})^{1/p_2} + (\sum_{j=1}^{k_{row}} (\|b^{(j)}\|_{p_1})^{p_2})^{1/p_2}$$

$$= \left(\sum_{j=1}^{k_{row}} \left(\sum_{i=1}^{k_{col}} a_{ij}{}^{p_1} \right)^{p_2/p_1} \right)^{1/p_2} + \left(\sum_{j=1}^{k_{row}} \left(\sum_{i=1}^{k_{col}} b_{ij}{}^{p_1} \right)^{p_2/p_1} \right)^{1/p_2}$$

$$= \|A\|_{AMD} + \|B\|_{AMD}$$

So $\|A+B\|_{AMD} \le \|A\|_{AMD} + \|B\|_{AMD}$, and $\|A\|_{AMD}$ is a matrix norm.

Definition 12.4 (Karhunan & Joutsensalo, 1995)**.** A *metric* in $\Re^{k_{col} \times k_{row}}$ is a function $f : \Re^{k_{col} \times k_{row}} \times \Re^{k_{col} \times k_{row}} \to \Re$ with the following properties:

$$f(A,B) \ge 0, \qquad A, B \in \Re^{k_{col} \times k_{row}} \tag{12.30}$$

$$f(A,B) = 0 \Leftrightarrow A = B \tag{12.31}$$

$$f(A,B) = f(B,A) \tag{12.32}$$

$$f(A,B) \le f(A,C) + f(C,B) \tag{12.33}$$

Theorem 3. $d_{AMD}(A, B)$ *is a distance metric.*

Proof: Function $\|A\|_{AMD}$ is a matrix norm, and it is easy to see that $d_{AMD}(A,B) = \|A-B\|_{AMD}$ is a distance measure derived from the matrix norm $\|A\|_{AMD}$. So the function $d_{AMD}(A, B)$ is a distance metric.

Corollary 12.1. The Frobenius distance measure $d_Y(A,B) = \left(\sum_{j=1}^{k_{row}} \sum_{i=1}^{k_{col}} (a_{ij} - b_{ij})^2 \right)^{1/2}$ is a special case of AMD metric with $p_1 = p_2 = 2$.

Corollary 12.2. The Yang's distance measure $d_Y(A,B) = \sum_{j=1}^{k_{row}} \left(\sum_{i=1}^{k_{col}} (a_{ij} - b_{ij})^2 \right)^{1/2}$ is a special case of AMD metric with $p_1 = 2$ and $p_2 = 1$.

Classifiers

We use two classifiers, the nearest neighbor (NN) and the nearest feature line (NFL), for image recognition. In NN classifier, the feature matrix is classified as belonging to the class with the nearest template. Given all the templates $\{M_{cl}, 1 \le c \le C, \ 1 \le l \le n_c\}$ and the query feature Y, the NN rule can be expressed as:

$$d(Y, M_{\hat{c}\hat{l}}) = \min_{\{1 \le c \le C, 1 \le l \le n_c\}} d(Y, M_{cl}) \Rightarrow Y \in w_{\hat{c}} \tag{12.34}$$

where C is the number of classes, n_c is the number of templates in class w_c and $d(Y, M_{cl})$ denotes the distance between Y and M_{cl}.

NFL is an extension of the NN classifier (Li & Lu, 1999; Chaudhuri, Murthy, & Chaudhuri, 1992). At least two templates are needed for each class in NFL classifier. The NFL classifier can extend the representative capacity of templates by using linear interpolation and extrapolation. Given two templates M_{cl} and M_{ck}, the distance between the feature point Y and the feature line $\overline{M_{cl}M_{ck}}$ is defined as:

$$d(Y, \overline{M_{cl}M_{ck}}) = d(Y, Y_p) \tag{12.35}$$

where $Y_p = M_{cl} + \mu (M_{ck} - M_{cl})$ and $\mu = (Y - M_{cl}) \cdot (M_{ck} - M_{cl}) / (M_{ck} - M_{cl}) \cdot (M_{ck} - M_{cl})$. Then, NFL determines the class $w_{\hat{c}}$ of the query feature Y according to:

$$d(Y, \overline{M_{\hat{c}\hat{l}}M_{\hat{c}\hat{k}}}) = \min_{\{1 \le c \le C, 1 \le l < k \le n_c\}} d(Y, \overline{M_{cl}M_{ck}}) \Rightarrow Y \in w_{\hat{c}}$$

Overview of PCA Techniques for 2D Image Transform

In this section, we introduce some basic conceptions on 2D image transform. Then we discuss two kinds of PCA for 2D transform, holistic PCA and 2D-KLT. Finally, we propose a face-specific 2D-KLT, BDPCA.

General Idea of 2D Image Transform

Two dimensional image transform has two major applications in image processing: image feature extraction and image dimensionality reduction. In Pratt (2001), *2D transform* is defined as follows:

Definition 12.5. The *2D transform* of the $m \times n$ image matrix $\mathbf{X}(j, k)$ results in a transformed image matrix $\mathbf{X}'(u, v)$ as defined by:

$$\mathbf{X}'(u,v) = \sum_{j=1}^{m}\sum_{k=1}^{n}\mathbf{X}(j,k)\mathbf{A}(j,k;u,v) \tag{12.36}$$

where $\mathbf{A}(i, j ; u, v)$ denotes the transform kernel. The inverse transform is defined as:

$$\tilde{\mathbf{X}}(i,j) = \sum_{u=1}^{m}\sum_{v=1}^{n}\mathbf{X}'(u,v)\mathbf{B}(i,j;u,v) \tag{12.37}$$

where $\mathbf{B}(i, j ; u, v)$ denotes the inverse transform kernel.

Definition 12.6. The transform is *unitary* if its transform kernels satisfy the following orthonormality constraints:

$$\sum_{u}\sum_{v}\mathbf{A}(j_1,k_1;u,v)\mathbf{A}^*(j_2,k_2;u,v) = \boldsymbol{\delta}(j_1-j_2,k_1-k_2) \tag{12.38}$$

$$\sum_{u}\sum_{v}\mathbf{B}(j_1,k_1;u,v)\mathbf{B}^*(j_2,k_2;u,v) = \boldsymbol{\delta}(j_1-k_2,k_1-k_2) \tag{12.39}$$

$$\sum_{j}\sum_{k}\mathbf{A}(j,k;u_1,v_1)\mathbf{A}^*(j,k;u_2,v_2) = \boldsymbol{\delta}(u_1-u_2,v_1-v_2) \tag{12.40}$$

$$\sum_{j}\sum_{k}\mathbf{B}(j,k;u_1,v_1)\mathbf{B}^*(j,k;u_2,v_2) = \boldsymbol{\delta}(u_1-u_2,v_1-v_2) \tag{12.41}$$

where $\mathbf{A}^*$ is the conjugation of $\mathbf{A}$.

Definition 12.7. The transform is *separable* if its kernels can be rewritten as:

$$\mathbf{A}(j, k ; u, v) = \mathbf{A}_{col}(j, u)\, \mathbf{A}_{row}(k,v) \tag{12.42}$$

$$\mathbf{B}(j, k ; u, v) = \mathbf{B}_{col}(j, u)\, \mathbf{B}_{row}(k,v) \tag{12.43}$$

The introduction of the terms "*unitary*" and "*separable*" is very important in 2D transform. For separable transform, the transformed matrix **X'** of the original image matrix **X** can be obtained by:

$$\mathbf{X}' = \mathbf{A}_{col}\mathbf{X}\mathbf{A}_{row}^{T} \tag{12.44}$$

and the inverse transformation can be given by:

$$\tilde{\mathbf{X}} = \mathbf{B}_{col}\mathbf{X}'\mathbf{B}_{row}^{T} \tag{12.45}$$

where $\mathbf{A}_{col}$ and $\mathbf{A}_{row}$ are column and row transform kernels, and $\mathbf{B}_{col}$ and $\mathbf{B}_{row}$ are column and row inverse transform kernels. If the separable transform is unitary, it is easy to obtain the inverse transform kernels by:

$$\mathbf{B}_{col} = \mathbf{A}_{col}^{\mathrm{T}} \text{ and } \mathbf{B}_{row} = \mathbf{A}_{row}^{\mathrm{T}} \tag{12.46}$$

Recently, some separable transform techniques, such as the Fourier transform and the wavelet transform (Mallat, 2002), have been applied to face recognition as pre-processors to transform the original image to its low-dimensional subspace (Lu, Plataniotis, & Venetsanopoulos, 2003; Zhang, Li, & Wang, 2004; Zhao, Chellappa, & Phillips, 1999).

Holistic PCA: Inseparable Image Model Based Technique

When handling an inseparable 2D transform, it is better to map the image matrix $\mathbf{X}(j, k)$ into its vector representation x in advance. Then the 2D transform of Equation 12.36 can be rewritten as:

$$x' = \mathbf{A}x \tag{12.47}$$

Holistic PCA transform, also known as K-L transform, is an important inseparable 2D image transform technique. In Holistic PCA, an image matrix **X** must be transformed into 1D vector x in advance. Then, given a set of N training images $\{x_1, x_2, \ldots, x_N\}$, the total covariance matrix $\mathbf{S}_t$ of PCA is defined by:

$$\mathbf{S}_t = \frac{1}{N}\sum_{i=1}^{N}(x_i - \bar{x})(x_i - \bar{x})^T \tag{12.48}$$

where $\bar{x}$ denotes the mean vector of all training images.

We then choose eigenvectors $\{v_1, v_2, \ldots, v_{d_{\mathrm{PCA}}}\}$ corresponding to the first d_{PCA} largest eigenvalues of $\mathbf{S}_t$ as projection axes. Generally, these eigenvectors can be calculated directly. However, for a problem like face recognition, it is difficult to solve the $\mathbf{S}_t$–matrix directly, because its dimensionality is always very high, requiring too many computations and too much memory.

Fortunately, the high-dimensionality problem can be addressed using the SVD technique. Let $\mathbf{Q} = [x_1 - \bar{x}, x_2 - \bar{x}, \ldots, x_N - \bar{x}]$; then Equation 12.48 can be rewritten as

$\mathbf{S}_t = \frac{1}{N}\mathbf{Q}\mathbf{Q}^T$. Next, we form the matrix $\mathbf{R} = \mathbf{Q}^T\mathbf{Q}$, which is an $N \times N$ semi-positive definite matrix. In a problem like face recognition, the number of training samples is much smaller than the dimensions of the image vector. Thus, the size of $\mathbf{R}$ is smaller than that of $\mathbf{S}_t$, and it is easier to obtain the eigenvectors of $\mathbf{R}$ than those of $\mathbf{S}_t$. S,o we first calculate the eigenvectors $\{\boldsymbol{\varphi}_1, \boldsymbol{\varphi}_2, \ldots, \boldsymbol{\varphi}_{d_{\text{PCA}}}\}$ of $\mathbf{R}$ corresponding to the first d_{PCA} largest eigenvalues. Then the ith eigenvector of $\mathbf{S}_t$, $\boldsymbol{\nu}_i$, can be obtained by:

$$\boldsymbol{\nu}_i = \frac{1}{\sqrt{\boldsymbol{\lambda}_i}}\mathbf{Q}\boldsymbol{\varphi}_i, \quad i = 1, \ldots, d_{\text{PCA}} \tag{12.49}$$

After the projection of sample x onto the eigenvector v_i:

$$y_i = v_i^T(x - \bar{x}), \quad i = 1, \cdots, d_{\text{PCA}} \tag{12.50}$$

we can form the PCA-transformed feature vector y = $[y_1, y_2, \ldots, y_{d_{\text{PCA}}}]^T$ of sample x. Correspondingly, the reconstructed image vector $\tilde{x}$ of the image vector x can be obtained by:

$$\tilde{x} = \bar{x} + \sum_{i=1}^{d_{\text{PCA}}} y_i w_i = \sum_{i=1}^{d_{\text{PCA}}} w_i^T(x - \bar{x})w_i \tag{12.51}$$

2D-KLT: Separable Image Model Based Technique

2D-KLT is a separable PCA technique. If the PCA kernel is separable, we can rewrite the 2D transform of Equation 12.36) as:

$$\mathbf{X}' = \mathbf{A}_{col}^T\mathbf{X}\mathbf{A}_{row} \tag{12.52}$$

where $\mathbf{A}_{row}$ and $\mathbf{A}_{col}$ are the row and column kernels that satisfy:

$$\mathbf{S}_t^{col}\mathbf{A}_{col} = \boldsymbol{\lambda}_{col}\mathbf{A}_{col} \tag{12.53}$$

$$\mathbf{S}_t^{row}\mathbf{A}_{row} = \boldsymbol{\lambda}_{row}\mathbf{A}_{row} \tag{12.54}$$

where $\mathbf{S}_t^{col}$ and $\mathbf{S}_t^{row}$ are the column and row total covariance matrices, $\boldsymbol{\lambda}_{col}$ and $\boldsymbol{\lambda}_{row}$ are two diagonal matrices.

Since one of the main applications of 2D-KLT is image compression (Yang, Yang, & Frangi, 2003), it is expected we would obtain an explicit universal form of the PCA kernel rather than an image or content-dependent kernel. For a 1D Markov process with correlation factor r, Ray and Driver (1970) gave the column eigenvalues and eigenvectors as (1970):

$$\lambda_k = \frac{1-r^2}{1-2r\cos\omega_k + r^2} \tag{12.55}$$

$$\psi_k(i) = \left(\frac{2}{m+\lambda_k}\right)^{1/2} \sin\left[\omega_k\left(i - \frac{(m+1)\pi}{2} + \frac{k\pi}{2}\right)\right], \quad i = 1, \ldots, m \tag{12.56}$$

where ω_k are the real positive roots of the transcendental equation for m = even:

$$\tan(m\omega) = \frac{(1-r^2)\sin\omega}{\cos\omega - 2r + r^2\cos\omega} \tag{12.57}$$

Next, we compare the computation complexity of holistic PCA and 2D-KLT. It is reasonable to use the number of multiplications as a measure of computation complexity involved in PCA and 2D-KLT transform. The 2D-KLT transform requires m^2n+n^2m multiplications, while holistic PCA transform requires $(mn)^2$ multiplications.

Like the Fourier and wavelet transform, 2D-KLT is a content-independent 2D image transform. When applied to face recognition, it is reasonable to expect that face-image-specific transform kernels can obtain better results.

BDPCA: A Face-Image-Specific 2D-KLT Technique

In this section, we propose a face-image-specific transform, BDPCA. Given a training set $\{\mathbf{X}_1, \ldots, \mathbf{X}_N\}$, N is the number of the training images, and the size of each image matrix is $m \times n$. By representing the ith image matrix $\mathbf{X}_i$ as an m-set of $1 \times n$ row vectors:

$$\mathbf{X}_i = \begin{bmatrix} x_i^1 \\ x_i^2 \\ \vdots \\ x_i^m \end{bmatrix} \tag{12.58}$$

we adopt Yang's approach (Liu & Wechsler, 2003; Yuen & Lai, 2002) to calculate the row total scatter matrix S_t^{row}:

$$\mathbf{S}_t^{row} = \frac{1}{Nm}\sum_{i=1}^{N}\sum_{j=1}^{m}(x_i^j - \overline{x^j})^T(x_i^j - \overline{x^j}) = \frac{1}{Nm}\sum_{i=1}^{N}(\mathbf{X}_i - \bar{\mathbf{X}})^T(\mathbf{X}_i - \bar{\mathbf{X}}) \tag{12.59}$$

where x_i^j denotes the jth row of $\mathbf{X}_i$, and $\overline{x_i^j}$ denotes the jth row of mean matrix $\bar{\mathbf{X}}$. We choose the row eigenvectors corresponding to the first k_{row} largest eigenvalues of S_t^{row} to construct the row projection matrix $\mathbf{W}_r$:

$$\mathbf{W}_r = [v_1^{row}, v_2^{row}, \ldots, v_{k_{row}}^{row}] \tag{12.60}$$

where v_i^{row} denotes the row eigenvector corresponding to the ith largest eigenvalues of $\mathbf{S}_t^{row}$.

Similarly, by treating an image matrix $\mathbf{X}_i$ as an n-set of $m \times 1$ column vectors:

$$\mathbf{X}_i = [x_i^1 \quad x_i^2 \quad \ldots \quad x_i^n] \tag{12.61}$$

we obtain the column total column scatter matrix $\mathbf{S}_t^{col}$:

$$\mathbf{S}_t^{col} = \frac{1}{Nn}\sum_{i=1}^{N}(\mathbf{X}_i - \bar{\mathbf{X}})(\mathbf{X}_i - \bar{\mathbf{X}})^T \tag{12.62}$$

Then, we choose the column eigenvectors corresponding to the first k_{col} largest eigenvalues of $\mathbf{S}_t^{col}$ to construct the column projection matrix $\mathbf{W}_c$:

$$\mathbf{W}_c = [v_1^{col}, v_2^{col}, \cdots, v_{k_{\text{col}}}^{col}] \tag{12.63}$$

where v_i^{col} denotes the row eigenvector corresponding to the ith largest eigenvalues of $\mathbf{S}_t^{col}$.

Finally, we use the transformation:

$$\mathbf{Y} = \mathbf{W}_c^T \mathbf{X} \mathbf{W}_r \tag{12.64}$$

to extract the feature matrix $\mathbf{Y}$ of image matrix $\mathbf{X}$.

Actually, BDPCA is also a generalization of Yang's 2DPCA, and 2DPCA can be regarded as a special case of BDPCA with $\mathbf{W}_{col} = \mathbf{I}_m$, where $\mathbf{I}_m$ denotes an m-by-m identity matrix (Liu & Wechsler, 2003; Yuen & Lai, 2002).

While holistic PCA needs to solve an $N \times N$ eigenvalues problem, BDPCA has the advantage that it only needs to solve an $m \times m$ and $n \times n$ matrix eigenvalue problem. The $N \times N$ eigenvalue problem requires $O(N^3)$ computation (Golub & Van Loan, 1996), but BDPCA's eigenvalue problem requires $O(m^3)+O(n^3)$ computation. Usually the number of training samples N is larger than $\max(m, n)$. Thus, comparing with holistic PCA, BDPCA saves on training time while also requiring less time for feature extraction. For example, holistic PCA requires $100mn$ multiplication to extract a 100-dimensional feature vector, but BDPCA requires just $10mn+100n$ multiplication to extract a 10×10 feature matrix.

TWO-DIRECTIONAL PCA PLUS LDA

BDPCA +LDA: A New Strategy for Facial Feature Extraction

Here we propose our BDPCA+LDA method for facial feature extraction. The first part presents the algorithm of BDPCA+LDA, and the second part gives a detailed comparison of the BDPCA+LDA and PCA+LDA frameworks.

BDPCA + LDA Technique

In this section, we propose a BDPCA+LDA technique for fast facial feature extraction. BDPCA+LDA is an LDA approach that is applied on a low-dimensional BDPCA subspace. Since less time is required to map an image matrix to BDPCA subspace, BDPCA+LDA is, at least, computationally faster than PCA+LDA.

BDPCA+LDA first uses BDPCA to obtain feature matrix **Y**:

$$\mathbf{Y} = \mathbf{W}_c^T \mathbf{X} \mathbf{W}_r \tag{12.65}$$

where $\mathbf{W}_c$ and $\mathbf{W}_r$ are the column and row projectors, **X** is an original image matrix and **Y** is a BDPCA feature matrix. Then, the feature matrix **Y** is transformed into feature vector y by concatenating the columns of **Y**. The LDA projector $\mathbf{W}_{\text{LDA}} = [\varphi_1, \varphi_2, \ldots, \varphi_m]$ is calculated by maximizing Fisher's criterion:

$$J(\boldsymbol{\varphi}) = \frac{\boldsymbol{\varphi}^T \mathbf{S}_b \boldsymbol{\varphi}}{\boldsymbol{\varphi}^T \mathbf{S}_w \boldsymbol{\varphi}} \tag{12.66}$$

where φ_i is the generalized eigenvector of $\mathbf{S}_b$ and $\mathbf{S}_w$ corresponding to the ith largest eigenvalue λ_i:

$$\mathbf{S}_b \varphi_i = \lambda_i \mathbf{S}_w \varphi_i \tag{12.67}$$

and $\mathbf{S}_b$ is the between-class scatter matrix of y:

$$\mathbf{S}_b = \frac{1}{N} \sum_{i=1}^{C} N_i (\boldsymbol{\mu}_i - \boldsymbol{\mu})(\boldsymbol{\mu}_i - \boldsymbol{\mu})^T \tag{12.68}$$

and $\mathbf{S}_w$ is the within-class scatter matrix of y:

$$\mathbf{S}_w = \frac{1}{N} \sum_{i=1}^{C} \sum_{j=1}^{N_i} (y_{i,j} - \boldsymbol{\mu}_i)(y_{i,j} - \boldsymbol{\mu}_i)^T \tag{12.69}$$

where μ_i is the mean vector of class i, N_i is the number of samples of class i, $y_{i,j}$ is the jth feature vector of class i, C is the number of classes and μ is the mean vector of all training feature vectors.

In summary, the main steps in BDPCA+LDA feature extraction are as follows: We first transform an image matrix **X** into BDPCA feature subspace **Y** by Equation 12.65, and map **Y** into its 1D representation, y. We then obtain the final feature vector z by:

$$z = \mathbf{W}_{\text{LDA}}^{\text{T}} y \tag{12.70}$$

Advantages over the Existing PCA + LDA Framework

In this section, we compare the BDPCA+LDA and PCA+LDA face recognition frameworks in terms of computation and memory requirements. What should be noted

is that the computation requirement consists of two parts: that involved in the training phase and that involved in the testing phase.

We first compare the computation requirement using the number of multiplications as a measure of computational complexity. In the training phase, there are two computational tasks: (a) calculation of the projector, and (b) projection of images into feature prototypes. To calculate the projector, the PCA+LDA method must solve an $N\times N$ eigenvalue problem and then a $d_{PCA}\times d_{PCA}$ generalized eigenvalue problem, where N is the size of the training set and d_{PCA} is the dimension of the PCA subspace. In contrast, BDPCA+LDA must solve both an $m\times m$, an $n\times n$ eigenvalue problem and a $d_{BDPCA}\times d_{BDPCA}$ generalized eigenvalue problem, where d_{BDPCA} is the dimension of BDPCA subspace. Since the complexity of an $M\times M$ eigenvalue problem is $O(M^3)$ (Golub & Van Loan, 1996), the complexity of the PCA+LDA projector-calculation operation is $O(N^3+d_{PCA}{}^3)$, whereas that of BDPCA+LDA is $O(m^3+n^3+d_{BDPCA}{}^3)$. Usually m, n, d_{PCA} and d_{BDPCA} are smaller than the number of training samples N. To calculate the projector, then, BDPCA+LDA requires less computation than PCA+LDA.

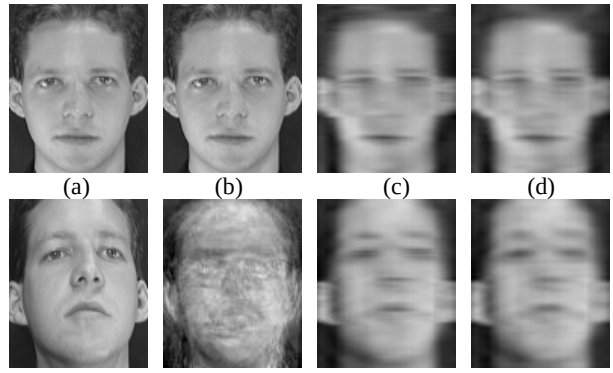
To project images into feature prototypes, we assume that the feature dimension of BDPCA+LDA and PCA+LDA is the same, d_{LDA}. The number of multiplications, thus, is $N_p\times(m\times n)\times d_{LDA}$ for PCA+LDA and is less than $N_p\times(m\times n\times\min(k_{row},k_{col})+(k_{col}\times k_{row})\times\max(m+d_{LDA}, n+d_{LDA}))$, where N_p is the number of prototypes. In this section, we use all the prototypes for training; thus, $N_p=N$. Generally $\min(k_{row}, k_{col})$ is much less than d_{LDA}. In the projection process, then, BDPCA+LDA also requires less computation than PCA+LDA.

In the test phase, there are two computational tasks: (c) the projection of images into the feature vector, and (d) the calculation of the distance between the feature vector and feature prototypes. In the testing phase, BDPCA+LDA requires less computation. There are a number of reasons for this. One is that, as discussed above, when projecting images into feature vectors, BDPCA+LDA requires less computation than PCA+LDA. Another reason is that, because the feature dimension of BDPCA+LDA and PCA+LDA is the same, in the similarity measure process the computational complexity of BDPCA+LDA and PCA+LDA are equivalent.

The memory requirements of the PCA+LDA and BDPCA+LDA frameworks mainly depend on the size of the projector and the total size of the feature prototypes. The size of the projector of PCA+LDA is $d_{LDA}\times m\times n$. This is because the PCA+LDA projector contains d_{LDA} fisherfaces, each of which is the same size as the original image. The BDPCA+LDA projector is in three parts, $\mathbf{W}_{col}$, $\mathbf{W}_{row}$ and $\mathbf{W}_{opt}$. The total size of the BDPCA+LDA projector is $(k_{col}\times m)+(k_{row}\times n)+(d_{LDA}\times k_{col}\times k_{row})$, which is much smaller than that of PCA+LDA. Finally, because these two methods have the same feature dimensions, BDPCA+LDA and PCA+LDA have equivalent feature prototype memory requirements.

We have compared the computation and memory requirements of the BDPCA+LDA and PCA+LDA frameworks, as listed in Table 12.7. We can see that the BDPCA+LDA framework is superior to the PCA+LDA in both computational and memory requirements.

Figure 12.3. Comparison of the reconstruction capability of PCA, 2DPCA and BD-PCA



(a), (e): Original images; (b), f): reconstruction images by PCA; (c), g): reconstruction images by 2DPCA; and (d), (h): reconstruction images by BD-PCA

EXPERIMENTAL RESULTS

To evaluate the efficiency of BDPCA using the AMD metric (BDPCA-AMD), we used two image databases, the ORL face database and the PolyU palmprint database. (PolyU, n.d.) For each database, we investigated the effect of AMD parameter, and compared the recognition performance of different distance measures. We also compared the recognition rate obtained using BDPCA-AMD with that obtained using some other popular image recognition techniques, such as eigenfaces, fisherfaces (Belhumeur, Hespanha, & Kriegman, 1997) and DLDA (Yu & Yang, 2001).

To test the efficiency of the BDPCA+LDA method, we make use of three face databases, the ORL (Olivetti, n.d.), UMIST (Zhang, Kong, You, & Wong, 2003; Graham & Allinson, 1998b) and FERET (Phillips, Moon, Rizvi, & Rauss, 2000; Phillips, 2001). We also compare the proposed method with several representative appearance-based approaches, including eigenfaces, fisherfaces and DLDA.

The experimental setup is as follows: Since our aim is to evaluate the effectiveness of feature extraction methods, we use a simple classifier, the NN classifier. For all our experiments, we randomly select n samples of each individual to construct the training set, and use the others as testing samples. In the following experiments, to reduce the variation of recognition results, we adopt the mean of 10 runs as the average recognition rate (ARR). All the experiments are carried out on an AMD 2500+ computer with 512Mb RAM and tested on the MATLAB platform (Version 6.5).

Experiments with the ORL Database for BDPCA

To test the performance of the proposed approach, a series of experiments are carried out using the ORL database. First, we give an intuitional illustration of BDPCA's reconstruction performance. Then, we evaluate the capability of BDPCA in solving the over-fitting problem. We also evaluate the effectiveness of the assembled matrix distance

metric and compare the recognition performance of the proposed approach with that of PCA (Draper, Baek, Bartlett, & Beveridge, 2003) and 2DPCA.

In the first set of experiments, we compare the reconstructed capability of PCA, 2DPCA and BDPCA (Jain, 1989). Figure 12.3 shows two original facial images and its reconstructed images by PCA, 2DPCA and BDPCA. Figure 12.3a is an original facial image from the training set. A satisfied reconstruction image of Figure 12.3a can be obtained using all of these three approaches, as shown in Figure 12.3b-d. The best reconstructed image is that reconstructed by PCA, as shown in Figure 12.3b. Figure 12.3e is a facial image from the testing set and its reconstructed images by PCA, 2DPCA and BDPCA are shown in Figure 12.3f-h. The quality of the reconstructed image by PCA deteriorates greatly, while both 2DPCA and BDPCA can obtain a satisfied reconstruction quality. Note that the feature dimensionality of 2DPCA is 8´112=896, much higher than that of BDPCA (8×30=240) and that of PCA (180). The experimental results indicate that for the training samples, PCA has the best reconstruction capability, but 2DPCA and BDPCA also can obtain satisfied reconstructed quality. For the testing samples, the reconstructed quality of PCA deteriorates greatly, while 2DPCA and BDPCA still have satisfied reconstruction performance. Besides, the feature dimensionality of BDPCA is much less than 2DPCA.

In the second set of experiments, we use the normalized MSE to evaluate BDPCA's capability in solving the over-fitting problem. Given the column projection W_{col} and row projection W_{row}, an original image X can be mapped into its BDPCA representation Y:

$$Y = W_{col}^T (X - \overline{X}) W_{row} \tag{37}$$

and the reconstructed image $\tilde{X}$ can be represented as:

$$\tilde{X} = \overline{X} + W_{col} Y W_{row}^T = \overline{X} + W_{col} W_{col}^T (X - \overline{X}) W_{row} W_{row}^T \tag{38}$$

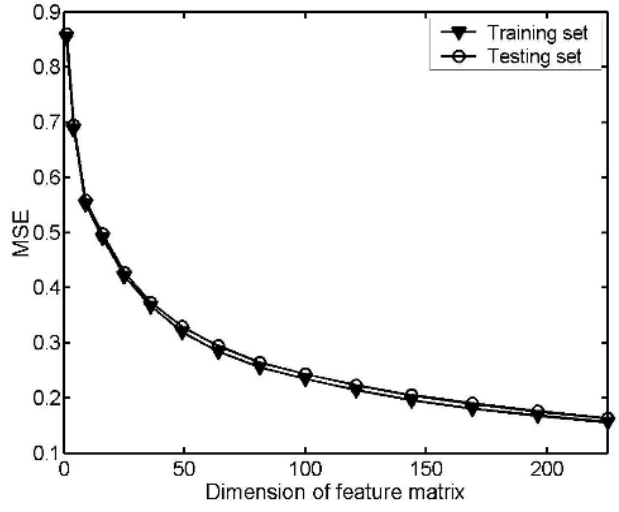
Then, the normalized MSE on the training set MSE^{train} can be defined as:

$$MSE^{train} = \sum_{i=1}^{N_1} \left\| X_i^{train} - \tilde{X}_i^{train} \right\|^2 \Big/ \sum_{i=1}^{N_1} \left\| X_i^{train} - \overline{X^{train}} \right\|^2 \tag{39}$$

where N_1 is the number of training samples, X_i^{train} is the ith training image matrix, $\tilde{X}_i^{train}$ is reconstructed image of X_i^{train}, and $\overline{X^{train}}$ is the mean matrix of all training images. Similarly, we can define the normalized MSE on the testing set MSE^{test} as:

$$MSE^{test} = \sum_{i=1}^{N_2} \left\| X_i^{test} - \tilde{X}_i^{test} \right\|^2 \Big/ \sum_{i=1}^{N_2} \left\| X_i^{test} - \overline{X^{test}} \right\|^2 \tag{40}$$

Figure 12.4. The BD-PCA's normalized MSE on the training set and the testing set as the function of feature dimension



where N_2 is the number of testing images, X_i^{test} is the ith testing image matrix, $\tilde{X}_i^{test}$ is the reconstructed image of X_i^{test}, and $\overline{X^{test}}$ is the mean matrix of all testing images.

By selecting the first five images per individual for training, we calculate MSE^{train} and MSE^{test} for given W_{col} and W_{row}, as shown in Figure 12.4. What to be noted is that we set the number of the row eigenvectors k_{row} equal to the number of column eigenvectors k_{col}, and, thus, the dimensionality of the feature matrix $L = k_{row} \times k_{row}$. Figure 12.4 shows that the difference of MSE^{train} and MSE^{test} is very small. Thus, BDPCA can solve the over-fitting problem successfully.

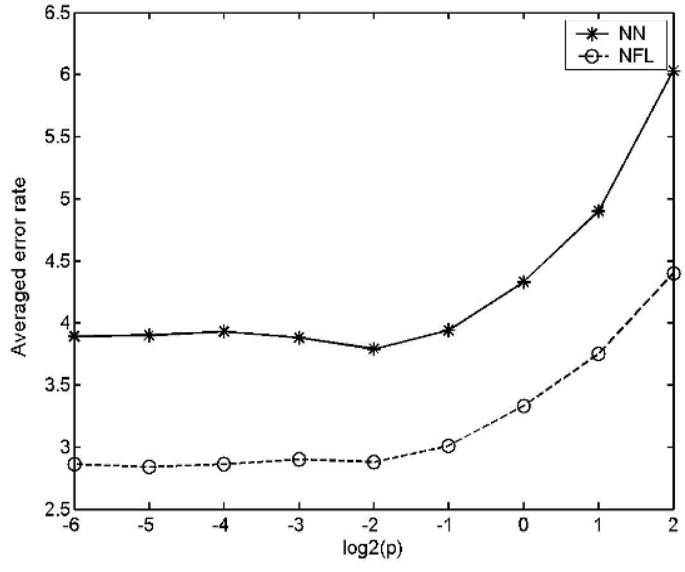
In all the following experiments, we randomly choose n samples per individual for training, resulting in a training set of $40\times n$ images and a testing set of $40\times(10-n)$ images with no overlap between the two sets. For BD-PCA, we choose first four row vectors as row projection matrix W_{row} (k_{row}=4), first 18 column vectors as column projection matrix W_{col} (k_{col}=18), and set the AMD parameter p_1=2. To reduce the variation of recognition results, an AER is adopted by calculating the mean of error rates over 20 runs.

In the third set of experiments, we study the effect of distance measures and the classifiers on the recognition performance of BDPCA. Figure 12.5 shows the effect of the assembled matrix distance parameter p_2 on recognition performance with five training samples per individual. The lowest AER can be obtained for both NN and NFL classifiers when $p_2 \le 0.25$. The AER increases with the augmentation of parameter p when $p_2 \le 0.25$. So, we determine the assembled matrix distance parameter $p_2 = 0.25$. Table 12.1 compares the AER obtained using the Frobenius distance, the Yang distance and the AMD measures. It can be observed that the AMD metric achieved the lower AER for both NN and NFL classifiers, and NFL with AMD measure has better recognition performance than

Table 12.1. Comparison of average error rates obtained using different distance measures and classifiers on the ORL database

Classifier	Frobenius	Yang	AMD
NN	4.90	4.33	3.78
NFL	3.75	3.33	2.88

Figure 12.5. Average error rates obtained using BDPCA with different p_2 values



NFL with the other two distance measures. In the following experiments, we use BDPCA-NN to denote BDPCA with AMD using the NN classifier and BDPCA-NFL to denote BDPCA with AMD using the NFL classifier.

Figure 12.6 depicts the AER with different n values. It is very interesting to point out that the improvement of BDPCA-NFL over BDPCA-NN is very small when the number of training samples $n \geq 7$. This observation indicates that NN can achieve a comparative recognition performance to NFL when the number of templates is enough.

In the fourth set of experiments, we carry out a comparative study on PCA and BDPCA with n=5. Figure 12.7 depicts the plot of the error rates of 20 runs obtained by PCA and BDPCA, while Table 12.2 lists the AER and variance of each method. It is obvious to see that BDPCA outperforms PCA in recognition performance and the variance for both NN and NFL classifiers. The AER of BDPCA is about 0.597 of that of PCA for the NN classifier, and 0.577 for the NFL classifier.

In the last set of experiments, the performance of BDPCA is compared with that of other appearance-based methods, with n=5. First, we implement two classical LDA-based

Figure 12.6. Comparison average error rate obtained using NN and NFL classifiers

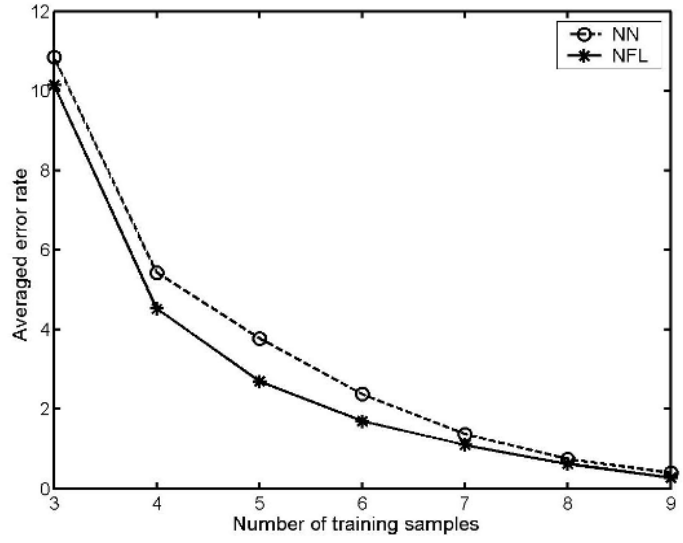
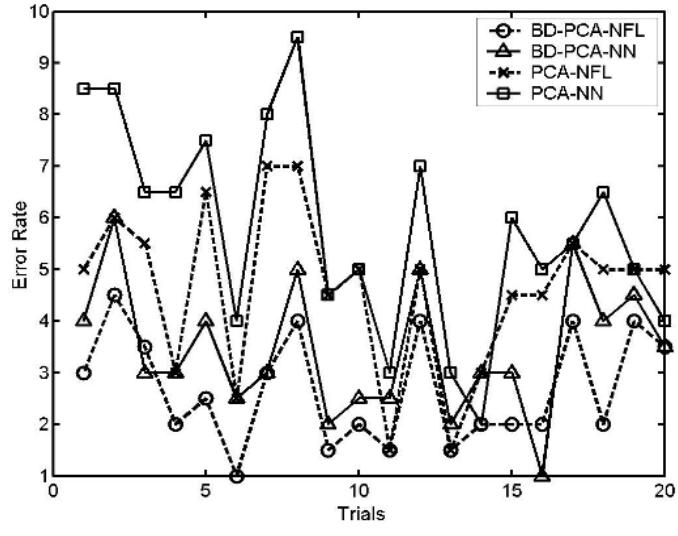


Figure 12.7. Plots of error rates for each runs



methods, fisherfaces (Belhumeur, Hespanha, & Kriegman, 1997) and DLDA (Yu & Yang, 2001). Table 12.4 shows the AER obtained using fisherfaces, DLDA and BDPCA. As references, we also compare the recognition performance of BDPCA with some recently reported results obtained by other appearance-based methods using the ORL database. The error rate is 4.05 for Ryu's SHC method (Ryu & Oh, 2002), 3.85 for Wang's CLSRD

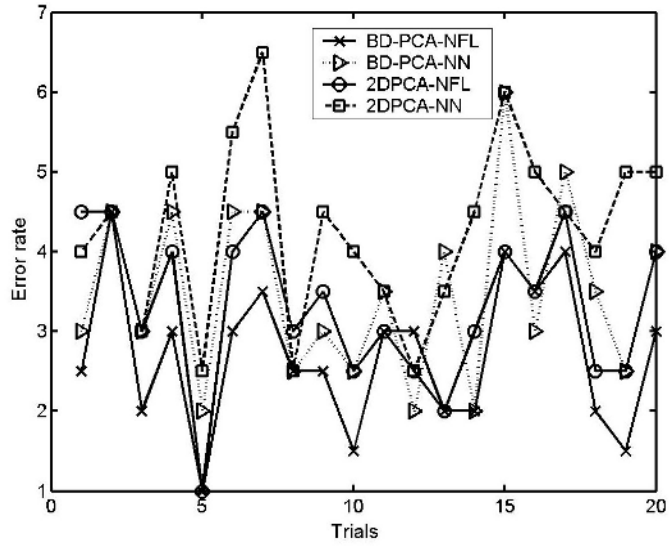
Table 12.2. Comparison of average error rates obtained using PCA and BDPCA on the ORL database

Methods	PCA-NN	PCA-NFL	BD-PCA-NN	BD-PCA-NFL
AER (%)	5.78	4.63	3.45	2.68
Variance	1.99	1.55	1.25	1.04

Table 12.3. Comparison of average error rates obtained using 2DPCA and BDPCA on the ORL database

Methods	2DPCA-NN	2DPCA-NFL	BD-PCA-NN	BD-PCA-NFL
AER(%)	4.28	3.30	3.48	2.70
Variance	1.10	0.94	1.08	0.90

Figure 12.8. Plots of error rates for each runs



(Wang & Zhang, 2004), 3.0 for Yang's complete PCA+LDA (Yang & Yang, 2003), 4.2 for Lu's DF-LDA (Lu, Plataniotis, & Venetsanopoulos, 2003), 4.9 for Liu's NKFDA (Liu, Wang, Li, & Tan, 2004), 4.2 for Song's LMLP (Song, Yang, & Liu, 2004) and 4.15 for Zheng's ELDA method (Zheng, Zhao, & Zou, 2004). Note that the reported results are obtained on the average of a different number of runs. While comparing with these results, BDPCA is still very effective and competitive.

Table 12.4. Comparison of average error rates obtained using different methods on the ORL database

Methods	Fisherfaces	D-LDA	BD-PCA-NN	BD-PCA-NFL
AER (%)	11.4	5.4	3.48	2.70

Experiments with the PolyU Palmprint Database for BDPCA

Palmprint sampling is low-cost, non-intrusive, and palmprint has a stable structural feature, making palmprint recognition the object of considerable recent research interest (Graham & Allinson, 1998a). Here we use the PolyU palmprint database (PolyU, n.d.) to test the efficiency of BD-PCA-AMD. The PolyU palmprint database contains 600 grayscale images of 100 different palms with six samples for each palm. Six samples from each of these palms were collected in two sessions, where the first three samples were captured in the first session and the other three in the second session. The average interval between the first and the second session was two months. In our experiments, sub-image of each original palmprint was cropped to the size of 128´128 and pre-processed by histogram equalization. Figure 12.9 shows 6 palmprint images of one palm. For the PolyU palmprint database, we choose the first 3 samples per individual for training, and thus use all the 300 images captured in the first session as training set and the images captured in the second session as testing set.

With the number of row vectors k_{row} = 13, the number of column vectors k_{col} = 15 and the AMD parameter p_1=1, we studied the effect of AMD parameter p_2. Figure 10 shows the error rate of BDPCA-AMD with different p_2 values. The lowest error rate can be obtained for both NN and NFL classifiers when $p_2 \leq 0.25$. So we set the AMD parameter $p_2 = 0.25$.

Figure 12.9. Six palmprint images of one palm in the PolyU palmprint database

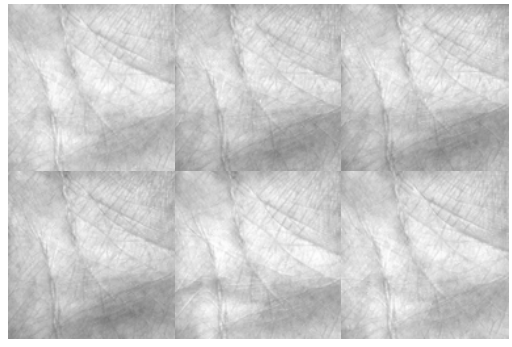


Figure 12.10. Error rates of BDPCA-AMD with different p_2 values

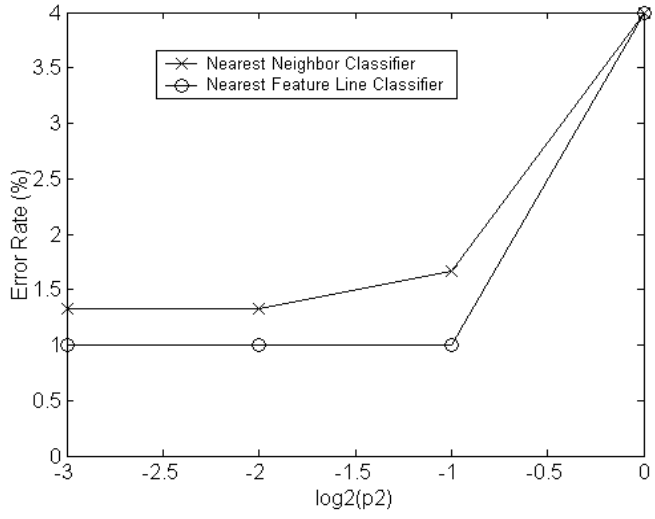


Figure 12.11. Error rates of BDPCA obtained with different k_{col} and k_{row} values

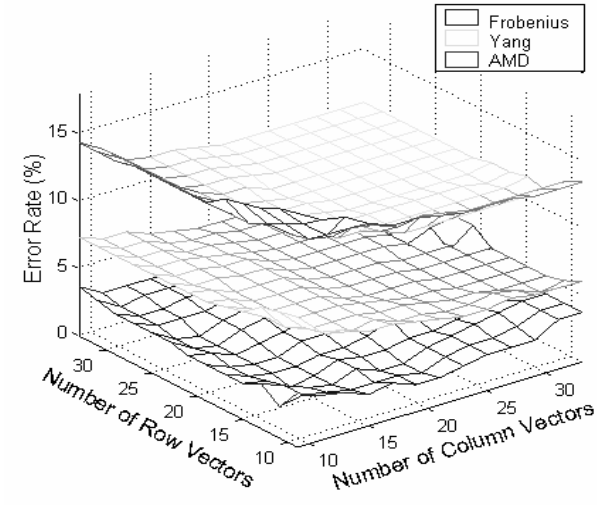


Figure 12.11 depicts the error rates of BDPCA obtained using the Frobenius, Yang and AMD distance measures with the NN classifier. The lowest error rate obtained using AMD measure is 1.33, lower than that obtained using the Frobennius and Yang distance measures. Table 12.5 compares the error rates obtained using different distance measures. It can be observed that the AMD metric achieved the lower AER for both NN and NFL classifiers, and NFL using AMD metric has better recognition performance than

Figure 12.12. Error rates of 2DPCA obtained with different number of principal component vectors

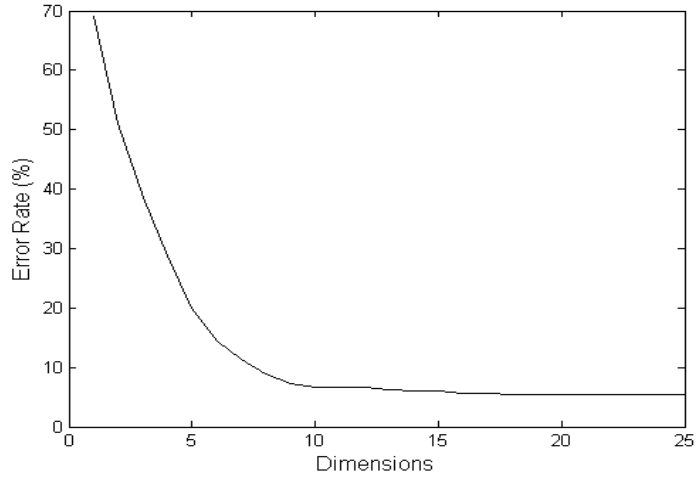


Table 12.5. Comparison of error rates obtained using different distance measures and classifiers. on the PolyU palmprint database

Classifiers	Frobenius	Yang	AMD
NN	11.00	4.67	1.33
NFL	11.00	4.33	1.00

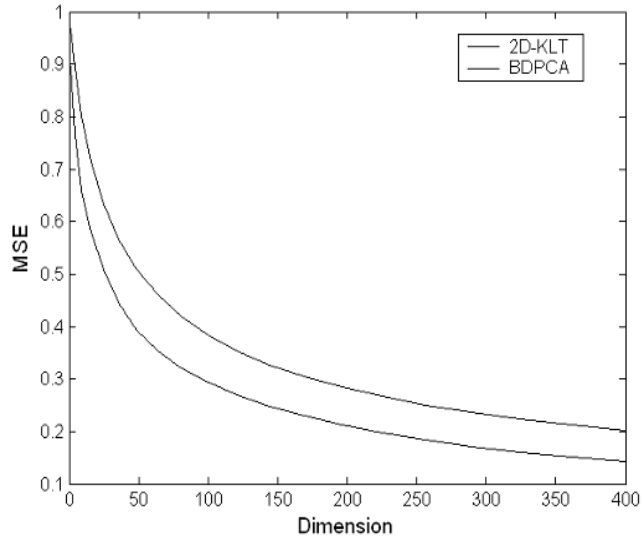
Table 12.6. Comparison of AER obtained using different methods on the PolyU palmprint database

Methods	Eigenfaces	Fisherfaces	D-LDA	BD-PCA-NFL
Error Rate (%)	11.33	6.67	6.00	1.00

conventional NFL. Figure 12.12 shows the recognition rate of Yang's 2DPCA obtained using different k_{row} values. The lowest error rate obtained using 2DPCA is 4.40, higher than that obtained using BDPCA-AMD.

Table 12.6 compares the error rates obtained using eigenfaces, fisherfaces, DLDA and BDPCA. The lowest error rate of BDPCA-NFL is 1.00, lower than that obtained using other three image recognition methods.

Figure 12.13. The curve of mean-square error of 2D-KLT and BDPCA



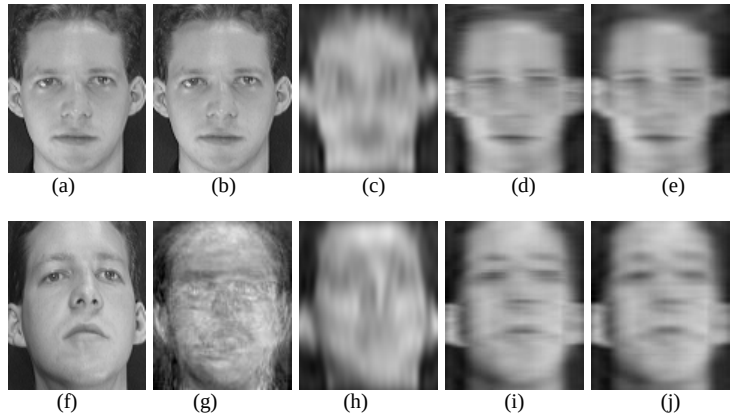
Experiments on the ORL Database for BDPCA+LDA

We use the ORL face database to test the performance of BDPCA+LDA in dealing with small light, expressions, scale and pose variation. The ORL database contains 400 facial images with 10 images per individual. Figure 12.1 shows the 10 images of one person. The images are collected with various age, light conditions, facial expressions, facial details (glasses/no glasses), scale and tilt (Olivetti, n.d.). To evaluate the performance of the proposed method, we first compare the mean-square error of 2D-KLT and BDPCA, and then compare the reconstruction performance of PCA, 2D-KLT, 2DPCA and BDPCA. Next, we present an intuitive illustration of discriminant vectors obtained using BDPCA+LDA. Finally, we carry out a comparison analysis of BDPCA+LDA and other facial feature extraction methods.

With reference to the comparison of the mean-square error (MSE) of 2D-KLT (r=0.9) and BDPCA, Figure 12.13 shows the curve of MSE corresponding to 2D-KLT and BDPCA. We can see that the MSE of BDPCA is lower than that of 2D-KLT and that the face-image-specific BDPCA represents facial images more effectively than the content-independent 2D-KLT.

We now intuitively compare the PCA, 2D-KLT, 2DPCA and BDPCA image reconstructions. Figure 12.14 shows two original facial images and their reconstructions. Figure 12.14a is a training image. We can see in Figures 12.14b-e that the quality of the PCA, 2DPCA and BDPCA reconstruction is satisfactory, and PCA is the best, but that of 2D-KLT is not. Figure 12.14f is a facial image from the testing set. Figures 12.14g-j are the reconstructed images. We can see that the quality of classical PCA has greatly deteriorated and that 2D-KLT is poor but 2DPCA and BDPCA still perform well. It should be noted that the feature dimension of 2DPCA is 896 (112×8), much higher than that of PCA (180), 2D-KLT (30×8=240) and BDPCA (30×8=240).

Figure 12.14. Comparisons of the reconstruction capability of four methods



(a), (f): the original images, reconstructed images by (b); (g): PCA; (c), (h): 2D-KLT; (d),(i): 2DPCA; and (e), (j): BD-PCA

Figure 12.15 presents an intuitive illustration of fisherfaces and BDPCA+LDA's discriminant vectors. Figure 12.15a shows the first five discriminant vectors obtained using fisherfaces, and Figure 12.15b depicts the first five discriminant vectors obtained using BDPCA+LDA. It can be observed that the appearance of fisherface's discriminant vectors is different from that of BDPCA+LDA. This establishes the novelty of the proposed LDA-based facial feature extraction technique.

BDPCA introduces two new parameters, the number of column eigenvectors k_{col} and that of row eigenvectors k_{row}. Table 12.8 depicts the effect of k_{col} and k_{row} on the ARR

Figure 12.15. An example of the discriminant vectors of (a) fisherfaces, and (b) BDPCA+LDA

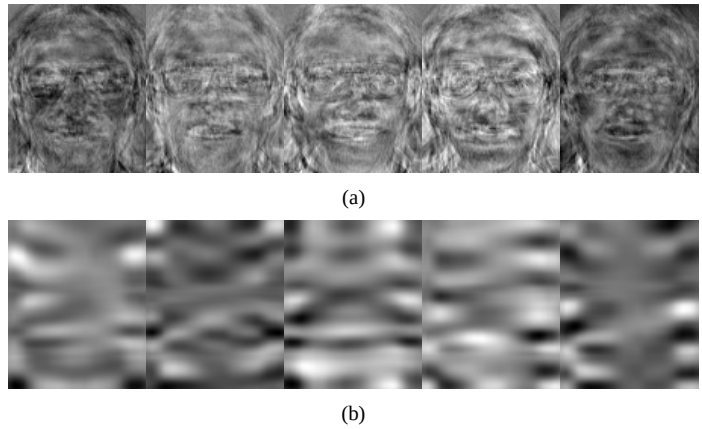


Table 12.7. Comparisons of memory and computation requirements of BDPCA+LDA and PCA+LDA

Method	Memory Requirements		Computation Requirements	
	Projector	Feature prototypes	Training	Testing
PCA+LDA	$(m\times n)\times d_{LDA}$ **Large**	$N\times d_{LDA}$ **Same**	a) Calculating the projector: $O(N_p^3+d_{PCA}^3)$ **Large** b) Projection: $N\times(m\times n)\times d_{LDA}$ **Large**	c) Projection: $(m\times n)\times d_{LDA}$ **Large** d) Distance calculation: $N\times d_{LDA}$ **Same**
BDPCA +LDA	$m\times k_{row}+ n\times k_{col}+ k_{col}\times k_{row}\times d_{LDA}$ **Small**	$N\times d_{LDA}$ **Same**	a) Calculating the projector: $O(m^3+n^3+d_{BDPCA}^3)$ **Small** b) Projection: $N\times[m\times n\times\min(k_{row}, k_{col}) +k_{col}\times k_{row}\times\max(m+d_{LDA}, n+d_{LDA})]$ **Small**	c) Projection: $m\times n\times\min(k_{row}, k_{col})+ k_{col}\times k_{row}\times[\max(m,n)+d_{LDA}]$ **Small** d) Distance calculation: $N\times d_{LDA}$ **Same**

obtained using BDPCA with the number of training samples $n_p = 5$. As Table 12.8 shows, the number of row eigenvectors k_{row} has an important effect on BDPCA's recognition performance and the maximum ARR is obtained when $k_{row} = 4$. When the number of column eigenvectors $k_{col} > 12$, k_{col} has little effect on the ARR of BDPCA. Table 3 shows the ARR of BDPCA+LDA with different k_{col} and k_{row} values and the minimum ARR, 97.1%, can be obtained when $k_{col} = 12$ and $k_{row} = 4$.

We now compare the computation and memory requirements of BDPCA+LDA and PCA+LDA (fisherfaces) with the number of training samples $n_p = 5$. Table 12.10 shows

Table12.8. Comparisons of ARR obtained using BDPCA with different parameters

ARR k_{row} \ k_{col}	1	6	12	18	24	30	112
1	0.138	0.851	0.905	0.924	0.920	0.920	0.919
4	0.616	0.942	0.949	**0.951**	0.949	0.950	0.950
8	0.719	0.937	0.941	0.941	0.943	0.941	0.940
12	0.734	0.941	0.943	0.942	0.942	0.942	0.941
16	0.742	0.936	0.944	0.941	0.942	0.941	0.940
20	0.741	0.934	0.942	0.944	0.943	0.942	0.940
92	0.741	0.936	0.944	0.945	0.944	0.942	0.941

Table 12.9. Comparisons of ARR obtained using BDPCA+LDA with different parameters

ARR k_{row} / k_{col}	4	8	10	12	15	20	25
2	0.937	0.957	0.957	0.954	0.958	0.944	0.940
3	0.932	0.957	0.958	0.967	0.957	0.948	0.940
4	0.947	0.958	0.970	**0.971**	0.968	0.959	0.939
6	0.954	0.955	0.968	0.960	0.950	0.933	0.896
8	0.958	0.955	0.965	0.956	0.940	0.906	–
10	0.944	0.954	0.959	0.934	–	–	–
12	0.940	0.942	0.938	0.864	–	–	–

Table 12.10. The total CPU time (s) for training and testing on the ORL database

Method	Time for Training (s)	Time for Testing (s)
PCA+LDA	46.0	5.2
BDPCA+LDA	17.5	3.9

that BDPCA+LDA is much faster than fisherfaces both for training and for testing. Table 12.11 compares BDPCA+LDA and fisherfaces in terms of memory and computational requirements and shows that BDPCA+LDA needs much less memory and has a lower computational requirement than fisherfaces.

Comparing the recognition performance of BDPCA+LDA with other feature extraction methods, such as eigenfaces, (Swets & Went, 1996; Torkkola, 2001) fisherfaces, D-LDA and 2DPCA, Figure 12.16 shows the ARR obtained by different approaches. BDPCA+LDA obtains the highest ARR for all n_p values.

To evaluate the recognition performance of BDPCA+LDA, we also compare its recognition accuracy with some recently reported results. Table 12.12 shows the reported recognition rates obtained by other LDA-based methods using the ORL database with the number of training samples $n_p = 5$. It should be noted that some results were evaluated based on performance of just one run (Yuen & Lai, 2002) and that some results were evaluated based on the average recognition rate of 5 runs or 10 runs (Pratt, 2001; Mallat, 2002). From Table 12.12, BDPCA+LDA is very effective and competitive in facial feature extraction.

Experiments on the UMIST Database for BDPCA+LDA

The UMIST face database is used to test the recognition performance of BDPCA+LDA where images contain a wide variety of poses. The UMIST repository is

Table 12.11. Comparisons of memory and computation requirements of BDPCA+LDA and PCA+LDA on the ORL database

Method	Memory Requirements			Computation Requirements	
	Projector	Feature prototypes	Total	Training	Testing
PCA+LDA	(112×92)×39 *=401856*	200×39 *=7800*	*409656*	a) Calculating the projector: $O(200^3+160^3)$? *12096000* b) Projection: 200×(112×92)×39 *=80371200* Total=*92467200*	c) Projection: (112×92)×39 *=401856* d) Distance calculation: 200×39 *=7800* Total=*409656*
BDPCA+LDA	112×12+92×4+(12×4)×39 *=3584*	200×39 *=7800*	*11384*	a) Calculating the projector: $O(112^3+92^3+(12\times4)^3)$? *2294208* b) Projection: 200×[112×92×4+12×4×(112+39)] *=9692800* Total=*11987008*	c) Projection: 112×92×4+4×12×(112+39) *=48464* d) Distance calculation: 200×39 *=7800* Total=*56264*

Table 12.12. Other results recently reported on the ORL database

Methods	Recognition Rate	Year
Complete PCA+LDA [24]	0.970	2003
DF-LDA [39]	0.958	2003
NKFDA [40]	0.951	2004
ELDA [41]	0.9585	2004
BDPCA+LDA	0.9707	

a multi-view database consisting of 564 images of 20 individuals. Each subject has provides between 19 and 48 samples and includes a wide range of viewing angles from profile to frontal views, and includes subjects of either sex, and of diverse appearance and ethnic backgrounds. The cropped images are 92×112. Figure 12.17. illustrates some samples of one person in the UMIST database. For this database, we set the number of column eigenvectors to $k_{\text{col}} = 10$, and the number of row eigenvectors to $k_{\text{row}} = 3$.

Figure 12.18 depicts the ARR obtained using eigenfaces, fisherfaces, D-LDA and BDPCA+LDA with differing numbers of training samples n_{p}. Again, the BDPCA+LDA

Figure 12.16. Comparisons of ARR obtained using different methods on the ORL database

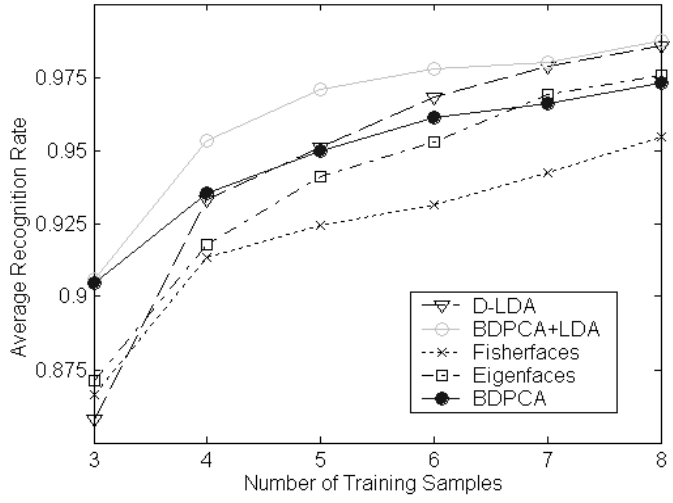


Figure 12.17. Ten images of one individual from the UMIST database

method has the highest ARR for all n_p values. Thus, BDPCA+LDA is an effective facial feature extraction technique where the poses in images are varied.

To further evaluate the recognition performance of BDPCA+LDA, we compare its recognition accuracy with some recently reported results. Table 12.13 lists the reported recognition rates obtained by other appearance-based methods using the UMIST database with different n_p values. It should be noted that some results were evaluated with the number of training samples $n_p = 6$ (Lu, Plataniotis & Venetsanopoulos, 2003), other results were evaluated with $n_p = 8$ (Lu, Plataniotis, & Venetsanopoulos, 2003), 9 (Gupta, Agrawal, Pruthi, Shekhar, & Chellappa, 2002) or 10 (Zhang, Li, & Wang, 2004); some results were evaluated based on average recognition rate of 5 runs (Lu, Plataniotis, & Venetsanopoulos, 2003), and other results were evaluated based on average recognition rate of 8 (Lu, Plataniotis, & Venetsanopoulos, 2003), 10 (Gupta, Agrawal, Pruthi,

Figure 12.18. Comparisons of ARR obtained using different methods on the UMIST database

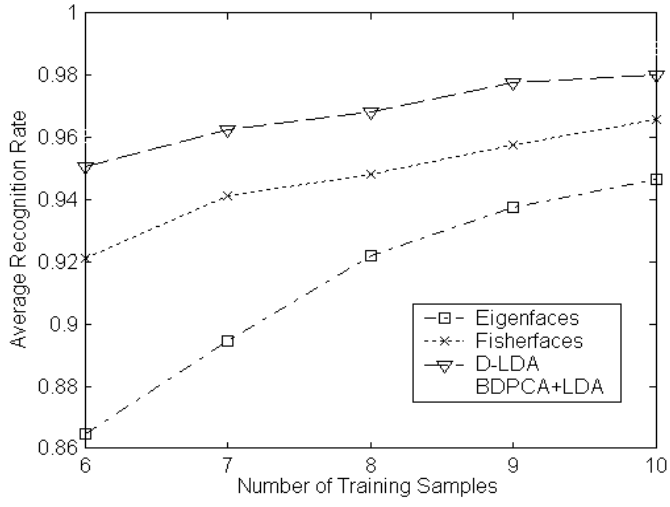


Table 12.13. Other results recently reported on the UMIST database

Methods	Number of training samples	Recognition Rate	Year
LDA+RBF SVM [9]	9	0.9823	2002
KDDA [10]	6	0.954	2003
DF-LDA [1]	8	0.978	2003
MLA [11]	10	0.9627	2004

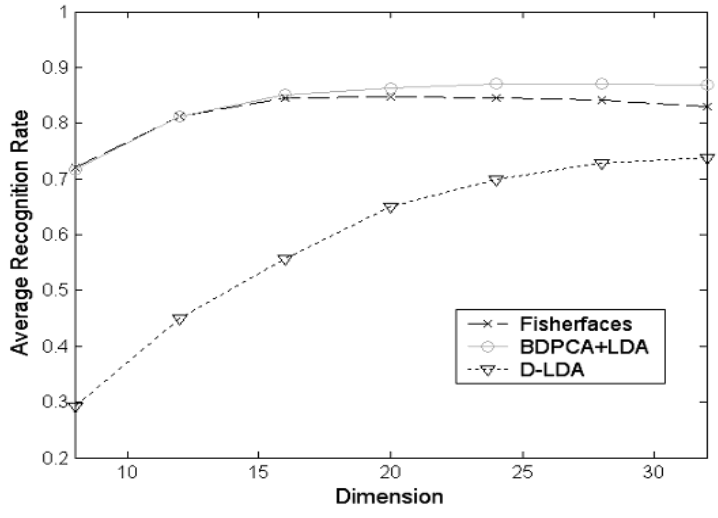
Shekhar, & Chellappa, 2002) or 100 (Zhang, Li, & Wang, 2004) runs. From Table 12.13, we can see that BDPCA+LDA is still very competitive while comparing with these results.

Experiments on the FERET Database for BDPCA+LDA

The FERET face image database is a result of the FERET program, which was sponsored by the Department of Defense through the DARPA Program. So far, it has become a standard database to test and evaluate face recognition algorithms. In this section, we choose a subset of the FERET database consisting of 1,400 images corresponding to 200 individuals (each individual has seven images, including a front image and its variations in facial expression, illumination, ±15° and ±30° pose). The facial portion of each original image was cropped to the size of 80×80 and pre-processed by histogram equalization. In our experiments, we randomly selected three images of each

Figure 12.19. Seven images of one individual from the FERET subset

Figure 12.20. Comparisons of ARR obtained using different methods on the FERET subset



subject for training, thus resulting in a training set of 600 images and a testing set of 800 images. Figure 12.19. illustrates the seven cropped images of one person.

Previous work on the FERET database indicates that the dimensionality of PCA subspace has an important effect on recognition accuracy of PCA+LDA (Zhao, Chellappa, & Phillips, 1999), and Yang has demonstrated that maximum recognition rate occurs with the number of discriminant vectors d_{LDA} within the interval from 6 to 26 (Yang, Yang, & Frangi, 2003). Here, we experimentally found that the maximum recognition accuracy can be obtained with $d_{\text{PCA}} = 100$ on the FERET subset.

Next, we compare the recognition accuracy of DLDA, fisherfaces and BDPCA+LDA. For BDPCA+LDA, we set the number of column eigenvectors $k_{\text{col}} = 15$ and that of row eigenvectors $k_{\text{row}} = 5$. Figure 12.20 plots the ARR as a function of the dimension of the feature subspace d_{LDA}. The maximum ARR of BDPCA+LDA is 0.8714, higher than that of D-LDA, and fisherfaces.

The FERET database is a much larger face database than the ORL and UMIST databases. Thus, we also presented a comparison analysis of BDPCA+LDA and fisherfaces using the FERET subset. Table 12.14 shows that the BDPCA+LDA framework

Table 12.14. The total CPU time(s) for training and testing on the FERET subset

Method	Time for Training (s)	Time for Testing (s)
PCA+LDA	254.2	36.2
BDPCA+LDA	57.5	26.3

Table 12.15. Comparisons of memory and computation requirements of BDPCA+LDA and PCA+LDA on the FERET subset

Method	Memory Requirements			Computation Requirements	
	Projector	Feature prototypes	Total	Training	Testing
Fisherfaces	(80×80)×24 =*153600*	600×24 =*14400*	*168000*	a)Calculating the projector: $O(600^3+100^3)$?*217000000* b)Projection: 600×(80×80)×24 =*92160000* Total=*309160000*	c)Projection: (80×80)×24 =*153600* d)Distance calculation: 600×24 =*14400* Total=*168000*
BDPCA +LDA	80×(5+15) +5×15×24 =*3400*	600×24 =*14400*	*17800*	a)Calculating the projector: $O(80^3+80^3+(15\times5)^3)$?*1445875* b)Projection: 600×[80×80×5+15×5 ×(80+24)] =*23880000* Total=*25325875*	c)Projection: 80×80×5+5×15 ×(80+24) =*39800* d)Distance calculation: 600×24 =*14400* Total=*54400*

is much faster than fisherfaces either in the training or testing phase. Compared with Table 12.10, we can see that much more training time is saved by the BDPCA+LDA framework for the FERET subset. This is because the computation complexity of BDPCA+LDA is $O(N)$ for training, while that of PCA+LDA is $O(N^3)$, where N is the size of training set. Table 15 compares the BDPCA+LDA and fisherfaces frameworks according to their memory and computation requirements. From this table, we can also see that BDPCA+LDA needs less memory and computation requirements than fisherfaces.

SUMMARY

In this chapter, we propose a BDPCA with assembled matrix distance measure method (BDPCA-AMD) for image recognition. The proposed method has some significant advantages. First, BDPCA is directly performed on image matrix, while classical PCA

is required to map an image matrix to a high-dimensional vector in advance. Second, BDPCA can circumvent classical PCA's over-fitting problem caused by the high dimensionality and SSS problem. Third, the feature dimensionality of BDPCA is much less than 2DPCA. Fourth, we present an assembled matrix distance metric to meet the fact that BDPCA feature is a matrix and apply the proposed distance metric to further improve the recognition performance of NN and NFL classifiers. BDPCA can achieve an AER of 3.45 using the ORL with five training samples per individual for the NN classifier; 2.68 for the NFL classifier. On the PolyU palmprint database, BDPCA-NN achieved an error rate of 1.33 and BDPCA-NFL achieved an error rate of 1.00.

In this chapter, we also propose a fast facial feature extraction technique, BDPCA+LDA, for face recognition. While comparing with the PCA+LDA (fisherfaces) framework, BDPCA+LDA has some significant advantages. First of all, BDPCA+LDA needs less computation requirement, either in the training or testing phases. The reason is twofold. On the one hand, compared to PCA+LDA, there are just some smaller eigen-problems required to be solved for BDPCA+LDA. On the other hand, BDPCA+LDA has a much faster speed for facial feature extraction. Second, BDPCA+LDA needs less memory requirement because its projector is much smaller than that of PCA+LDA. Third, BDPCA+LDA is also superior to PCA+LDA with respect to recognition accuracy.

Three face databases, ORL, UMIST and FERET, are employed to evaluate BDPCA+LDA. Experimental results show that BDPCA+LDA outperforms PCA+LDA on all three databases. From Table 12.7, we find that the computation complexity of BDPCA+LDA is much less sensitive to the size of training set compared with PCA+LDA. Therefore, when applied to a larger face database, BDPCA+LDA would be very efficient, because much less computation requirements are required in the training phase.

REFERENCES

Baeka, J., & Kimb, M. (2004). Face recognition using partial least squares components. *Pattern Recognition*, *37*, 1303-1306.

Bartlett, M. S., Movellan, J. R., & Sejnowski, T. J. (2002). Face recognition by independent component analysis. *IEEE Transactions on Neural Networks*, *13*(6), 1450-1464.

Belhumeur, P. N., Hespanha, J. P., & Kriegman, D. J. (1997). Eigenfaces vs. fisherfaces: Recognition using class specific linear projection. *IEEE Trans. Pattern Analysis and Machine Intelligence*, *19*(7), 711-720.

Chaudhuri, D., Murthy, C. A., & Chaudhuri, B. B. (1992). A modified metric to compute distance. *Pattern Recognition*, *25*, 667-677.

Chellappa, R., Wilson, C. L., & Sirohey, S. (1995). Human and machine recognition of faces: A survey. *Proceedings of the IEEE*, *83*, 705-740.

Chen, L. F., Mark Liao, H. Y., Ko, M. T., Lin, J. C., & Yu, G. J. (2000). A new LDA-based face recognition system which can solve the small sample size problem. *Pattern Recognition*, *33*, 1713-1726.

Chen, S., Liu, J., & Zhou, Z-H. (2004). Making FLDA applicable to face recognition with one sample per person. *Pattern Recognition*, *37*, 1553-1555.

Chen, S., Zhang, D., & Zhou, Z-H. (2004). Enhanced $(PC)^2A$ for face recognition with one training image per person. *Pattern Recognition Letters*, *25*, 1173-1181.

Chen, S., & Zhu, Y. (2004). Subpattern-based principal component analysis. *Pattern Recognition*, *37*, 1081-1083.

Chien, J. T., & Wu, C. C. (2002). Discriminant waveletfaces and nearest feature classifiers for face recognition. *IEEE Transaction on Pattern Analysis and Machine Intelligence*, *24*(12), 1644-1649.

Deerwester, S., Dumais, S. T., Furnas, G. W., Landauer, T. K., & Harshman, R. (1990). Indexing by latent semantic analysis. *Journal of the American Society of Information Science*, *41*, 391-407.

Draper, B. A., Baek, K., Bartlett, M. S., & Beveridge, J. R. (2003). Recognition faces with PCA and ICA. *Computer Vision and Image Understanding*, *91*, 115-137.

Etemad, K., & Chellappa, R. (1996). Face recognition using discriminant eigenvectors. *Proceedings of the IEEE International Conference on Acoustic, Speech, and Signal Processing* (pp. 2148-2151).

Fukunaga, K. (1990). *Introduction to statistical pattern recognition* (2nd ed.). San Diego, CA: Academic Press.

Garthwaite, P.M. (1994). An interpretation of partial least squares. *Journal of the American Statistical Association, 89*, 122-127.

Golub, G. H., & Van Loan, C. F. (1996). *Matrix computation* (3rd ed.). Baltimore: The Johns Hopkins University Press.

Gottumukkal, R., & Asari, V. K. (2004). An improved face recognition technique based on modular PCA approach. *Pattern Recognition Letters*, *25*, 429-436.

Graham, D. B., & Allinson, N. M. (1998a). Characterizing virtual eigensignatures for general purpose face recognition. In H. Wechsler, P. J. Phillips, V. Bruce, F. Fogelman-Soulie & T. S. Huang (Eds.), *Face recognition: From theory to application* (pp. 446-456). Computer and Systems Sciences.

Graham, D. B., & Allinson, N. M. (1998b). *The UMIST face database.* Retrieved from http://images.ee.umist.ac.uk /danny/database.html

Gupta, H., Agrawal, A. K., Pruthi, T., Shekhar, C., & Chellappa, R. (2002). An experimental evaluation of linear and kernel-based methods for face recognition. In *Proceedings of the 6th IEEE Workshop on Applications of Computer Vision, (WACV'02)* (pp. 13-18).

Habibi, A., & Wintz, P. A. (1971). Image coding by linear transformation and block quantization. *IEEE Transactions on Communication Technology*, *19*(1), 50-62.

Huber, R., Ramoser, H., Mayer, K., Penz, H., & Rubik, M. (2005). Classification of coins using an eigenspace approach. *Pattern Recognition Letters*, *26*, 61-75.

Hyvarinen, A. (2001). *Independent component analysis*. New York: J. Wiley.

Jain, A. K. (1989). *Fundamentals of digital image processing*. Upper Saddle River, NJ: Prentice Hall.

Jing, X., Tang, Y., & Zhang, D. (2005). A Fourier-LDA approach for image recognition. *Pattern Recognition*, *38*, 453-457.

Karhunan, J., & Joutsensalo, J. (1995). Generalization of principal component analysis, optimization problems, and neural networks. *Neural Networks*, *8*(4), 549-562.

Kirby, M., & Sirovich, L. (1990). Application of the KL procedure for the characterization of human faces. *IEEE Transaction on Pattern Analysis and Machine Intelligence*, *12*(1), 103-108.

Li, S., & Lu, J. (1999). Face recognition using the nearest feature line method. *IEEE Transactions on Neural Networks*, *10*(2), 439-443.

Liu, C., & Wechsler, H. (2003). Independent component analysis of Gabor features for face recognition. *IEEE Transaction on Neural Networks*, *14*(4), 919-928.

Liu, W., Wang, Y., Li, S. Z., & Tan, T. (2004). Null space-based kernel Fisher discriminant analysis for face recognition. *Sixth IEEE International Conference on Automatic Face and Gesture Recognition* (pp. 369-375).

Lu, G., Zhang, D., & Wang, K. (2003). Palmprint recognition using eigenpalms features. *Pattern Recognition Letters*, *24*, 1463-1467.

Lu, J., Plataniotis, K. N., & Venetsanopoulos, A. N. (2003a). Face recognition using LDA-based algorithms. *IEEE Transactions on Neural Networks*, *14*(1), 195-200.

Lu, J., Plataniotis, K. N.,& Venetsanopoulos, A. N. (2003b). Face recognition using kernel direct discriminant analysis algorithms. *IEEE Trans. Neural Networks*, *14*(1), 117-126.

Mallat, S. (2002). *A wavelet tour of signal processing* (2nd ed.). New York: Academic Press.

Moon, H., & Phillips, J. (1998). Analysis of PCA-based face recognition algorithms. In K. W. Boyer & P. J. Phillips (Eds.), *Empirical evaluation techniques in computer vision*. Los Alamitos, CA: IEEE Computer Society Press.

Navarrete, P., & Ruiz-del-Solar, J. (n.d.). Eigenspace-based recognition of faces: Comparisons and a new approach. In *Proceedings of the International Conference on Image Analysis and Processing ICIAP 2001* (pp. 42-47).

ORL Face Database. (2002). AT&T Research Laboratories. *The ORL Database of Faces.* Retrieved from www.uk.research.att.com/facedatabase.html

Perlibakas, V. (2004). Distance measures for PCA-based face recognition. *Pattern Recognition Letters*, *25*, 711-724.

Phillips, P. J. (2001). *The Facial Recognition Technology (FERET) database.* Retrieved from www.itl.nist.gov/iad /humanid/feret

Phillips, P. J., Moon, H., Rizvi, S. A., & Rauss, P. J. (2000). The FERET evaluation methodology for face-recognition algorithms. *IEEE Transaction on Pattern Analysis and Machine Intelligence*, *22*(10), 1090-1104.

Pratt, W. K. (2001). *Digital image processing* (3rd ed.). New York: John Wiley & Sons.

Ray, W. D., & Driver, R. M. (1970). Further decomposition of the Karhunen-Loeve series representation of a stationary random process. *IEEE Transactions on Information Theory*, *16*(6), 663-668.

Ryu, Y-S., & Oh, S-Y. (2002). Simple hybrid classifier for face recognition with adaptively generated virtual data. *Pattern Recognition Letters*, *23*, 833-841.

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

Song, F., Yang, J-Y., & Liu, S. (2004). Large margin linear projection and face recognition. *Pattern Recognition*, *37*, 1953-1955.

Swets, D. L., & Weng, J. (1996). Using discriminant eigenfeatures for image retrieval. *IEEE Transaction on Pattern Analysis and Machine Intelligence*, *18*, 8, 831-836.

Torkkola, K. (2001). Linear discriminant analysis in document classification. In *Proceedings of the IEEE ICDM Workshop Text Mining*.

Toygar, O., & Acan, A. (2004). Multiple classifier implementation of a divide-and-conquer approach using appearance-based statistical methods for face recognition. *Pattern Recognition Letters*, *25*, 1421-1430.

Turk, M., & Pentland, A. (1991). Eigenfaces for recognition. *Journal of Cognitive Neuroscience*, *3*(1), 71-86.

Wang, H., & Zhang, L. (2004). Linear generalization probe samples for face recognition. *Pattern Recognition Letters*, *25*, 829-840.

Wu, J., & Zhou, Z-H. (2002). Face recognition with one training image per person. *Pattern Recognition Lett*ers, *23*, 1711-1719.

Wu, X., Zhang, D., & Wang, K. (2003). fisherpalms based palmprint recognition. *Pattern Recognition Letters*, *24*, 2829-2838.

Yambor, W. S., Draper, B. S., & Beveridge, J. R. (2002). Analyzing PCA-based face recognition algorithm: Eigenvector selection and distance measures. In H. Christensen & J. Phillips (Eds.), *Empirical evaluation methods in computer vision*. Singapore: World Scientific Press.

Yang, J., & Yang, J-Y. (2002). From image vector to matrix: A straightforward image projection technique – IMPCA vs. PCA. *Pattern Recognition*, *35*, 1997-1999.

Yang, J., & Yang, J-Y. (2003). Why can LDA be performed in PCA transformed space? *Pattern Recognition*, *36*, 563-566.

Yang, J., Yang, J.Y., & Frangi, A.F. (2003). Combined fisherfaces framework. *Image and Vision Computing*, *21*, 1037-1044.

Yang, J., Zhang, D., Frangi, A.F., & Yang, J-Y. (2004). Two-dimensional PCA: A new approach to appearance-based face representation and recognition. *IEEE Transaction on Pattern Analysis and Machine Intelligence*, *26*(1), 131-137.

Yu, H., & Yang, J. (2001). A direct LDA algorithm for high-dimensional data with application to face recognition. *Pattern Recognition*, *34*, 2067-2070.

Yuen, P. C., & Lai, J. H. (2002). Face representation using independent component analysis. *Pattern Recognition*, *35*, 1247-1257.

Zhang, D. (2004). *PolyU palmprint database.* Biometrics Research Centre, Hong Kong Polytechnic University. Retrieved from http://www.comp.polyu.edu.hk/~biometrics

Zhang, D., Kong, W., You, J., & Wong, M. (2003). On-line palmprint identification. *IEEE Transaction on Pattern Analysis and Machine Intelligence*, *25*(9), 1041-1050.

Zhang, J., Li, S. Z., & Wang, J. (2004). Nearest manifold approach for face recognition. *Proceedings of the 6th IEEE International Conference on Automatic Face and Gesture Recognition (FGR'04)* (pp. 223-228).

Zhao, W., Chellappa, R., & Phillips, P. J. (1999). *Subspace linear discriminant analysis for face recognition* (Technical report CAR-TR-914). College Park: Center for Automation Research, University of Maryland.

Zhao, W., Chellappa, R., Phillips, P. J., & Rosenfeld, A. (2003). Face recognition: A literature survey. *ACM Computing Surveys*, *35*(4), 399-458.

Zheng, W., Zhao, L., & Zou, C. (2004). Locally nearest neighbor classifiers for pattern classification. *Patte*nt algorithm to solve the small sample size problem for LDA. *Pattern Recognition*, *37*, 1077-1079.

Chapter XIII

Feature Fusion Using Complex Discriminator

ABSTRACT

This chapter describes feature fusion techniques using complex discriminator. After the introduction, we first introduce serial and parallel feature fusion strategies. Then, the complex linear projection analysis methods, complex PCA and complex LDA, are developed. Next, some feature preprocessing techniques are given. The symmetry property of parallel feature fusion is analyzed and revealed. Then, the proposed methods are applied to biometric applications, related experiments are performed and the detailed comparison analysis is exhibited. Finally, a summary is given.

INTRODUCTION

In recent years, data fusion has been developed rapidly and applied widely in many areas, such as object tracking and recognition (Chiang, Moses, & Potter, 2001; Peli, Young, Knox, et al., 1999), pattern analysis and classification (Doi, Shintani, Hayashi, et al., 1995; Gunatilaka & Baertlein, 2001; Young & Fu, 1986), image processing and understanding (Ulug & McCullough, 1999; Chang & Park, 2001), and so forth. In this chapter, we pay most attention to the data fusion techniques used for pattern classification problems.

In practical classification applications, if the number of classes and multiple feature sets of pattern samples are given, how to achieve a desirable recognition performance based on these sets of features is a very interesting problem. Generally speaking, there

exist three popular schemes. In the first one, the information derived from multiple feature sets is assimilated and integrated into a final decision directly. This technique is generally referred to as *centralized data fusion* (Peli, Young, Knox, et al., 1999) or *information fusion* (Dassigi, Mann, & Protopoescu, 2001) and is widely adopted in many pattern recognition systems (Li, Deklerck, Cuyper, et al., 1995). In the second, the individual decisions are made first based on different feature sets, and then they are reconciled or combined into a global decision. The technique is generally known as *distributed data fusion* or *decision fusion* (Peli, Young, Knox, et al., 1999). In the third scheme, the given multiple feature sets are used to produce new fused feature sets, which are more helpful to the final classification (Ulug & McCullough, 1999). The technique is usually termed *feature fusion*.

As a matter of fact, feature fusion and decision fusion are two levels of data fusion. In some cases, they are involved in the same application system (Gunatilaka & Baertlein, 2001; Jimenez, 1999). But, in recent years, decision level fusion, represented by multi-classifier or multi-expert combination strategies, has been of major concern (Huang & Suen, 1995; Constantinidis, Fairhurst, & Rahman, 2001). In contrast, feature level fusion has probably not received the amount of attention it deserves. However, feature level fusion plays a very important role in the process of data fusion. The advantage of feature level fusion lies in two aspects: First, it can derive the most discriminatory information from original multiple feature sets involved in fusion; Second, it enables eliminating redundant information resulting from the correlation between distinct feature sets and making the subsequent decision in real time possible. In a word, feature fusion is capable of deriving and gaining the most effective and least-dimensional feature vector sets that benefit the final decision.

In general, the existing feature fusion techniques for pattern classification can be subdivided into two basic categories. One is feature selection-based, and the other is feature extraction-based. In the former, all feature sets are first grouped together and then a suitable method is used to select most discriminative features from them. Zhang presented a fused method based on dynamic programming (Zhang, 1998); Battiti gave a method using supervised neural network; and recently, Battiti (1994), Shi and Zhang provided a method based on support vector machines (SVM) (Shi & Zhang, 1996). In the latter, the multiple feature sets are combined into one set of feature vectors that are input into a feature extractor for fusion (Liu & Wechsler, 2000). The classical feature combination method is to group two sets of feature vectors into one union-vector (or super-vector). Recently, a new feature combination strategy; that is, combining two sets of feature vectors into one complex vector, was developed (Yang & Yang, 2002; Yang, Yang, Zhang, & Lu, 2003; Yang, Yang, & Frangi, 2003). The feature fusion method based on union-vector is referred to as *serial feature fusion,* and that based on complex vector is called *parallel feature fusion*.

In this chapter, our focus is on feature level fusion. The distinction of *feature combination* and *feature fusion* is specified, and the notions of *serial feature fusion* and *parallel feature fusion* are given. The basic idea of parallel feature fusion is: The given two sets of original feature vectors are first used to form a complex feature vector space, and then traditional linear projection methods, such as PCA (see Chapter II) and LDA (see Chapter III) are generalized for feature extraction in such a space. The proposed parallel feature fusion techniques are applied to face recognition. The experimental

results on the AR face database and FERET database indicate that the classification accuracy is increased after parallel feature fusion, and the proposed parallel fusion strategy outperforms the classical serial feature fusion strategy.

SERIAL AND PARALLEL FEATURE FUSION STRATEGIES

Suppose A and B are two feature spaces defined on pattern sample space Ω. For an arbitrary sample $\xi \in \Omega$, the corresponding two feature vectors are $\alpha \in A$ and $\beta \in B$. The serial combined feature of ξ is defined by $\gamma = \begin{pmatrix} \alpha \\ \beta \end{pmatrix}$. Obviously, if feature vector α is n-dimensional and β is m-dimensional, then the serial combined feature γ is $(n + m)$-dimensional. All serial combined feature vectors of pattern samples form a $(n + m)$-dimensional serial combined feature space.

Here, we intend to combine two feature vectors by a complex vector rather than a super-vector. In more details, let α and β be two different feature vectors of a same sample ξ, the complex vector $\gamma = \alpha + i\beta$ (i is imaginary unit) is used to represent the combination of α and β and is named parallel combined feature of ξ. Note that if the dimension of α and β is not equal, pad the lower-dimensional one with zeros until its dimension is equal to the other ones' before combination. For example, $\alpha = (a_1, a_2, a_3)^T$ $\beta = (b_1, b_2)^T$, β is first turned into $(b_1, b_2, 0)^T$ and the resulting combination form is denoted by $\gamma = (a_1 + ib_1, a_2 + ib_2, a_3 + i0)^T$.

Let us define the parallel combined feature space on Ω as $C = \{\alpha + i\beta \mid \alpha \in A, \beta \in B\}$. Obviously, it is an n-dimensional complex vector space, where $n = \max\{\dim A, \dim B\}$. In the space, the inner product is defined by:

$$(X, Y) = X^H Y \tag{13.1}$$

where $X, Y \in C$, and H is the denotation of conjugate transpose.

The complex vector space defined the above inner product is usually called unitary space. In unitary space, the measurement (norm) can be introduced as follows:

$$\| Z \| = \sqrt{Z^H Z} = \sqrt{\sum_{j=1}^{n} (a_j^2 + b_j^2)} \tag{13.2}$$

where $Z = (a_1 + ib_1, \ldots, a_n + ib_n)^T$.

Correspondingly, the distance (called unitary distance) between the complex vectors Z_1 and Z_2 is defined by:

$$\| Z_1 - Z_2 \| = \sqrt{(Z_1 - Z_2)^H (Z_1 - Z_2)} \tag{13.3}$$

If samples are directly classified based on the serial combined feature $\gamma = \begin{pmatrix} \alpha \\ \beta \end{pmatrix}$ or the parallel combined feature $\gamma = \alpha + i\beta$, and Euclidean distance is adopted in the serial combined feature space while unitary distance is used in parallel combined feature space, then it is obvious that the two kinds of combined features are equivalent in essence; that is, they will result in the same classification results. However, the combined feature vectors are always high dimensional and contain much redundant information and some conflicting information, which is unfavorable to recognition. Consequently, in general, we would rather perform the classification after the process of *feature fusion* than after the process of *feature combination*.

In our opinion, feature fusion includes feature combination but more than it. That is to say, the fusion is a process of reprocessing the combined features; that is, after dimension reduction or feature extraction, the favorable discriminatory information remains and, at the same time, the unfavorable redundant or conflicting information is eliminated. According to this opinion, there are two strategies of feature fusion based on two methods of feature combination:

1. **Serial feature fusion**. Serial feature fusion is a process of feature extraction based on the serial feature combination method, and the resulting feature is called serial fused feature.
2. **Parallel feature fusion**. Parallel feature fusion is a process of feature extraction based on the parallel feature combination method, and the resulting feature is called parallel fused feature.

As we know, the parallel combined feature vector is a complex vector. Now, a problem is: How do we perform the feature extraction in complex feature space? In the following section, we will discuss the linear feature extraction techniques in complex feature space.

COMPLEX LINEAR PROJECTION ANALYSIS

Fundamentals

In the unitary space, the between-class scatter (covariance) matrix, within-class scatter matrix and total scatter matrix, respectively, are defined in another representative form as follows:

$$\mathbf{S}_b = \sum_{i=1}^{L} P(\omega_i)(m_i - m_0)(m_i - m_0)^H \tag{13.4}$$

$$\mathbf{S}_w = \sum_{i=1}^{L} P(\omega_i) E\left\{(X - m_i)(X - m_i)^H / \omega_i\right\} \tag{13.5}$$

$$\mathbf{S}_t = \mathbf{S}_b + \mathbf{S}_w = E\left\{(X - m_0)(X - m_0)^H\right\} \tag{13.6}$$

where L denotes the number of pattern class; $P(\omega_i)$ is the prior probability of class i; $m_i = E\{X/\omega_i\}$ is the mean vector of class i; $m_0 = E\{X\} = \sum_{i=1}^{m} P(\omega_i) m_i$ is the mean of all training samples.

From Equation 13.4 to 13.6, it is obvious that $\mathbf{S}_w$, $\mathbf{S}_b$ and $\mathbf{S}_t$ are semi-positive definite Hermite matrices. What is more, $\mathbf{S}_w$ and $\mathbf{S}_t$ are all positive definite matrix when $\mathbf{S}_w$ is nonsingular. In this chapter, we assume $\mathbf{S}_w$ is nonsingular.

Lemma 13.1 (Ding & Cai, 1995). Each eigenvalue of Hermite matrix is a real number.

Since $\mathbf{S}_w$, $\mathbf{S}_b$ and $\mathbf{S}_t$ are semi-positive definite Hermite matrices, it is easy to get the following conclusion:

Corollary 13.1. The eigenvalues of $\mathbf{S}_w$, $\mathbf{S}_b$ or $\mathbf{S}_t$ are all non-negative real numbers.

Complex PCA

Taking $\mathbf{S}_t$ as generation matrix, we present the PCA technique in complex feature space. Suppose that the orthogonal eigenvectors of $\mathbf{S}_t$ are $\xi_1, \ldots, \xi_n$, and the associated eigenvalues are $\lambda_1, \ldots, \lambda_n$, which satisfy $\lambda_1 \geq \ldots \geq \lambda_n$. Choosing the first d-maximal eigenvectors $\xi_1, \ldots, \xi_d$ as projection axes, thus, the complex PCA transform can be defined: by:

$$Y = \Phi^H X, \text{ where } \Phi = (\xi_1, \ldots, \xi_d) \tag{13.7}$$

We also call it complex principle component analysis (CPCA).

In fact, classical PCA is only a special case of the CPCA. That is to say, the theory of PCA developed in complex feature space is more significant, and it undoubtedly suits the case in real feature space.

Complex LDA

In unitary space, the Fisher discriminant criterion function can be defined by:

$$J_f(\varphi) = \frac{\varphi^H \mathbf{S}_b \varphi}{\varphi^H \mathbf{S}_w \varphi} \tag{13.8}$$

where j is an n-dimensional nonzero vector.

Since $\mathbf{S}_w$ and $\mathbf{S}_b$ are semi-positive definite, for any arbitrary j, we have $\varphi^H \mathbf{S}_b \varphi \geq 0$ and $\varphi^H \mathbf{S}_w \varphi \geq 0$. Hence, the values of $J_f(\varphi)$ are all non-negative real numbers. It means that the physical meanings of Fisher criterion defined in unitary space is same as that defined in Euclidian space.

If $\mathbf{S}_w$ is nonsingular, Fisher criterion is equivalent to the following function:

$$J(\varphi) = \frac{\varphi^H \mathbf{S}_b \varphi}{\varphi^H \mathbf{S}_t \varphi} \tag{13.9}$$

For convenience, in this chapter, we use the above criterion function instead of Fisher criterion defined in Equation 13.8.

Recently, Jin and Yang suggested the theory on the uncorrelated LDA (ULDA), and used it to solve face recognition and handwritten digit recognition problems successfully (Jin, Yang, Hu, & Lou, 2001; Jin, Yang, Tang, & Hu, 2001). The most outstanding advantage of ULDA is that it can eliminate the statistical correlation between the components of pattern vector. In this chapter, we try to further extend Jin's theory and enable it to suit for feature extraction in the combined complex feature space (unitary space). Now, we describe the uncorrelated discriminant analysis in unitary space in details.

The uncorrelated discriminant analysis aims to find a set of vectors $\varphi_1, \ldots, \varphi_d$, which maximizes the criterion function $J(\varphi)$ under the following conjugate orthogonality constraints:

$$\varphi_j^H \mathbf{S}_t \varphi_i = \delta_{ij} = \begin{cases} 1 & i = j \\ 0 & i \neq j \end{cases} \quad i, j = 1, \ldots, d \tag{13.10}$$

More formally, the uncorrelated discriminant vectors $\varphi_1, \ldots, \varphi_d$ can be chosen in this way: φ_1 is selected as Fisher optimal projective direction. And after determining the first k discriminant vectors $\varphi_1, \ldots, \varphi_k$, the $(k+1)$th discriminant vector φ_{k+1} is the optimal solution of the following optimization problem:

$$\text{Model 1} \begin{cases} \max\left(J(\varphi)\right) \\ \varphi_j^H \mathbf{S}_t \varphi = 0, j = 1, \ldots, k \\ \varphi \in C^n \end{cases} \tag{13.11}$$

where C^n denotes n-dimensional unitary space.

Now, we discuss how to find the optimal discriminant vectors. Since $\mathbf{S}_b$, $\mathbf{S}_t$ are Hermite matrices and $\mathbf{S}_t$ is positive definite, according to the theory in document, it is easy to draw the following conclusion (Ding & Cai, 1995):

Theorem 13.1. Suppose that $\mathbf{S}_w$ *is nonsingular, there exist* n *eigenvectors* $X_1, \ldots, X_n$ *corresponding to eigenvalues* $\lambda_1, \ldots, \lambda_n$ *of the eigenequation* $\mathbf{S}_b X = \lambda \mathbf{S}_t X$, *such that:*

$$X_i^H \mathbf{S}_t X_j = \begin{cases} 1 & i = j \\ 0 & i \neq j \end{cases} \quad i, j = 1, \ldots, n \tag{13.12}$$

$$\text{and } X_i^H \mathbf{S}_b X_j = \begin{cases} \lambda_i & i = j \\ 0 & i \neq j \end{cases} \quad i, j = 1, \ldots, n \tag{13.13}$$

The vectors $X_1, \ldots, X_n$ *satisfying the constraint 13.12 are called* $\mathbf{S}_t$*-orthogonal. Since* $\mathbf{S}_b$ *is semi-positive definite, by Theorem 1, it is not hard to get the following corollaries.*

Corollary 13.2. The generalized eigenvalues $\lambda_1, \ldots, \lambda_n$ of $\mathbf{S}_b X = \lambda \mathbf{S}_t X$ are all non-negative real numbers, in which there exist q positive ones, where q = **rank** ($\mathbf{S}_b$).

Corollary 13.3. The values of criterion function $J(X_j) = \lambda_j, j = 1, \ldots, n$.

Corollary 13.4. The eigenvectors $X_1, \ldots, X_n$ of $\mathbf{S}_b X = \lambda \mathbf{S}_t X$ are linearly independent, and C^n = **span** $\{X_1, \ldots, X_n\}$.

Without loss generality, suppose that the eigenvalues of $\mathbf{S}_b X = \lambda \mathbf{S}_t X$ satisfy $\lambda_1 \geq \ldots \geq \lambda_n$.

Reviewing **Proposition 11.1**, it indicates that in the unitary space, the uncorrelated optimal discriminant vectors can be selected as $X_1, \ldots, X_d$, which are the $\mathbf{S}_t$-orthogonal eigenvectors corresponding to the first d maximal eigenvalues of $\mathbf{S}_b X = \lambda \mathbf{S}_t X$. By Corollary 13.1 and physical meanings of Fisher criterion, the total number of effective discriminant vectors is at most q = **rank** ($\mathbf{S}_b$) $\leq L - 1$, where L is pattern class number.

The detailed algorithm for the calculation of $X_1, \ldots, X_d$ is described as follows:

First get the prewhitening transformation matrix W, such that $W^H \mathbf{S}_t W = I$. In fact, $W = U\Lambda^{-\frac{1}{2}}$, where U = $(\xi_1, \ldots, \xi_n)$, Λ = **diag** $(a_1, \ldots, a_n)$, $\xi_1, \ldots, \xi_n$ are eigenvectors of $\mathbf{S}_t$, and $a_1, \ldots, a_n$ are associated eigenvalues.

Next, let $\tilde{\mathbf{S}}_b = W^H \mathbf{S}_b W$, and calculate its orthonormal eigenvectors $\xi_1, \ldots, \xi_n$. Suppose the associated eigenvalues satisfy $\lambda_1 \geq \ldots \geq \lambda_n$, then, the optimal projection axes are:

$$X_1 = W\xi_1, \ldots, X_d = W\xi_d$$

In unitary space, since the uncorrelated optimal discriminant vectors $X_1, \ldots, X_d$ ($d \leq q$) satisfy constraints 13.10 and 13.12, and λ_j is a positive real number, the physical meanings of the fused feature being projected onto X_j is specific; that is, the between-class scatter is λ_j and the total scatter is 1.

In unitary space, the uncorrelated discriminant vectors $X_1, \ldots, X_d$ form the following transformation:

$$Y = \Phi^H X, \text{ where } \Phi = (X_1, \ldots, X_d) \tag{13.14}$$

which is used for feature extraction in parallel combined feature space.

As a comparison, the method addressed in document is only a special case of complex LDA (Jin, Yang, Hu, & Lou, 2001a; Jin, Yang, Tang, & Hu, 2001b). That is to say, the theory of the uncorrelated discriminant analysis developed in complex vector space is more significant, and it undoubtedly suits the case in real vector space.

FEATURE PREPROCESSING TECHNIQUES

The difference of feature extraction or measurement might lead to the numerical unbalance between the two features α and β of a same pattern. For instance, given two feature vectors $\alpha = (10, 11, 9)^T$ and $\beta = (0.1, 0.9)^T$ corresponding to one sample, assume

they are combined as $\gamma = \alpha + i\beta$ or $\gamma = \begin{pmatrix} \alpha \\ \beta \end{pmatrix}$, which implies that the feature α play a more important role than β in the process of fusion. Consequently, to counteract the numerical unbalance between features and gain satisfactory fusion performance, our suggestion is to initialize the original feature α and β, respectively, before the combination.

What's more, when the dimensions of α and β are unequal, after being initialized, we think the higher-dimensional one is still more powerful than the lower-dimensional one. The reason is that in the course of linear feature extraction after combination, the higher-dimensional one plays a more important role in the scatter matrices. So, to eliminate the unfavorable effect resulting from unequal dimension, our suggestion is to adopt the weighted combination form. The serial combination is formed by $\gamma = \begin{pmatrix} \alpha \\ \theta\beta \end{pmatrix}$ or $\gamma = \begin{pmatrix} \theta\alpha \\ \beta \end{pmatrix}$, while the parallel combination is formed by $\gamma = \alpha + i\theta\beta$ or $\gamma = \theta\alpha + i\beta$, where the weight θ is called combination coefficient.

It is easy to prove that the weighted combined feature satisfies the following properties:

Property 13.1. If $\theta \neq 0$, the parallel combined feature $\gamma = \alpha + i\theta\beta$ is equivalent to $\gamma = (1/\theta)\,\alpha + i\beta$, and the serial combined feature $\gamma = \begin{pmatrix} \alpha \\ \theta\beta \end{pmatrix}$ is equivalent to $\gamma = \begin{pmatrix} \frac{1}{\theta}\alpha \\ \beta \end{pmatrix}$.

Property 13.2. While $\theta \to 0$, the fused feature $\gamma = \alpha + i\theta\beta$ is equivalent to the single feature α. While $\theta \to \infty$ ($\theta \neq \infty$), $\gamma = \alpha + i\theta\beta$ is equivalent to the single feature β.

Two feature-preprocessing methods are introduced respectively as follows.

Preprocessing Method I

- **Step 1.** Let $\bar{\alpha} = \frac{\alpha}{\|\alpha\|}$, $\bar{\beta} = \frac{\beta}{\|\beta\|}$; that is, turn *a* and *b* into unit vectors, respectively.
- **Step 2.** Suppose the dimension of *a* and *b* is *n* and *m*, respectively, if *n* = *m*, let *q* =1; Otherwise, suppose *n* > *m*, let $\theta = \frac{n^2}{m^2}$, and the parallel combination form is $\gamma = \bar{\alpha} + i\theta\bar{\beta}$ while the serial combination form is $\gamma = \begin{pmatrix} \bar{\alpha} \\ \theta\bar{\beta} \end{pmatrix}$.

In Step 2, the evaluation of the combination coefficient $\theta = \frac{n^2}{m^2}$ is attributed to the following reason. When the length of two feature vectors is unequal, since the size of scatter matrix generated by feature vector α is $n \times n$, and the size of scatter matrix generated

by β is $m \times m$, so the combination coefficient θ is considered to be the square of n/m.

Preprocessing Method II

Firstly, respectively initialize α and β by:

$$Y = \frac{X - \mu}{\sigma} \tag{13.15}$$

where X denotes sample feature vector, μ denotes the mean vector of training samples; $\sigma = \frac{1}{n}\sum_{j=1}^{n}\sigma_j$, where n is the dimension of X, and σ_j is the standard deviation of the jth feature component of the training samples.

However, if the above feature initialization method is employed, it is more difficult to evaluate the combination coefficient θ when the dimension of α and β are unequal. Here, we give the selection scope by experience. Suppose the combination form of α and β is denoted by $\gamma = \alpha + i\theta\beta$ or $\gamma = \begin{pmatrix} \alpha \\ \theta\beta \end{pmatrix}$, the dimension of α and β is n and m respectively and $n > m$, let $\delta = \frac{n}{m}$, then, θ can be selected between δ and δ^2.

How to determine a proper combination coefficient θ for the optimal fusion performance is still a problem that deserves to study on.

SYMMETRY PROPERTY OF PARALLEL FEATURE FUSION

For parallel feature combination, two kinds of feature vectors α and β of a sample can be formed by $\alpha + i\theta\beta$ or $\beta + i\alpha$. But, based on the two combination forms $\alpha + i\beta$ and $\beta + i\alpha$, after feature extraction via complex linear projection analysis, there is still a question: Are the final classification results identical with a same classifier? If the results are identical, we think that the parallel feature fusion satisfies a property of symmetry. That is to say, the result of parallel fusion is independent to the sequence of feature in combination. That is expected. Otherwise, if parallel feature fusion does not satisfy this property — that is, a different sequence of combined features induces different classification result — it makes the problem more complicated. Fortunately, we can prove theoretically that parallel feature fusion satisfies a desirable property of symmetry. Now, taking CPCA (one of the feature extraction techniques mentioned above) as an example, we give the proof of symmetry in parallel feature fusion.

Suppose two feature spaces defined on pattern sample space Ω are respectively defined by $C_1 = \{\alpha + i\beta \mid \alpha \in A, \beta \in B\}$, $C_2 = \{\beta + i\alpha \mid \alpha \in A, \beta \in B\}$.

Lemma 13.2. In unitary space, construct two matrices $H(\alpha, \beta) = (\alpha + i\beta)(\alpha + i\beta)^H$ and $H(\alpha, \beta) = (\beta + i\alpha)(\beta + i\alpha)^H$, then $H(\alpha, \beta) = \overline{H(\alpha, \beta)}$, where α and β are n-dimensional real vectors.

Proof: Suppose $\alpha = (\alpha_1, \ldots, \alpha_n)^T$, $\beta = (b_1, \ldots, b_n)^T$, then:

$$[H(\alpha, \beta)]_{kl} = (a_k + ib_k)\overline{(a_l + ib_l)} = (a_k a_l + b_k b_l) + i(a_l b_k - a_k b_l)$$

$$[H(\beta, \alpha)]_{kl} = (b_k + ia_k)\overline{(b_l + ia_l)} = (a_k a_l + b_k b_l) - i(a_l b_k - a_k b_l)$$

So $[H(\beta, \alpha)]_{kl} = \overline{[H(\alpha, \beta)]_{kl}}$

Hence $H(\beta, \alpha) = \overline{H(\alpha, \beta)}$.

Suppose that the within-class, between-class and total scatter matrices in combined feature space C_i (i = 1, 2) are defined by $\mathbf{S}_w^i$, $\mathbf{S}_b^i$ and $\mathbf{S}_t^i$ (i = 1, 2), respectively; by Lemma 13.2, it is easy to prove that they satisfy the following properties.

Property 13.3. $\mathbf{S}_w^2 = \overline{\mathbf{S}_w^1}$; $\mathbf{S}_b^2 = \overline{\mathbf{S}_b^1}$; $\mathbf{S}_t^2 = \overline{\mathbf{S}_t^1}$.

Property 13.4. Suppose that ξ is the eigenvector of $\mathbf{S}_t^1$ ($\mathbf{S}_w^1$ or $\mathbf{S}_b^1$) corresponding to the eigenvalue λ, then $\overline{\xi}$ is the eigenvector of $\mathbf{S}_t^2$($\mathbf{S}_w^2$ or $\mathbf{S}_b^2$) corresponding to a same eigenvalue λ.

Proof: $\mathbf{S}_t^1\xi = \lambda\xi \Rightarrow \overline{\mathbf{S}_t^1\xi} = \overline{\lambda\xi} \Rightarrow \overline{\mathbf{S}_t^1}\,\overline{\xi} = \overline{\lambda}\,\overline{\xi}$

By Property 13.3 and Corollary 13.2, we have $\mathbf{S}_t^2 = \overline{\mathbf{S}_t^1}$ and $\overline{\lambda} = \lambda$,
Hence $\mathbf{S}_t^2\overline{\xi} = \lambda\overline{\xi}$.

By Property 13.2, we can draw the following conclusion. If $\xi_1, \ldots, \xi_d$ are projection axes of CPCA in combined feature space C_1, then, $\overline{\xi}_1, \ldots, \overline{\xi}_d$ are projection axes of CPCA in combined feature space C_2.

Lemma 13.3. In combined feature space C_1, if the projection of vector $x + iy$ onto the axe ξ is $p + iq$, then, in combined feature space C_2, the projection of sample $y + ix$ onto the axe $\overline{\xi}$ is $q + ip$.

Proof: Denote $\xi = (a_1 + ib_1, \ldots, a_n + ib_n)^T$, then:

$$\xi^H (x - iy) = (a_1 - ib_1, \ldots, a_n - ib_n)(x_1 + iy_1, \ldots, x_n + iy_n)^T$$

$$= \sum_{j=1}^{n} (a_j x_j + b_j y_j) + i \sum_{j=1}^{n} (a_j y_j - b_j x_j)$$

$$= p + iq$$

$$\overline{\xi}^H (y + ix) = (a_1 + ib_1, \ldots, a_n + ib_n)(y_1 + ix_1, \ldots, y_n + ix_n)^T$$

$$= \sum_{j=1}^{n} (a_j y_j - b_j x_j) + i \sum_{j=1}^{n} (a_j x_j + b_j y_j)$$

$$= q + ip$$

So the proposition holds.

According to Lemma 13.3, it is not difficult to draw the conclusion:

Property 13.5. In combined feature space C_1, $x + iy \xrightarrow{\text{CPCA}} u + iv$
In combined feature space C_2, $y + ix \xrightarrow{\text{CPCA}} v + iu$
In unitary space, by the norm defined in Equation 13.2, we have $\| u + iv \| = \| v + iu \|$.

By the unitary distance defined in Equation 13.16, if $Z_1^1 = u_1 + iv_1$, $Z_2^1 = u_2 + iv_2$; $Z_1^2 = v_1 + iu_1$, $Z_2^2 = v_2 + iu_2$, then $\| Z_1^1 - Z_2^1 \| = \| Z_1^2 - Z_2^2 \|$. That is to say, the distance between two complex vectors depends on the values of their real part and the imaginary part but is independent to their sequence.

In summary, we draw the conclusion:

Theorem 13.3. In unitary space, the parallel feature fusion based on CPCA has a property of symmetry.

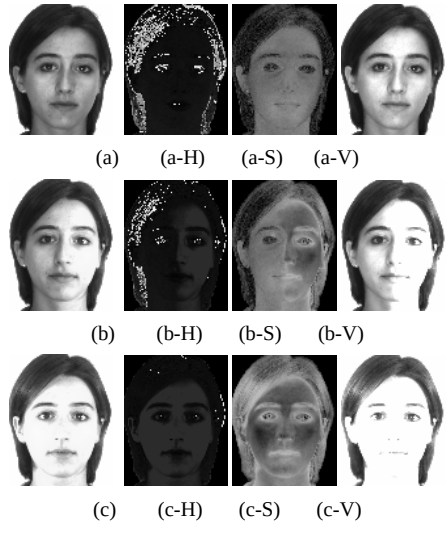
Similarly, we can prove that the parallel feature fusion based on complex LDA satisfies the property of symmetry as well. That is to say, based on the parallel feature combination form $\alpha + i\beta$ or $\beta + i\alpha$, after feature extraction via linear projection analysis, the classification results are identical. In other words, the fusion result is independent to parallel feature combination form.

BIOMETRIC APPLICATIONS

Complex PCA-Based Color Image Representation

By far, numerous techniques have been developed for face representation and recognition. But, almost all of these methods are based on grayscale (intensity) face

Figure 13.1. Three images under different illumination conditions and their corresponding hue (H), saturation (S) and value (V) component images



images. Even if the color images are available, the usual way is to convert them into grayscale images and base on them to recognize. In the process of image conversion, some useful discriminatory information contained in the face color itself may be lost. If we characterize a color image using color model, such as HSV (or HSI), there are three basic color attributes — hue, saturation and intensity (value). Converting color images into grayscale ones means that the intensity component is merely employed while the two other components are discarded. Is there some discriminatory information in hue and saturation components? If so, how do we make use of this discriminatory information for recognition? And, as we know, the intensity component is sensitive to illumination conditions, which leads to the difficulty of recognition based on grayscale images. Now, another issue is: Can we combine the color components of image effectively to reduce the disadvantageous effect resulting from different illumination conditions as far as possible? In this section, we try to answer these questions.

Since it is generally considered that the HSV model is more similar to the human perception of color, this color model is adopted in this chapter. The common RGB model can be converted into HSV by the formulations provided in Wang and Yuan (2001). Figure 13.1 shows the three HSV components — hue, saturation and (intensity) value — corresponding to image (a), (b) and (c), respectively. From Figure 13.1, it is easy to see that the illumination conditions of image (a), (b) and (c) are different and the component hue is most sensitive to lighting variation. So, we decide to use the saturation and value components to represent face. These two components can be combined by a complex matrix:

$$\text{Complex-matrix} = \mu_1 S + i\,\mu_2 V \tag{13.16}$$

where i is imaginary unit, μ_1 and μ_2 are called combination parameters.

Note that the parameters μ_1 and μ_2 are introduced to reduce the effect of illumination variations. Here, we select $\mu_1 = 1/m_1$, $\mu_2 = 1/m_2$, where m_1 is the mean of all elements of component S, and m_2 is the mean of all elements of component V.

Then, we use the complex PCA technique for feature extraction. Since n-dimensional image vectors will result in an $n \times n$ covariance matrix $\mathbf{S}_t$, if the dimension of image vector is very high, it is very difficult to calculate $\mathbf{S}_t$'s eigenvectors directly. As we know, in face recognition problems, the total number of training samples m is always much smaller than the dimension of image vector n, so, for computational efficiency, we suggest to use the following technique to get the $\mathbf{S}_t$'s eigenvectors.

Let $Y = (X_1 - \bar{X}, \ldots, X_m - \bar{X})$, $Y \in R^{n \times m}$, then $\mathbf{S}_t$ can also be denoted by $\mathbf{S}_t = \frac{1}{M} YY^H$.

Form matrix $R = Y^H Y$, which is a $m \times m$ non-negative definite Hermite matrix. Since R's size is much smaller than that of $\mathbf{S}_t$, it is much easier to get its eigenvectors. If we work out R's orthonormal eigenvectors $v_1, v_2, \ldots, v_m$, and suppose the associated eigenvalues satisfy $\lambda_1 \geq \lambda_2 \geq \ldots \geq \lambda_m$, then it is easy to prove that the orthonormal eigenvectors of $\mathbf{S}_t$ corresponding to non-zero eigenvalues $\lambda_1, \lambda_2, \ldots, \lambda_m$ are:

$$\xi_i = \frac{1}{\sqrt{\lambda_i}} Y v_i \ , i = 1, \ldots, r \ (r \leq m - 1) \tag{13.17}$$

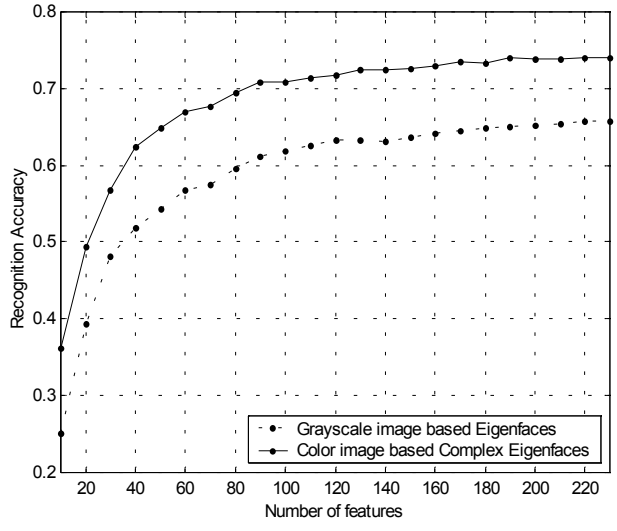
This complex PCA-based face recognition technique can be called complex eigenfaces.

Finally, we test our idea using AR face database, which was created by Aleix Martinez and Robert Benavente in CVC at the U.A.B (Martinex & Benavente, 1998). This database contains more than 4,000 color images corresponding to 126 people's faces (70 men and 56 women). Images feature frontal view faces with different facial expressions, illumination conditions and occlusions (sun glasses and scarf). The pictures were taken at the CVC under strictly controlled conditions. No restrictions on wear (clothes, glasses, etc.), make-up, hair style and so forth were imposed on participants. Each person participated in two sessions, separated by two weeks' (14 days) time. The same pictures were taken in both sessions. Each section contains 13 color images. Some examples are shown in Web page http://rvl1.ecn.purdue.edu/~aleix/aleix_face_DB.html.

Figure 13.2. The training and testing samples of the first man in the database, where (1-1) and (1-14) are training samples; the remaining are testing samples

1-1 1-5 1-6 1-7 1-14 1-18 1-19 1-20

Figure 13.3. Comparison of the proposed color image based complex eigenfaces and the traditional grayscale image based eigenfaces under a nearest-neighbor classifier



In this experiment, 120 different individuals (65 men and 55 women) are randomly selected from this database. We manually cut the face portion from the original image and resize it to be 50×40 pixels. Since the main objective of this experiment is to compare the robustness of face representation approaches in variable illumination conditions, we use the first image of each session (Nos. 1 and 14) for training, and the other images (Nos. 5, 6, 7 and Nos. 18, 19, 20), which are taken under different illumination conditions and without occlusions, are used for testing. The training and testing samples of the first man in the database are shown in Figure 13.2.

The images are first converted from RGB space to HSV space. Then, the saturation and value components of each image are combined by Equation 13.1 to represent face. In the resulting complex image vector space, the developed complex eigenfaces technique is used for feature extraction. In the final feature space, a nearest-neighbor classifier is employed. When the number of selected features varies from 10 to 230 with an interval of 10, the corresponding recognition accuracy is illustrated in Figure 13.3.

For comparison, another experiment is performed using the common method. The color images are first converted to gray-level ones by adding the three color channels; that is, $I = \frac{1}{3}(R + G + B)$. Then, based on these grayscale images, the classical eigenfaces technique is used for feature extraction and a nearest-neighbor classifier is employed for classification (Turk & Pentland, 1991). The recognition accuracy is also illustrated in Figure 13.3.

From Figure 13.3, it is obvious that the proposed color image-based complex eigenfaces is superior to the traditional grayscale image-based eigenfaces. The top recognition accuracy of the complex eigenfaces reaches 74.0%, which is an increase of

8.3% compared to the eigenfaces (65.7%). This experimental result also demonstrates that color image-based face representation and recognition is more robust to illumination variations.

Complex LDA-Based Face Recognition

In this section, a complex LDA-based combined fisherfaces framework (coined complex fisherfaces) is developed for face image feature extraction and recognition (Yang, Yang, & Frandi, 2003). In this framework, PCA and KPCA are both used for feature extraction in the first phase (Schölkopf, Smola, & Müller, 1998). In the second phase, PCA-based linear features and KPCA-based nonlinear features are integrated by complex vectors, which are fed into a feature fusion container called complex LDA for a second feature extraction. Finally, the resulting complex LDA transformed features are used for classification.

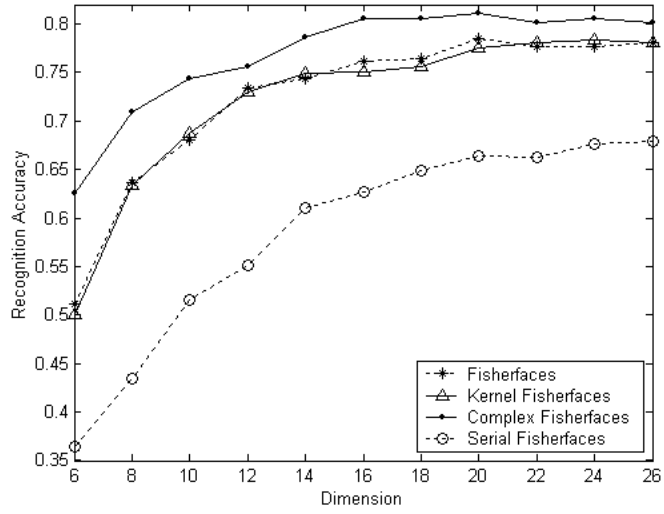
For the current PCA plus LDA two-phase algorithms, to avoid the difficulty of the within-class scatter matrix being singular in the LDA phase, a feasible way is to discard the subordinate components (those corresponding to small eigenvalues) in the PCA phase (Swets & Weng, 1996; Belhumeur, Hespanha, & Kriegman, 1997). Generally, only m principal components of PCA (KPCA) are retained and used to form the feature vectors, where m is generally subject to $c \leq m \leq M - c$. Liu and Wechsler pointed out that a constraint that just make the within-class scatter matrix nonsingular still cannot guarantee good generalization for LDA (Liu & Wechsler, 2000, 2001). This is because the trailing eigenvalues of the within-class scatter matrix tend to capture more noise if they were too small. Taking the generalization of LDA into account, we select $m = c$ components of PCA (KPCA) in the above framework.

Suppose the PCA-based feature vector is denoted by α and the KPCA-based feature vector is denoted by β. They are both c-dimensional vectors. After the normalization process using Preprocessing Method II (Note that Preprocessing Method I was shown as not very satisfying for face recognition, although it is effective for handwritten character recognition (Yang, Yang, Zhang, & Lu, 2003)), we get the normalized feature vectors $\bar{\alpha}$ and $\bar{\beta}$. Combining them by a complex vector — that is, $\gamma = \bar{\alpha} + i\bar{\beta}$ — we get a c-dimensional combined feature space. In this space, the developed complex LDA is exploited for feature extraction and fusion. This method is named *complex fisherfaces*.

Besides, we can combine the normalized feature vectors $\bar{\alpha}$ and $\bar{\beta}$ by a super-vector $\gamma = \begin{pmatrix} \bar{\alpha} \\ \bar{\beta} \end{pmatrix}$. The traditional LDA is then employed for a second feature extraction. The method is called *serial fisherfaces*. Here, it should be pointed out that *serial fisherfaces* is different from the method proposed in Liu and Wechsler (2000), where the two sets of features involved in fusion are shape- and texture-based features. Since extraction of the shape- and texture-based features needs manual operations, Liu's combined Fisher classifier method is semi-automatic, while *serial fisherfaces* and *complex fisherfaces* are both automatic methods.

The proposed algorithm was applied to face recognition and tested on a subset of the FERET database (Phillips, Moon, Rizvi, & Rauss, 2000). This subset includes 1,400 images of 200 individuals (each individual has 7 images). It is composed of the images

Figure 13.4. Recognition rates of fisherface, kernel fisherface, complex fisherfaces and serial fisherfaces on Test 1



whose names are marked with two-character strings: "*ba,*" "*bj,*" "*bk,*" "*be,*" "*bf,*" "*bd*" and "*bg.*" This subset involves variations in facial expression, illumination and pose. The facial portion of each original image was cropped based on the location of eyes, and the cropped image was resized to 80×80 pixels and pre-processed by histogram equalization.

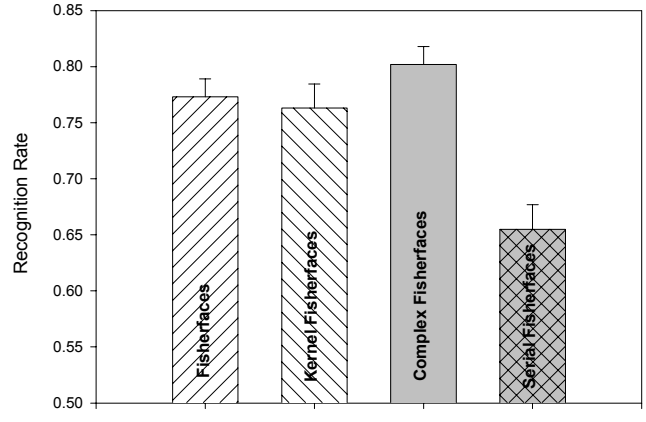
In our experiment, three images of each subject are randomly chosen for training, while the remaining images are used for testing. Thus, the total number of training samples is 600 and the total number of testing samples is 800. Fisherfaces (Belhumeur, Hespanha, & Kriegman, 1997), kernel fisherfaces (Yang, 2002), complex fisherfaces and serial fisherfaces, respectively, are used for feature extraction. Like that in complex fisherfaces and serial fisherfaces, 200 principal components ($m = c = 200$) are chosen in the first phase of fisherfaces and kernel fisherfaces. Yang has demonstrated that a second- or third-order polynomial kernel suffices to achieve good results with less computation than other kernels (Yang, 2002). So, for consistency with Yang's studies, the polynomial kernel $k(\mathbf{x}, \mathbf{y}) = (\mathbf{x} \cdot \mathbf{y} + 1)^q$ is adopted ($q = 2$) here for all kernel-related methods. Finally, a minimum (unitary) distance classifier is employed. The classification results are illustrated in Figure 13.4. Note that in Figure 13.4, for each method, we only show the recognition rates within the interval where the dimension (the number of features) varies from 6 to 26. This is because the maximal recognition rates of fisherfaces, kernel fisherfaces and complex fisherfaces all occur within this interval, and their recognition rates begin to reduce after the dimension is more than 26. Here, we pay less attention to serial fisherfaces, because its recognition rates are overall less than 70%.

From Figure 13.4, we can see that the performance of complex fisherfaces is consistently better than fisherfaces and kernel fisherfaces. Fisherfaces can only utilize the linear discriminant information, while kernel fisherfaces can only utilize the nonlinear discriminant information. In contrast, complex fisherfaces can make use of these two kinds of discriminant information, which turn out to be complimentary for achieving a

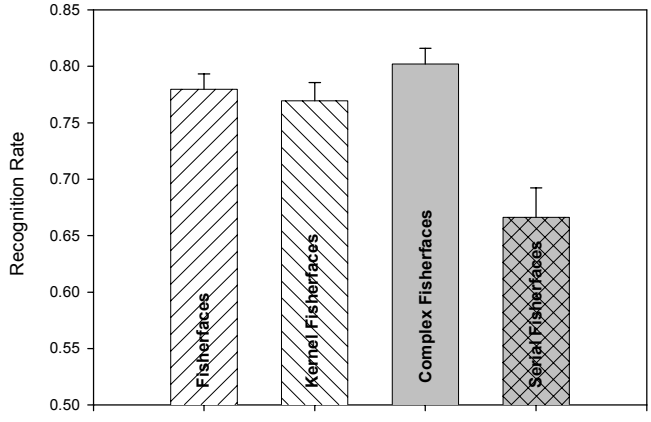
Table 13.1. The average recognition rates (%) of eigenfaces, kernel eigenfaces, fisherfaces, kernel fisherfaces, complex fisherfaces, and serial fisherfaces across 10 tests and four dimensions (18, 20, 22, 24)

eigenfaces	kernel eigenfaces	fisherfaces	kernel fisherfaces	complex fisherfaces	serial fisherfaces
17.53	16.94	77.87	77.16	80.30	66.96

Figure 13.5. The mean and standard deviation of the recognition rates of fisherface, kernel fisherface, complex fisherfaces, and serial fisherfaces across 10 tests when the dimension=18, 20, 22, 24, respectively

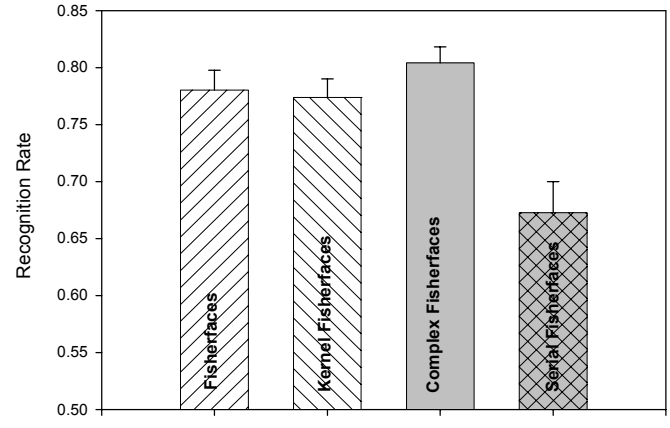


(dimension = 18)

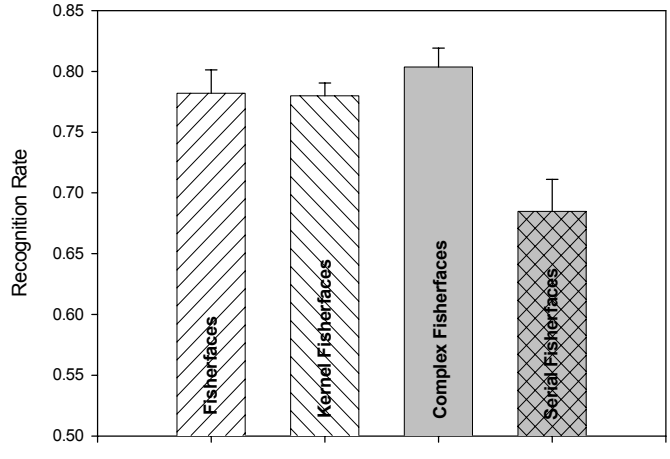


(dimension = 20)

Figure 13.5. The mean and standard deviation of the recognition rates of fisherface, kernel fisherface, complex fisherfaces, and serial fisherfaces across 10 tests when the dimension=18, 20, 22, 24, respectively (cont.)



(dimension = 22)



(dimension = 24)

better result. The performance of serial fisherfaces is not satisfying. Its recognition rates are even lower than those of fisherfaces and kernel fisherfaces.

Now, a question is: Is the above result with respect to the choice of training set? In other words, if another set of training samples was chosen at random, could we obtain a similar result? To answer this question, we repeat the above experiment 10 times. In each time, the training sample set is selected at random so that the training sample sets are different for 10 tests (Correspondingly, the testing sets are also different). For each method and four different dimensions (18, 20, 22, 24, respectively), the mean and standard deviation of the recognition rates across 10 tests are illustrated in Figure 13.5. Note that

we chose dimension = 18, 20, 22, 24, because it can be seen from Figure 13.4 that the maximal recognition rates of fisherfaces, kernel fisherfaces and complex fisherfaces all occur in the interval where the dimension varies from 18 to 24. Also, for each method mentioned above, the average recognition rates across 10 tests and four dimensions are listed in Table 13.1. Moreover, Table 13.1 also gives the results of eigenfaces and kernel eigenfaces.

Figure 13.5 shows that complex fisherfaces outperforms fisherfaces and kernel fisherfaces irrespective of different training sample sets and varying dimensions. The mean recognition rate of complex fisherfaces is more than 2% higher than those of fisherfaces and kernel fisherfaces, and its standard deviation is always between or less than those of fisherfaces and kernel fisherfaces. However, serial fisherfaces does not perform well across all these trials and dimensional variations. Its mean recognition rate is lower, while its standard deviation is larger than other methods. Table 13.1 shows fisherfaces, kernel fisherfaces and complex fisherfaces are all significantly superior to eigenfaces and kernel eigenfaces. This indicates that LDA (or KFD) is really helpful for improving the performance of PCA (or KPCA) for face recognition. Why does complex fisherfaces perform better than serial fisherfaces? In our opinion, the underlying reason is that the parallel feature fusion strategy based on complex vectors is more suitable for the SSS problem like face recognition than the serial strategy based on super-vectors. For SSS problems, the higher the dimension of feature vector, the more difficult it is to evaluate the scatter matrices accurately. If we combine two sets of features of a sample serially by a super-vector, the dimension of feature vector will increase double. For instance, the resulting super-vector $\gamma = \begin{pmatrix} \alpha \\ \beta \end{pmatrix}$ is 400 dimensional after two 200-dimensional α and β are combined in the above experiment. Thus, after serial combination, it becomes more difficult to evaluate the scatter matrices (based on a relatively small number of training samples) than before. Whereas, the parallel feature fusion strategy based on complex vectors can avoid this difficulty, since the dimension keeps invariable after combination. In our experiments, using parallel strategy, the size of the scatter matrices is still 200×200, just like before fusion. However, the size of the scatter matrices becomes 400×400 as the serial strategy is used. Taking the characteristic of complex matrices into account, the amount of data needing evaluation using complex (parallel) combination is only half of that using serial combination. Note that for an $l \times l$ complex matrix, there are $2l^2$ real numbers involved, since one complex element is actually formed by two real numbers. Consequently, it is much easier to evaluate the corresponding scatter matrices using complex fisherfaces than serial fisherfaces.

Actually, a more specific explanation can be given from the *spectrum magnitude criterion* point of view. Liu and Wechsler thought the trailing eigenvalues of the within-class scatter matrix should not be too small (2000, 2001) for good generalization. Let us calculate 10 smallest eigenvalues of the within-class scatter matrix in two different fusion spaces and list them in Table 13.2. Table 13.2 shows the trailing eigenvalues of the within-class scatter matrix in the serial fusion space is much less than those in the complex (parallel) fusion space. This fact indicates that complex fisherfaces should have better generalization than serial fisherfaces.

Table 13.2. Ten smallest eigenvalues of the within-class scatter matrix in two different fusion spaces

Strategy of Combination	1	2	3	4	5	6	7	8	9	10
serial fusion (Unit: 1e-5)	2.45	2.26	2.14	1.94	1.90	1.72	1.69	1.44	1.37	1.13
complex fusion (Unit: 1e-3)	15.1	13.6	11.7	9.5	7.9	7.0	5.8	4.7	3.9	2.6

SUMMARY

A new feature fusion strategy, *feature parallel fusion*, is introduced in this chapter. The complex vector is used to represent parallel combined feature, and traditional linear projection methods, such as PCA and LDA, are generalized for feature extraction in the complex feature space. In fact, the traditional projection methods are special cases of the complex projection methods developed in this chapter.

The idea and theory proposed in this chapter enrich the content of feature level fusion. By far, two styles of feature fusion techniques come into being. One is the classical serial feature fusion, and the other is the presented parallel feature fusion. As a comparison, an outstanding advantage of parallel feature fusion is that the dimensional increase is avoided after feature combination. Thus, on the one hand, much computational time is saved in the process of subsequent feature extraction. On the other hand, the difficulty of within-class scatter matrix being singular is avoided in the case that the sum of dimension of two sets of feature vectors involved in combination is larger than the total number of training samples, which provides convenience for subsequent LDA-based linear feature extraction.

The experiments on the AR face database show that complex PCA is effective for color facial image representation. The experiments on a subset of FERET database indicate that the recognition accuracy is increased after the parallel fusion of PCA and KPCA features, and complex fisherfaces-based parallel fusion is better than serial fisherfaces-based serial fusion for face recognition. We also give the reason why serial fisherfaces could not achieve a satisfying performance. serial fisherfaces combine two sets of features via super-vectors, which doubles the dimension. The increase of dimension makes it more difficult to evaluate the scatter matrices accurately and renders the trailing eigenvalues of the within-class scatter matrix too small. These small trailing eigenvalues capture more noises and make the generalization of serial fisherfaces poor. In contrast, complex fisherfaces can avoid these disadvantages and, thus, has a good generalization.

In conclusion, this chapter provides a new and effective means of feature level fusion. The developed parallel feature fusion techniques have practical significance and wide applicability. It deserves to be emphasized that the parallel feature fusion method has more intuitive physical meaning when applied to some real-life problems. For

example, in object recognition problems, if the intensity image and range image of object are captured at the same time, and they are the same size and well matched, we can combine them by a complex matrix, in which each element contains intensity information as well as range information. After parallel feature extraction, the resulting low-dimensional complex feature vectors, which contain the fused information, are used for recognition. This is a problem deserving further exploration.

REFERENCES

Battiti, R. (1994). Using mutual information for selecting features in supervised neural net learning. *IEEE Transactions on Neural Network*, *5*(4), 537-550.

Belhumeur, P. N., Hespanha, J. P., & Kriegman, D. J. (1997). Eigenfaces vs. fisherfaces: Recognition using class specific linear projection. *IEEE Transaction on Pattern Analysis and Machine Intelligence, 19*(7), 711-720.

Chang, I. S., & Park, R-H. (2001). Segmentation based on fusion of range and intensity images using robust trimmed methods. *Pattern Recognition*, *34*(10), 1952-1962.

Chiang, H-C., Moses, R. C., & Potter, L. C. (2001). Model-based Bayesian feature matching with application to synthetic aperture radar target recognition. *Pattern Recognition*, *34*(8), 1539-1553.

Constantinidis, A. S., Fairhurst, M. C., & Rahman, A. R. (2001). A new multi-expert decision combination algorithm and its application to the detection of circumscribed masses in digital mammograms. *Pattern Recognition*, *34*(8), 1528-1537.

Dassigi, V., Mann, R. C., & Protopoescu, V. A. (2001). Inforamtion fusion for text classification-an experimental comparison. *Pattern Recognition*, *34*(12), 2413-2425.

Ding, X-R., & Cai, M-K. (1995). *Matrix theory in engineering*. Tianjin: University Press.

Doi, N., Shintani, A., Hayashi, Y., Ogihara, A., & Shinobu, T. (1995). A study on month shape features suitable for HMM speech recognition using fusion of visual and auditory information. *IEICE Trans. Fundamentals*, *E78-A*(11), 1548-1552.

Gunatilaka, A. H., & Baertlein, B. A. (2001). Feature-level and decision-level fusion of noncoincidently sampled sensors for land mine detection. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, *23*(6), 577-589.

Huang, Y. S., & Suen, C. Y. (1995). A method of combining multiple experts for the recognition of unconstrained handwritten numerals. *IEEE Transactions on Pattern Analysis and Machine Intelligence, 17*(1), 90-94.

Jimenez, L. O. (1999). Classification of hyperdimensional data based on feature and decision fusion approaches using projection pursuit, majority voting, and neural networks. *IEEE Transaction on Geoscience and Remote Sensing*, *37*(3), 1360-1366.

Jin, Z., Yang, J., Hu, Z., & Lou, Z. (2001a). Face recognition based on the uncorrelated discrimination transformation. *Pattern Recognition, 33*(7), 1405-1467.

Jin, Z., Yang, J., Tang, Z., & Hu, Z. (2001b). A theorem on the uncorrelated optimal discrimination vectors. *Pattern Recognition, 33*(10), 2041-2047.

Li, H., Deklerck, R., Cuyper, B. D., Nyssen, E., & Cornelis, J. (1995). Object recognition in brain CT-scans: Knowledge-based fusion of data from multiple feature extractors. *IEEE Transactions on Medical Imaging*, *14*(2), 212-228.

Liu, C-J., & Wechsler, H. (2000). Robust coding schemes for indexing and retrieval from large face databases. *IEEE Transactions on Image Processing*, *9*(1), 132-137.

Liu, C-J., & Wechsler, H. (2001). A shape- and texture-based enhanced Fisher classifier for face recognition. *IEEE Transactions on Image Processing*, *10*(4), 598-608.

Martinez, A. M., & Benavente, R. (1998). The AR face database. *CVC Technical Report #24*.

Peli, T., Young, M., Knox, R., Ellis, K., & Bennett, F. (1999). Feature level sensor fusion. *Proceedings of the SPIE Sensor Fusion: Architectures, Algorithms and Applications III*, *3719* (pp. 332-339).

Phillips, P. J., Moon, H., Rizvi, S. A., & Rauss, P. J. (2000). The FERET evaluation methodology for face-recognition algorithms. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, *22*(10), 1090-1104.

Schölkopf, B., Smola, A., & Müller, K. R. (1998). Nonlinear component analysis as a kernel eigenvalue problem. *Neural Computation*, *10*(5), 1299-1319.

Shi, Y., & Zhang, T. (2001). Feature analysis: Support vector machines approaches. *SPIE Conference on Image Extraction, Segmentation, and Recognition*, *4550*, 245-251.

Swets, D. L., & Weng, J. (1996). Using discriminant eigenfeatures for image retrieval. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, *18*(8), 831-836.

Turk, M., & Pentland, A. (1991). Eigenfaces for recognition. *Journal of Cognitive Neuroscience, 3*(1).

Ulug, M. E., & McCullough, C. L. (1999). Feature and data level fusion of infrared and visual images. *SPIE Conference on Sensor Fusion: Architecture.*

Yang, J., & Yang, J. Y. (2002). Generalized K-L transform based combined feature extraction. *Pattern Recognition*, *35*(1), 295-297.

Yang, J., Yang, J. Y., & Frangi, A. F. (2003b). Combined fisherfaces framework. *Image and Vision Computing*, *21*(12), 1037-1044.

Yang, J., Yang, J. Y., Zhang, D., & Lu, J. F. (2003a). Feature fusion: Parallel strategy vs. serial strategy. *Pattern Recognition*, *36*(6), 1369-1381.

Yang, Y., & Yuan, B. (2001). A novel approach for human face detection from color images under complex background. *Pattern Recognition*, *34*(10), 1983-1992.

Yang, M. H. (2002). Kernel eigenfaces vs. kernel fisherfaces: Face recognition using kernel methods. In *Proceedings of the Fifth IEEE International Conference on Automatic Face and Gesture Recognition (RGR'02)*, 215-220.

Young, T., & Fu, K-S. (1986). *Handbook of pattern recognition and image processing* (pp. 78-81). Academic Press.

Zhang, Z-H. (1998). An information model and method of feature fusion. *International Conference on Signal Processing*, *2*, 1389-1392.

About the Authors

David Zhang graduated in computer science from Peking University (1974). He earned his MSc and PhD in computer science from the Harbin Institute of Technology (HIT) (1982 and 1985, respectively). From 1986 to 1988, he was a postdoctoral fellow at Tsinghua University and then an associate professor at the Academia Sinica, Beijing. In 1994, he received his second PhD in electrical and computer engineering from the University of Waterloo, Ontario, Canada. Currently, he is a chair professor at The Hong Kong Polytechnic University, where he is the founding director of the Biometrics Technology Centre (UGC/CRC) supported by the Hong Kong SAR government. He also serves as an adjunct professor at Tsinghua University, Shanghai Jiao Tong University, Beihang University, HIT and the University of Waterloo. He is founder and editor-in-chief of the *International Journal of Image and Graphics (IJIG),* book editor of the *Kluwer International Series on Biometrics (KISB),* and program chair of the *International Conference on Biometrics Authentication (ICBA).* He is also the associate editor of more than 10 international journals, including *IEEE Transactions on SMC-A/SMC-C* and *Pattern Recognition*. He is the author of more than 10 books and 160 journal papers related to his research areas. These include biometrics, image processing, and pattern recognition. He is a current Croucher senior research fellow and distinguished speaker of the IEEE Computer Society.

Xiaoyuan Jing graduated in computer application from the Jiangsu University of Science and Technology (1992). He earned his MSc and PhD in pattern recognition from the Nanjing University of Science and Technology (1995 and 1998, respectively). From 1998 to 2001, he was a manager of the Image Technology Department of E-Com Company. From 2001 to 2004, he was an associate professor at the Institute of Automation, Chinese Academia of Sciences, Beijing, and a visiting scholar at Hong Kong Polytechnic University and Hong Kong Baptist University. Currently, he is a professor and doctor supervisor at ShenZhen Graduate School of Harbin Institute of Technology, Shenzhen, China. He serves as a member of the Intelligent Systems Applications Committee of IEEE Computational Intelligence Society. He is a reviewer of several international journals such as *IEEE Transactions* and *Pattern Recognition*. His research interests include pattern recognition, computer vision, image processing, information fusion, neural network and artificial intelligence.

Jian Yang was born in Jiangsu, China, June 1973. He earned his BS in mathematics at the Xuzhou Normal University (1995). He then completed an MS degree in applied mathematics at the Changsha Railway University (1998) and his PhD at the Nanjing University of Science and Technology (NUST) in the Department of Computer Science on the subject of pattern recognition and intelligence systems (2002). In 2003, he was a postdoctoral researcher at the University of Zaragoza. In the same year, he was awarded the RyC program Research Fellowship, sponsored by the Spanish Ministry of Science and Technology. Currently, he is a professor in the Department of Computer Science of NUST and a postdoctoral research fellow at The Hong Kong Polytechnic University. He is the author of more than 30 scientific papers in pattern recognition and computer vision. His current research interests include pattern recognition, computer vision and machine learning.

Index

D

E

F

G

H

I

K

L

M

N

O

P

Q

R

S

T

U

V

W

Y