Palmprint Identification Based on Generalization of IrisCode

by

Adams Wai Kin KONG

A thesis
presented to the University of Waterloo
in fulfillment of the
thesis requirement for the degree of
Doctor of Philosophy
in
Electrical and Computer Engineering

Waterloo, Ontario, Canada, 2007

Author's Declaration

I hereby declare that I am the sole author of this thesis. This is a true copy of the thesis, including any required final revisions, as accepted by my examiners.

I understand that my thesis may be made electronically available to the public.

Abstract

The development of accurate and reliable security systems is a matter of wide interest, and in this context biometrics is seen as a highly effective automatic mechanism for personal identification. Among biometric technologies, [1]IrisCode developed by Daugman in 1993 is regarded as a highly accurate approach, being able to support real-time personal identification of large databases. Since 1993, on the top of IrisCode, different coding methods have been proposed for iris and fingerprint identification. In this research, I extend and generalize IrisCode for real-time secure palmprint identification.

PalmCode, the first coding method for palmprint identification developed by me in 2002, directly applied IrisCode to extract phase information of palmprints as features. However, I observe that the PalmCodes from the different palms are similar, having many 45^{o} streaks. Such structural similarities in the PalmCodes of different palms would reduce the individuality of PalmCodes and the performance of palmprint identification systems. To reduce the correlation between PalmCodes, in this thesis, I employ multiple elliptical Gabor filters with different orientations to compute different PalmCodes and merge them to produce a single feature, called Fusion Code. Experimental results demonstrate that Fusion Code performs better than PalmCode. Based on the results of Fusion Code, I further identify that the orientation fields of palmprints are powerful features. Consequently, Competitive Code, which uses real parts of six Gabor filters to estimate the orientation fields, is developed. To embed the properties of IrisCode, such as high speed matching, in Competitive Code, a novel coding scheme and a bitwise angular distance are proposed. Experimental results demonstrate that Competitive Code is much more effective than other palmprint algorithms.

Although many coding methods have been developed based on IrisCode for iris and palmprint identification, we lack a detailed analysis of IrisCode. One of the aims of this research is to provide such analysis as a way of better understanding IrisCode, extending the coarse phase representation to a precise phase representation, and uncovering the relationship between IrisCode and other coding methods. This analysis demonstrates that IrisCode is a clustering process with four prototypes; the locus of a Gabor function is a two-dimensional ellipse with respect to a phase parameter and the bitwise hamming distance can be regarded as a bitwise angular distance. In this analysis, I also point out that the theoretical evidence of the imposter binomial distribution of IrisCode is incomplete. I use this analysis to develop a precise phase representation which can enhance iris recognition accuracy and to relate IrisCode and other coding methods. By making use of this analysis, principal component analysis and simulated annealing, near optimal filters for palmprint identification are sought. The near optimal filters perform better than Competitive Code in term of *d*' index.

Identical twins having the closest genetics-based relationship are expected to have maximum similarity in their biometrics. Classifying identical twins is a challenging problem for some automatic biometric systems. Palmprint has been studied for personal identification for many years. However, genetically identical palmprints have not been studied. I systemically examine Competitive Code on genetically identical palmprints for automatic personal identification and to uncover the genetically related palmprint features. The experimental results show that the three principal lines and some portions of weak lines are

[1] In this thesis, IrisCode interchangeably refers to the method and features of iris recognition developed by Daugman.

genetically related features but our palms still contain rich genetically unrelated features for classifying identical twins.

As biometric systems are vulnerable to replay, database and brute-force attacks, such potential attacks must be analyzed before they are massively deployed in security systems. I propose projected multinomial distribution for studying the probability of successfully using brute-force attacks to break into a palmprint system based on Competitive Code. The proposed model indicates that it is computationally infeasible to break into the palmprint system using brute-force attacks. In addition to brute-force attacks, I address the other three security issues: template re-issuances, also called cancellable biometrics, replay attacks, and database attacks. A random orientation filter bank (ROFB) is used to generate cancellable Competitive Codes for templates re-issuances. Secret messages are hidden in templates to prevent replay and database attacks. This technique can be regarded as template watermarking. A series of analyses is provided to evaluate the security levels of the measures.

Acknowledgements

During my three-year study at the University of Waterloo, I received an incredible amount of support and encouragement, both technical and moral, from numerous professors, researchers, friends and family.

Firstly, I would like to give thanks to my advisor, Prof. Mohamed Kamel. I learned a lot from our discussion and his positive attitude. One of his suggestions that I will always remember is “learn from comments and improve your work.” This simple suggestion is applicable not only to research but also other aspects of my life.

Secondly, I would like to give thanks to my co-advisor, Prof. David Zhang. Through his network in the biometric community, I met top researchers from academy and industry at several conferences and seminars. Both Prof. Kamel and Prof. Zhang had given me academic freedom which I am especially grateful for since it allowed me to conduct my research with freedom of creativity.

This thesis analyzes IrisCode which was developed by Prof. John Daugman and generalizes IrisCode for palmprint identification. Within my study, Prof. Daugman spent numerous hours discussing his works, IrisCode and Gabor filters with me. He was kind enough to send the preprocessed iris images to me and patiently explained his unpublished algorithms. I had originally sent the manuscript version of Chapter 5 of this thesis to Prof. Daugman for his interest. However, I was pleasantly surprised when he had returned with several pages of insightful comments. In the original manuscript, I had proved that the locus of Gabor phase is an ellipse; however, Prof. Daugman believed that it should be a circle. After re-investigating the matter, I found that under some suitable parameterization, the locus of Gabor phase can be a circle [Appendix 2]. At the end of his e-mail, he wrote “Congratulations on your excellent and superb paper! Best regards, John”. I will never forget this appreciation because it was the best reward for my PhD study. I would like to give especial thanks to the superstar of iris recognition, my research mentor and my close friend, Prof. Daugman.

Another professor, also a member of my examination committee, who played an important role in my studies is Prof. Andrew Wong. He spent countless coffee breaks sharing his academic and industrial experience as well as explaining his research on pattern discovery and computer vision to me. Prof. Wong once said, “Adams, I know that you are a rising star in academies.” Although I cannot agree with him, I must thank him for all the encouragement that he has provided. Other than research, Prof. Wong taught me many Bible stories and he was able to apply those stories to everyday life. He said that “even though everyday we may see or encounter unfairness, injustice, violence and evil in the world (including academic communities) keep in mind that ultimately, we are not in control of things on earth and that our Lords will come and make the final judgement.” His words are invaluable to me.

I also would like to thank my examination committee members Prof. Bir Bhanu, Prof. Otman Basir and Prof. Daniel Stashuk for carefully reading and commenting on my thesis.

I should thank Dr King Hong Cheung, my research partner for trusting my research directions and for sharing our happiness and frustrations both academically and personally.

I would like to thank my PAMI friends Gireesh Dharwarkar, Gary Li, Masoud Makrehchi, Bakkama Srinath Reddy, Yanmin Sun, Patrick Tsui, Ehsan Mohammadi Arvacheh and Shahed Shahir for discussing various research topics.

I have to express my gratitude to Water Yan, Vinci Yan, Janet Yan, Joyce Yan, Cynthia Lam, Eric Lam, Kalvin Lam, Germaine Choi and Jackie Cheng for their concerns, kindness and traditional Chinese dinners and for their prayers.

I would like to express my sincere gratitude to my family members, dad, mon, Kin and Yan for their love and support from the day I came to the world.

During my PhD studies, a woman passed many lonely days in a city 13,000 km away from the University of Waterloo. She sacrificed so much and provided continuous support so as to allow her man to chase his dream. Her endless love and patience are memorable. The most sincere thanks are to my beloved fiancée, Lan.

To my Lord

Table of Contents

List of Figures

List of Tables

Chapter 1 Introduction to Biometrics

Biometrics refers to the technology for personal identification or authentication based on our physiological and/or behavioral characteristics. Biometrics overcomes the weaknesses of traditional personal identification schemes including token-based approaches and knowledge-based approaches. In this chapter, a brief introduction to biometrics is presented. Further information can be found at [79, 152].

1.1 Introduction

The rapid growth in the use of Internet applications and the great concern of security require reliable and automatic personal identification. Traditional automatic personal identification schemes can be divided into two categories: knowledge-based, such as a password and token-based, such as a physical key, an ID card and a passport. However, these approaches have limitations. In the knowledge-based approach, to some extent, the "knowledge" can be guessed, forgotten or shared. In the token-based approach, the "token" can be easily stolen or lost. These weaknesses generate serious financial damage to companies and societies. The following are some interesting statistics:

- According to Nilson report, in 2005, MasterCard, Vise, American Express and Discover incurred US$ 1.14 billion in fraud losses. [153]
- Between 20% and 50% of all helpdesk calls are for password resets and each password reset costs about US$70 [154].
- In 2005, 9.3 million US citizens suffered from identity theft. The loss is US$54.4 billion [155].

These figures strongly indicate that we need a more effective and reliable solution for human identity management. Biometrics is regarded as the potential solution. Many large-scale human identity management projects such as US-visit program, EU biometric passport and Hong Kong identity card involve biometrics. Biometrics is also applied to many other applications including access control, computer login and financial transactions authentication. As a result, the biometric market is expanding drastically. In 2003, the annual global biometric revenues were US$719 million only while the projected revenues in 2008 are US$4,639 million. This dramatic increase in demand requires a matching increase in biometrics research [165].

1.2 Biometric Traits

Many human characteristics proposed as biometric traits have both advantages and disadvantages. There is no so-called optimal biometric trait. The selection of biometric traits depends on requirements of applications. In this Section, a brief summary of different biometric traits is presented.

- **Deoxyribonucleic Acid** (DNA), a nucleic acid containing all genetic instructions for development of organs, is commonly applied to forensic applications such as criminal investigation and corpse identification. Everyone has unique DNA pattern, except identical twins. DNA can be extracted from blood, hair, and skin etc, which can always be collected in crime senses. One of major concerns of using DNA for personal identification is privacy since DNA can be collected unintentionally and contains all genetic information including genetic disorder.

- **Face** is a widely acceptable biometric trait, which can be captured from distance and even without users' cooperation. Nevertheless, face contains limited information for personal identification. Identical twins have very similar facial features. Another inherent difficulty of using face for personal identification is that face images of a person can change a lot due to facial expression, capture environment and aging. As a result, current face recognition systems cannot support high security applications but it is an important component in surveillance systems.

- **Fingerprint,** the pattern on fingertips, is the most mature biometric technology. Fingerprints have been used for personal identification for many centuries and current automatic fingerprint systems have achieved high performance. These patterns containing rich features including orientation field, minutiae points and pores are unique for everyone. Even identical twins sharing the same DNA also have different fingerprints. The other advantage of fingerprint recognition is that fingerprint scanners are inexpensive and small, which can be embedded in laptops, mobile phones and personal digital assistants. Fingerprints are selected for many large-scale human identity management projects including the US-visit program and the Hong Kong identity card. Although fingerprint recognition has many advantages, it is still not perfect. A small portion of the population cannot provide clear fingerprint images due to aging and genetic problems.

- **Iris**, the color pattern in eyes bounded by pupil and limbus, is a highly informative biometric trait. This pattern is unique and stable in whole lifetime. Current iris recognition systems can support real-time large-scale identification up to million records and capture iris images up to 3m. Almost all the commercial iris recognition systems are based on IrisCode developed by Daugman. Chapter 5 presents a detailed discussion and an analysis of IrisCode.

- **Hand geometry** systems recognize a person based on the measurements taken from a user's hand such as width and length of fingers and size of palms. Hand geometry is a widely acceptable and robust biometric. It is commonly applied to access control. Nevertheless, the geometric features have only limited information so it is suitable only for verification, 1-to-1 comparisons. The size of the capture device is another problem limiting its applications.

- **Palmprint,** the inner surface of palms, has rich features including principal lines, wrinkles, minutiae points, singular points and texture. These features can be used for uniquely identifying a person. Currently, there are two types of palmprint research, high resolution approach and low resolution approach. High resolution approach is suitable for forensic applications while low resolution approach is suitable for commercial applications. Chapter 2 gives detailed discussion about palmprint recognition.

- **Signature**, a behavioral biometric, is widely accepted in governmental, legal and commercial transactions. Each person can have several signatures for different applications. Nevertheless, a signature cannot uniquely identify a person. Many factors can influence the consistency of signatures such as emotional and physical conditions. Furthermore, professional forgers are capable of reproducing signatures to fool recognition systems.

- **Voice** regarded as a combination of behavioral and physiological biometrics is based on the size and shape of the appendages that generate sound. Voice recognition is commonly applied to phone-based applications and therefore, no extra input sensor is required. However, voice recognition faces several difficulties. Voice is neither distinctive nor stable. Current voice recognition systems cannot separate identical twins. Voice also changes due to medical condition, emotional state and aging.
- **Other biometrics** including gaits, lip prints, brain signals, ears, teeth, retinas, odor, keystrokes, heights, weights and genders have been proposed. They have different characteristics and different potential applications.

1.3 Terminology

In this Section, a set of terms commonly used in the biometric community is listed. Figure 1.1 illustrates some of them.

- **Genuine user** is a user who registers in a biometric system
- **Imposter user** is a person who does not register in the system but attempts to use it.
- **Template** is a representation of biometric features stored in a database of the system.
- **Matching score** is a numerical value that represents similarity or dissimilarity between two biometric signals.
- **Genuine matching score** is a matching score which is generated by matching two biometric signals (e.g. face images) from the same biometric trait (e.g. face).
- **Imposter matching score** is a matching score which is generated by matching two biometric signals from two different biometric traits.
- **Genuine distribution** is a distribution of the genuine matching scores.
- **Imposter distribution** is a distribution of the imposter matching scores.
- **Verification system** is a biometric system which performs one-to-one matching. Users are required to provide user identities, smart cards or other tokens to retrieve their templates in the database. According to the matching score generated by comparing an input biometric signal and the retrieved template, the system either accepts or rejects that they are from the same biometric trait
- **Close-set identification system** is a biometric system which performs one-to-many matching. An input biometric signal is matched with all the templates in the database. The system ranks the templates according to their matching scores.

- **Open-set identification system** is a combination of a close-set identification system and a verification system. It retrieves the first rank of the template and then performs verification. In this thesis, I generally refer identification to open-set identification.
- **Genuine acceptance rate (GAR)** is the probability or the percentage of a verification system correctly verifying a genuine user.
- **False acceptance rate (FAR)** is the probability or the percentage of a verification system recognizing an imposter user as a genuine user.
- **False rejection rate (FRR)** is the probability or the percentage of a verification system recognizing a genuine user as an imposter user.
- **Threshold** a numerical value, determines whether to accept two biometric signals from the same trait or not. For dissimilarity measures, if a matching score of two biometric signals is greater than a threshold, they are considered to be from two different traits. Otherwise, they are considered to be from the same trait. For similarity measure, if the matching score of two biometric signals is greater than the threshold, they are considered from the same trait. Otherwise, they are considered from different traits.
- **Receiver operating characteristic (ROC) curve** is a plot of genuine acceptance rate or false rejection rate against false acceptance rate for all possible operating points.

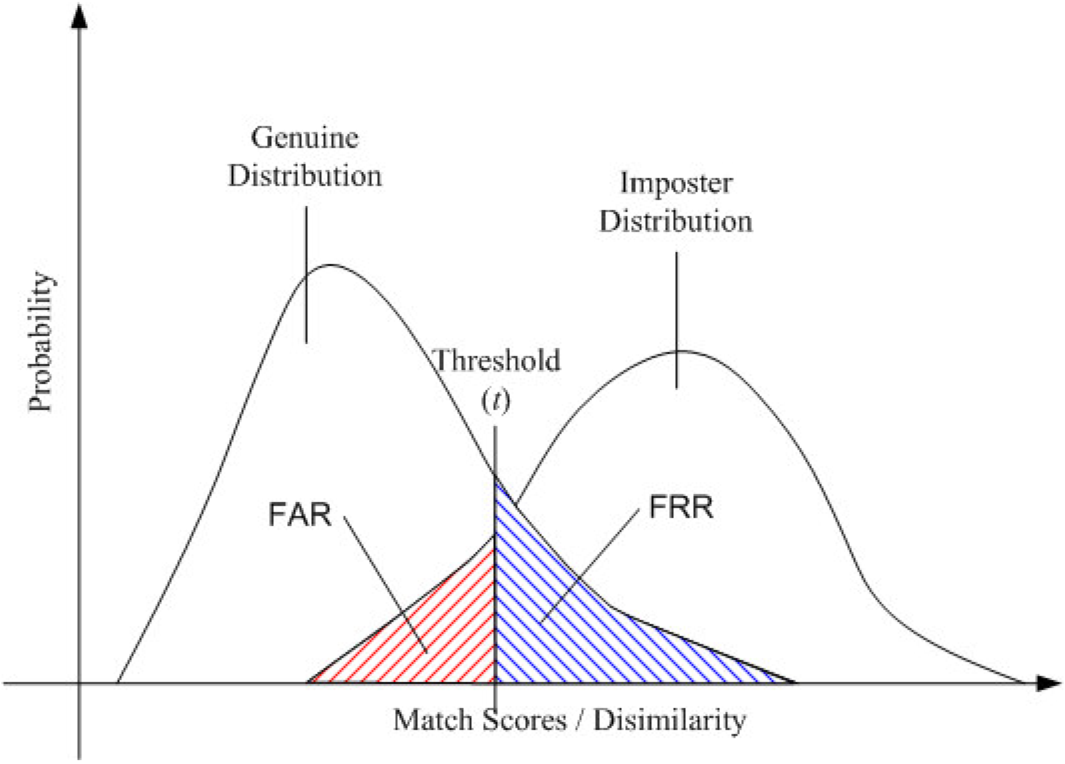


Figure 1.1 Illustration of common terms used in biometric community

1.4 Design of Biometric System

Five objectives, cost, user acceptance and environment constraints, accuracy, computation speed and security should be considered when designing a biometric system. They are inter-related, as is shown in Figure 1.2. Reducing accuracy can increase speed. Typical examples are hierarchical approaches. Reducing user acceptance can improve accuracy. For instance, users are required to provide more samples for training the system. Increasing cost can enhance security. More sensors can be embedded to collect different signals for liveness detection. In some applications, some environmental constraints such as memory usage, power consumption, size of templates, and size of devices have to be factored into a design. A biometric system installed in a PDA (personal digital assistant) requires low power and memory usage, but these requirements are not essential for access control. A practical biometric system should balance all these aspects.

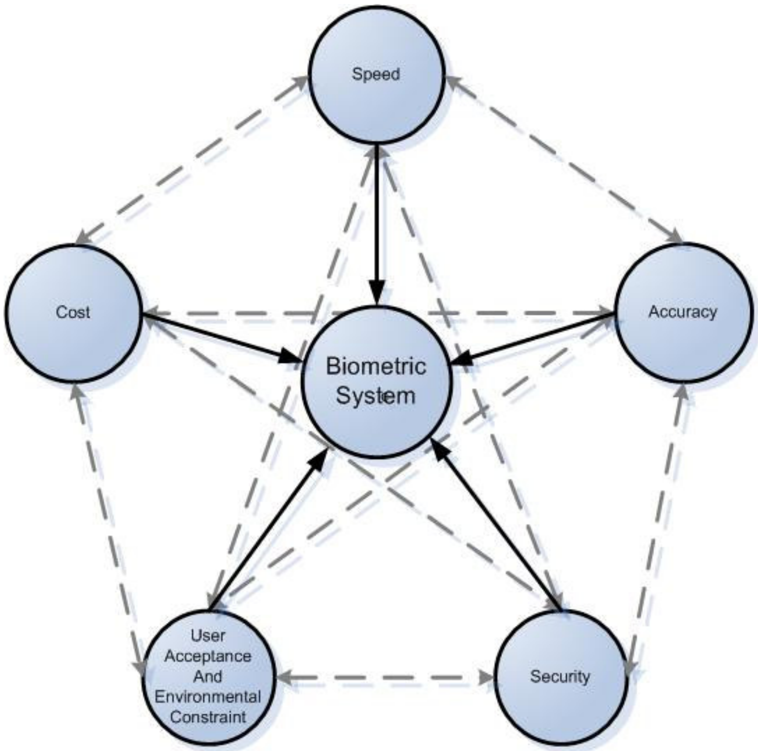


Figure 1.2 The inter relationships between different objectives for designing a biometric system

1.5 Biometric Security

Biometrics is much more secure than traditional authentication methods. They are not, however, invulnerable. For example, they are open to database, replay, and brute-force attacks. Figure 1.3 shows a number of points, Points 1-8, all being vulnerable points as identified by [26, 95, 145]. The potential attack points are between and on the common components of a biometric system, input sensor, feature extractor, matcher and database and are especially open to attack when biometric systems are employed on remote, unattended applications, giving attackers enough time to make complex and numerous attempts to break in. At Point 1, a system can be spoofed using fake biometrics such as artificial gummy fingerprints and face masks [146]. At Point 2, it is possible to avoid liveness tests in the sensors by using a pre-recorded biometric signal such as a fingerprint image. This is a so-called replay attack. At Point 3, the original output features can be replaced with a predefined feature by using a Trojan horse to override the feature extraction process. At Point 4, it is possible to use both brute-force and replay attacks, submitting on the one hand numerous synthetic templates or, on the other, prerecorded templates. At Point 5, original matching scores can be replaced with preselected matching scores by using a Trojan horse. At Point 6, it is possible to insert templates from unauthorized users into the database or to modify templates in the database. At Point 7, replay attacks are once again possible. At Point 8, it is possible to override the system's decision output and to collect the matching scores to generate the images in the registered database [150].

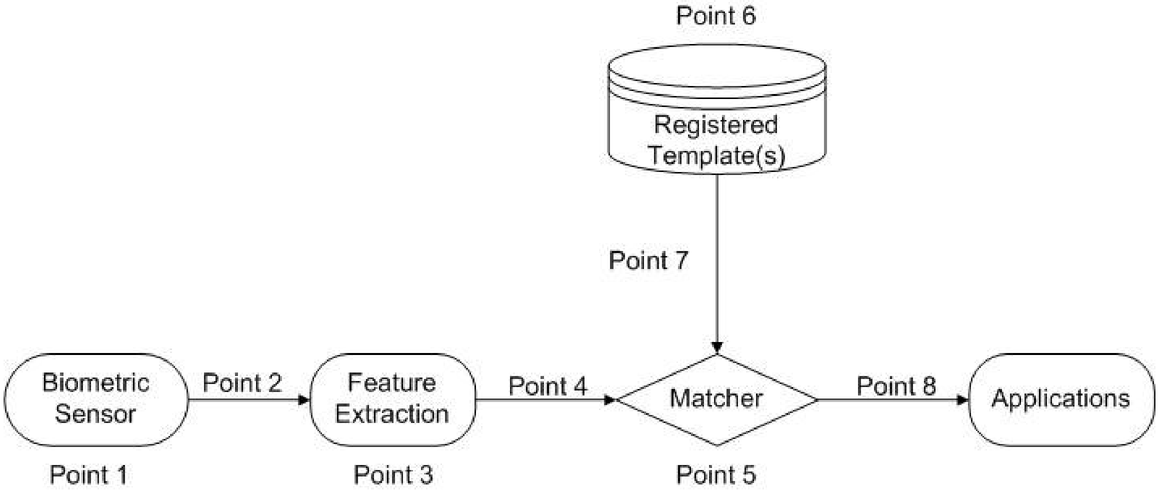


Figure 1.3 Potential attack points in a biometric system [26, 95, 145]

1.6 Problems Raised by Biometrics

In addition to the potential illegitimate access by imposters, biometric systems raise issues including unintended functions, unintended applications and template sharing [156].

- **Unintended functions**: Our biometric traits contain rich private information, which can be extracted from biometrics for non-authentication purposes. DNA containing all genetic information including sex, ethnicity, physical disorder and mental illness can be employed for discrimination. Certain patterns in palm lines also associate with mental disorders such as Down syndrome and schizophrenia [36]
- **Unintended applications**: Some biometric traits can be collected without user cooperation. Face and iris are two typical examples. Governments and organizations can employ them for tracking.

- **Template sharing:** Biometric templates in databases of authorized agents are possible to be shared by unauthorized agents.

1.7 Motivation and Summary of the Work

Palmprint recognition has been studied for many years and a real-time large-scale palmprint identification algorithm has not been developed. IrisCode developed in 1993, and continuously modified, by Daugman demonstrates the ability for real-time large-scale iris identification. To develop such an algorithm for palmprint recognition, in this thesis, I investigate the properties of palmprints and analyze IrisCode.

PalmCode, a palmprint identification algorithm directly applying IrisCode to palmprints, was developed by me in 2002 [1, 7]. Even though PalmCode is regarded as an important algorithm in palmprint research, the accuracy of PalmCode can support only 1-to-100 palmprint identification. In this thesis, I firstly identify that the performance of PalmCode is suppressed by its highly correlated features. To break this correlation, I propose a new coding algorithm called Fusion Code, which combines different PalmCodes generated by Gabor filters with different directions. Experimental results show that Fusion Code performs better than PalmCode. Nevertheless, both Fusion Code and PalmCode are based on the phase features in palmprints. To develop an accurate palmprint identification algorithm, effective palmprint features have to be identified. Based on the results of Fusion Code, I identify that orientation field of palmprints is a powerful feature. This feature is ignored by previous palmprint research. Using orientation field as a feature, I develop a new coding algorithm, called Competitive Code. The original bitwise hamming distance used in IrisCode, PalmCode and Fusion Code is not suitable for comparing orientation fields. For high speed matching, I develop a new coding scheme and bitwise angular distance specifically for Competitive Code. Comparing other palmprint algorithms including Fusion Code and PalmCode, Competitive Code is the best in terms of accuracy.

Although the performance of Competitive Code is excellent, a general theory for the coding methods is still missing. Since all the coding methods including those proposed by other authors are developed on the bases of IrisCode, analyzing IrisCode is essential for establishing such a general coding framework. This analysis demonstrates that all coding methods are in fact clustering processes; bitwise angular distance and bitwise hamming distance are equivalent; the locus of a Gabor function is a two-dimensional ellipse with respect to a phase parameter and a Gabor filter can be regarded as a steerable filter [129]. Furthermore, I show that the theoretical evidence of binomial imposter distribution of IrisCode is not enough. As a result, the theoretical imposter distributions of other coding methods are possible to follow other distributions, not binomial distribution. Based on this analysis, I develop a learning algorithm to automatically generate the filters for the coding methods.

In addition to development of palmprint identification algorithms, studying identical twins' palmprints are also important. Classifying identical twins is a challenging problem for some automatic biometric systems. However, this issue is ignored in the previous palmprint studies. I examine Competitive Code on genetically identical palmprints and show that Competitive Code can effectively classify them. Furthermore, I identify that the three principal lines and some portions of wrinkles are genetically dependent features but palmprints contain rich genetically unrelated features.

Figure 1.3 illustrates that biometric systems are vulnerable to various attacks. To develop an actual palmprint identification system, security issues should be addressed. In the last part of this thesis, I address four security issues: template re-issuances, also called cancellable biometrics, brute-force attacks, replay attacks, and database attacks. A random orientation field is inserted in the feature extractor of Competitive Code for template re-issuances. This random orientation field can generate much more than enough cancellable Competitive Code. A projected multinomial distribution is proposed for studying the probability of successfully using brute-force attacks to break into a palmprint system based on Competitive Code. This model indicates that it is computationally infeasible to break into the palmprint system using brute-force attacks. One time pad regarded as perfect secrecy is used to defend against replay attacks. Secret messages are hidden in templates for detecting database attacks. This technique can be regarded as template watermarking. A series of analyses is provided to evaluate the security levels of the measures.

1.8 Organization of the Thesis

The rest of this thesis is organized as follows. Chapter 2 gives a comprehensive survey of palmprint recognition. Chapter 3 presents Fusion Code. Chapter 4 describes Competitive Code and compares it with other palmprint algorithms. Chapter 5 analyzes IrisCode and reveals the relationship between different coding methods. Chapter 6 further generalizes the coding methods and mentions an algorithm for learning the filters in the coding methods. Chapter 7 studies the genetically identical palmprints. Chapter 8 reports the analysis of brute-fore break-ins and proposes security measures for defending against replay and database attacks and analyzes their effectiveness. Chapter 9 points out some further directions. Chapter 10 offers conclusive remarks. Figure 1.4 illustrates the organization of this thesis.

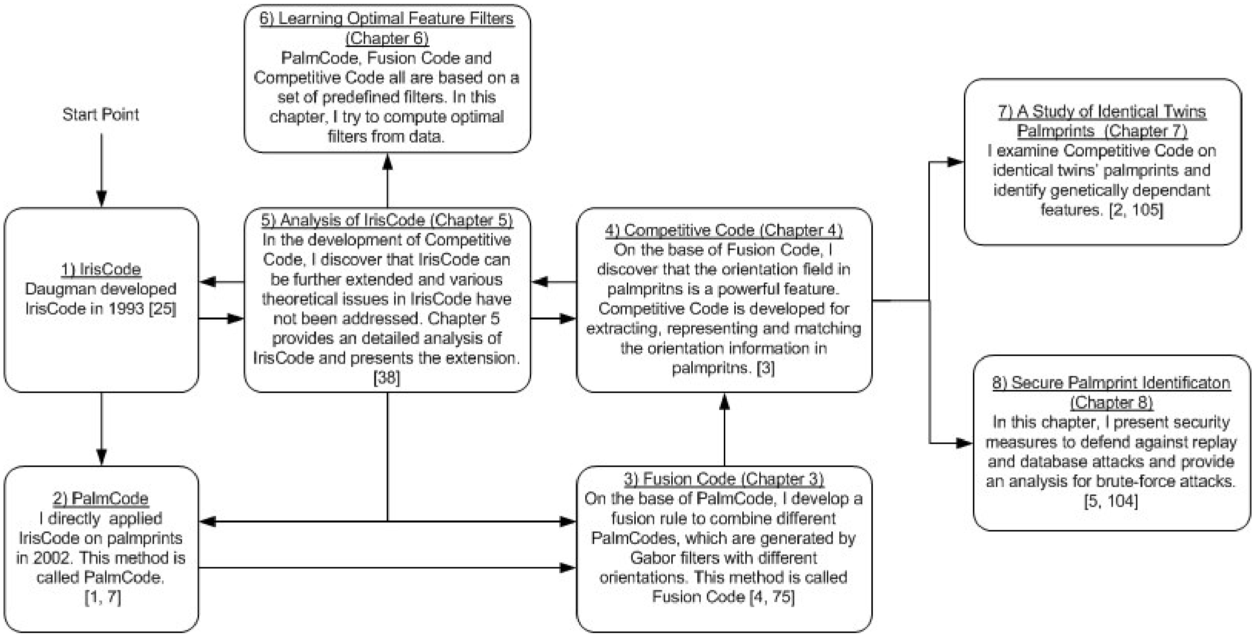

Start Point
1) IrisCode
Daugman developed IrisCode in 1993 [25]
2) PalmCode
I directly applied IrisCode on palmprints in 2002. This method is called PalmCode. [1, 7]
3) Fusion Code (Chapter 3)
On the base of PalmCode, I develop a fusion rule to combine different PalmCodes, which are generated by Gabor filters with different orientations. This method is called Fusion Code [4, 75]
4) Competitive Code (Chapter 4)
On the base of Fusion Code, I discover that the orientation field in palmpritns is a powerful feature. Competitive Code is developed for extracting, representing and matching the orientation information in palmpritns. [3]
5) Analysis of IrisCode (Chapter 5)
In the development of Competitive Code, I discover that IrisCode can be further extended and various theoretical issues in IrisCode have not been addressed. Chapter 5 provides an detailed analysis of IrisCode and presents the extension. [38]
6) Learning Optimal Feature Filters (Chapter 6)
PalmCode, Fusion Code and Competitive Code all are based on a set of predefined filters. In this chapter, I try to compute optimal filters from data.
7) A Study of Identical Twins Palmprints (Chapter 7)
I examine Competitive Code on identical twins' palmprints and identify genetically dependant features. [2, 105]
8) Secure Palmprint Identificaton (Chapter 8)
In this chapter, I present security measures to defend against replay and database attacks and provide an analysis for brute-force attacks. [5, 104]

Figure 1.4 The organization of the thesis, where the links show the relationships between chapters and other research works.

Chapter 2 Survey of Palmprint Recognition

Palmprint recognition has been investigated over the past eight years. During this period, many different problems related to palmprint recognition have been addressed. Researchers have focused on developing accurate verification algorithms. Various feature extraction and matching algorithms have been proposed. To achieve high verification accuracy, researchers combine different biometric traits with palmprints and combine different features in palmprints. Researchers also address a more challenging problem, real-time palmprint identification in large databases. In this context, both accuracy and recognition speed are important. Recently, the biometric community has also emphasized on the security of biometric systems. Pioneers have proposed some measures to protect palmprint systems. In addition to summarizing the current palmprint research, other related issues including performance evaluation and privacy involved with palmprints are discussed. The aims of this chapter are to give an overview of the current palmprint research.

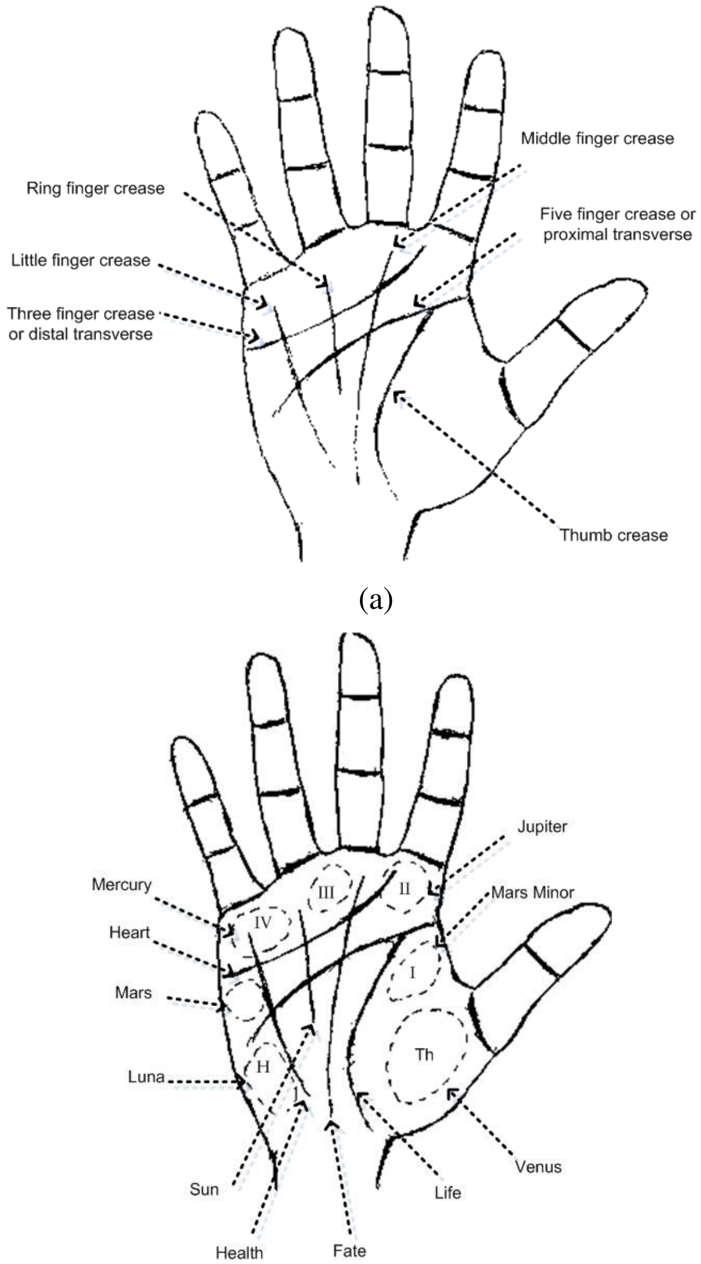
2.1 Introduction

Palmprint, the inner surface of our palm normally contains three flexion creases, secondary creases and ridges. The flexion and secondary creases are also called principal lines and wrinkles, respectively. The flexion creases and the main creases are formed between the 3rd and 5th months after conception [36] and superficial lines appear after birth. These creases are not genetically deterministic. Even identical twins who share the same DNA sequences have different palmprints [2, Chapter 7]. These non-genetically deterministic and complex patterns have rich information for personal identification.

Human beings were interested in the palm lines for fortune telling long time ago. In this century, scientists discovered that the palm lines were associated with some genetic diseases including Down syndrome, Aarskog syndrome, Cohen syndrome and fetal alcohol syndrome [68]. Scientists and fortunetellers name the lines and the regions differently, as shown in Figure 2.1 [30].

There are two types of palmprint recognition research, high resolution and low resolution approaches. High resolution approach employs high resolution images while low resolution approach employs low resolution images. High resolution approach is suitable for forensic applications such as criminal detection [24], while low resolution is more suitable for civil and commercial applications such as access control. Generally speaking, high resolution refers to 400 dpi or more and low resolution refers to 150 dpi or less. Figure 2.2 illustrates a part of a high resolution palmprint image and a low resolution palmprint image. In high resolution images, researchers can extract ridges, singular points and minutia points as features while in low resolution images, they generally use principal lines, wrinkles and texture. At the beginning of palmprint research, the high-resolution approach was the focus [69-70] but almost all current research is focused on the low resolution approach because of the potential applications. In this chapter, we concentrate only on the low resolution approach since it is the current focus.

The rest of this chapter is organized as the follows. Section 2.2 provides an overview of the current palmprint research including, palmprint scanners, preprocessing, feature extraction, matching, fusion, identification in large databases and security. Section 2.3 discusses performance evaluation, comparison and privacy issues related to palmprint recognition. Section 2.4 offers some concluding remarks.



(a)

(b)

Figure 2.1 Definitions of palm lines and regions (a) from scientists and (b) from fortune-tellers.

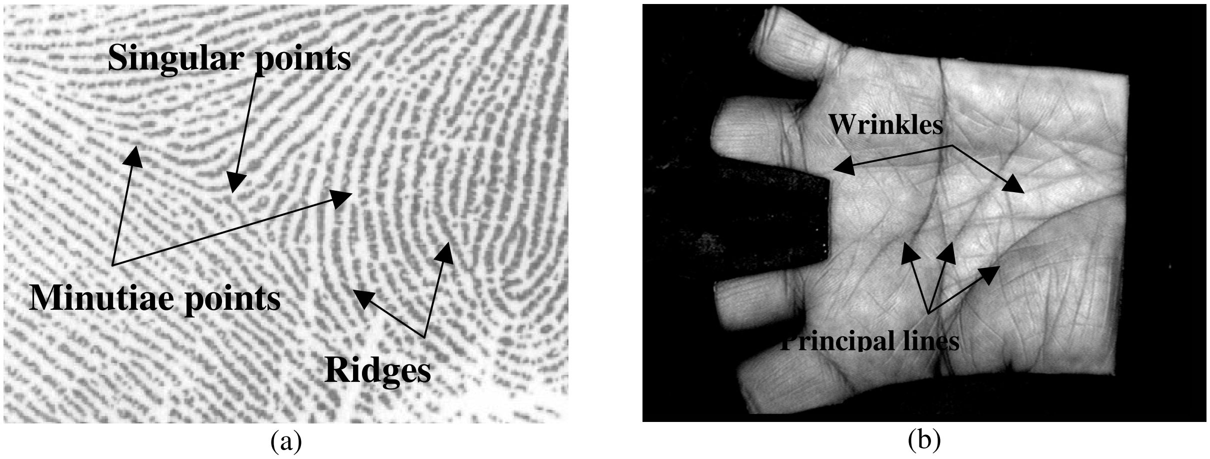


Figure 2.2 Palmprint features in (a) a high resolution image and (b) a low resolution image

2.2 Current Research

2.2.1 Overview

A palmprint recognition system generally consists of four parts: palmprint scanner, preprocessing, feature extraction and matcher. Palmprint scanner is to collect palmprint images. Preprocessing is to setup a coordinate system to align palmprint images and to segment a part of palmprint image for feature extraction. Feature extraction is to obtain effective features from the preprocessed palmprints. Finally, a matcher compares two palmprint features.

2.2.2 Palmprint Scanners

Researchers utilize four different types of sensors to collect palmprint images, CCD-based palmprint scanners, digital cameras, digital scanners and video cameras. So far, only two research teams have CCD-based palmprint scanners [7, 9]. Figure 2.3 shows a CCD-based palmprint scanner developed by the Hong Kong Polytechnic University. Generally speaking, CCD-based palmprint scanners capture high quality palmprint images and align palms accurately since the scanners have pegs for guiding placement of hands. Digital scanners are cost-effective to collect palmprint images. However, they cannot support real-time verification because of the scanning time. Digital cameras and video cameras are two ways to collect palmprint images without contact. Figure 2.4(a) is a palmprint image collected by a CCD-based palmprint scanner and Figure 2.4(b) is a palmprint image collected by a digital scanner.

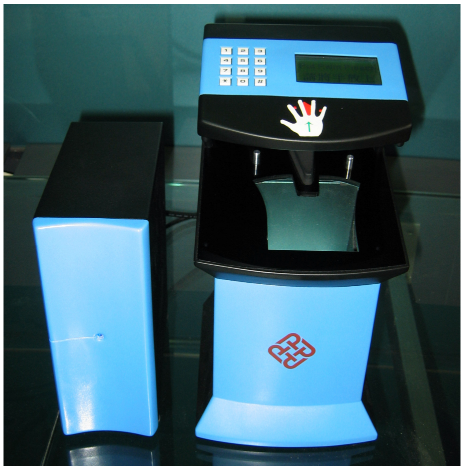
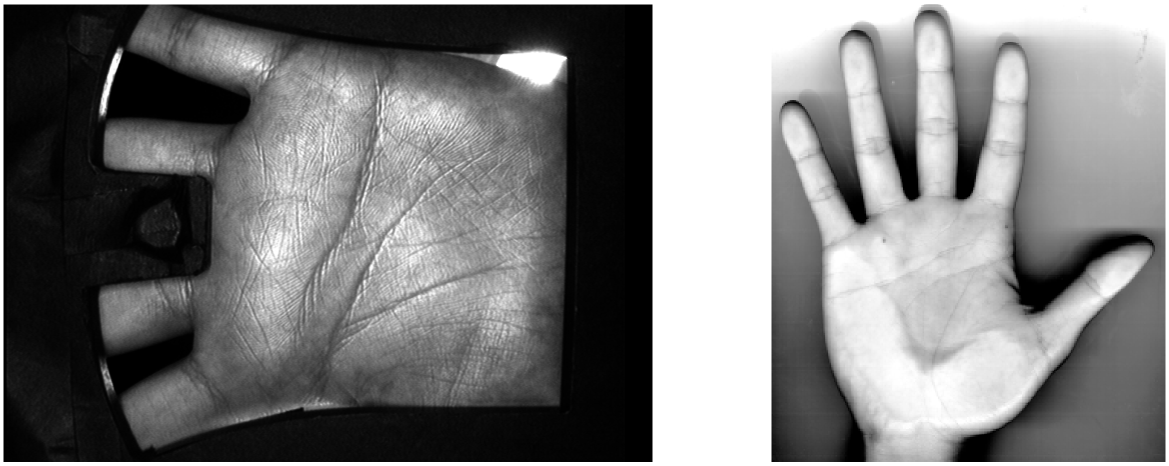


Figure 2.3 A CCD-based palmprint scanner



(a)

(b)

Figure 2.4 Two palmprints collected by (a) a CCD-based palmprint scanner, and (b) a digital scanner

2.2.3 Preprocessing

Preprocessing is used to align different palmprint images and to segment the central parts for feature extraction. Most of the preprocessing algorithms employ the key points between fingers to set up a coordinate system. Preprocessing involves generally five common steps, 1) binarizing the palm images, 2) extracting the contour of hand and/or fingers, 3) detecting the key points, 4) establishing a coordination system and 5) extracting the central parts. Figure 2.5(a) illustrates the key points and Figure 2.5(b) shows a preprocessed image. The first and second steps in all the preprocessing algorithms are similar. However, the third step has several different implementations including tangent-based [7], wavelet-based [10] and bisector-based [16, 48] to detect the key points between fingers. Furthermore, Han detects points in the middle of fingers and constructs lines passing through fingertips and the points to setup a coordinate system [9]. All these approaches utilize only the information on the boundaries of fingers, while Kumar et al. propose to use all information in palms [50]. They fit an ellipse to a binary palmprint image. According to the orientation of the ellipse, a coordinate system is established. After obtaining the coordinate systems, central parts of palmprints are segmented. Most of the preprocessing algorithms segment square regions for feature extraction, but some of them segment circular [61] and half elliptical regions [41].

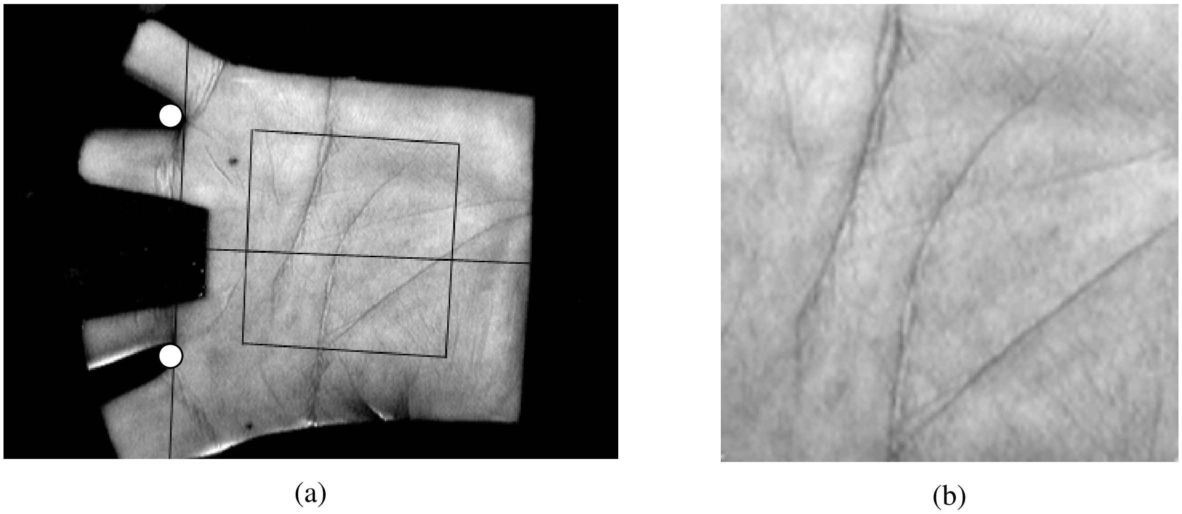


Figure 2.5 Illustration of preprocessing. (a) the key points based on finger boundary and (b) the central parts for feature extraction.

2.2.4 Feature Extraction and Matching

Comparing with image collection and preprocessing, the research of feature extraction and matching is more diverse. Feature extraction algorithms can be classified into five categories, line-based, subspace-based, local statistical-based, global statistical-based and coding-based approaches. However, some of them cannot be classified.

2.2.4.1 Line-Based Approach

Palm lines are obvious features in palmprints. Researchers employ existing edge detection methods and develop edge detectors to extract the palm lines [34, 44, 52-53, 58-60]. The extracted palm lines are either matched directly or represented in other formats for effective matching. Although at the beginning of palmprint research, some researchers concentrate on line-based approach, it is not the focus of current palmprint research since it is difficult to accurately extract palm lines from low-resolution palmprint images.

2.2.4.2 Subspace-Based Approach

Subspace-based approach (also called appearance-based approach in the literature of face recognition) involves generally in principal component analysis (PCA), linear discriminant analysis (LDA) and independent component analysis (ICA) [8, 12-13, 18, 42, 62-63, 66-67]. The subspace coefficients are considered as features. Various distance measures and classifiers are used to compare the features. In addition to applying PCA, LDA and ICA directly to palmprint images, researchers embed wavelets, discrete cosine transform (DCT) and kernels in their methods [8, 18, 39, 42]. A comprehensive comparison can be found in [8]. Researchers also develop new subspace algorithms and examine them on palmprints [71-73]. Generally speaking, subspace-based approach does not make use of any prior knowledge of palmprints.

2.2.4.3 Statistical Approach

Statistical approach can be further divided into local and global statistical approaches. Local statistical approach transforms images into another domain and then divides the transformed images into small regions [10, 15, 33, 45, 50-51, 63-64]. Local statistics such as means and variances of each small region are calculated and regarded as features. Gabor filters, wavelets and Fourier transforms have been examined. The small regions are commonly square but some of them are elliptical and circular [29, 64]. According to the collected papers, so far, no one investigates high order statistics for this approach. In addition to local statistics, researchers also employ global statistics, which are computed from whole transformed images [11, 14, 46, 49, 54]. Moments, centers of gravity and densities are considered as the global statistical features.

2.2.4.4 Coding Approach

Coding approach encodes filter responses as features [1, 3-4, 7, 56, 75]. Gabor filters are commonly applied in this approach. Phase [1, 4, 7, 75, Chapter 3] and orientation [3, 56, Chapter 4] features have been encoded. The encoding process is to construct a bitwise representation for high speed matching. The high speed matching is performed by bitwise hamming distance or bitwise angular distance [3]. These two bitwise distances are equivalent [38]. In fact, these coding methods are clustering processes [38, Chapter 5] and can be considered as extensions of IrisCode [25, Chapter 5].

Although Wu et al. use the term, *code* to describe their methods, I do not regard them as coding methods since their feature representation and matching functions are not bitwise [57, 74]. In fact, the design of their methods is flawed. They use code words 1, 2, 3, 4 to represent four directions, 0, π/4, π/2 and 3π/4, respectively and use hamming distance (non-bitwise) to match the code words. According to their hamming distance, both distance between code words 1 and 3 and distance between code words 1 and 2 are one. They ignore the distance between the codes. Competitive code and its generalization have taken into account this issue [3, 38]. However, Wu et al. ignore my works.

2.2.4.5 Other Approaches

In addition to the previous approaches, some algorithms are difficult to be classified [9, 31-32, 43, 47, 55, 78]. These algorithms combine several image-processing methods to extract palmprint features and employ standard classifiers such as neural networks to make the final decision. In addition, Kumar and his coworkers apply correlation filter for palmprint recognition [43]. In fact, correlation filter is a classifier.

2.2.4.6 Matching

Many existing classifiers including neural networks [10], hidden Markov models [48] and correlation filters [43] and various measures including cosine measure, weight Euclidean distance, Euclidean distance and hamming distance have been examined. Only limited researchers try to develop special distances or classifiers for palmprint recognition. Bitwise angular distance may be the only one [3].

2.2.5 Fusion

Fusion is a promising approach to increase accuracy [77]. Many biometric traits including finger surface [19, 39], face [20, 62, 66] and hand shape [17, 39, 50, 61, 76] have been combined with palmprints at score level or at representation level. Combining other hand features such as hand geometry and finger surface with palmprints has an inherent advantage since these features and palmprints can extract from a single hand image. Only one sensor is needed. Researchers have examined various fusion rules including sum, maximum, average, minimum, support vector machines and neural networks. In addition to combining different biometric traits with palmprints, researchers also fuse different features including appearance-based, line and texture features from palmprints [21, 29]. Kumar et al. even fuse user identity [62]. Table 2.1 summarizes the existing fusion approaches. Although fusion is an effective way to increase accuracy, it generally increases computation cost and template sizes and reduces user acceptance.

Table 2.1 Summary of palmprint fusion

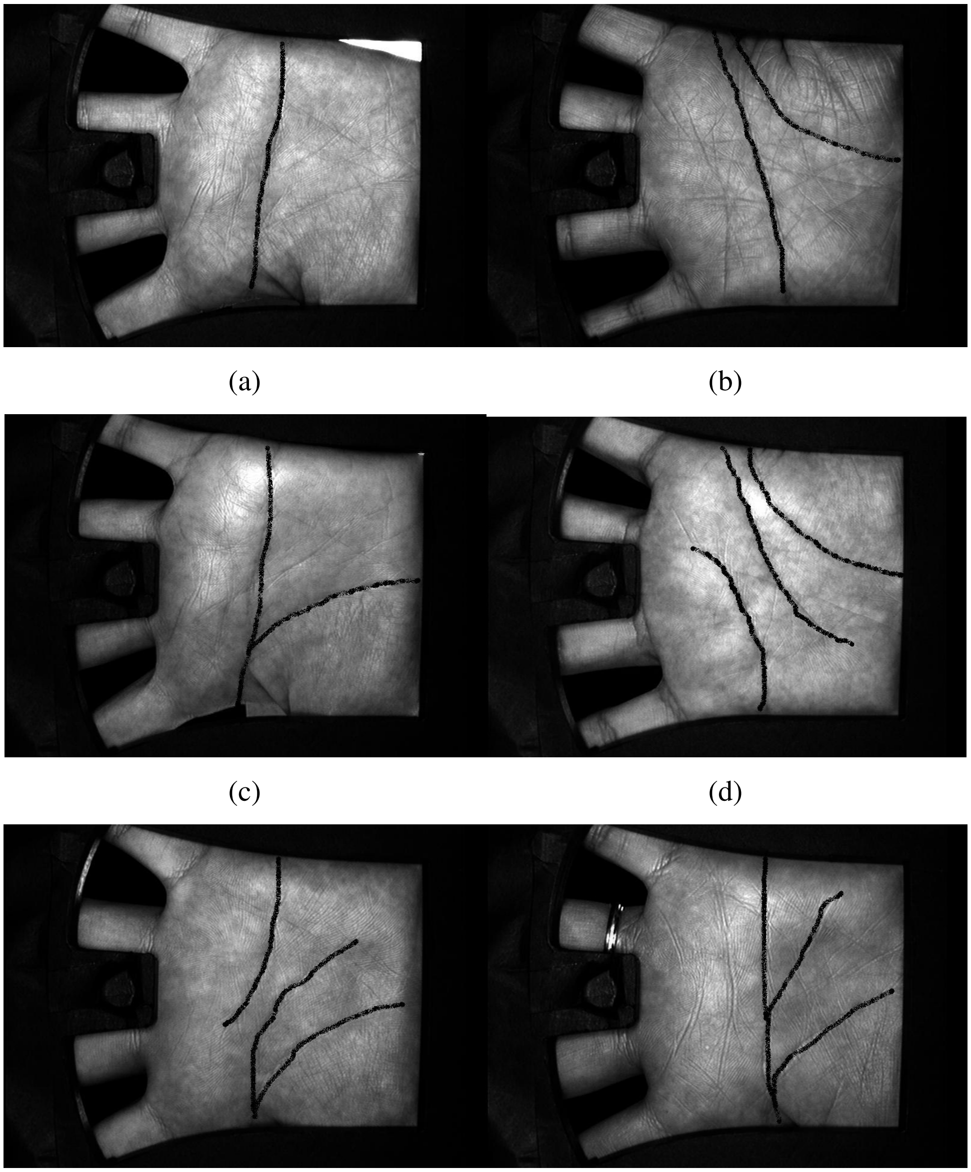
Biometric traits and features	Level of fusion	Ref
Hand geometry and palmprint	Score	17
Finger + palmprint	Score	19
Face + palmprint	Score	20
Gabor + Line features + PCA features from palmprints	Score	21
Gabor + Line + Haar wavelet features from palmprints	Score/decision	29
Hand geometry + palmprint + knuckleprint	Feature	39
Hand geometry + palmprint	Feature/score	50
Face + palmprint + Claimed identity	Score	62
Face + palmprint	Feature	66
Hand geometry + palmprint	Feature/score	76

2.2.6 Identification in Large Databases

Real-time identification in large databases is a more challenging problem. Three different approaches, hierarchy, classification and brute-force, have been proposed to attack this problem. Hierarchical approach employs simple but computationally effective features to retrieve a sub-set of the templates in a given database for further comparison [14-16]. Although hierarchical approach increases matching speed, it sacrifices accuracy. Target palmprints are possible to be removed by the classifiers using the simple features.

Classification is another approach to address this problem. Each palmprint in a given database is assigned to a class. Wu et al. define six classes based on number of principal lines and number of their intersections [22]. The six classes are illustrated in Figure 2.6. However, using their definitions and technique for identification is ineffective because the six classes are highly unbalanced e.g. about 80% of palmprints belonging category 5 shown in Figure 2.6(e) and their algorithm has high bin error of 4%.

The last one is brute-force approach [1, 3-4, 7, 56]. Brute-force approach uses one matching function to search entire databases. The advantage of this approach is to avoid introducing errors from the classification or hierarchical systems. The major challenge is to design the matching function and to identify effective features for this matching function. Daugman, the inventor of IrisCode, has demonstrated that bitwise hamming distance can achieve this goal, real-time brute-force identification in large databases [25]. Following the idea of IrisCode, PalmCode and Fusion Code employ bitwise hamming distance to match encoded phase information [1, 4, 7]. Bitwise angular distance in Competitive Code is designed to match encoded orientation features [3].



(a) (b)

(c) (d)

(e) (f)

Figure 2.6 The six classes of palmprints defined by Wu et al. [22]

2.2.7 Security

Ratha and his coworkers pinpoint that biometric systems are vulnerable to many attacks including replay, database and brute-force attacks [26]. Comparing verification, fusion and identification, only limited works are related to palmprint security.

I analyze the probability of successfully using brute-force attack to break into a palmprint identification system [5] and propose cancellable palmprints for template re-issuance and template watermarking to defend replay attacks and database attacks [Chapter 8]. Sun et al. apply watermarking techniques to hid finger features in palmprint images for secure identification [40]. Connie et al. combine pseudo-random keys and palmprint features to generate cancellable palmprint representation [27] and claim that their method can achieve zero equal error rates. This result is based on an assumption that the pseudo-random keys are never lost or shared [6]. It is in fact an unrealistic assumption. In addition to palmprints, they publish a series of papers based on this assumption and report zero equal error rates for different biometric traits [28]. A detailed analysis is given in [6].

2.3 Other Issues Related to Palmprint Recognition

2.3.1 Performance Evaluation

Evaluating biometric systems is an important topic [23]. Some researchers report impressive results even using simple methods for palmprint recognition. For example, Wu et al. report an identification rate of 99.55% in a database with 3,000 images from 300 different palms based on LDA [12]; Lu et al. report an identification rate of 99.15% in a database with 3056 images from 382 palms based on PCA [13]. However, Jing et al. report identification rates of only 71.34% for PCA and 90.91% for LDA from a database with 3040 images from 190 palms [18]. It should be pointed out that Jing et al's implementations are slightly different from Wu et al.'s and Lu et al.'s implementations. However, it is not the major problem. The major performance differences are due to the difference in their evaluation schemes. Jing et al.'s database contains palmprints collected from two occasions, but Wu et al. and Lu et al. employ palmprints collected in the same occasion. In actual applications, input palmprints and templates in a database are always collected at different occasions. For reliable performance evaluation, palmprint systems should be examined by palmprints collected from different occasions. Moreover, potential imposters would not provide their palmprints to train the systems in actual applications, so palmprints for training the systems and for evaluation should be collected from different palms. Gibbons et al. demonstrate that the performance of open systems trained by imposter's biometrics is overestimated [37]. Systems should also be examined by genetically identical palmprints since some features including the principal lines are genetically dependent [2, Chapter 7]. The algorithms highly relying on the principal lines may not able to classify genetically identical palmprints [14]. More detailed discussion about evaluating palmprint systems can be found at [35].

2.3.2 Privacy

Biometric traits contain information not only for personal identification but also for other applications. For example, deoxyribonucleic acid (DNA) and retina are useful for diagnosing genetic problems and diabetes, respectively. Palmprints are also related to some genetic disorders. Most of the previous medical research concentrates on the abnormal flexion creases, Simian line and Sydney line shown in Figure 2.7 [68]. These abnormal palm lines always associate with Down syndrome, Aarskog syndrome, Cohen syndrome and fetal alcohol syndrome [68] but about 3% of normal population has abnormal flexion creases. Medical researchers also discover the association between density of secondary creases and schizophrenia [36]. To protect our private information in palmprints, databases have to store encrypted templates only since the line features are possible to be reconstructed from raw templates. Both traditional encryption techniques and cancellable biometrics can be used for encryption. The difference between these two approaches is that cancellable biometrics performs matching in transform domains while traditional encryption techniques require decryption before matching. In other words, decryption is not necessary for cancellable biometrics. When matching speed is an issue, e.g. identification in a large database, cancellable biometrics is more suitable for hiding the privacy information.

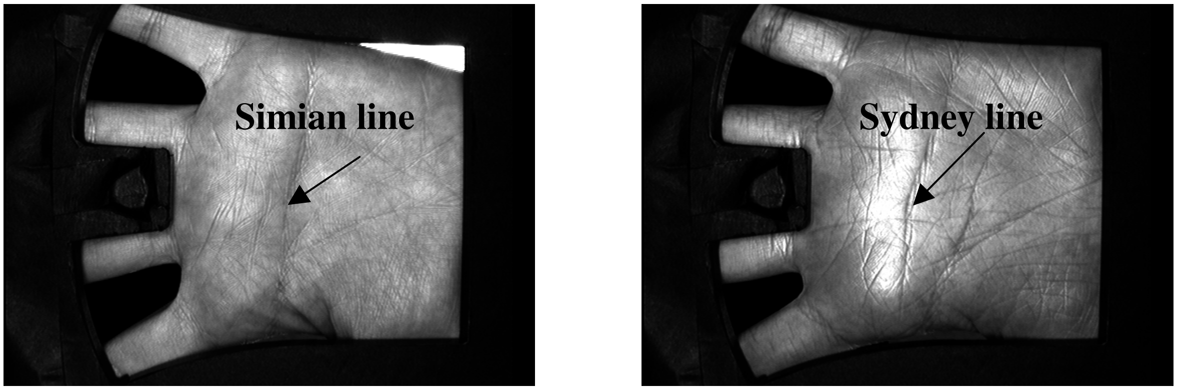


Figure 2.7 Abnormal palmprints.

2.4 Conclusion

In this chapter, a summary of the current palmprint research including sensors, preprocessing, feature extraction, matching, identification in large databases, fusion and security is presented and the related issues including privacy and performance evaluation are discussed.

Chapter 3 Palmprint Identification Using Feature-Level Fusion

In this chapter, I propose a feature-level fusion approach for improving the efficiency of PalmCode. Multiple elliptical Gabor filters with different orientations are employed to extract the phase information on a palmprint image, which is then merged according to a fusion rule to produce a single feature called the Fusion Code. The similarity of two Fusion Codes is measured by their normalized hamming distance. A dynamic threshold is used for the final decisions. Comparing the previous non-fusion approach, PalmCode and the proposed method, improvements in verification and identification are ensured.

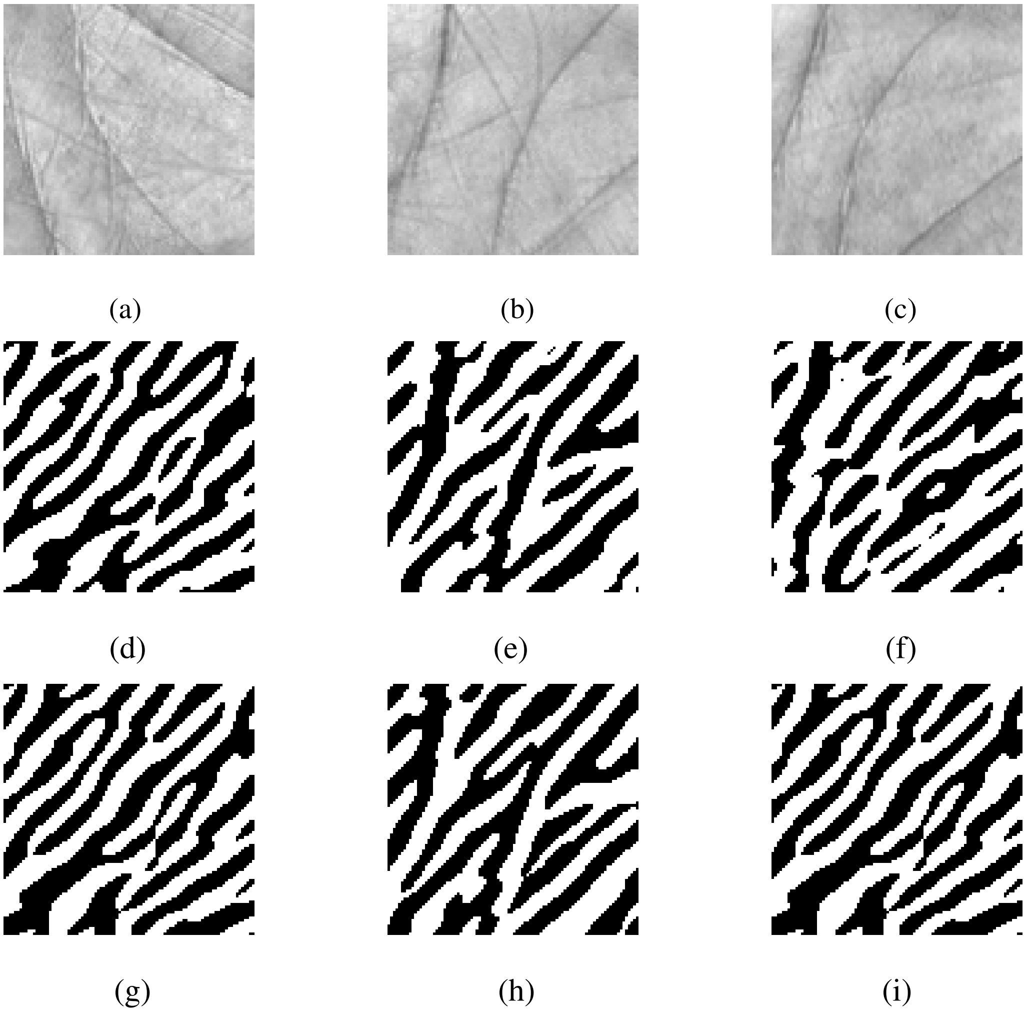
3.1 Introduction

I directly applied IrisCode to palmprints in 2002 [1]. This algorithm is referred to PalmCode in biometric community. The experimental results show that the accuracy of PalmCode can support only middle size databases [7]. To further enhance the performance, in this chapter, I firstly analyze PalmCodes and then propose a new coding method called, Fusion Code. The following is a brief summary of PalmCode. A detailed implementation can be found at [1, 7].

1. An adjusted circular Gabor filter is applied to the preprocessed palmprint images.
2. The signs of the filtered images are coded as a feature vector, and
3. Two PalmCodes are measured using the normalized hamming distance.

Figure 3.1 (d)-(i) show three PalmCodes derived from the three different palms in Figure 3.1(a)-(c). We can observe that the PalmCodes from the different palms are similar, having many 45^{o} streaks. Intuitively, we might conclude that such structural similarities in the PalmCodes of different palms would reduce the individuality of PalmCodes and the performance of the palmprint identification system.

To reduce the correlation between PalmCodes, in this chapter, a fusion rule is employed to select one of elliptical Gabor filters for coding the phase information. To further enhance the performance of the system, I replace the fixed threshold used in PalmCode by a dynamic threshold for the final decisions.



(a) (b) (c)

(d) (e) (f)

(g) (h) (i)

Figure 3.1 Three typical samples of PalmCodes: (a)-(c) original images, (d)-(f) real parts of PalmCodes, (g)-(i) imaginary parts of PalmCode.

A palmprint identification system based on Fusion Code consists of two parts: a palmprint scanner for on-line palmprint image acquisition and an algorithm for real-time palmprint identification. The system structure is illustrated in Figure 3.2. The four main steps in the system are as follows.

1) Transmit a palmprint image to a computer from a palmprint scanner.

2) Determine the two key points between the fingers and extract the central parts based on the coordinate system established by the key points. As a result, different palmprint images are aligned.

3) Convolute the central parts using a number of Gabor filters. Merge the filter outputs, then code the phases as a feature vector called Fusion Code.

4) Use the normalized hamming distance to measure the similarity of two Fusion Codes and use a dynamic threshold for the final decision.

In this chapter, the preprocessing algorithm to segment the central parts of palmprints described in [7] is employed. The proposed method will directly operate on the central parts of palmprints.

This chapter is organized as follows. Section 3.2 presents the step-by-step implementation of Fusion Codes. Section 3.3 presents the bitwise hamming distance for matching two Fusion Codes and the dynamic threshold for final decision. Section 3.4 provides a series of experimental results including, verification and identification. Section 3.5 discusses the assumption for the development of the dynamic threshold. Section 3.6 offers concluding remarks.

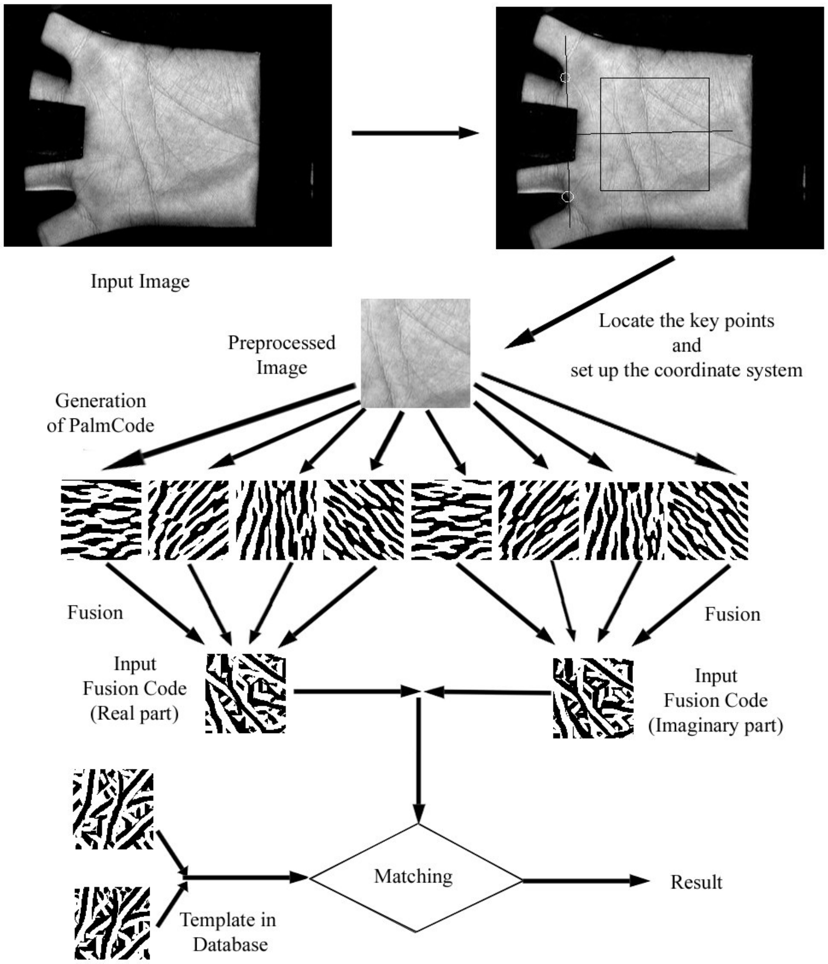

Input Image
Locate the key points and set up the coordinate system
Preprocessed Image
Generation of PalmCode
Fusion
Fusion
Input Fusion Code (Real part)
Input Fusion Code (Imaginary part)
Matching
Result
Template in Database

Figure 3.2 Block diagram of palmprint identification system based on Fusion Code.

3.2 Implementation of Fusion Code

3.2.1 Filtering

First, a preprocessed palmprint image is passed to a Gabor filter bank. The filter bank contains a number of Gabor filters, which have the following general formula:

$$G(x,y,\theta,u,\sigma,\beta)=\frac{1}{2\pi\sigma\beta}\exp\left\{-\pi\left(\frac{x'^2}{\sigma^2}+\frac{y'^2}{\beta^2}\right)\right\}\exp(2iux'), \tag{3.1}$$

where, $x'=(x-x_0)cos\theta+(y-y_0)sin\theta$, $y'=-(x-x_0)sin\theta+(y-y_0)cos\theta$, (x_0, y_0) is the center of the function, u is the radial frequency in radians per unit length and θ is the orientation of the Gabor function in radians. σ and β are the standard deviations of the elliptical Gaussian along x and y axes, respectively. As in the implementation of PalmCode, the Gabor filters are adjusted to zero DC (direct current). The parameter θ in the Gabor filters is $j\pi /v$, where j=0, 1,...,v-1 and v is the total number of Gabor filters in the bank. The other parameters are optimized for d' index defined as $d'=|\mu_1-\mu_2|/\sqrt{(\sigma_1^2+\sigma_2^2)/2}$, where μ_1 and μ_2 are the means of genuine and imposter distributions, respectively and σ_1 and σ_2 are their standard deviations. For convenience, we use G_j, to represent the Gabor filters.

3.2.2 Fusion Rule Design and Feature Coding

The filtered images contain two kinds of information: magnitude M_j and phase P_j, which are defined as

$$M_j(x,y)=\sqrt{G_j * I(x,y)\times\overline{G_j * I(x,y)}}, \tag{3.2}$$

and

$$P_j(x,y)=\tan^{-1}\left(\frac{i(\overline{G_j * I(x,y)}-G_j * I(x,y))}{G_j * I(x,y)+\overline{G_j * I(x,y)}}\right), \tag{3.3}$$

where "—" represents complex conjugate, "*" is an operator of convolution and I is a preprocessed palmprint image. Because of the zero DC Gabor filters, both of phase and magnitude are independent of the DC of the image. DC relies on the brightness of the capturing environment. Phase is also independent of the contrast of the image but the magnitude is not. These properties can be observed from the following equations.

Let AI be a preprocessed image, where A, a positive number, controls the contrast of the image. The magnitude and the phase of the filtered palmprint image are:

$$AM_j(x,y)=\sqrt{G_j * AI(x,y)\times\overline{G_j * AI(x,y)}}, \tag{3.4}$$

and

$$P_j(x,y) = \tan^{-1}\left(\frac{i(\overline{G_j * AI(x,y)} - G_j * AI(x,y))}{G_j * AI(x,y) + \overline{G_j * AI(x,y)}} \right), \tag{3.5}$$

respectively. As a result, since the PalmCode only uses the phase information, it is stable for two properties: variation in the contrast, and the *DC* of palmprint images. To design a fusion coding scheme inheriting these two properties, I employ the magnitude for fusion and the phase for the final feature. Thus, I propose a fusion rule:

$$k = \arg\max_j (M_j(x,y)), \tag{3.6}$$

and coding equations:

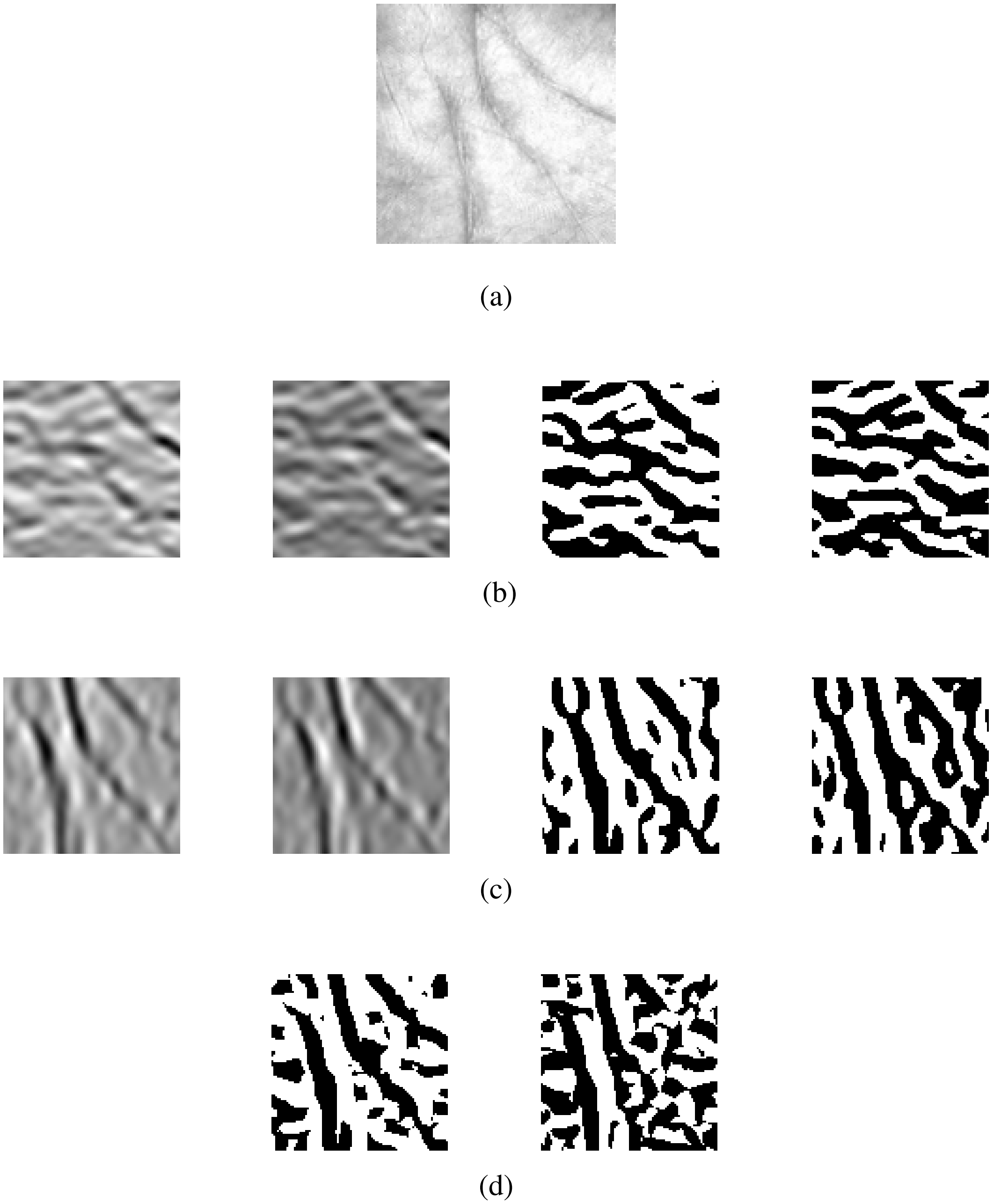
$$(h_r, h_i) = (1,1) \quad if \quad 0 \le P_k(x,y) < \pi/2, \tag{3.7}$$

$$(h_r, h_i) = (0,1) \quad if \quad \pi/2 \le P_k(x,y) < \pi, \tag{3.8}$$

$$(h_r, h_i) = (0,0) \quad if \quad \pi \le P_k(x,y) < 3\pi/2, \tag{3.9}$$

$$(h_r, h_i) = (1,0) \quad if \quad 3\pi/2 \le P_k(x,y) < 2\pi, \tag{3.10}$$

where h_r and h_i are bits in the real and the imaginary parts of the Fusion Code. A Fusion Code is illustrated in Figure 3.3, which is generated by two elliptical Gabor filters.



(a)

(b)

(c)

(d)

Figure 3.3 Procedure of how the Fusion Code is generated: (a) original palmprint image, (b)-(c) real parts (Column 1) and imaginary parts (Column 2) of the filtered images, and real parts (Column 3) and imaginary parts (Column 4) of PalmCodes and (d) Fusion Code.

3.3 Comparisons of Fusion Codes

In terms of the feature format, the proposed Fusion Code is exactly the same as that of the PalmCode. Consequently, the normalized hamming distance for the PalmCode is still useful for the Fusion Code. To describe the matching process clearly, a feature vector that consists of two feature matrices, a real one and an imaginary one is used to represent Fusion Code. A normalized hamming distance is adopted to determine the similarity measurement for palmprint matching. Let P and Q be two palmprint feature vectors. The normalized hamming distance can be described as:

$$s=\frac{\sum_{i=1}^{N}\sum_{j=1}^{N}P_M(i,j)\cap Q_M(i,j)\cap\left((P_R(i,j)\otimes Q_R(i,j)+P_I(i,j)\otimes Q_I(i,j))\right)}{2\sum_{i=1}^{N}\sum_{j=1}^{N}P_M(i,j)\cap Q_M(i,j)}, \tag{3.11}$$

where P_R (Q_R), P_I (Q_I) and $P_M(Q_M)$ are the real part, the imaginary part, and the mask of $P(Q)$, respectively. The mask is used to denote the non-palmprint pixels such as the boundary of the device which result from incorrect placement of user's hand. The result of the Boolean operator XOR ($\otimes$) is equal to zero, if and only if the bit, $P_{R(I)}(i,j)$, is equal to $Q_{R(I)}(i,j)$. The symbol $\cap$ represents the AND operator, and the size of the feature matrices is $N\times N$. It is noted that s is between 1 and 0. For the best matching, the normalized hamming should be zero. Because of imperfect preprocessing, one of the features is translated vertically and horizontally and matched again. The ranges of the vertical and the horizontal translations are defined from –2 to 2. The minimum s value obtained from the translated matching is treated as the final matching score.

In the original PalmCode, a fixed threshold is used to make the final decision. If the minimal translated matching score is greater than the fixed threshold, t_s the system rejects the statement that two PalmCodes are from the same palm; otherwise, the system accepts it. When the effective matched bits are different, the fixed threshold has different statistical confidences for different decisions. To take into account this point, I use the following dynamic threshold

$$t_d=\mu_s-(\mu_s-t_s)\times\sqrt{2048/m} \tag{3.12}$$

Eq. 3.12 is derived based on the assumption that the imposter matching score, s follows binominal distribution, $B(n,\ \mu_s)$, where n is the degrees-of-freedom and μ_s is the probability of success in each Bernoulli trail. Mathematically, the distribution is defined as

$$f(s)=\frac{n!}{S!(n-S)!}\mu_s^S(1-\mu_s)^{n-S}, \tag{3.13}$$

where S is the integer part of $n\times s$. In Section 3.5, this assumption will be discussed. Let s_1 be an imposter matching score and the corresponding number of effective matched bits is m. It is also assumed that s_1 follows $B(a,\mu_s)$ and its degrees-of-freedom, a is proportional to m. If n and a are large enough, the binomial distributions can be approximated by the normal distributions and obtain the following equation

$$\int_{-\infty}^{\frac{t_s-\mu_s}{\sqrt{\mu_s(1-\mu_s)/n}}}\frac{s-\mu_s}{\sqrt{\mu_s(1-\mu_s)/n}}ds=\int_{-\infty}^{\frac{t_d-\mu_s}{\sqrt{\mu_s(1-\mu_s)/a}}}\frac{s_1-\mu_s}{\sqrt{\mu_s(1-\mu_s)/a}}ds_1\,. \tag{3.14}$$

Since both $\frac{s-\mu_s}{\sqrt{\mu_s(1-\mu_s)/n}}$ and $\frac{s_1-\mu_s}{\sqrt{\mu_s(1-\mu_s)/a}}$ follow standard normal distributions, we have

$$\frac{t_s - \mu_s}{\sqrt{\mu_s(1-\mu_s)/n}} = \frac{t_d - \mu_s}{\sqrt{\mu_s(1-\mu_s)/a}} . \tag{3.15}$$

Simplifying Eq. 3.15 and using the assumption that the degrees-of-freedom is proportional to the number of the effective matched bits, Eq. 3.12 can be obtained. For matching two non-translated and clear palmprints, the number of matched bits should be 2048. IrisCode also uses a similar dynamic threshold but all the mathematical derivations have not been disclosed clearly [86].

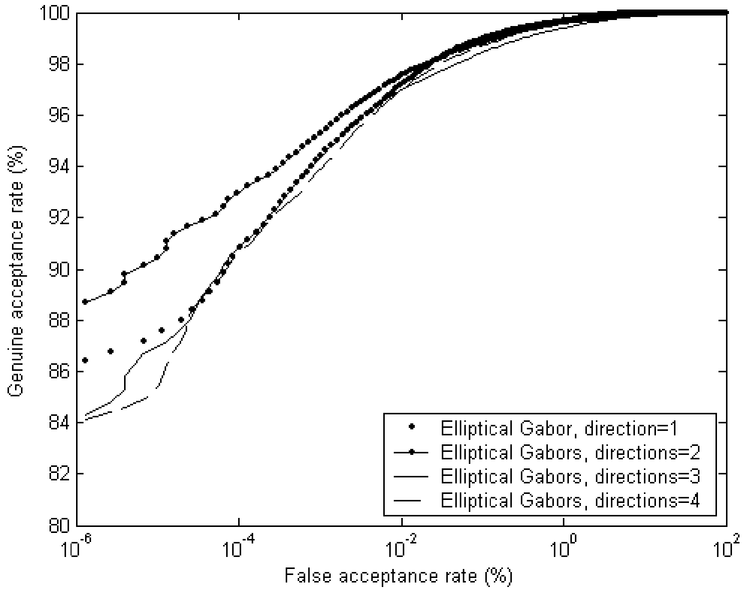
3.4 Experimental Results and Comparisons

3.4.1 Palmprint Database

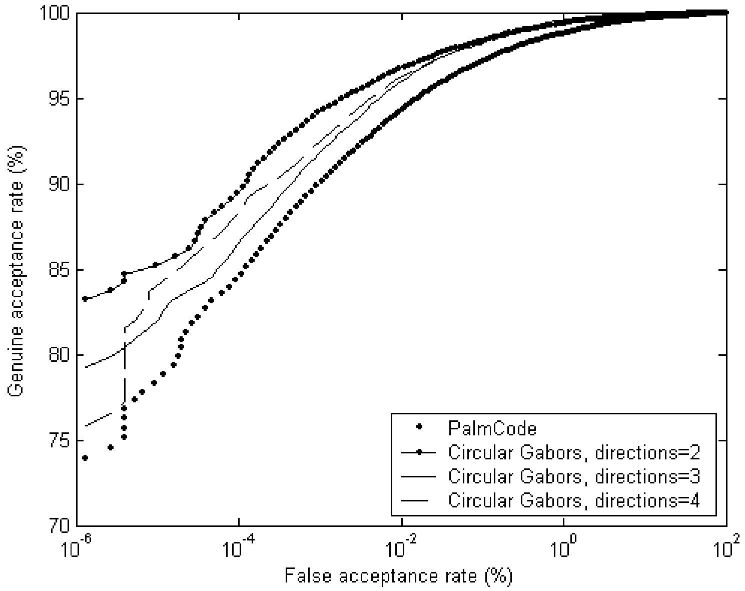
Palmprint images from 284 individuals using the palmprint capture device as described in [7] are collected. In this dataset, 186 people are male, and the age distribution of the subjects is: about 89% are younger than 30, about 10% are aged between 30 and 50, and about 1% are older than 50. The palmprint images were collected on two separate occasions, at an interval of around two months. On each occasion, the subject was asked to provide about 10 images each of the left palm and the right palm. Therefore, each person provided around 40 images, resulting in a total number of 11,074 images from 568 different palms in our database. The average time interval between the first and second occasions was 73 days. The maximum and the minimum time intervals were 340 days and 1 day, respectively. The size of all the test images used in the following experiments was 384×284 with a resolution of 75dpi. The database is divided into two datasets, training and testing. Testing set contained 9,599 palmprint images from 488 different palms and training set contained the rest of them. The training set is used to adjust the parameters of the Gabor filters only. All the experiments were conducted on the testing set. I should emphasize that matching palmprints from the same sessions was not counted in the following experiments. In other words, the palmprints from the first session were only matched with the palmprints from the second session. A matching is counted as a genuine matching if two palmprint images are from the same palm; otherwise it is counted as an imposter matching. Number of genuine and imposter matching are 47,276 and 22,987,462 respectively.

3.4.2 Comparisons of Different Types and Different Numbers of Gabor Filters

In this experiment, different numbers of elliptical and circular Gabor filters are examined. The parameters in the elliptical Gabor filters are optimized based on *d'* index. For the circular Gabor filters, I use the previous parameters [7, 75] for these comparisons. Figure 3.4(a) shows the four ROC curves obtained from elliptical Gabor filters. Each of the ROC curve represents different numbers of Gabor filters used in the fusion rule. Figure 3.4(b) shows the results obtained from the circular Gabor filters. In this test, the static threshold is used, rather than the dynamic threshold. According to Figure 3.4, we have two observations. 1) The elliptical Gabor filters perform better than the circular Gabor filters. 2) Using two filters for fusion is the best choice for both cases. The first observation can be easily understood. The elliptical Gabor filters have more parameters so that they can be well tuned for palmprint features. The reason for the second observation is not obvious. Therefore, another set of experiments is conducted. In this set of experiments, the elliptical case is considered only. First of all, the imposter distributions without considering translated matching are plotted in Figure 3.5(a). We can see that the imposter distributions from two to four filters are very similar. Their means, μ_s are 0.497 and standard deviations, σ_s are around 0.0258. However, the imposter distribution from a single filter has a relatively large variance. If binomial distribution is used to model the imposter distributions, the imposter distributions from two to four filters have around 370 degrees-of-freedom. However, the imposter distribution from the single filter only has 250 degrees-of-freedom. The degrees-of-freedoms are estimated by $\mu_s(1-\mu_s)/\sigma_s^2$. These values demonstrate that using more than two filters cannot improve the imposter distributions but increasing number of filters from one to two can get a great improvement. Although increasing number of filters can reduce the variances of the imposter distributions, it would adversely influence the genuine distributions. Given two patches of palmprints from the same palm and same location, if the number of filters is increased, the fusion rule has high probability to select different filters for coding. To demonstrate this phenomenon, all the palmprints from the same hand is matched. If the fusion rule selects the same filter, the matching distance of these local patches is zero; otherwise it is one. Then, the local matching distances are summed as a global matching distance for comparing two palmprints. The global matching distance is normalized by the matching area as Eq. 3.11. In other words, the matching function is still a hamming distance. Figure 3.5(b) shows the cumulative distributions of the genuine hamming distances. We see that the fusion rule using four filters is the easiest to select different filters. When the hamming distance is shorter than 0.3, the fusion rule using three filters performs better than that using two filters. It contradicts our expectation. The reason is that the direction of one of the three filters is close to one of the principal lines. Thus, it provides an extra robustness to the filter selection. Nevertheless, when the hamming distance is longer than 0.3, fusion rule using two filters performs better. This range is more important since false acceptance tends to happen in that region. Combining the influences for the imposter and genuine distributions, the best choice is to employ two filters for fusion. In the following experiments, I study only the two elliptical filters case.

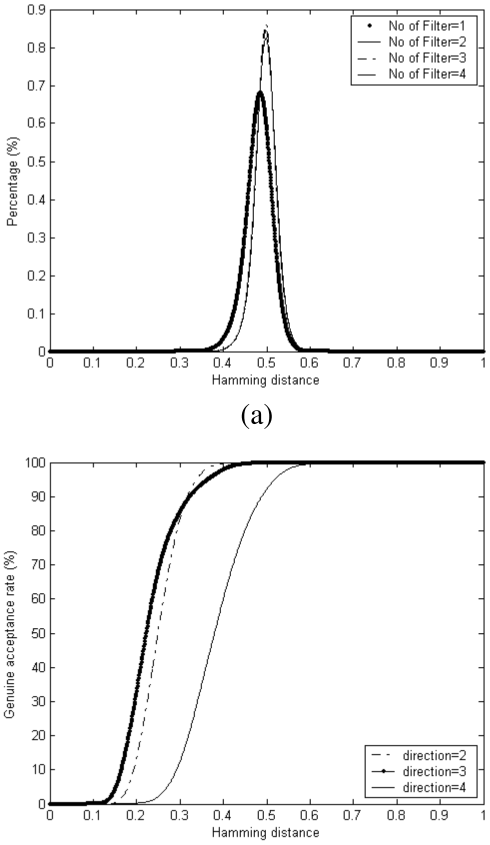


(a)



(b)

Figure 3.4 Comparisons between different numbers of filters used in fusion, (a) elliptical Gabor filters and (b) circular Gabor filters.



(a)

(b)

Figure 3.5 Analysis of different numbers of filters for fusion. (a) Comparison between imposter distributions using different numbers of elliptical Gabor filters for fusion. (b) The cumulative distributions of hamming distance for studying the fusion rules selecting different filters for coding.

3.4.3 Comparison of Static and Dynamic Thresholds

In this experiment, the proposed dynamic threshold and original static threshold are compared. For graphical presentation convenience, I dynamically scale the hamming distances rather than the threshold. In fact, they have the same effect. Figure 3.6 shows their ROC curves. We can see that dynamic threshold effectively improves the accuracy. Combining all the proposed improvements including elliptical Gabor filters, fusion rule and dynamic threshold, the proposed method obtains around 15% improvement for genuine acceptance rate when the false acceptance rate is 10^{-6}%. Table 3.1(a) lists some false acceptance rates and false rejection rates and the corresponding thresholds. The results demonstrate that the proposed method is comparable with the previous palmprint approaches and other hand-based biometric technologies, including hand geometry and fingerprint verification [7, 82]. It is also comparable with other fusion approaches [84-85]. A detailed comparison between Fusion Code and other palmprint algorithms can be found in Chapter 4.

Table 3.1 Genuine and false acceptance rates with different threshold values, (a) Verification results and (b) 1-to-488 identification results

(a)

Threshold	False Acceptance Rate	False Rejection rate
0.317	1.2×10^{-5}%	7.77%
0.324	1.3×10^{-4}%	6.07%
0.334	1.0×10^{-3}%	4.15%
0.350	1.0×10^{-2}%	2.32%

(b)

Threshold	False Acceptance Rate	False Rejection Rate
0.309	6.91×10^{-3}%	4.56%
0.315	1.38×10^{-2}%	3.67%
0.323	1.24×10^{-1}%	2.61%
0.333	9.68×10^{-1}%	1.74%

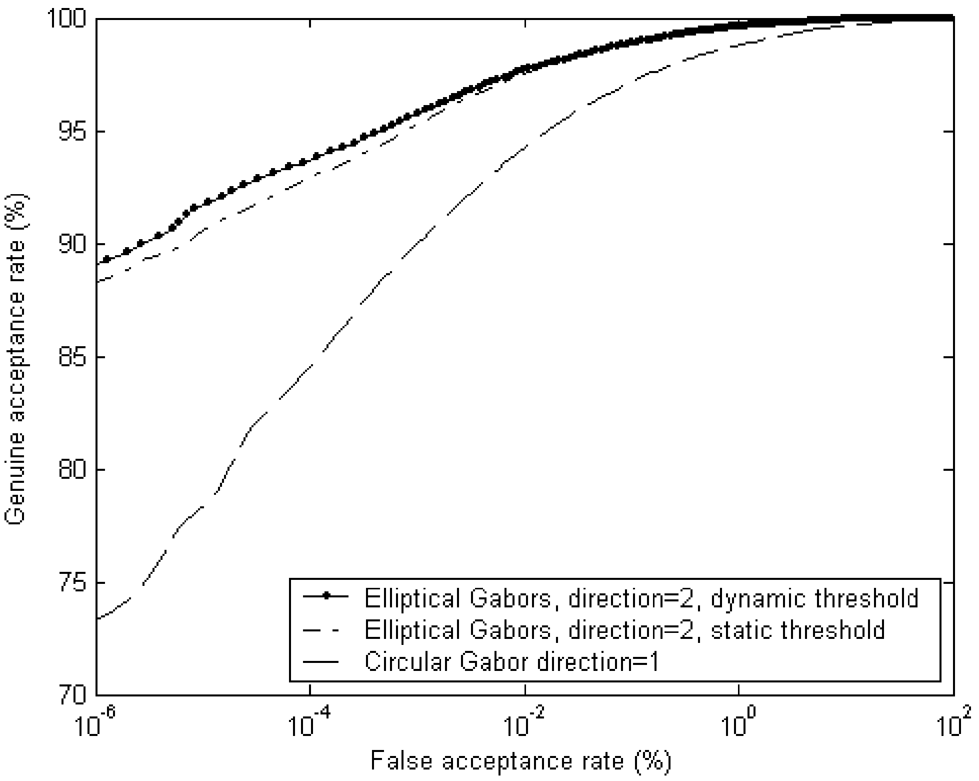


Figure 3.6 Comparison between dynamic and static thresholds

3.4.4 Identification

Identification is a one-against-many, ***M*** comparisons process. To establish the identification accuracy of the proposed method, we need to specify ***M***. In the following identification test, I set ***M***=488, which is the total number of different palms in the testing database. Generally, a practical biometric identification system stores several users' templates in its database for training the system so that the system can recognize noise or deformed signals. The original testing database is divided into the registering database and the identification database. Three palmprint images collected on the first occasion are selected for the registering database and all the palmprint images collected on the second occasion are put in the identification database. The registering and identification databases contain 1,464 and 4,821 palmprint images, respectively. Each palmprint image in the identification database is compared with all images in the registering database. Since each palm has three palmprint images in the registering database, each testing image can generate three correct verification hamming distances. The minimum of them is regarded as a correct identification hamming distance. Similarly, each testing image can generate 1,461 incorrect verification hamming distances. The minimum of them is regarded as an incorrect identification hamming distance. Thus, both the numbers of the correct and incorrect identification hamming distances are 4,821. To obtain more statistically reliable results by generating more incorrect and correct identification hamming distances, this identification test selecting other palmprint images collected on the first occasion for the registering database is repeated three times. The genuine and imposter identification distributions are generated by 14,463 correct and 14,463 incorrect identification hamming distances, respectively. The corresponding ROC curve is depicted in Figure 3.7. As the verification test shown in Figure 3.6, the ROC curve of PalmCode is also plotted for comparison. Table 3.1(b) provides the numerical values of false rejection and false acceptance rates with the corresponding thresholds for this test. The ROC curve of Fusion Code and the table show that in 1-to-488 identification, the proposed method can operate at a genuine acceptance rate of 96.33% and the corresponding false acceptance rate is 1.38×10^{-2}%. Comparing the two ROC curves, there is no doubt that the proposed Fusion Code is much better than PalmCode.

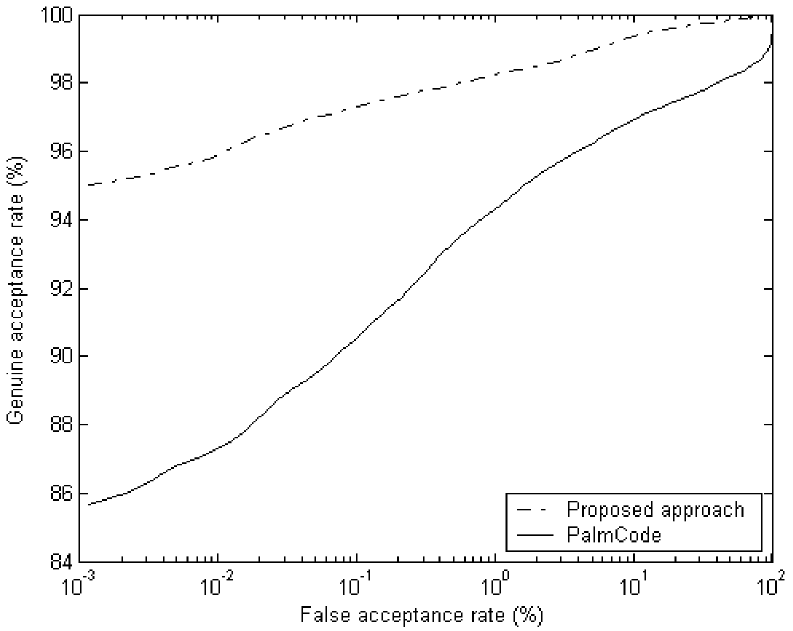


Figure 3.7 1-to-488 identification results

3.5 Discussion on the Imposter Distribution

In Section 3.3, the assumption that the imposter distribution follows binomial distribution is used to develop the dynamic threshold. To examine this assumption, the cumulative binomial probabilities are plotted against the observed cumulative probabilities. This plot is shown in Figure 3.8. If the assumption was valid, the plot would give a straight line, as the reference line in Figure 3.8. This figure shows that the observed imposter distribution is close to the binomial distribution in many regions. However, if we use Kolmogorov-Smirnov test to compare the two distributions, the test rejects that they are from the same distribution [87]. Although this assumption is not true, the dynamic threshold still effectively improves the accuracy. The further discussion of the imposter distributions of coding methods is given in Chapter 5.

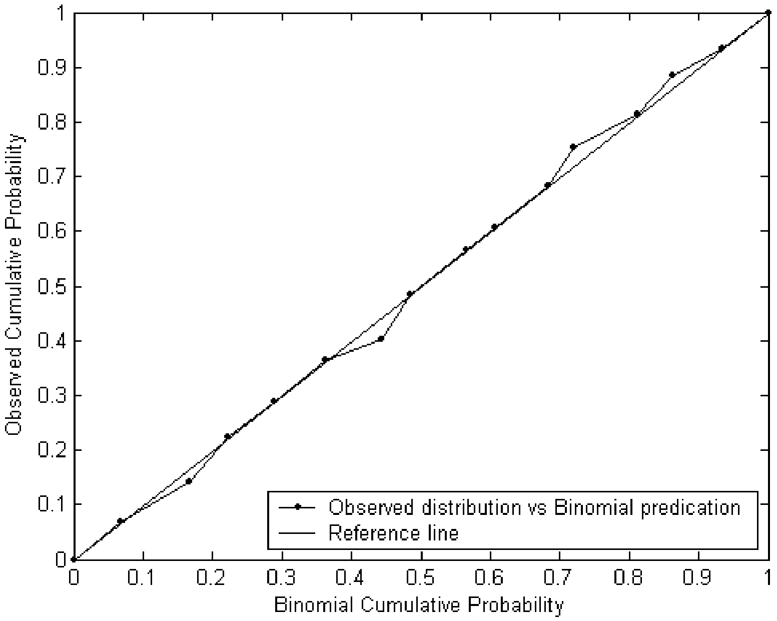


Figure 3.8 Plot of the observed cumulative probability versus the predicated binomial cumulative probability.

3.6 Conclusion

In this chapter, a feature-level coding scheme for palmprint identification is presented. On the top of PalmCode [7], a number of improvements for developing Fusion Code is made. 1) The circular Gabor filter in PalmCode is replaced by a bank of elliptical Gabor filters. 2) A feature level fusion scheme is proposed to select a filter output for feature coding. 3) The static threshold in PalmCode is replaced by the dynamic threshold. A series of experiments has been conducted to verify the usefulness of each improvement.

In the testing database containing 9,599 palmprint images from 488 different palms, the proposed method achieves around 15% verification improvement for genuine acceptance rate when the false acceptance rate is 10^{-6}%. This result is also comparable with those of other hand-based biometrics technologies, such as hand geometry, fingerprint verification and of other fusion approaches. For 1-to-488 identification, our method can operate at a low false acceptance rate (1.38×10^{-2}%) and a reasonable genuine acceptance rate (96.33%).

Chapter 4 Competitive Coding Scheme for Palmprint Identification

IrisCode, PalmCode and Fusion Code all are based on phase information for iris or palmprint identification. These two different biometric traits in fact should have different discriminative information. In this chapter, an effective palmprint feature, the orientation field of palmprints constituted by palm lines, is revealed. Using this feature, a coding method called Competitive Code is developed, which utilizes multiple 2-D Gabor filters to extract the orientation fields, a novel coding scheme to generate a bitwise feature representation and bitwise angular distance to compare two feature codes. The experimental results demonstrate that Competitive Code performs better than other palmprint recognition algorithms including PalmCode and Fusion Code.

4.1 Introduction

A palmprint contains various features, including principal lines, wrinkles, ridges, minutiae points, singular points and texture. Lines and texture are the most clearly observable features in low-resolution palmprint images (such as 100 dpi). Lines are more appealing than texture for the human vision. When human beings compare two palmprint images, they instinctively compare line features. This action motivates me to develop a coding scheme for the palm lines.

A line contains various information including type, width, position, magnitude and orientation. There are two types of lines: positive (brighter pixels in the center of the line) and negative (darker pixels in the center of the line) [159]. All the lines in palmprints are of the negative lines category. Palm lines do have a certain width. Generally, principal lines are wider than wrinkles but this information was not regarded as a useful feature in the previous palmprint research. On top of types and width, line position is often considered as an important feature, especially for the line-based approach [Chapter 2]. Magnitude of the lines has also been investigated in the previous palmprint research [10]. It is worthwhile to note that no previous palmprint research has investigated the orientation information of the palm lines for palmprint identification/verification before the conference version of this chapter [3].

This chapter considers only feature extraction and matching. The detailed information about palmprint capture and preprocessing can be referred to the previous chapters. The rest of this chapter is organized in the following sections. Section 4.2 presents the extraction of the orientation field based on a winner-take-all rule. Section 4.3 describes the angular matching for comparing the codes and its effective implementation. Section 4.4 reports experimental results including verification, comparison and matching speed. Section 4.5 summarizes the main results of this chapter and offers concluding remarks.

4.2 Extracting Orientation Field of Palmprints

Some tunable filters are appropriate for capturing the orientation information from palmprints. Gabor filters are a good choice. Based on the neurophysiological evidence from the visual cortex of mammalian brains and wavelet theory, Lee reformed the Gabor functions as the following form [103]:

$$\psi(x, y, x_o, y_o, \omega, \theta, \kappa) = \frac{\omega}{\sqrt{2\pi}\kappa} e^{-\frac{\omega^2}{8\kappa^2}(4x'^2 + y'^2)} \left(e^{\omega x'} - e^{-\frac{\kappa^2}{2}} \right), \tag{4.1}$$

where $x'=(x-x_0)\cos\theta+(y-y_0)\sin\theta$, $y'=-(x-x_0)\sin\theta+(y-y_0)\cos\theta$; (x_0, y_0) is the center of the function; ω is the radial frequency in radians per unit length and θ is the orientation of the Gabor functions in radians. The κ is defined by $\kappa = \sqrt{2\ln 2}\left(\frac{2^\delta + 1}{2^\delta - 1}\right)$, where δ is the half-amplitude bandwidth of the frequency response, which, according to neurophysiological findings, is between 1 and 1.5 octaves [103]. When σ and δ are fixed, ω can be derived from $\omega=\kappa/\sigma$. These neurophysiology-based Gabor functions are the same as the general Gabor functions but the choices of parameters are limited by neurophysiological findings and the DC of the functions are removed. Since palm lines are negative type [159], only the negative real part of the Gabor, which is defined as

$$\psi_R(x, y, x_o, y_o, \omega, \theta, \kappa) = \frac{-\omega}{\sqrt{2\pi}\kappa} e^{-\frac{\omega^2}{8\kappa^2}(4x'^2 + y'^2)} \left(\cos(\omega x') - e^{-\frac{\kappa^2}{2}} \right), \tag{4.2}$$

is needed. Using these filters, the orientation of a region of palmprint can be estimated using a rule,

$$j = \arg\max_p \int\int I(x, y)\psi_R(x, y, x_o, y_o, \omega, \theta_p, \kappa)dxdy. \tag{4.3}$$

where j is called the winning index, an integer representation of the orientation and I is a preprocessed image. Since it is a winner-take-all rule, I call this rule as competitive rule. According to the neurophysiological findings, the simple cells are sensitive to specific orientations with approximate bandwidths of $\pi/6$ [103]. Thus, six Gabor filters with orientations, $\theta_p=p\pi/6$, where $p=\{0, 1,\ldots,5\}$ are selected for the competition. The competitive rule is applied to code each sample point to obtain feature vectors with the same dimension. The proposed feature and the algorithm are named as *Competitive Code*. An example of Competitive Code is illustrated in Figure 4.1.

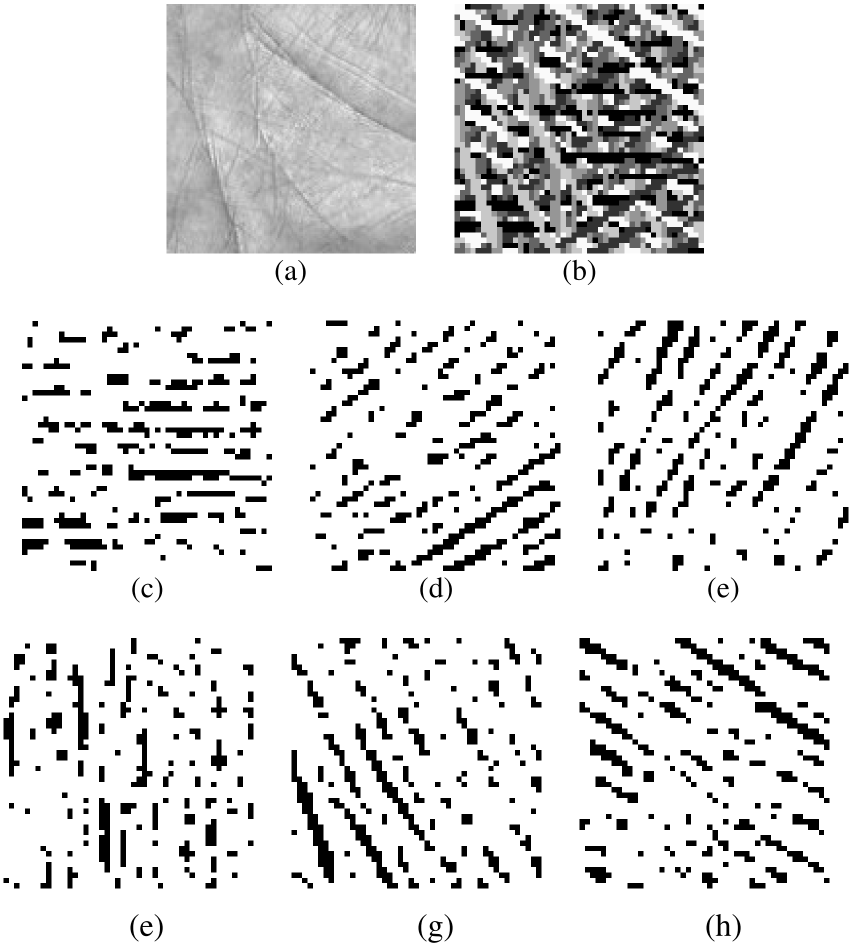


Figure 4.1 An example of Competitive Code. (a) Preprocessed image. (b) Competitive code. (c)-(h) are the winning indexes 0, 1, 2, 3, 4 and 5, respectively.

4.3 Angular Matching with Effective Implementation

Identifying a palm from a large database in real-time is a major challenge in designing matching algorithms. Some have tried to exploit hierarchical and classification approaches [14-16, 22] to increase matching speed but have sacrificed accuracy. To achieve the goal of real-time identification and to overcome the problems of previous approaches, following the idea of IrisCode [25], Boolean operators are exploited to design a matching scheme for Competitive Code, which can support real-time brute force searching in a large database.

The winning indexes are integer representations of the local orientations of palmprints. When comparing two winning indexes, it is natural to employ angular distance. The angular distances between different winning indexes are given in Table 4.1. Summing up all the angular distances at different positions, the angular distance between two Competitive Codes can be defined as

$$A_f(P,Q)=\sum_{x=1}^{N}\sum_{y=1}^{N}A(P_{x,y},Q_{x,y}), \qquad (4.4)$$

where $P_{x,y}$ $(Q_{x,y})$ is a winning index of Competitive Code $P(Q)$ at position (x, y); N by N is size of Competitive Code, and $A(P_{x,y},Q_{x,y})$ is the angular distance between the two winning indexes.

To employ Boolean operators for rapid comparison, the integer representation is converted into bitwise representation. The coding scheme is given in Table 4.2, where three bits are used to represent one winning index. Using bitwise representation, angular distance is defined as

$$A_f(P,Q)=\sum_{x=1}^{N}\sum_{y=1}^{N}\sum_{i=1}^{3}P_i^b(x,y)\otimes Q_i^b(x,y), \qquad (4.5)$$

where $P_i^b\,(Q_i^b)$ is the i^{th} bit plane of $P(Q)$ and $\otimes$ is bitwise exclusive OR. In addition to employing the Boolean operators for rapid comparison, I take in advance of the 32-bit architecture of current computers to further speed up the matching process. This is the major reason that N is set to 32. Using the bitwise implementation is not only effective for matching but also for storage. If an 8-bit integer is used to represent a winning index, the size of a Competitive Code is 1024 bytes. Now, bitwise representation needs only 384 bytes.

Sometimes, a preprocessed palmprint image contains non-palmprint pixels as described in [7]. One bit mask is used to denote the non-palmprint pixels. Finally, the angular distance is defined as

$$A_f=\frac{\sum_{x=1}^{N}\sum_{y=1}^{N}\sum_{i=1}^{3}(P_M(x,y)\cap Q_M(x,y))\cap(P_i^b(x,y)\otimes Q_i^b(x,y))}{3\sum_{x=1}^{N}\sum_{y=1}^{N}P_M(x,y)\cap Q_M(x,y)}, \qquad (4.6)$$

where P_M and Q_M are the masks of P and Q, respectively and $\cap$ is bitwise AND. The range of A_f is between 0 and 1 and the angular distance is zero for perfect matching. Twenty-five translated Competitive Codes generated by shifting the preprocessing image are computed for alignment imperfections. Thus, 25 angular distances are obtained when two palmprints are matched. The minimum of these distances is considered as the final angular distance, A_F.

Table 4.1 All possible angular distances between different winning indexes

Angular Distance		Winning indexes, $P_{x,y}$					
$A(P_{x,y}, Q_{x,y})$		0	1	2	3	4	5
Winning indexes $Q_{x,y}$	0	0	1	2	3	2	1
	1	1	0	1	2	3	2
	2	2	1	0	1	2	3
	3	3	2	1	0	1	2
	4	2	3	2	1	0	1
	5	1	2	3	2	1	0

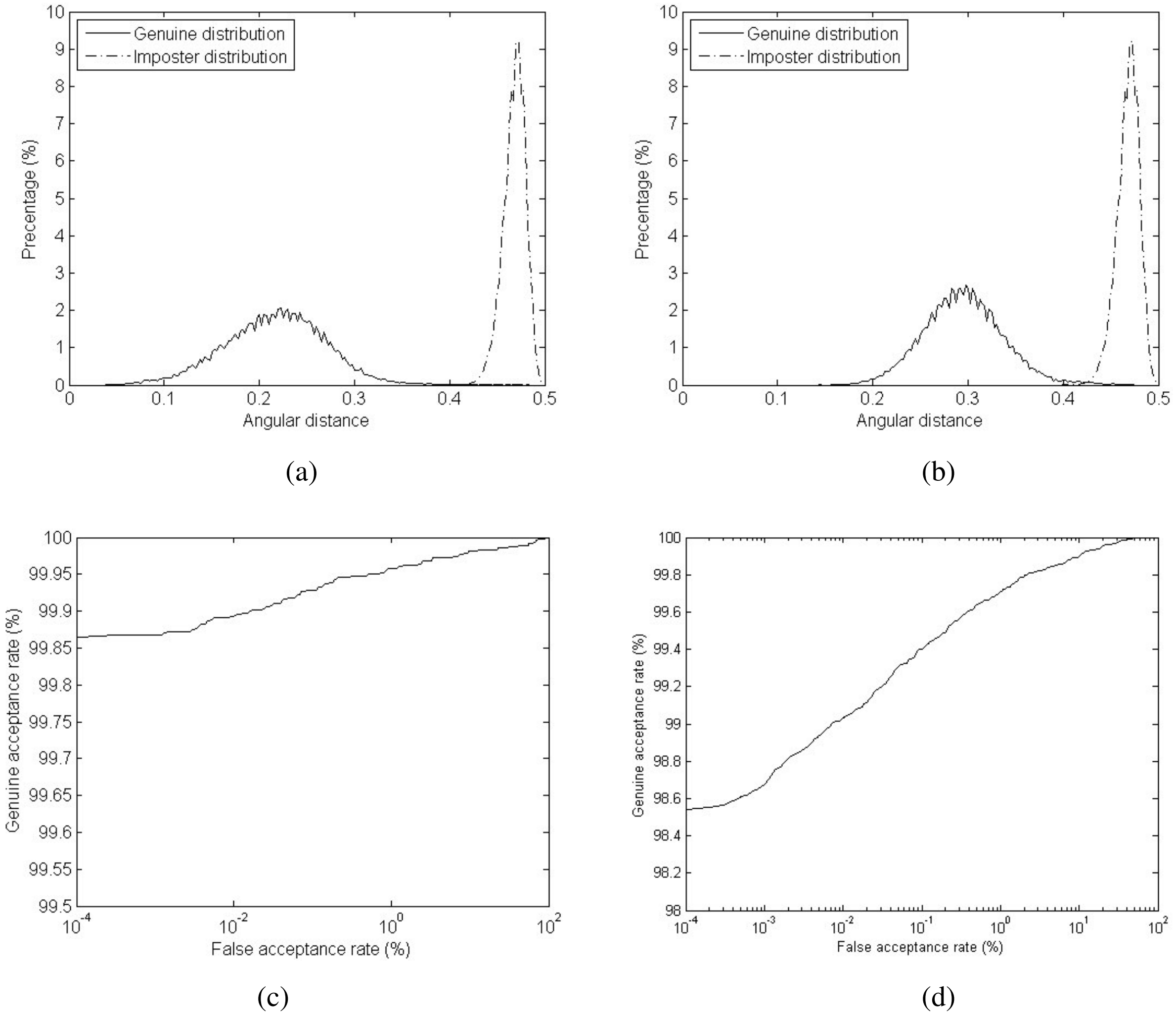
Table 4.2 Bitwise representation of the Competitive Code

Winning indexes	Bit 1	Bit 2	Bit 3
0	0	0	0
1	0	0	1
2	0	1	1
3	1	1	1
4	1	1	0
5	1	0	0

4.4 Experimental Results

4.4.1 Verification

To estimate the verification accuracy of the method, each of the palmprint images was compared with all of the palmprint images in the database described in Chapter 3. To clearly demonstrate the performance of the proposed method, two different comparison schemes are defined. The first scheme is to match the palmprint images from the same occasions. The second scheme is to match palmprint images from different occasions. The number of genuine and imposter matchings from comparisons on the same occasion are 48,707 and 30,605,025, respectively and the number of genuine and imposter matchings from comparisons on different occasions are 54,081 and 30,603,388 respectively. Figure 4.2(a) and (b) show the genuine and imposter distributions estimated by the genuine and imposter matchings and Figure 4.2(c) and (d) show their corresponding receiver operating characteristic (ROC) curves. The genuine and imposter distributions in Figure 4.2(a) and (b) are well separated. Table 4.3 lists some thresholds and the corresponding false acceptance rates (FAR) and false rejection rates (FRR). From the ROC curves, we know that it is more difficult to match palmprints from different occasions but the Competitive Code still can operate at an extremely low false acceptance rate of 1×10^{-4}% and a reasonably high genuine acceptance rate of 98.5%.



(a)

(b)

(c)

(d)

Figure 4.2 Verification test results. Genuine and imposter distributions of comparisons made on (a) the same and (b) different occasions. (c) and (d) are their corresponding ROC curves.

Table 4.3 False Acceptance Rate (FAR) and False Rejection Rate (FRR) at different thresholds in verification test

Matching on the same occasion			Matching on the different occasions		
Threshold	FAR (%)	FRR (%)	Threshold	FAR (%)	FRR (%)
0.398	1×10^{-4}	0.137	0.400	1×10^{-4}	1.487
0.402	1×10^{-3}	0.120	0.403	1×10^{-3}	1.336
0.412	1×10^{-2}	0.095	0.412	1×10^{-2}	0.981
0.424	1×10^{-1}	0.0595	0.412	1×10^{-1}	0.610

4.4.2 Comparison

To demonstrate the accuracy of the proposed method, the palmprint verification methods developed by Zhang and his co-workers and published in major journals [4, 7, 12-13, 15-16, 18] are re-implemented for comparisons. I do not compare the algorithms developed by other researchers since their palmprint images have different resolutions or contain all hand geometric features for fusion. Most importantly, the comparison may not be fair for them since the quality of my re-implementation may not be as high as the original implementation, especially setting the parameters for a different dataset. More detailed discussion about fair comparison can be found in Section 5.6. The preprocessing algorithm described in [7] is employed to obtain the preprocessed images for this comparison so that we can clearly observe the performance differences related to feature extraction, which is the major difference between the algorithms. For the hierarchical approaches [15-16], only the last recognition modules is implemented since the retrieval modules are used to speed up the identification process. Four palmprint images from the first occasion are used to train the subspace approaches [12-13, 18]. The ROC curves of the different methods are given in Figure 4.3. It is clear that the Competitive Code is the best in terms of accuracy. Several approaches including [12-13, 15, 18] can perform well for the same occasion comparison but they cannot survive comparison across different occasions. It should be noted that the best approaches are the three coding methods, Competitive Code, Fusion Code and PalmCode [7], whose designs are based on highly localized and robust feature representation.

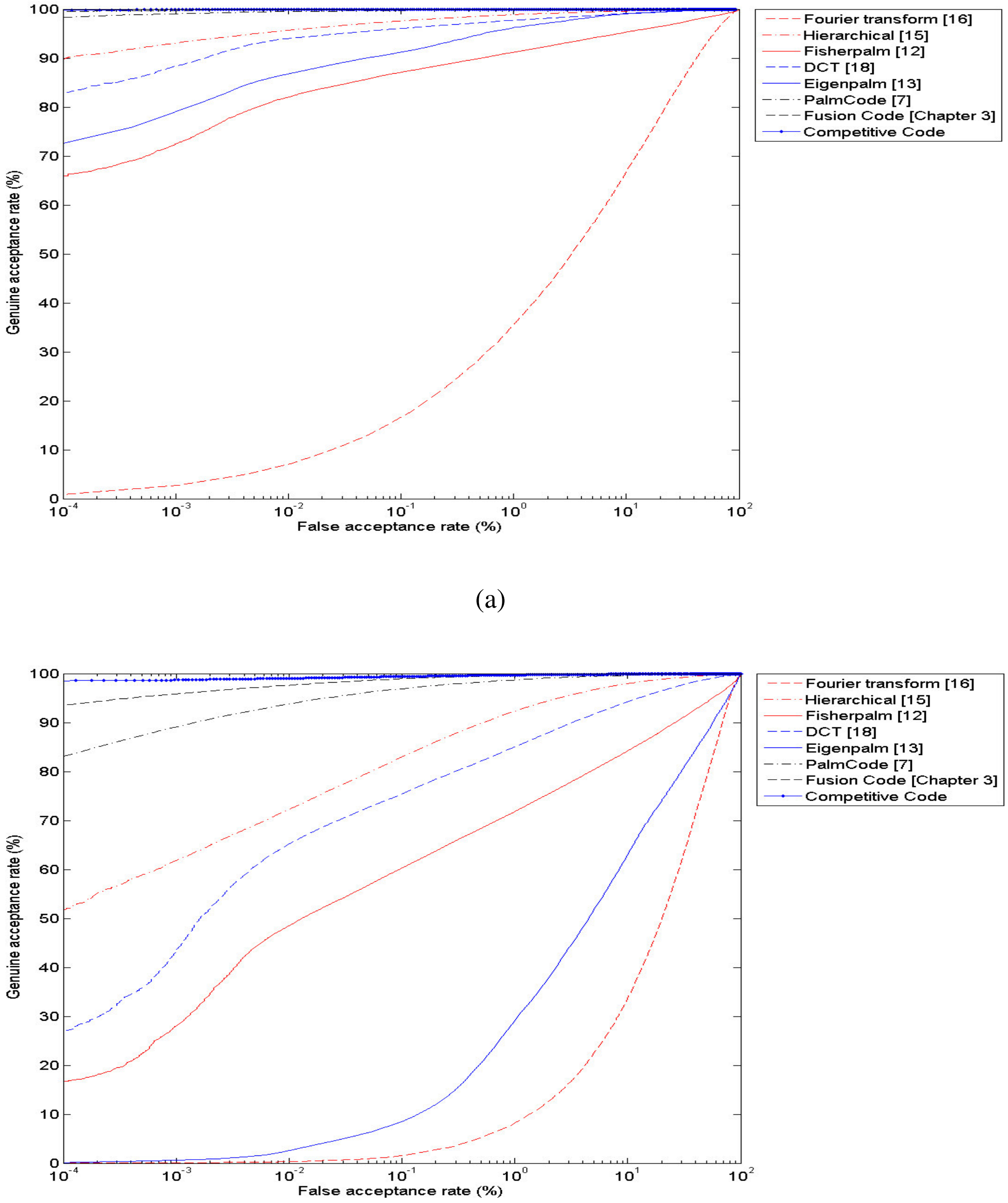
Even though Competitive Code is a coarse representation of local orientation, completely ignoring the magnitude information, it still performs better than other methods. Competitive Code and all the coding methods designed by me exploit the position information. However, the subspace methods depend only on the training images. They have not taken advances from this prior information. Furthermore, the subspace methods require estimating a huge covariance matrix. For images of size 64 by 64, the size of covariance matrix is 4096 by 4096. The number of parameters that need to be estimated is 8,390,656, much more than the images in the database. The accuracy of this estimation is questionable. Although the use of phase in PalmCode and Fusion Code is powerful for iris recognition [25], phase is not as discriminative as orientation for palmprints. In addition, the size of Competitive Code is 384 bytes while both the sizes of PalmCode and Fusion Code are 256 bytes. Thus, Competitive Code can store more discriminative information. The method described in [16] extracts the features from the frequency domain, completely ignoring the position information. Furthermore, the dimension of its feature vector is very low.

4.4.3 Speed

The computation time of the major component of the method is given in Table 4.4. These times are estimated using an ASUS notebook embedded Intel Pentium III Mobile processor (933MHz). Using bitwise angular distance, 38,500 comparisons per second can be made. On a 3-GHz computer, 100,000 comparisons per second can be made. It should be noted that I do not completely optimize the program code for preprocessing and feature extraction. It is possible to further improve the computation speed.

Table 4.4 Execution time of a system using Competitive Code

Operations	Time
Preprocessing	267ms
Feature extraction and Coding	178 ms
Angular matching	26 µs



(a)

(b)

Figure 4.3 Comparisons of the proposed method and other palmprint verification approaches. (a) and (b) are the results of matching palmprints from the same and different occasions, respectively. (color figure)

4.4.4 Comparison with Bitwise Angular Distance and Integer Angular Distance

To demonstrate the necessity of using XOR, the integer and bitwise angular distances are compared. The integer angular distance is $A_f(P,Q)=\sum_{x=1}^{N}\sum_{y=1}^{N}A(P(x,y),Q(x,y))$ where P and Q are two Competitive Codes and *A(P(x, y), Q(x, y))* is the angular distance between two winning indexes. In this experiment, a look up table is used to implement *A(P(x, y),Q(x, y))*. The bitwise angular distance, $A_f(P,Q)=\sum_{x=1}^{N}\sum_{y=1}^{N}\sum_{i=1}^{3}P_i^b(x,y)\otimes Q_i^b(x,y)$ is given Eq. 4.5. In this experiment, the masks and translated matching are not considered. The testing environment is a COMPAQ, Presario R3000 laptop embedded Mobile Inter Pentium 4 processor with 2.4GHz. Visual C++ is used to implement all the programming code. Bitwise angular distance needs only 0.7µs for one comparison while integer angular distance needs 10.6µs. Bitwise angular distance is 1500% faster than integer angular distance. The experimental result demonstrates the necessity of the bitwise representation.

4.5 Conclusion

A novel feature extraction method called Competitive Code for palmprint identification is presented in this chapter. This method extracts the orientation information from the palmprints and stores it in a feature code. Angular distance with an effective implementation is developed for comparing two feature codes. Using an ASUS laptop embedded Intel Pentium III Mobile processor (933MHz) and the bitwise representation, angular matching can make over 38,500 comparisons in about 1s. Total execution time for verification is about 0.5s, which is fast enough for real-time applications. In addition to matching speed, Competitive Code has been compared with other palmprint recognition algorithms including PalmCode and Fusion Code, described in the previous chapters. This comparison demonstrates that Competitive Code performs better than the others.

Chapter 5 An Analysis of IrisCode

IrisCode is an iris recognition algorithm developed in 1993 and continuously improved by Daugman. It has been extensively applied to commercial iris recognition systems. IrisCode representing an iris based on coarse phase has a number of properties including rapid matching, binomial imposter distribution and a predictable false acceptance rate. Because of its successful applications and these properties, many similar coding methods have been developed for iris and palmprint identification as described in the previous chapters. However, we lack a detailed analysis of IrisCode. The aim of this chapter is to provide such an analysis as a way of better understanding IrisCode, extending the coarse phase representation to a precise phase representation, and uncovering the relationship between IrisCode and other coding methods. This analysis demonstrates that IrisCode is a clustering process with four prototypes; the locus of a Gabor function is a two-dimensional ellipse with respect to a phase parameter and the bitwise hamming distance can be regarded as a bitwise angular distance. In this analysis, I also point out that the theoretical evidence of the imposter binomial distribution of IrisCode is incomplete and use this analysis to develop a precise phase representation which can enhance accuracy. I relate IrisCode and other coding methods and discuss the issues of making fair comparisons between IrisCode and other iris recognition methods.

5.1 Introduction

Various biometric systems have been developed for governmental and commercial applications. Most of these systems can verify, 1-to-1 match or identify a person in a small database, 1-to-many match. Real time large-scale identification is still a challenging problem in terms of matching speed and accuracy. Of existing biometric technologies, IrisCode developed in 1993 and continuously improved by Daugman [25, 86] is able to identify a person in an extremely large database in real time and its false acceptance rate is always zero [106-107]. As a result, it has been extensively deployed in commercial iris recognition systems for various security applications. Following the key idea of IrisCode, researchers have developed different coding methods for iris and palmprint recognition [4, 7, 56, 108-116].

Firstly a brief computational summary is given for those who are not familiar with IrisCode. Two dimensional Gabor filters with zero DC are applied to an iris image in dimensionless polar coordinate system, $I(\rho, \phi)$. Using the following inequalities, the complex Gabor response is encoded into two bits.

$$h_{\mathrm{Re}} = 1 \quad if \quad \mathrm{Re}\left(\int_\rho \int_\phi I(\rho,\phi) e^{-(r_0-\rho)^2/\alpha^2} e^{-(\theta_o-\phi)^2/\beta^2} e^{-i\omega(\theta_0-\phi)} \rho d\rho d\phi \right) \geq 0, \tag{5.1}$$

$$h_{\mathrm{Re}} = 0 \quad if \quad \mathrm{Re}\left(\int_\rho \int_\phi I(\rho,\phi) e^{-(r_0-\rho)^2/\alpha^2} e^{-(\theta_o-\phi)^2/\beta^2} e^{-i\omega(\theta_0-\phi)} \rho d\rho d\phi \right) < 0, \tag{5.2}$$

$$h_{\mathrm{Im}} = 1 \quad if \quad \mathrm{Im}\left(\int_{\rho}\int_{\phi} I(\rho,\phi) e^{-(r_0-\rho)^2/\alpha^2} e^{-(\theta_o-\phi)^2/\beta^2} e^{-i\omega(\theta_o-\phi)} \rho d\rho d\phi \right) \geq 0, \tag{5.3}$$

$$h_{\mathrm{Im}} = 0 \quad if \quad \mathrm{Im}\left(\int_{\rho}\int_{\phi} I(\rho,\phi) e^{-(r_0-\rho)^2/\alpha^2} e^{-(\theta_o-\phi)^2/\beta^2} e^{-i\omega(\theta_0-\phi)} \rho d\rho d\phi \right) < 0, \tag{5.4}$$

where r_0, θ_0, ω, α and β are the parameters of the Gabor filters [86]. Bitwise hamming distance is used to measure the difference between two IrisCodes. The first version of bitwise hamming distance is defined as $HD = \sum_{i=1}^{2048} (A_i \otimes B_i)/2048$, where A_i and B_i are the bits in two IrisCodes; $\otimes$ represents bitwise operator, XOR. The current IrisCode uses a mask to denote the corrupted bits from eyelashes, reflection, eyelids and low signal-to-noise ratio [86]. The hamming distance between two IrisCodes is redefined as

$$HD = \frac{\sum_{i=1}^{2048} ((A_i \otimes B_i) \cap (A_i^M \cap B_i^M))}{\sum_{i=1}^{2048} (A_i^M \cap B_i^M)}, \tag{5.5}$$

where A^M and B^M are the masks of IrisCodes, A and B, respectively and $\cap$ represents bitwise operator AND.

It is noted that IrisCode has a number of desirable properties. It is robust against local brightness and contrast variations because of the zero DC Gabor filters and the coding scheme. In rotating between any adjacent phase quadrants, only a single bit in IrisCode changes. It can enhance the robustness of the genuine distribution. In rotating between one phase quadrant to the opposite phase quadrant, both two bits in IrisCode change. In other words, the distances between phase quadrants are retained. This representation is referred to cyclic representation. One of the well-known properties of IrisCode is the [2]binomial imposter distribution with high degrees-of-freedom. Making use of this property, the decision threshold is dynamically changed according to a predictable false acceptance rate from the binomial imposter distribution. This is the reason why the false match rate of IrisCode is always zero [107]. Based on the binomial imposter distribution, some researchers have developed an iris individuality model [123]. The bitwise hamming distance is the key to high speed matching. IrisCode can perform 1 million comparisons per second using a computer with a 3G Hz processor.

It is generally believed that the cores of IrisCode are the operators, "≥" and "<" in the Eqs. 5.1-5.4 and the bitwise hamming distance. Using these two operators, each feature value is represented by one bit and two encoded features are compared by the bitwise hamming. Researchers replace the Gabor filters in IrisCode with different filters and transformations including [3]quadratic spline wavelet, Haar wavelet frame, log Gabor filters, independent component analysis, directional filter banks and dissociated tripole filters [108-116] to develop new coding methods for iris recognition. Based on this understanding, some

[2] The binomial imposter distribution with high degrees of freedom does not mean high accuracy [128].

[3] The coding scheme used in [108] does not involve the two operators, "≥" and "<". In fact, it can be rewritten based on these two operators. Readers can refer to appendix 1.

researchers claim that IrisCode is a local ordinal feature [109]. Some researchers further believe that the imposter distribution of their coding method also follows binomial distribution [117, 164]. However, this understanding of IrisCode is incomplete and limits the design of new coding schemes for feature representation. A weakness of this understanding is that each filter response or coefficient provides only one bit of information. It lacks representational flexibility. Furthermore, some claims are controversial. Developing an iris individuality model with solid theoretical foundation also requires a complete understanding [123].

A detailed analysis of IrisCode is important for understanding IrisCode, for designing new coding schemes, and for clarifying the relationship between IrisCode and other coding methods. Nevertheless, such an analysis has not been found in the literature. In this chapter, I aim to investigate the relationship between IrisCode and clustering processes, the property of the Gabor function and the relationship between the bitwise hamming distance and bitwise angular distance. The theoretical foundation of the binomial imposter distribution is also discussed. Making use of this analysis, an algorithm for precise phase representation with effective filtering and matching is developed. Finally, I study the relationship between IrisCode and other coding methods and offer a note on the comparison of IrisCode.

The rest of this chapter is organized as follows. Section 5.2 uncovers the properties of IrisCode for understanding. Section 5.3 presents an algorithm for precise phase representation. Section 5.4 discusses the theoretical evidence of the binomial imposter distribution. Section 5.5 links up the relationship between IrisCode and different coding methods. Section 5.6 mentions some issues about the comparison of IrisCode. Section 5.7 offers some concluding remarks.

5.2 Understanding IrisCode from Clustering Point of View.

In this section, I demonstrate that IrisCode is a clustering process and study the properties of the Gabor function. The relationship between bitwise hamming distance and bitwise angular distance is presented in Section 5.3.

5.2.1 IrisCode — A Clustering Process

Let $M_R(\rho,\ \phi)$ and $M_I(\rho,\ \phi)$ be the real and imaginary parts of a Gabor filter. For convenience, let M_R be $M_R(\rho,\ \phi)$. The same notations are used for other symbols. The definitions of M_R and M_I are given below:

$$M_R(\rho,\phi) = e^{-(r_0-\rho)^2/\alpha^2} e^{-(\theta_o-\phi)^2/\beta^2} \left(\cos(-\omega(\theta_0-\phi))\right), \tag{5.6}$$

$$M_I(\rho,\phi) = e^{-(r_0-\rho)^2/\alpha^2} e^{-(\theta_o-\phi)^2/\beta^2} \left(\sin(-\omega(\theta_0-\phi))\right). \tag{5.7}$$

DC term of Gabor filter in Eq. 5.6 is not removed since in this chapter, it is assumed that the DC of iris patch for filtering has been removed.

A continuous periodic function,

$$Z(\varphi) = (\cos(\varphi)M_R + \sin(\varphi)M_I), \tag{5.8}$$

with respect to the parameter, φ, where $\varphi\in[0, 2\pi)$ is defined. $Z(\varphi)$ is called filter-generating function. Using $Z(\varphi)$, four filters by substituting 5π/4, 7π/4, π/4, and 3π/4 to φ can be obtained. The four filters are

$$Z_0 = Z(5\pi/4) = (-M_R - M_I)/\sqrt{2}\,, \tag{5.9}$$

$$Z_1 = Z(7\pi/4) = (M_R - M_I)/\sqrt{2}\,, \tag{5.10}$$

$$Z_2 = Z(\pi/4) = (M_R + M_I)/\sqrt{2}\,, \tag{5.11}$$

$$Z_3 = Z(3\pi/4) = (-M_R + M_I)/\sqrt{2}\,. \tag{5.12}$$

These four filters are shown in Figure 5.1 and will later be regarded as the cluster centers. I define $j = \arg\max_i \int_\rho\int_\phi \rho I Z_i d\rho d\phi / \|\rho I\| \|Z_i\|$ as a clustering criterion, where j is called the winning index and I is the iris image in the dimensionless polar coordinate system. The clustering criterion is the cosine measure between Z_i and ρI. It can be rewritten as

$$j = \arg\max_i \left(\int_\rho\int_\phi \rho I Z_i d\rho d\phi \right), \tag{5.13}$$

since the four filters have the same power, i.e., $\int_\rho\int_\phi Z_i^2 d\rho d\phi = C$, where C is a constant, which can be proved by using the orthogonal property between M_R and M_I i.e., $\int_\rho\int_\phi M_I M_R d\rho d\phi = 0$. The winning index is an integer representation of φ. For fast matching, the winning index has to be encoded. Table 5.1 gives the coding table under the heading "Coded winning indexes". The difference between two encoded winning indexes is also measured by their bitwise hamming distance same as IrisCode. Table 5.1 compares bits of IrisCode and the binary representation of the winning indexes, demonstrating their equivalence. In the other words, IrisCode is a clustering process and the cosine measure is the clustering criterion.

Table 5.1 Comparison of IrisCode, winning index and coded winning index

IrisCode		Winning index	Coded winning indexes	
h_{Im}	h_{Re}		Bit 2	Bit 1
0	0	0	0	0
0	1	1	0	1
1	1	2	1	1
1	0	3	1	0

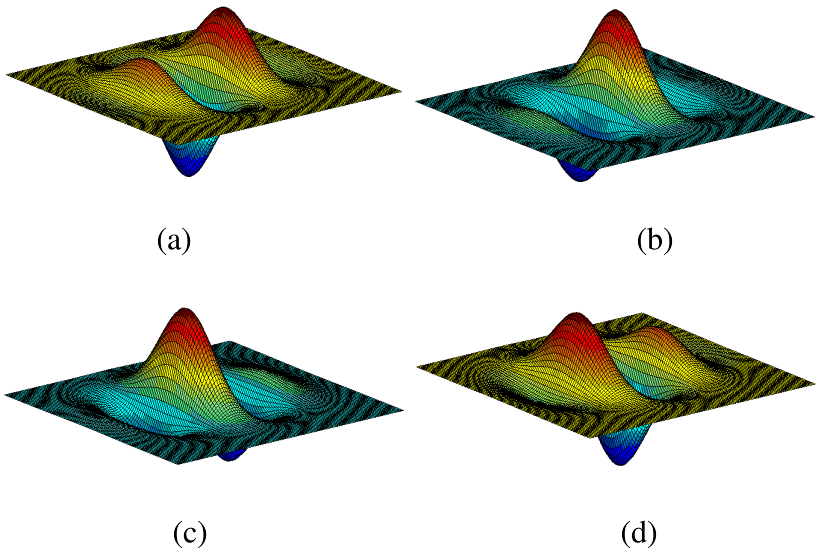


Figure 5.1 The four filters used in IrisCode, a) Z_0, b) Z_1, c) Z_2 and d) Z_3.

5.2.2 Properties of the Gabor Function

Let us study the physical meaning of φ in the filter generating function. Reordering the terms in Eq. 5.8, $Z(\varphi)$ can be rewritten as

$$Z(\varphi) = e^{-(r_0-\rho)^2/\sigma^2} e^{-(\theta_0-\phi)^2/\beta^2} \cos(-\omega(\theta_0 - \phi) - \varphi)\,. \tag{5.14}$$

It is clear that φ is the phase of a Gabor function and the filter generating function can be rewritten as a Gabor function. According to Eqs. 5.9-5.12 and 5.14, it is very clear that IrisCode is a periodic feature.

Now, we study the locus of the filter generating function, also the Gabor function in Eq. 5.14 with respect to the phase parameter. Discretizing M_R, M_I and Z, three vectors, $\vec{M}_R$, $\vec{M}_I$ and $\vec{Z}(\varphi)$ can be obtained respectively and two orthonormal vectors, $\vec{v}_R = \vec{M}_R / \|\vec{M}_R\|$ and $\vec{v}_I = \vec{M}_I / \|\vec{M}_I\|$ are defined. Using these two vectors, $\vec{Z}(\varphi)$ can be rewritten as

$$\vec{Z}(\varphi) = (\cos(\varphi)\|\vec{M}_R\|\vec{v}_R + \sin(\varphi)\|\vec{M}_I\|\vec{v}_I)\,, \tag{5.15}$$

a linear combination of $\vec{v}_R$ and $\vec{v}_I$. The coordinate of $\vec{Z}(\varphi)$ in the two dimensional space spanned by $\vec{v}_R$ and $\vec{v}_I$ is $(\|M_R\|\cos(\varphi), \|M_I\|\sin(\varphi))$ satisfying the following equality

$$\frac{(\|M_R\|\cos(\varphi))^2}{\|M_R\|^2} + \frac{(\|M_I\|\sin(\varphi))^2}{\|M_I\|^2} = 1. \tag{5.16}$$

Obviously, the locus of $\vec{Z}(\varphi)$ with respect to φ is an ellipse on the two dimensional space. Under suitable parameterization, the locus can be further constraint to a circle. Appendix 2 gives the mathematical condition.

5.3 Precise Phase Representation

To design a precise phase representation, I first take more sample points from the filter generating function to generate more prototypes for the clustering process. Uniform sampling is used to obtain the filters, i.e., $Z(i\pi / n + \eta)$, where i=0, 1,…2n-1 and η is an offset. The number of prototypes being 2n is for the design of bitwise angular distance, where n is called the order of coding scheme. For IrisCode, n is 2 and η is 5π/4. It should be noted that the clustering criterion is independent of the contrast of the image. The brightness of the iris is normalized. To embed the other inherent properties of IrisCode such as rapid matching and cyclic representation, I design a novel coding scheme to encode the winning indexes and a distance measure for rapid matching. In the following discussion, the offset η is ignored without loss of generality.

5.3.1 Angular Distance

Since $\vec{Z}(\varphi)$ is on a two dimensional ellipse and φ is the phase, the distance between $\vec{Z}(\omega)$ and $\vec{Z}(\gamma)$ can be measured by the angular distance between ω and γ defined as $\min(|\omega-\gamma|, 2\pi-|\omega-\gamma|)$. If uniform sampling is used to obtain $\vec{Z}(\omega)$ and $\vec{Z}(\gamma)$ *i.e.,* $\omega = 2\pi p/2n$ and $\gamma = 2\pi q/2n$ where *p* and *q* are two integers between 0 and 2*n*-1, the angular distance can be rewritten as $\min\left(\frac{\pi}{n}|p-q|, \frac{\pi}{n}(2n-|p-q|)\right)$. The angular distance can be further simplified as

$$\min(|p-q|, (2n-|p-q|)), \tag{5.17}$$

if π/n is defined as one unit distance. As a result, the angular distance between $\vec{Z}(\omega)$ and $\vec{Z}(\gamma)$ is defined only based on their winning indexes, *p* and *q*, the integer presentation of their phases. The angular distance between any two adjacent winning indexes is 1, as in IrisCode. The cyclic representation has been embedded in the precise phase representation.

5.3.2 Bitwise Matching and Coding Scheme

The bitwise hamming distance supporting high speed matching is one of the keys to large-scale iris identification in real time. However, the winning index and angular distance in Eq. 5.17 are integer representation. To embed high speed matching to the precise phase representation, I design a new coding scheme to encode the integer winning indexes and develop a bitwise angular distance. A coding table $A = [a_{i,j}]$ is designed for this purpose, where $1 \le i \le n$; $1 \le j \le 2n$ and $a_{i,j}$ defined as in Figure 5.2.

```
if    $j \le n$    and    $1 \le i < j$,
          $a_{i,j} = 1$
elseif    $j > n$    and    $j - n \le i \le n$,
          $a_{i,j} = 1$
else
          $a_{i,j} = 0$
```

Figure 5.2 Pseudo code of the coding table

For illustration, two coding tables for *n*=3 and 4 are given in Table 5.2. We can observe a structure in these coding tables. Each winning index is represented by the column of *A* with the result that *n* bits are used to encode one winning index.

It should be noted that when *n*=1 and *n*=2, the bitwise winning indexes form a [4]cyclic code [132] and a gray code. However, when *n* is greater than two, the bitwise winning indexes do not form a cyclic code since cyclically shifting a bitwise winning index can generate a code that does not exist in the coding table. For instance, (1 0 0) is a code in Table 5.2a but (0 1 0) does not exist in this table. They also do not form a gray code since gray code uses *n* bits two to represent 2^n integers. For example, (1 0 1) is a code in gray code but it does not exist in Table 5.2a. Cyclic code and gray code are not essential properties for precise phase representation since cyclic representation is embedded in it.

To achieve high speed matching, a bitwise matching for the encoded winning indexes is needed. I discover that angular distance and bitwise hamming distance have an equivalent relationship, i.e. $\sum_{i=1}^{n} a_{i,j} \otimes a_{i,j+k} = \min(k, 2n-k)$. The mathematical proof is given in Appendix 3. This bitwise hamming distance for precise phase representation is referred to as the bitwise angular distance since it measures the angle between two prototypes in the two dimensional ellipse.

Table 5.2 The coding tables for (a) *n*=3 and (b) *n*=4

(a)

Winning index	0	1	2	3	4	5
Bit 0	0	1	1	1	0	0
Bit 1	0	0	1	1	1	0
Bit 2	0	0	0	1	1	1

(b)

Winning index	0	1	2	3	4	5	6	7
Bit 0	0	1	1	1	1	0	0	0
Bit 1	0	0	1	1	1	1	0	0
Bit 2	0	0	0	1	1	1	1	0
Bit 3	0	0	0	0	1	1	1	1

[4] The definitions of cyclic code [132] and cyclic representation are different.

5.3.3 Steerable Filtering

An algorithm for precise phase representation that embeds the properties of IrisCode including fast matching and cyclic representation has been developed. However, the number of filtering increases with respect to the precision of the phase. 2*n* filtering is required if the precision of phase is *n* bits. When *n* is large, the computation cost increases dramatically. To alleviate this problem, an effective filtering approach is needed. Let us consider the clustering criterion, $j = \arg\max_i \int_\rho \int_\phi \rho I Z(i\pi / n) d\rho d\phi / \|\rho I\| \|Z(i\pi / n)\|$, where $\|\rho I(\rho, \phi)\|$ is independent of i and $\|Z(i\pi / n)\|$ can be pre-computed. These two terms would not dramatically increase the computational burden even when *n* is large. The major computation cost comes from $\int_\rho \int_\phi \rho I Z(i\pi / n) d\rho d\phi$.

Substituting Eq. 5.8 into $\int_\rho \int_\phi \rho I Z(i\pi / n) d\rho d\phi$, we have

$$\int_\rho \int_\phi \rho I (\cos(i\pi / n) M_R + \sin(i\pi / n) M_I) d\rho d\phi$$

$$= \cos(i\pi / n) \int_\rho \int_\phi \rho I M_R d\rho d\phi + \sin(i\pi / n) \int_\rho \int_\phi \rho I M_I d\rho d\phi \qquad (5.18)$$

Eq. 5.18 shows that $Z(\varphi)$ is a steerable filter [129] and only two filters, M_R and M_I are needed for any precision of the phase. The computation complexity of filtering has been successfully reduced from *O*(*n*) to *O*(1). If the locus of $\vec{Z}(\varphi)$ is a circle, the solution phase, $\arg\max_\varphi \int_\rho \int_\phi \rho I Z(\varphi) d\rho d\phi / \|\rho I\| \|Z(\varphi)\|$ can be computed by $\varphi = \tan^{-1}\left(\int_\rho \int_\phi \rho I M_I d\rho d\phi \Big/ \int_\rho \int_\phi \rho I M_R d\rho d\phi \right)$. The mathematical proof is given in Appendix 4.

5.3.4 Re-implementation of IrisCode.

To examine the precise phase representation, I re-implement IrisCode including pupil, limbus and eyelid detection, eyelash segmentation, normalization, coding and matching for comparison. Accurately re-implementing IrisCode is highly difficult since it is a complex computer vision system. Moreover, some parts have not been disclosed clearly e.g. the computational details of eyelash segmentation. In my re-implementation, I have made number of modifications of the preprocessing since the [5]CASIA iris database is used [118], which has different characteristics from Daugman's iris databases. This issue will be discussed in Section 5.6. The CASIA iris database is selected for this study because it is the most widely employed public iris database.

[5] I would like to thank CASIA for sharing the iris database for this research.

5.3.4.1 Iris Segmentation

In the re-implementation, the pupil location based on its gray levels is firstly estimated. To localize a pupil boundary, the integro-differential operator, $\max_{(r,x_o,y_o)}\left|G_\sigma(r)*\frac{\partial}{\partial r}\oint_{r,x_o,y_o}\frac{I(x,y)}{2\pi r}ds\right|$ reported in Daugman's publications [25] is then applied. This boundary is used to initialize an active contour that can be used to accurately estimate the pupil boundary. The center of the mass of the final contour is regarded as the center of the pupil for normalization. All the current commercial iris recognition systems using IrisCode have employed an active contour for pupil detection [119] but the implementations may not be the same.

After detecting the pupil, the integro-differential operator is used to detect upper and lower eyelids by replacing the circular path to a spline with four control points. The eyelashes are detected based on two simple criterions. A simple threshold is used to detect the eyelashes. If the distance between eyelashes detected by the first criterion and upper eyelid is shorter than a certain threshold, all the pixels between them are regarded as eyelashes [120].

To make use of the detected eyelashes and eyelids for limbus localization, the integro-differential operator is modified as follows:

$$\max_{(r,x_o,y_o)}\left|G_\sigma(r)*\frac{\partial}{\partial r}\oint_{S_L(r,x_o,y_o)}\frac{I(x,y)}{L(S_L)}ds+G_\sigma(r)*\frac{\partial}{\partial r}\oint_{S_R(r,x_o,y_o)}\frac{I(x,y)}{L(S_R)}ds\right|. \tag{5.19}$$

where, S_L and S_R are the parts of left and right circular paths controlled by r, x_0 and y_0 and $L(S_L)$ and $L(S_R)$ are the lengths of S_L and S_R, respectively. Any pixels above the upper eyelid, below the lower eyelid or corrupted by eyelashes are not included in these two paths. Therefore, the lengths of S_L and S_R can be different. The term $2\pi r$ in the original integro-differential operator is replaced with $L(S_L)$ and $L(S_R)$. Since the boundary of the limbus is weak, the parameters in Eq. 5.19 are limited by the parameters of pupil. Figure 5.3 shows a segmented iris.

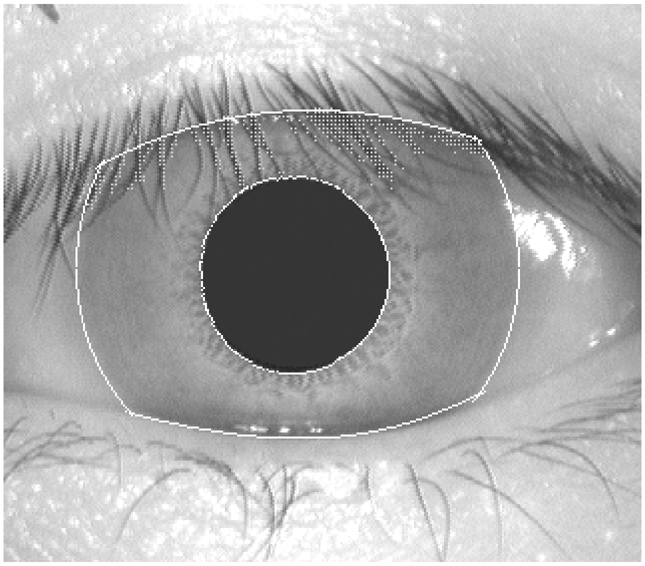


Figure 5.3 A segmented iris

5.3.4.2 Normalization and Filtering

After segmenting an iris, the dimensionless polar coordinate system is used to normalize an iris. Many researchers would display the normalized iris as a rectangular sheet. Strictly speaking, the normalized iris is the surface of a cylinder and it has only two cuts from the pupil and limbus boundaries. The size of a normalized iris is 64 by 512 pixels.

To each sample point, two Gabor filters are applied. In total, sixteen Gabor filters with different parameters, determined by 140 different irises from twenty persons, are employed. In parameter training, the Gabor filters have two sizes, 8 by 105 and 24 by 105 pixels. The smaller one is applied to the sample points close to pupil and limbus boundaries. The angular sampling frequency is 8 pixels. The *d'* index defined as $d' = \left|\mu_g - \mu_i\right| / \sqrt{(\sigma_g^2 + \sigma_i^2)/2}$, where $\mu_{g(i)}$ and $\sigma_{g(i)}^2$ are the mean and variance of a genuine (imposter) distribution, is employed as the objective function for training. The parameters of each filter are selected from 13,468 sets of parameters. In the training, one set of parameters for the Gabor filter closest to the pupil boundary is selected first. Then, I keep the first optimized parameters and select another set of parameters for the same sample points. The other sets of parameters are selected in the same way. For precise phase representation, the parameters trained for IrisCode are used.

5.3.4.3 Matching

A raw hamming distance (HD_{raw}) is computed from Eq. 5.5. The number of effective bit matchings is different in each comparison because of eyelashes and eyelids. To obtain the same decision confidence, Daugman rescales the hamming distance with the following equation,

$$HD_{norm} = 0.5 - (0.5 - HD_{raw})\sqrt{m/911}\,, \tag{5.20}$$

where m is the number of actually compared bits and 911 is the average number of actually compared bits [106]. In the CASIA iris database, the average number of actually compared bits is 894 from a preliminary training so 911 in Eq. 5.20 is replaced with 894 for training and testing. For precise phase presentation, normalized hamming distance is defined as

$$HD_{norm} = 0.5 - (0.5 - HD_{raw})\sqrt{(m/(n894/2))}\,, \tag{5.21}$$

where, n is the order of the coding scheme.

5.3.5 Experimental Results

CASIA iris database contains 756 images from 108 irises. 140 images have been used to train the parameters. The remainders are used for the following testing. In the following experiments, *d*' index and Receiver Operating Characteristic (ROC) curves are used as performance indexes.

5.3.5.1 Validation of Precise Phase Representation

To validate the precise phase representation, the following experiment is designed. When n=2, the performance of precise phase representation and IrisCode should be the same. Figure 5.4 shows the *d*' indexes of the two approaches, where x-axis represents number of Gabor filters used to compute the *d*' indexes. First two Gabor filters are applied to the region close to pupil boundaries. The last two Gabor filters are applied to the region close to limbus boundaries. From this figure, we see that the performance of the two approaches is nearly identical. The slight difference between them is due to the different approaches to the normalization of brightness variations. We also observe that the performance improves when number of Gabor filters is between 1 and 8 but the performance degrades when the number of Gabor filters is between 9 and 16. The best performance is achieved when IrisCode is computed from the half of the normalized iris which is close to the pupil boundary. The performance degradation is due to poor limbus detection. The two iris images in Figure 5.5 show that eyelashes can significantly affect the accuracy of limbus detection. Although some processes are designed for detecting eyelashes, they are not perfect. Furthermore, the contrast of eyelashes is much stronger than that of the limbus. This is another reason why the integro-differential operator cannot provide good performance in this database. Therefore, to obviate the contrast variation, some researchers use the Hough transform to detect the limbus [108]. In the following experiments, the first eight Gabor filters are used to compare IrisCode and precise phase representation since they provide the best results for IrisCode.

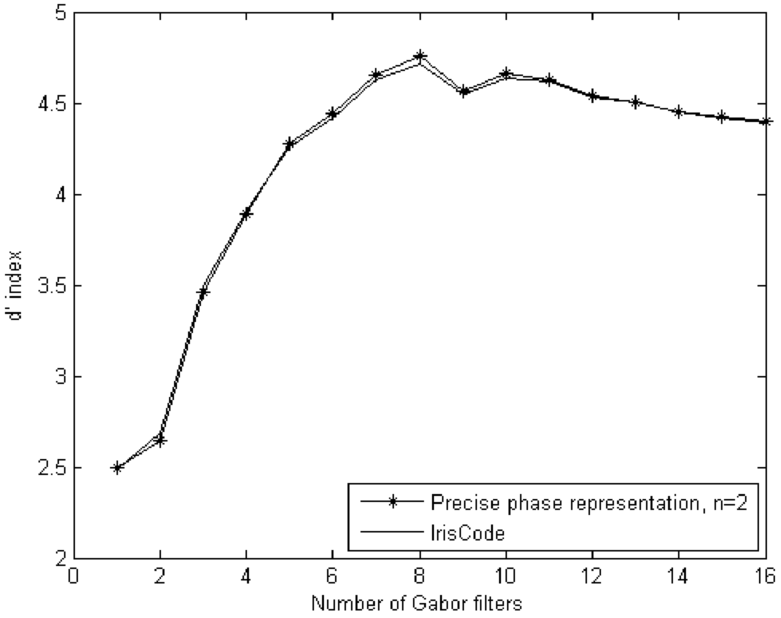


Figure 5.4 Validation of precise phase representation

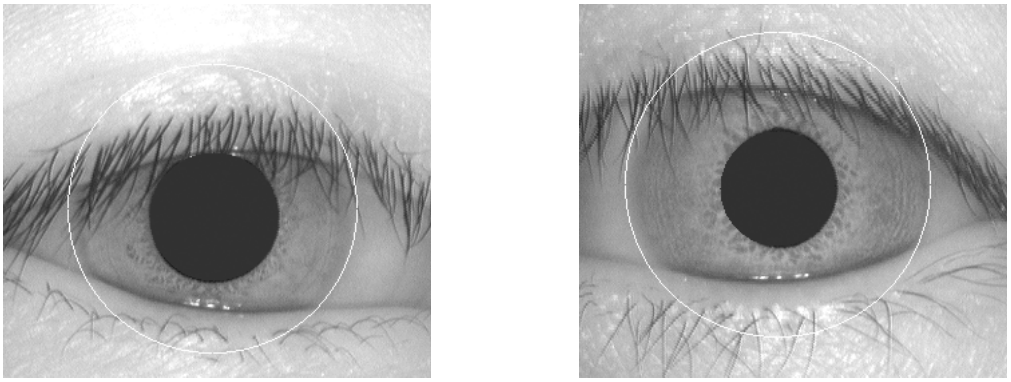
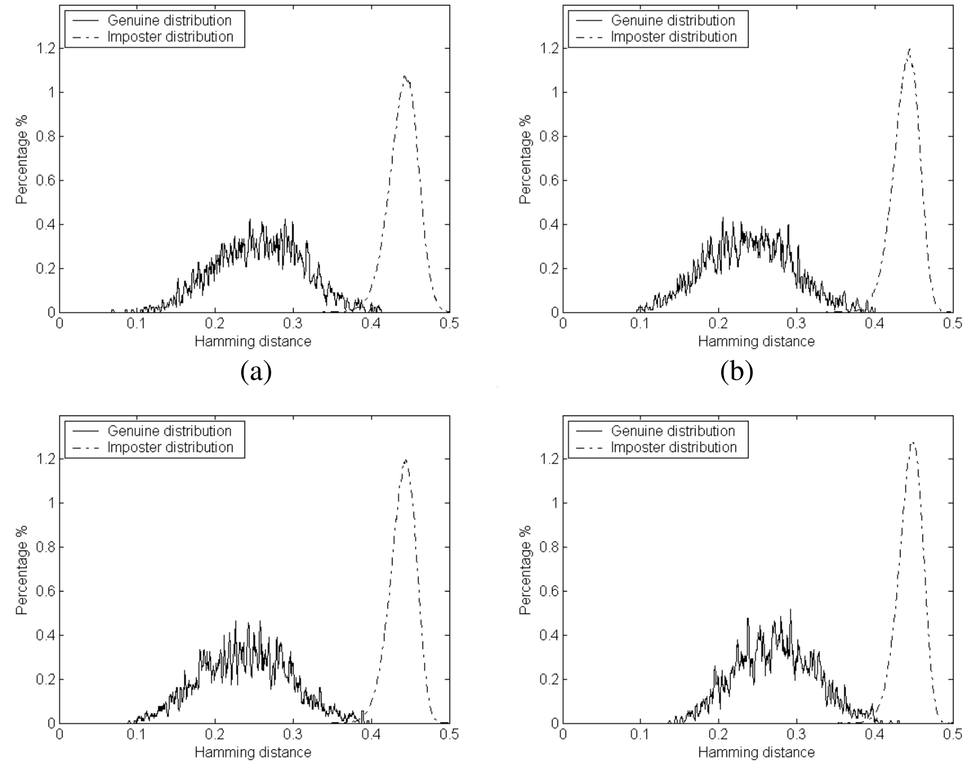


Figure 5.5 Poor limbus detection

5.3.5.2 Comparison of IrisCode and Precise Phase Representation

To examine the effectiveness of precise phase representation, IrisCode is compared with precise phase representation for *n*=3, 4, and 5. I match irises from the same persons and from different persons to respectively obtain 1,848 genuine matchings and 187,572 imposter matchings for each representation. The genuine and imposter matchings are used to estimate the genuine and imposter distributions. Figure 5.6(a)-(b) respectively show the genuine and imposter distributions of IrisCode and precise phase representation for n=3, 4 and 5. For comparison, Figure 5.7 depicts the corresponding ROC curves. As can be seen in Figure 5.7, the precise phase representation of order 4 always provides the most accurate performance. Its equal error rate is 0.32%. Comparing IrisCode and precise phase representation of order 4, precise phase representation of order 4 has 3.1% improvement in genuine acceptance rate when the false acceptance rate is 5.3×10^{-4}%. However, increasing the precision of the phase does not always improve the accuracy. It is also the case in Eigenface that increasing the number of principal components does not always increase accuracy. It should be noted that the matching speed of precise phase representation is slower than IrisCode. Roughly speaking, the matching speed of precise phase representation of order 4 is half of that of IrisCode since IrisCode uses only two bits to represent one filter response while precise phase representation of order 4 requires four bits to represent one filter response. For some applications such as identifying a person in a residence building for access control, one million comparisons per second is much more than enough. Precise phase representation can be used to achieve high accuracy for these applications. I do not claim that precise phase representation performs better than IrisCode. Nevertheless, it is a flexible representation to balance speed and accuracy.



(a)

(b)

(c)

(d)

Figure 5.6 Genuine and imposter distributions for (a) IrisCode, (b)-(d) precise phase representation for n=3-5, respectively

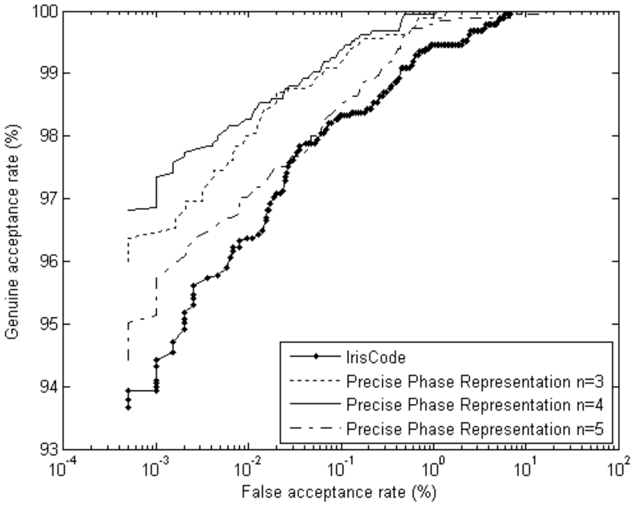


Figure 5.7 ROC curves of IrisCode and precise phase representation for *n*=3, 4 and 5.

5.4 Theoretical Evidence of Binomial Imposter Distribution

In the previous sections, I investigated a number of properties of IrisCode and used them to develop an algorithm for precise phase representation with effective filtering and rapid matching. However, I have not touched upon the most important property in IrisCode, the binomial imposter distribution. This is used to predicate the false acceptance rate under different thresholds and different matching condition such as variation of actually compared bits. The binomial imposter distribution of IrisCode has been experimentally validated on a large database [106]. Some may expect that the imposter distributions of the similar coding methods designed for iris and palmprint [4, 7, 56, 108-116] and the precise phase representation would also follow a binomial distribution since, as in IrisCode, all their feature values are binary and two feature codes are compared using a hamming distance.

In practice, the imposter distribution of IrisCode is binomial. I am interested in its theoretical foundation. Let us review the three assumptions of a binomial distribution. A random variable *X* following a binomial distribution should fulfill the following conditions.

1. X is defined as $\sum_{i=1} T_i$, where T_i is a Bernoulli variable.
2. All T_i has the same probabilities of success, *p*.
3. All T_i are independent.

(2) is referred to a stationary condition and (3) is referred to an independent condition. Obviously, the hamming distances of IrisCode and other coding methods satisfy condition (1). Daugman validated condition (2) in 1993 [25]. However, IrisCode violates the independent condition because of the inherent correlation between the texture features and the convolution of Gabor filters. If the sum of correlated and stationary Bernoulli variables followed a binomial distribution unconditionally, an imposter distribution

of IrisCode would have to be binomial. However, no such mathematical theorem has been discovered. Even if the correlation is of the first order Markovian type, the distribution of the sum of correlated stationary Bernoulli trails can be bimodal and trimodal shapes [121-122]. This shows that the theoretical foundation of the binomial imposter distribution is incomplete. As a result, it cannot be expected that the imposter distributions of the other coding methods and precise phase representation also follow binomial distributions. Clearly, it is very hard to develop a reasonable iris individuality model [123] if the binomial assumption is used.

Under some conditions and making use of the central limit theorem, the sum of correlated and stationary Bernoulli trails could be approximated by a normal distribution [90]. A normal distribution can be approximated by a binomial distribution when the number of Bernoulli trials is large. This may be the reason why the imposter distribution of IrisCode follows a binomial distribution. However, these conditions have not been validated.

5.5 The Relationship between IrisCode and Other Coding Methods.

Many coding methods have been developed for iris and palmprint identification that are quite similar [4, 7, 56, 108-116]. The most common approach is to replace the Gabor filters in IrisCode with other linear transforms or filters. According to this analysis, IrisCode is a clustering process with four prototypes. As it happens, most the other coding methods can also be regarded as a clustering process but with two prototypes. Let us formally define these approaches. Let F be a linear filter used in their coding methods. Their coding schemes can be summarized in the following equations,

$$h = 1 \quad if \quad \int_{\rho}\int_{\phi} FId\rho d\phi \geq 0, \tag{5.22}$$

$$h = 0 \quad if \quad \int_{\rho}\int_{\phi} FId\rho d\phi < 0, \tag{5.23}$$

where h is a resultant bit. This coding scheme is referred to a standard coding scheme. To uncover the relationship between IrisCode and the standard coding scheme, a filter generating function $(-1)^{\upsilon+1}F$, where $\nu \in \{0, 1\}$ is defined. This filter generating function can generate only two filters, F and –F. Since these two filters have the same power, i.e., $\|F\| = \|-F\|$, the clustering criterion can be rewritten as

$$j = \arg\max_{i}\left(\int_{\rho}\int_{\phi}(-1)^{i+1}FId\rho d\phi\right). \tag{5.24}$$

If j=0, then we have $\int_{\rho}\int_{\phi} -FId\rho d\phi > \int_{\rho}\int_{\phi} FId\rho d\phi$ and $0 > \int_{\rho}\int_{\phi} FId\rho d\phi$. If j=1, we have $\int_{\rho}\int_{\phi} FId\rho d\phi > \int_{\rho}\int_{\phi} - FId\rho d\phi$ and $\int_{\rho}\int_{\phi} FId\rho d\phi > 0$. Using the first order coding scheme defined in Figure 5.2 to encode the winning index j, we obtain Eqs. 5.22 and 5.23.

In addition to the standard coding scheme, other coding methods based on Gabor filters and log Gabor filters employ the order 2 coding scheme in precise phase representation [110, 113]. I also developed a Competitive Code for palmprint identification described in Chapter 4 [3]. Its filter generating function is the negative real part of a Gabor function. Different values are assigned to the orientation parameter so as to generate six filters and use order 3 coding scheme to encode the winning indexes. Bitwise angular distance is employed to measure two different Competitive Codes.

The standard coding method, IrisCode, and Competitive Code respectively employ the order 1, 2 and 3 coding schemes in precise phase representation. Their differences are in their filter generating functions. It is the fact that most of the existing coding methods including standard coding method, IrisCode, Competitive Code and precise phase representation are under the same framework given in Figure 5.8. The cores of this framework include filter generating function, clustering, coding scheme and bitwise angular matching. This framework is different from the previous frameworks [56, 109]. It is more detailed. Furthermore, it should be mentioned that both loci of the filter generating functions of the standard coding method and IrisCode are always on two-dimensional planes. However, the locus of the filter generating function of Competitive Code is on a higher dimensional plane.

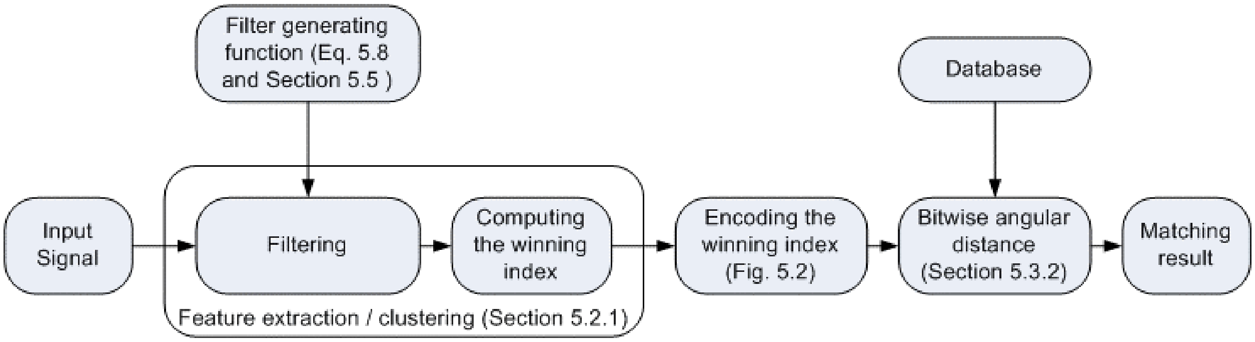

Filter generating function (Eq. 5.8 and Section 5.5)
Database
Input Signal
Filtering
Computing the winning index
Feature extraction / clustering (Section 5.2.1)
Encoding the winning index (Fig. 5.2)
Bitwise angular distance (Section 5.3.2)
Matching result

Figure 5.8 A common framework employed by most of the existing coding methods.

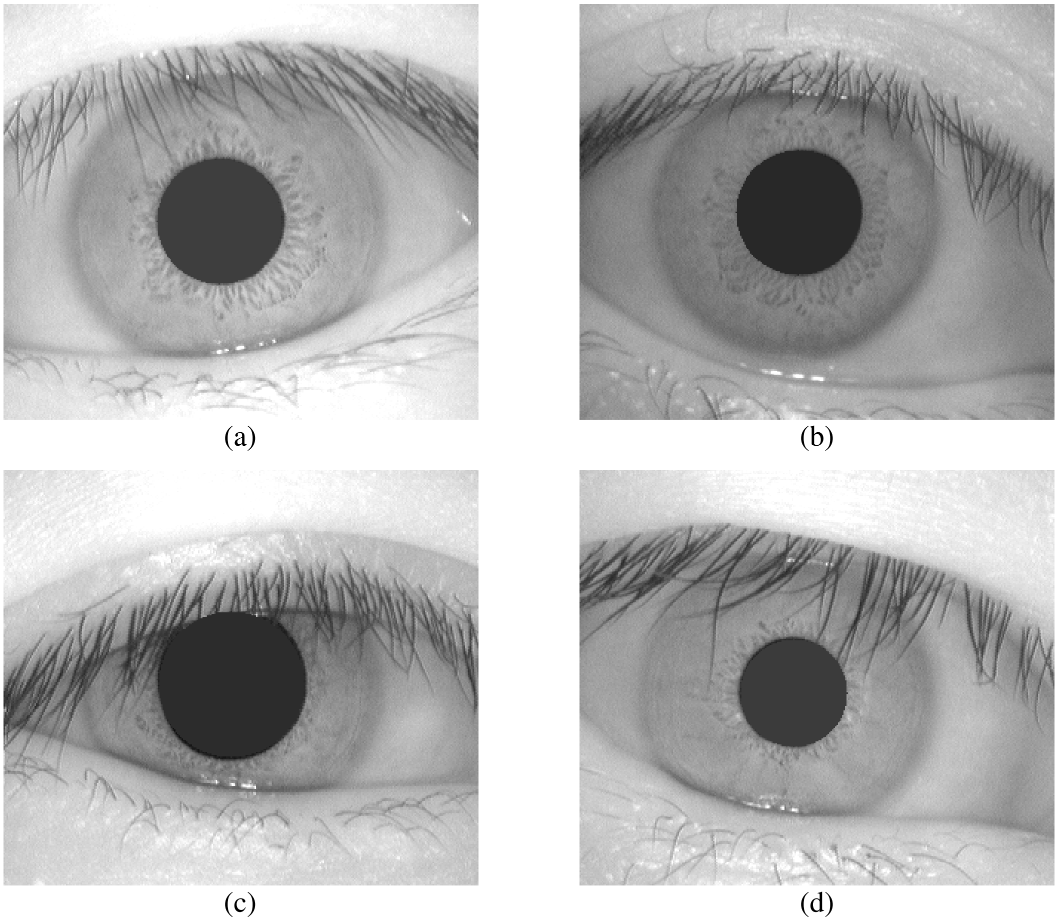
5.6 Comparison of IrisCode

Many researchers have re-implemented IrisCode for comparison. I strongly believe that these researchers have tried their best to make a fair comparison yet there are three particular issues which should be considered when comparing IrisCode with other iris algorithms: the differing properties of different iris scanners, the quality of re-implementation, and issues of identification versus verification.

5.6.1 Iris Scanners

Different iris images captured using different iris scanners have different properties. Let us compare the iris images in the CASIA database and those captured using commercial iris scanners. Figure 5.9 shows four iris images from the CASIA database and Figure 5.10 shows four iris images from commercial iris scanners. The images captured with commercial iris images are provided by Daugman. They are rescaled for display but their intensity values are not modified. The CASIA images have an image size of 320 by

280 while Daugman's have an image size of 640 by 480. We can make several observations about these two sets of images. Note that the images in Figure 5.10 include a greater width of the eyelids. Daugman uses this greater width for more accurate estimation of the eyelid/iris boundaries [124]. The contrast of the limbus in Figure 5.10 is also stronger than that in Figure 5.9. It is also noted that each image from the CASIA database has a black circle in the pupil and some of these black circles cover the part of iris and eyelids, potentially affecting the accuracy of pupil detection. Figure 5.9(c) provides an example of this. The black circles also cover the specular reflection. Therefore, automatic detection of the pupils is easier. It is important to be aware of these differences in the properties of the CASIA images and Daugman's images because Daugman designed IrisCode based on the properties in his images. Directly applying IrisCode on other images may not produce an optimal performance.



(a) (b)

(c) (d)

Figure 5.9 Four irises images from CASIA database

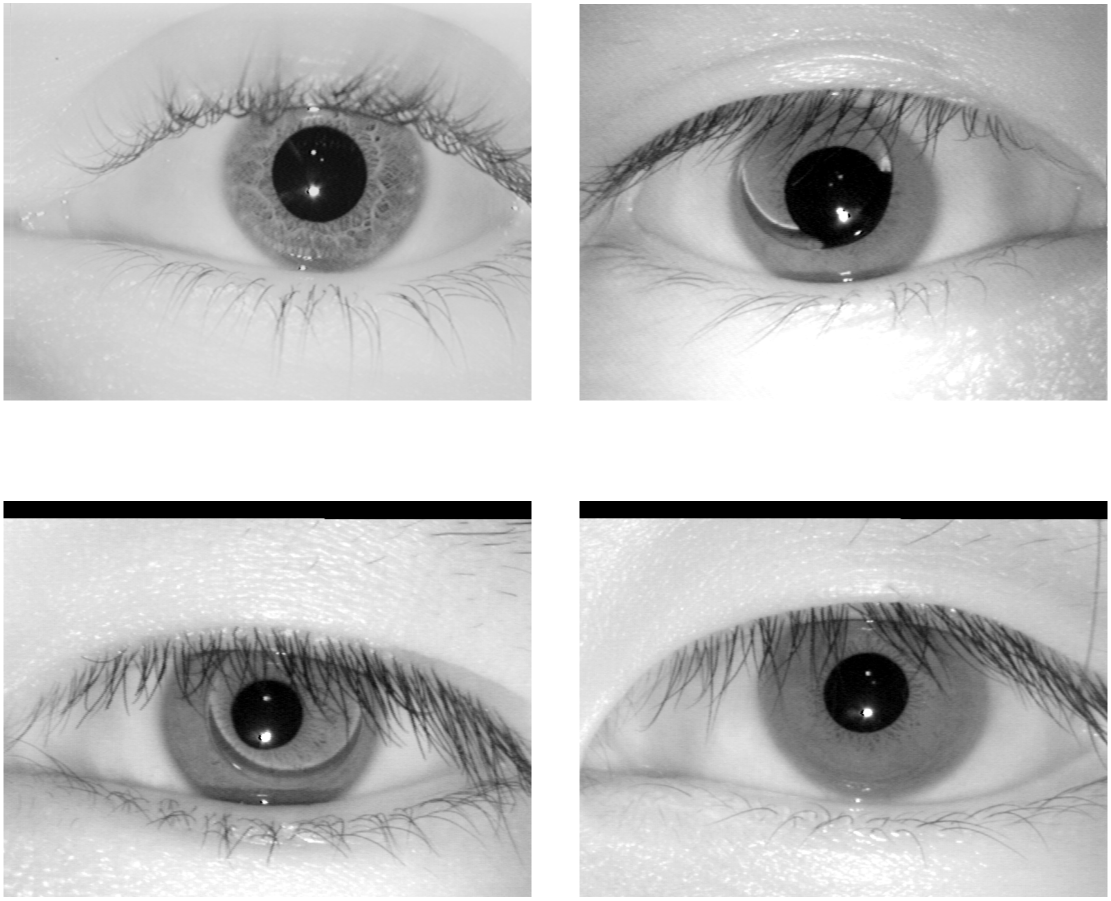


Figure 5.10 Four irises images from commercial iris scanners.

5.6.2 Quality of Re-implementation

IrisCode is a commercial algorithm and as a result it is not possible to obtain the source code and compare it with executable program. Researchers are restricted to re-implementing IrisCode based on available knowledge. With this, it is known that there are some implementations that are especially problematic. For instance, some researchers use Gabor filters having various orientations [110, 115, 125, 163] whereas IrisCode uses only Gabor filters that have a single orientation in the dimensionless polar coordinate system. Another problem is that some parts of IrisCode, including eyelids and eyelashes detection, cannot be accurately re-implemented since Daugman has not clearly disclosed them in his publications. Another difficulty for accurate re-implementation is that of choosing the parameters for preprocessing. Because of the variation of re-implementations, different researchers report different accuracy, even for the same iris database [130-131].

5.6.3 Identification versus Verification

When comparing iris algorithms it is important to bear in mind whether their application will focus on identification or verification. Some iris algorithms have recently reported high accuracy, such as phase matching and correlation filter approaches [126-127, 163] but as far as matching is concerned their computational cost is much higher than that of IrisCode and other coding methods. Such approaches are more suitable for verification or identification in small databases where speed is not an issue. In contrast, coding methods such as IrisCode are designed for large-scale real-time identification where matching speed is of the essence. Matching speed of different coding methods can be very different. Precise phase representation is one of the examples.

5.7 Conclusion

IrisCode first appeared thirteen years ago yet to my knowledge this is the first report to provide a detailed analysis of this method. The analysis made here makes a number of contributions. It presents a complete analysis of IrisCode, demonstrating that IrisCode is a clustering process and that the locus of a Gabor function is on a two dimensional ellipse with respect to the phase parameter. It also proves the equivalent relationship between the bitwise hamming distance and bitwise angular distance and shows that Gabor function can be considered as a steerable filter. It then uses these properties and this relationship to develop a precise phase representation algorithm. This algorithm inherits the properties of IrisCode including robustness against brightness and contrast variations and rapid matching based on bitwise operators and cyclic representation. The experiments showed that given the same quality of preprocessing precise phase representation is more accurate than IrisCode. Precise phase representation is a flexible representation for balancing the tradeoff between matching speed and identification accuracy. This chapter has also presented theoretical evidence regarding the binomial imposter distribution of IrisCode and has determined that the theoretical evidence is incomplete. Finally, using the filter generating function and the coding scheme defined in this study, I have shown the relationships between IrisCode and other iris and palmprint recognition coding methods.

Chapter 6 Learning Optimal Feature Filters

The filters in the coding methods also regarded as cluster centers in Chapter 5, play an important role in the coding framework for palmprint and iris identification. The current selection of the filters is completely based on human intelligence. In this chapter, I propose a learning algorithm for automatically generating the filters for palmprint identification and further extend the framework.

6.1 Introduction

The coding methods described in the previous chapters require a set of filters to generate the bitwise feature codes. Currently, all the researchers design or select their filters completely based on human intelligence, a methodology which is relatively ah-hoc. To reduce human involvement in the design process and to automatically extract information from images, a learning algorithm for automatically generating the filters is proposed. Only palmprints will be considered here, since they are the main focus of this thesis.

The coding methods for palmprint and iris identification can be regarded as clustering processes. Although there are numerous clustering algorithms such as K-mean, fuzzy c-mean and self-organizing map are proposed for classification, knowledge mining and knowledge representation [161], the existing clustering algorithms are not suitable for generating the filters since their objective functions do not the optimize recognition performance. As a result, a new learning algorithm is needed.

In designing such a learning algorithm, several aspects must be considered. First of all, an effective filter representation is needed. Using the [6]standard basis to represent a filter, v_i requires large number of parameters. For example, if a coding method requires eight filters and the size of the filters is 25 by 25, the number of parameters required is 5,000. More parameters can enhance the representation power but may inadvertently cause over-learning. Therefore, a more effective representation is essential. In addition to representation, selection of a searching algorithm for estimating optimal parameters is equally important since it directly influences learning speed and memory requirement. This selection is especially vital for large training databases.

The rest of this chapter is organized as follows. Section 6.2 presents the filter representation. Section 6.3 provides an effective searching algorithm based on simulated annealing. Section 6.4 extends the coding framework from the perspective of topological maps. Section 6.5 reports the experimental results. Section 6.6 offers concluding remarks.

[6] Standard basis is an orthonormal basis where each vector has only one non-zero element, and that element is equal to one.

6.2 Filter Representation

A filter v_i with size s by s can be represented by a linear combination of a set of basis vectors, $\{\xi_k\}$, i.e. $v_i = \sum_k a_{ik}\xi_k$, where $a_{ik}s$ are the filter parameters. If optimal filters can be approximated by a subspace basis, the number of parameters can be reduced. Let $I_p = \sum_k b_k \xi_k$ be a patch of an image. For discrete filters and discrete images, the clustering criterion described in Section 5.2.1 can be written as $j = \arg\max_i \left(< I_p, v_i > / \|I_p\| \|v_i\|\right)$, where $< I_p, v_i >= \sum_k a_{ik} b_k$, $\|v_i\| = \sqrt{\sum_k a_{ik}^2}$ and $\|I_p\| = \sqrt{\sum_k b_k^2}$. If $b_k \approx 0$ for all $k>J$, $< I_p, v_i > \approx \sum_{k=1}^{J} a_{ik} b_k$. In the other words, v_i can be approximated by the span of $< \xi_1, \xi_2, \ldots, \xi_J >$ and the representation of v_i can be compressed.

According to this analysis, the coefficients of the image b_k directly affect the solution subspace of v_i. In the following derivation, the variance of b_k is used as a criterion to generate the subspace. Since $\max_k Var(b_k) = \max_k E(b_k - E(b_k))^2$ and $b_k = \xi_k^T I_p$, the corresponding ξ_k can be obtained by $\arg\max_{\|\xi_k\|=1} E\left(\xi_k^T I_p - E(\xi_k^T I_p)\right)^2$. Clearly, ξ_k is a principal component and therefore, principal component analysis can be used for compressing the representation of the filters.

The previous coding methods employ zero DC filters to eliminate brightness variation. However, the filter $v_i = \sum_k a_{ik}\xi_k$ generated by principal components is not zero DC since ξ_k is not zero DC in general. To generate zero DC filters and principal components, the DC of training images is removed.

In addition to zero DC, a Gaussian function is applied to all training images to weight the importance of pixels in different positions for imperfect alignment of image patches. The Gaussian function is applied before removing the DC of the training images. Once the basis vectors $\{\xi_k\}$ are generated, all the filters v_i in the coding methods are represented by $v_i = \sum_{k=1}^{J} a_{ik}\xi_k$, where J is a predefined parameter. Figure 6.1 illustrates the first eight principal components learned from the training images described in Section 6.5.

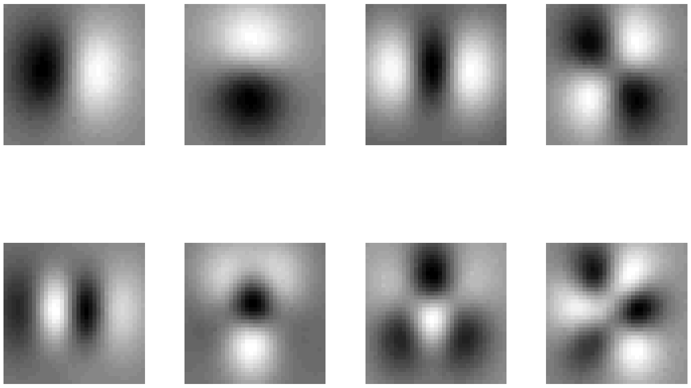


Figure 6.1 The first eight principal components using Gaussian function with standard derivation 11.

6.3 An Optimization Algorithm Based on Simulated Annealing and Effective Filtering

Selecting optimization algorithms depends on objective functions. In the following experiments, *d*' index is employed as an objective function. There are numerous optimization algorithms but not all of them are suitable for this application. For example, optimization algorithms such as gradient descent which involve the derivative of the objective function are not applicable since the bitwise operators in the matching function are not differentiable. Even though many stochastic optimization algorithms such as simulated annealing and genetic algorithms [162] do not require the derivative of the objective function, computation complexity is another problem. In each iteration, these stochastic optimization algorithms generate at least one set of potential solutions (the filters) and evaluate them based on the objective function. In the other words, the filters should be applied to whole training datasets in each iteration. It is a very time consuming process. To speed up this process, an effective filtering scheme is proposed.

6.3.1 An Effective Filtering Scheme

The proposed filtering scheme makes use of the information in the current iteration to improve filtering speed in the next iteration. Let $v_i = \sum_{k=1}^{J} a_{ik} \xi_k$ be a filter in the current iteration and $< I_p, v_i >= \sum_{k=1}^{J} a_{ik} < I_p, \xi_k >$, where $< I_p, \xi_k >$ being independent of iterations, can be pre-computed and stored in main memory of a computer. In stochastic optimization algorithms, solutions in the next iteration generally depend on the solutions in the current iteration. It is assumed that only a single filter, v_i is modified and only M coefficients in v_i are changed. Without loss of generality, let the M coefficients be a_{i1}, a_{i2},… and a_{iM}. The new filter, $v_i^{'}$ is defined as $v_i^{'} = \sum_{k=1}^{M} a_{ik}^{'} \xi_k + \sum_{k=M+1}^{J} a_{ik} \xi_k$, where $a_{ik}^{'}$ s are the new coefficients generated for the next iteration. The corresponding inner product $< I_p, v_i^{'} >$ can be obtained in a very effective way. Consider

$$\begin{aligned}
&< I_p, v_i^{'} >= \sum_{k=1}^{M} a_{ik}^{'} < I_p, \xi_k > + \sum_{k=M+1}^{J} a_{ik} < I_p, \xi_k > \\
&= \sum_{k=1}^{M} a_{ik}^{'} < I_p, \xi_k > + \sum_{k=M+1}^{J} a_{ik} < I_p, \xi_k > + \sum_{k=1}^{M} a_{ik} < I_p, \xi_k > - \sum_{k=1}^{M} a_{ik} < I_p, \xi_k > \\
&=< I_p, v_i > + \sum_{k=1}^{M} (a_{ik}^{'} - a_{ik}) < I_p, \xi_k > \qquad (6.1)
\end{aligned}$$

Therefore, computing $< I_p, v_i^{'} >$ requires only M subtractions, M multiplications and one addition. If M is equal to 1, only three operations are needed. It should be emphasized that $< I_p, v_i >$ and $< I_p, \xi_k >$ should be stored in main memory rather than a hard disk since transferring data from a hard disk to a processor is much slower than that from main memory to a processor.

Using the same approaches, the norms of filters can also be updated effectively. Consider

$$\begin{aligned}
&\left\| v_i^{'} \right\|^2 =< v_i^{'}, v_i^{'} > \\
&= \sum_{k=1}^{M} a'^{2}_{ik} + \sum_{k=M+1}^{J} a_{ik}^2 \\
&= \sum_{k=1}^{M} a'^{2}_{ik} + \sum_{k=M+1}^{J} a_{ik}^2 + \sum_{k=1}^{M} a_{ik}^2 - \sum_{k=1}^{M} a_{ik}^2 \\
&= \left\| v_k \right\|^2 + \sum_{k=1}^{M} (a_{ik}^{'\,2} - a_{ik}^2) \qquad (6.2)
\end{aligned}$$

6.3.2 Using Simulated Annealing Optimization

Although the effective filtering scheme can speed up the filtering process, the memory requirement increases with respect to the number of potential solutions in each iteration. Therefore, population based optimization algorithms such as genetic algorithms are not suitable. Simulated annealing having only one potential solution in each iteration is applicable [160]. The computational details of simulated annealing and an effective filtering scheme are summarized in Figure 6.2. To control over-learning, early stopping is applied [160].

```
$M$ = The number of parameters to be changed in the next iteration
$M_{max}$ = The maximum number of parameters to be changed in each iteration
$T_0$ = Initial annealing temperature
$T$ = Annealing temperature
$\eta$ = Learning rate
$\Gamma$= Maximum number of iteration.

Initialization
1.    Compute all $< I_p, \xi_k >$ and store them in main memory.
2.    Randomly generate the parameters, $a_{ik}$ and compute the norms of the filters $\|v_i\|$.
3.    Compute all $< I_p, v_i >= \sum_{k=1}^{J} a_{ik} < I_p, \xi_k >$ and store them in main memory.
4.    Compute the winning indexes and perform matching.
5.    Compute $d'$ index ($d'_{old}$) .
6.    Set $T$=$T_0$ and $M$=1.

Iteration
1.    Randomly select one of the filters, $v_i$ and randomly change $M$ of its coefficients from $a_{ik}$
      to $a'_{ik}$. $a'_{ik}$s are randomly generated.
2.    Compute all $< I_p, v_i^{'} >$ using Eq. 6.1 and compute $\|v_i^{'}\|^2$ using Eq. 6.2
3.    Compute the winning indexes and perform matching.
4.    Compute $d'$ index ($d'_{new}$) using new $v_i$.
5.    Accept the new parameters if $d'_{\text{new}}$ is greater than $d'_{\text{old}}$ or $\gamma < \exp\left(\frac{d_{new}^{'} - d_{old}^{'}}{T}\right)$, where $\gamma$
      follows an uniform distribution bounded by 0 and 1. Otherwise reject the new parameters
      and keep the original parameters.
6.    Update the annealing temperature according to $T = \eta T$ .
7.    Set $< I_p, v_i >=< I_p, v_i^{'} >$ , $\|v_i\| = \|v_i^{'}\|$ , $a_{ik} = a_{ik}^{'}$ and $M$=1 if the new parameters are
      accepted.
8.    Set $M$=$M$+1 if the new parameters are rejected and $M \le M_{\max}$ .
9.    Stop the iteration if the number of iteration is equal to $\Gamma$.
10.   Otherwise, continue.
```

Figure 6.2 The pseudo-code of the proposed optimization algorithm using simulated annealing and the effective filtering scheme.

6.4 Using the View of Topological Map to Extend the Coding Framework

In Chapter 5, I have shown that Gabor phase regarded as a one-dimensional periodic feature is always on the circumference of an ellipse. The encoder given in Figure 5.2 is especially designed for this feature. To further extend the encoder for other features, another view is provided to understand the framework. The filters can be considered as synaptic weight vectors in a topological map and the relationship between winning indexes can be considered as the relationship between the neurons. Figure 6.3 illustrates this concept. The elliptically topological map presented in Chapter 5 is possible to be replaced with other forms of topological maps such as the hypercube. If the relationship between neurons can be measured by bitwise hamming distance, high speed matching can be guaranteed.

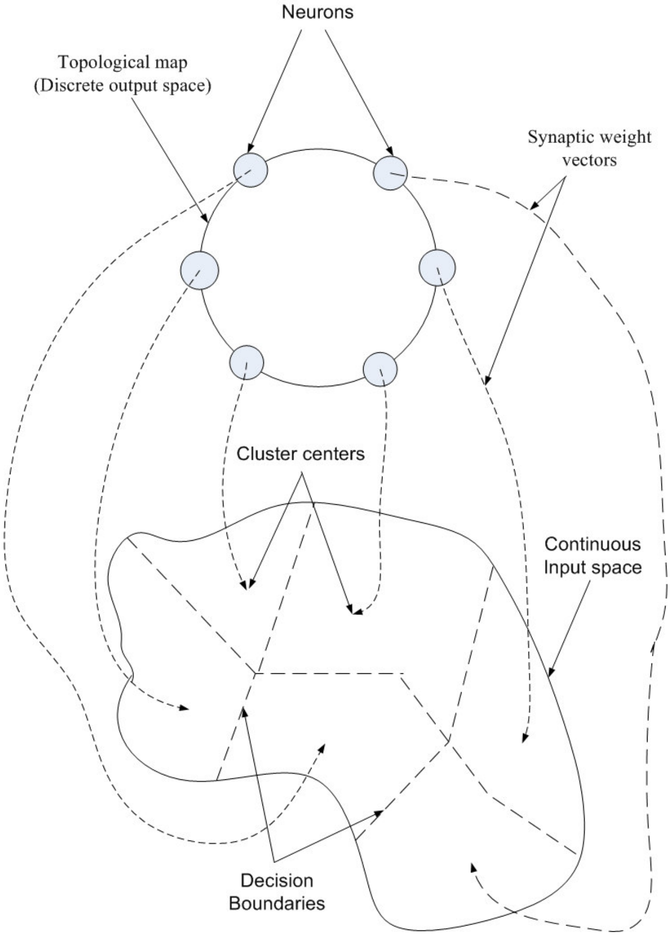


Figure 6.3 Using topological map to extend the coding framework

6.5 Experimental Results

The palmprint database described in Chapter 3 is divided into three datasets: training set, validation set and testing set. Training and validation sets both contain 1,600 palmprints from 160 palms. Each palm in training set or validation set has 10 images, 5 from the first occasion and 5 from the second occasion. The testing set contains 4,835 images from 248 palms. These three datasets do not overlap in terms of palms and images. In other words, images from the same palm would belong to the same datasets. The proposed learning algorithm has several parameters including the number of principal components for representing the filters, the variance of the Gaussian function for preprocessing the data, the structure of the topological map and the parameters listed in Figure 6.2. In the following experiments, 3-bit gray code given in Table 6.1 is used to represent a cube illustrated in Figure 6.4. The parameters in Figure 6.2, M_{max}, T_0, η and Γ are 5, 18000, 0.998 and 18000, respectively. Gaussian functions with standard derivations 11 and 13 are examined and the size of image patch, I_p is 35 by 35. The principal components are estimated from 30,000 image patches from the training set and the numbers of principal components to be examined are 20, 30 and 40. Table 6.2 lists the parameters and the corresponding *d*' indexes from the testing set. Comparing the *d*' index of Competitive Code of 5.63, the proposed learning algorithm is effective in terms of *d*' index. The greatest one is 6.08 from experiment (c) and the corresponding filters are given in Figure 6.5. Figure 6.6 compares their ROC curves. The ROC curve of Competitive Code is also plotted for comparison. In the range of the false acceptance rates between 5×10^{-1}% and 100%, the learned filters from experiment (c) perform better than Competitive Code. However, in the range of false acceptance rates being smaller than 5×10^{-1}%, Competitive Code performs better. Some may expect that the learned filters should perform better than Competitive Code at any false acceptance rate. It should be emphasized that the objective function of the learning algorithm is *d*' index, not ROC curve. As a result, the learned filters perform better than Competitive Code in terms of *d*' index while the learned filters do not always perform better than Competitive Code in terms of genuine acceptance rates. Some may suggest using genuine acceptance rates at a low false acceptance rate e.g. 10^{-4}% as an objective function. However, this statistics is difficult to estimate reliably since the number of false matching at that range is very limited. In this case, even though early stopping is applied, it is still hard to control over-learning.

Table 6.1 3-bit grey code.

Winning indexes	0	1	2	3	4	5	6	7
Bit 1	0	1	1	0	0	1	1	0
Bit 2	0	0	1	1	1	1	0	0
Bit 3	0	0	0	0	1	1	1	1

Table 6.2 The experimental parameters and the corresponding *d*' indexes.

Experiment ID	a	b	c	d	e	f
Standard derivation of the Gaussian	11	11	11	13	13	13
Number of principal components	20	30	40	20	30	40
d' index	5.94	6.01	6.08	5.90	6.01	5.9

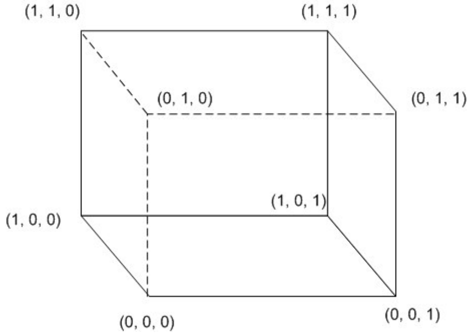


Figure 6.4 The topological map of 3-bit gray code

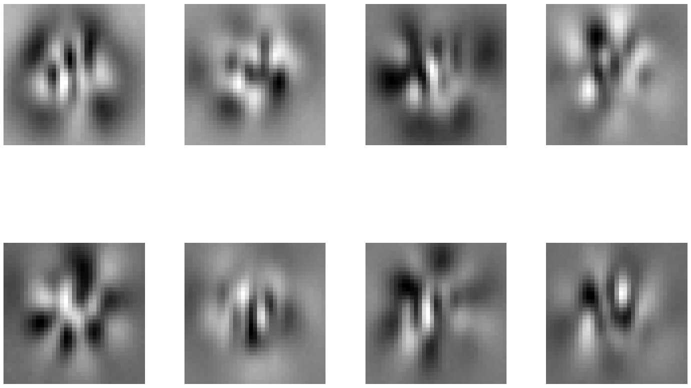


Figure 6.5 The resultant filters from the experiment (c)

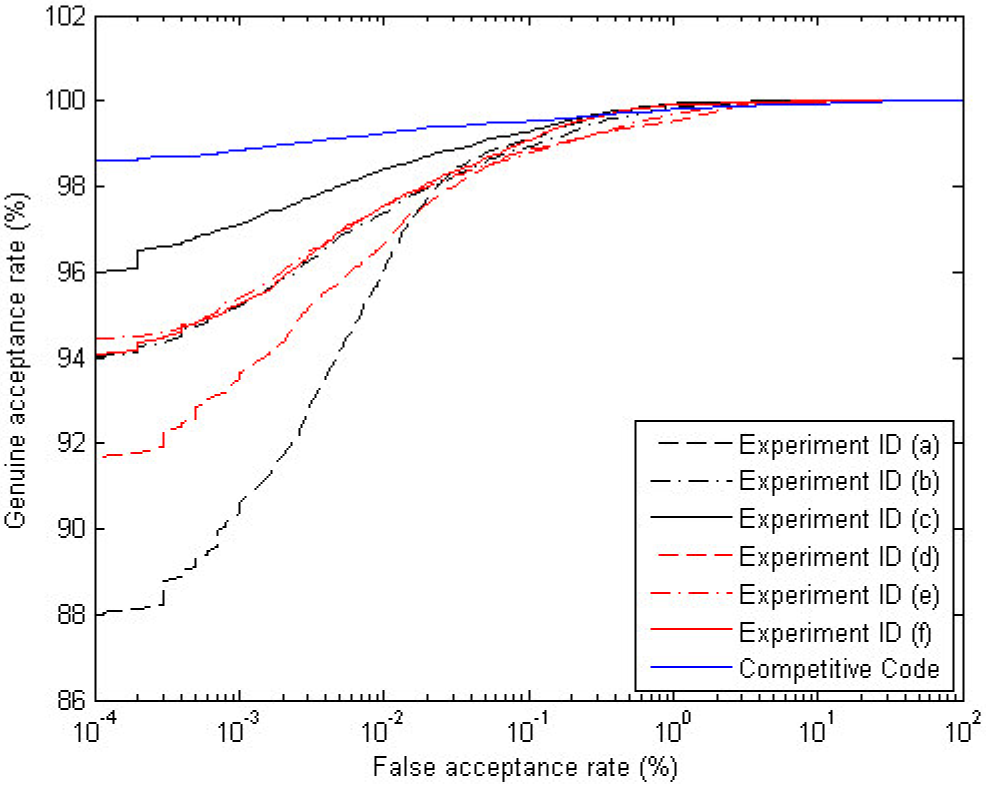


Figure 6.6 The ROC curves of the learned filters and Competitive Code. (color figure)

6.6 Conclusion

In this chapter, I attempt to design a learning algorithm to automatically generate filters for the coding framework. The learning algorithm has several components including the principal components for representing the filters, a Gaussian function for imperfect alignment, the effective filtering scheme for increasing the learning speed, a simulated annealing algorithm for estimating the parameters and an early stopping for preventing over-learning. In addition to the learning algorithm, the framework is further extended based on the concept of topological maps. The experimental results demonstrate that using *d*' index as a performance index, the learned filters perform better than Competitive Code.

Chapter 7 A Study of Identical Twins' Palmprints

The performance of biometric systems highly depends on the distinctive information in the biometrics. Identical twins having the closest genetics-based relationship are expected to have maximum similarity in their biometrics. Classifying identical twins is a challenging problem for some automatic biometric systems. Palmprint has been studied for personal identification for many years. Most of the previous research concentrates on algorithm development. This chapter systemically examines palmprints from the same DNA for automatic personal identification and to uncover the genetically related palmprint features. The experimental results show that the three principal lines and some portions of weak lines are genetically related features but our palms still contain rich genetically unrelated features for classifying identical twins.

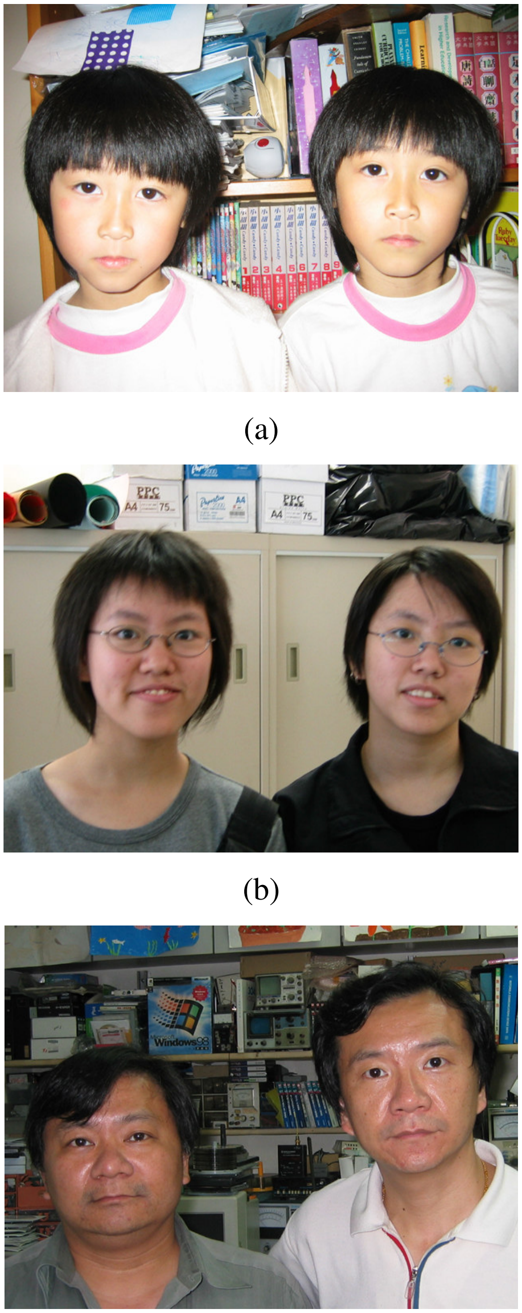
7.1 Introduction

Biometric technologies verifying different people are based on the distinctive information in their biometric traits. However, not all biometrics provide sufficient distinctive information to classify identical twins, who have the same genetic expression. Studying identical twins' biometrics is an important topic for biometric authentication.

There are two types of twins, dizygotic and monozygotic twins. Dizygotic twins result from two different fertilized eggs resulting in different Deoxyribo Nucleic Acid (DNA). Monozygotic twins, also called identical twins are the results of a single fertilized egg splitting into two individual cells and developing into two individuals. Therefore, identical twins have the same genetic expressions. The frequency of identical twins is about 0.4% across different populations [132]. Some researchers believe that this is the performance limit of face recognition systems [133].

7.1.1 From DNA to biometrics

DNA contains all the genetic information required to create an organ of a species. The mapping from DNA to the actual expression of an organ is very complex. Firstly, the genetic information is copied from DNA molecule into RNA (Ribo Nucleic Acid) molecule. Then, the RNA is converted into amino acids and the amino acids are converted into functioning proteins. The proteins are assembled to be an organ. In this decoding process, the final products are affected by not only genetic information but other factors as well. As a result, identical twins sharing the same genetic expression have many different biometrics, including fingerprint, iris and retina [134-136]. In fact, some biometrics such as face continually change after we are born. The changes depend on environmental factors such as living style, diet and climate. These environmental factors make identical twins more different as they age. Figure 7.1 shows three pairs of identical twins at different ages. The oldest twins in Figure 7.1(c) are the most distinguishable.



(a)

(b)

(c)

Figure 7.1 Three pairs of identical twins at different ages

7.1.2 Problems of confusion of twins' identities

In spite of the fact that the biometrics of identical twins are affected by many factors, some of them such as facial features are still very similar. Some identical twins share not only similar facial features but also the same signatures. Confusion over their identities has made it difficult for others to know who owns what and who does what. As a result, some identical twins partake in commercial scams such as fraudulent insurance compensation. Most importantly, if one of the identical twins commits a serious crime, their unclear identities cause confusion and uncertainty in court trials.

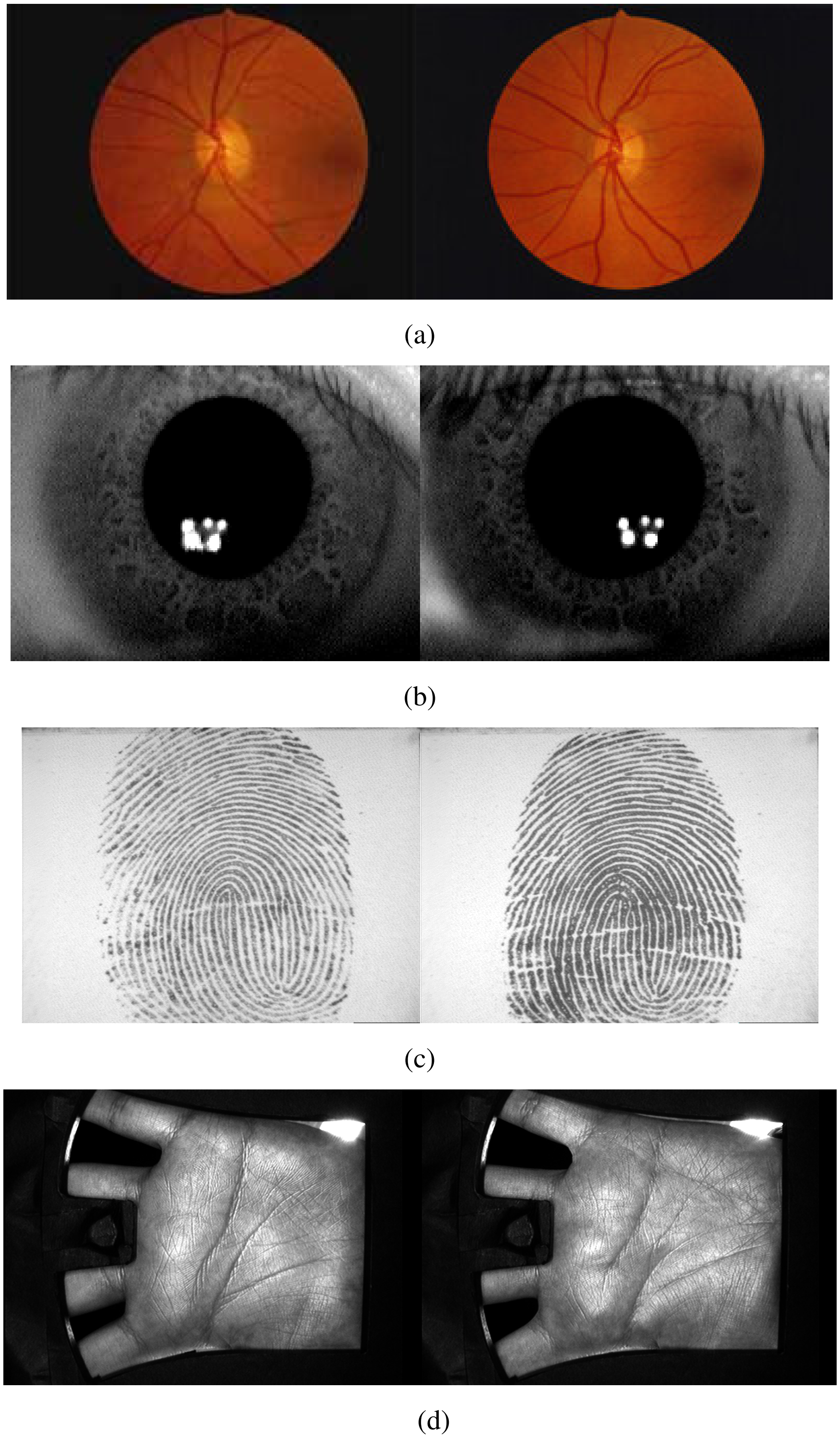
7.1.3 Motivations

Classifying identical twins is crucial for all biometric authentication systems. The systems, which cannot handle identical twins, have an obvious security flaw. Figure 7.2 depicts the retinas, irises, fingerprints and palmprints of identical twins. The iris and palmprint images are collected using the devices described in [7]; the retina images are obtained from Retinal Technologies, (http://www.retinaltech.com/technology.html) with permission to reprint; and the fingerprint images are collected using a standard optical fingerprint scanner. Figure 7.2 shows that the retinas, irises and palmprints are distinguishable to human vision. For the fingerprints, to differentiate the images one must pay close attention to the minutiae points (end points and bifurcation points), commonly utilized in fingerprint systems. Based on the position and direction of the minutiae points, the twins' fingerprints are also distinguishable.

In many cases, biometrics are proposed by medical doctors or ophthalmologists [136] but almost all the biometric systems are designed by engineers. The features discovered by doctors or ophthalmologists and the features applied to authentication systems may not be the same. The iris is a typical example [136-137]. Therefore, the experimental results or observations given by doctors or ophthalmologists about identical twins may not be applicable to automatic biometric systems. Therefore, it is essential to test automatic biometric systems on identical twins. There has been research in iris, face, voice and fingerprint authentication for identical twins [134, 137-141]; however, identical twins' palm is ignored. This chapter aims at examining an automatic palmprint system on identical twins and identifying their genetically related features.

Competitive Code is employed for this study because it is most accurate palmprint identification algorithm according to the comparison in Chapter 4 and because the winning indexes in Competitive Code describing the orientation of palm lines can be used to identify the genetically related features in palmprints. In this chapter, one of Competitive Code is translated vertically and horizontally to match another Competitive Code as described in [3] and Chapter 3 to handle imperfect preprocessing.

The rest of this chapter is organized as follows. Section 7.2 presents the genetically identical palmprint databases. Section 7.3 reports the experimental results and analysis. Section 7.4 offers some concluding remarks.



(a)

(b)

(c)

(d)

Figure 7.2 Different identical twin's biometrics, a) retina, b) iris, c) fingerprint and d) palmprint.

7.2 Genetically identical palmprint databases

There are two possible ways to obtain palmprints generated from the same genetic information. Identical twins' palms are one of them. Left and right palms from the same persons are the other. They are generated form the same gene, HOXD13 gene as identical twins. To make this study more complete, palmprints are collected in both ways. Figure 7.3 shows four palmprints from a pair of identical twins. Three of them are similar for human vision but they are still distinguishable. The other is relatively different.

For this study, two palmprint databases, twin database and general database are prepared. The twin database contains 1028 images collected from 53 pairs of identical twins' palms. The images are collected over half year. Around 10 images are collected from each palm. The age range of the subjects is between 6 and 45. All the images are collected by the palmprint capture device described in [7]. The image size is 384×284 with 75 dpi.

The general database contains 7,967 palmprint images from 200 subjects. The palmprint images were collected on two separate occasions. The average time interval between the first collection and second collection was around two months. On each occasion, each subject was asked to provide about 10 images, each of the left palm and the right palm. The image size is also 384×284 and their resolution is 75 dpi.

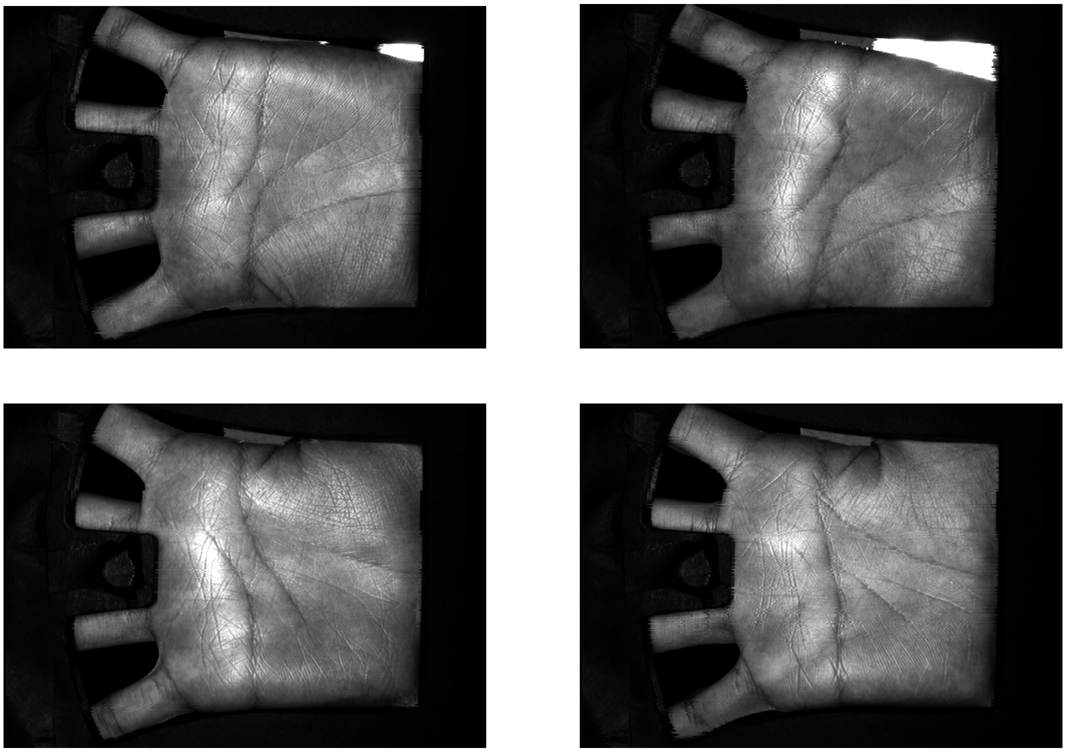
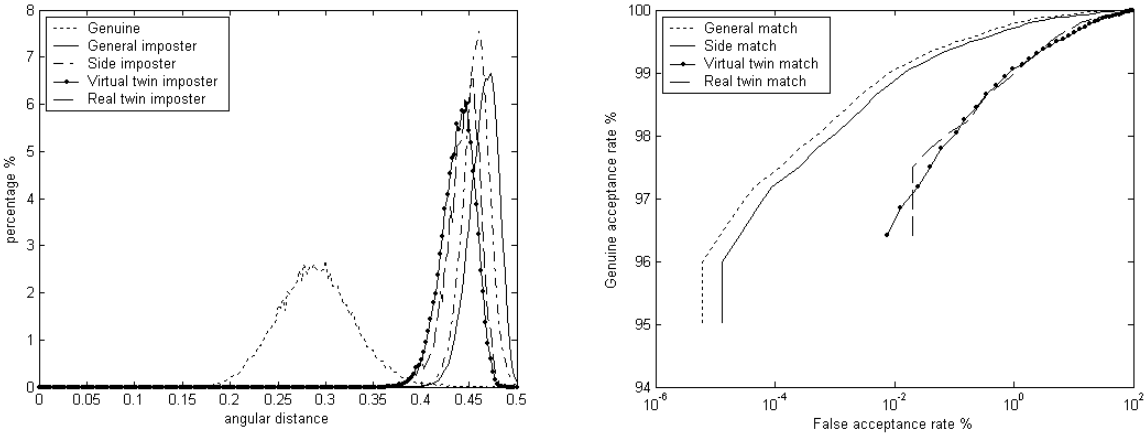


Figure 7.3 Four palmprints from a pair of identical twins.

7.3 Experimental Results and Analysis

To study the similarity between identical twins' palmprints and to obtain a twin imposter distribution, the palmprints from the pairs of identical twins' palms (real twin match) are matched. The palmprints from the same palms in the general database are also matched to obtain a genuine distribution of normal (non-identical) palms. Similarly, the palmprints from different palms in the general database are matched to obtain an imposter distribution of normal palms (general match). In addition, different person's left palmprints are matched and different person's right palmprints are matched to obtain a side imposter distribution (side match). The left and right palmprints from the same persons are also matched (virtual twin match). For virtual twin match, one of the images is flipped to match the other. For the palmprints in the general database, the palmprints from different occasions are matched only. The total number of genuine matchings, general imposter matchings, side imposter matchings, virtual twin imposter matchings and twin imposter matchings are 39,673, 15,828,599, 7,894,462, 39,671 and 4,900, respectively; Figure 7.4(a) shows these distributions. The genuine distribution along with the four imposter distributions in Figure 7.4(a) is used to generate the Receiver Operating Characteristics (ROC) curves given in Figure 7.4(b). These figures show that palmprints generated from the same genetic information are significantly correlated and this correlation is neither due to matching between right palms or matching between left palms. However, they still have enough non-genetically related information for classification. For example, if we set the threshold at 0.3725 along with the genuine acceptance rate of 97%, the corresponding false acceptance rate for general imposters is 4.4×10^{-5}% and the corresponding false acceptance rate for identical twin imposters is 2.0×10^{-2}%.



(a) (b)

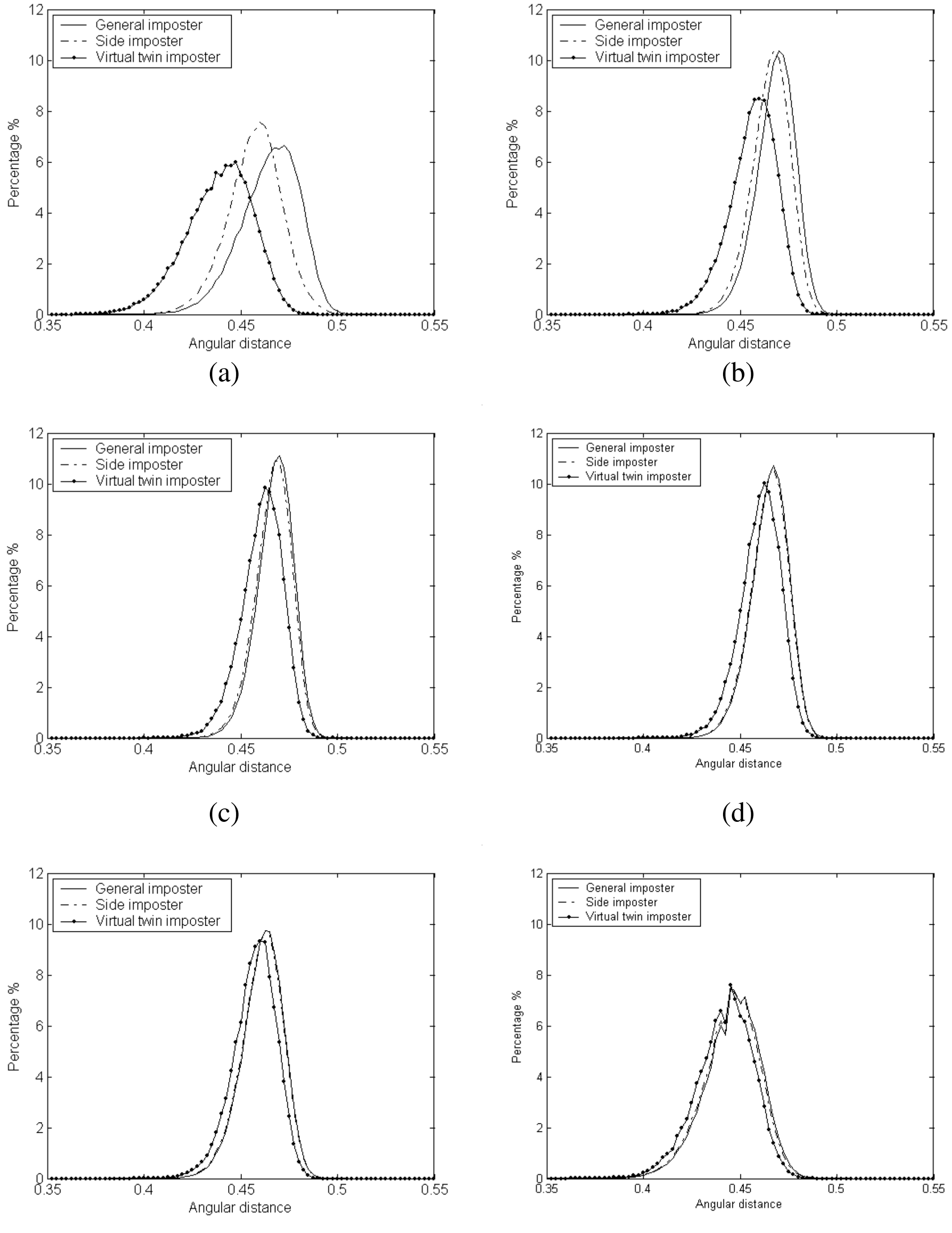
Figure 7.4 Experimental results. (a) Distributions of real twin imposter, virtual twin imposter, side imposter, general imposter and genuine and (b) the corresponding ROC curves.

7.3.1 Identifying the Genetically Related Features.

The previous experimental results demonstrate that palmprints generated from the same genetic information have correlated features. From the observation of Figure 7.3, it is believed that the strong lines including the principal lines are the genetically related features. The following experiment quantitively justifies this observation.

In this experiment, the features points (the winning indexes) associating with the strong lines are successively removed and the rest of feature points are used to perform matching. Using the filter response given in Eq. 4.3, it is easy to identify the strong lines. The corresponding winning indexes are denoted by the masks in Eq. 4.6. The masks are superimposed to identify the locations of the strong feature points. To provide statistically reliable results, virtual twin matchings are used to investigate the correlated features in this experiment since the virtual twin matchings are more. Virtual twin matchings and real twin matchings are equivalent from genetic point of view since our left and right palms are generated from the same gene as identical twins' palms.

The most 100, 200, 300, 400 and 600 significant winning indexes are removed to generate the general imposter, side imposter and virtual twin imposter distributions shown in Figure 7.5(b)-(f), respectively. Figure 7.5(a) shows the original imposter distributions for comparisons. To measure the dissimilarity between virtual twin imposter distribution, *p* and general imposter distribution *q*, Bhattacharyya distance defined as, $B(p,q) = -\log\left(\int \sqrt{p(x)q(x)}dx\right)$ is used. Figure 7.6(a) shows their Bhattacharyya distances. Similarly, Figure 7.6(b) shows Bhattacharyya distances between side and general imposter distributions. Figure 7.6(a) illustrates that the dissimilarity of the two imposter distributions decreases when the number of removed winning indexed increases. Comparing the Bhattacharyya distances in Figure 7.6(a) and (b), we know that even though more than half of the significant winning indexes are removed, the Bhattacharyya distance between virtual twin and general imposter distributions is relatively large. There are two reasons. Some weak lines are also genetically related. Figure 7.7 shows a pair of palms from the same person for illustration. The small portions are enhanced for visualization. In addition to the correlation of the weak lines, some winning indexes having weak response are generated from principal lines. To identify the location of the significant winning indexes, the masks of all left palmprints in the general database are superimposed. Figure 7.8 gives the distributions of the significant winning indexes. The first 100 significant winning indexes associate with the three principal lines. However, we do not observe any clear structure from others significant winning indexes. According to this finding, the pervious palmprint recognition methods exploiting only the strong lines as features may not be suitable for classifying identical twins [14]. In addition, this finding gives a quantitative evidence for the usage of principal lines for genetic research. However, most of the genetic research about palm lines concentrates only on Simian and Sydney lines shown in Figure 2.7. [68, 101, 142-143].



(a)

(b)

(c)

(d)

(e)

(f)

Figure 7.5 (a)-(f), virtual twin, side and general imposter distributions obtained by removing the most 0, 100, 200, 300, 400, and 600 significant winning indexes, respectively.

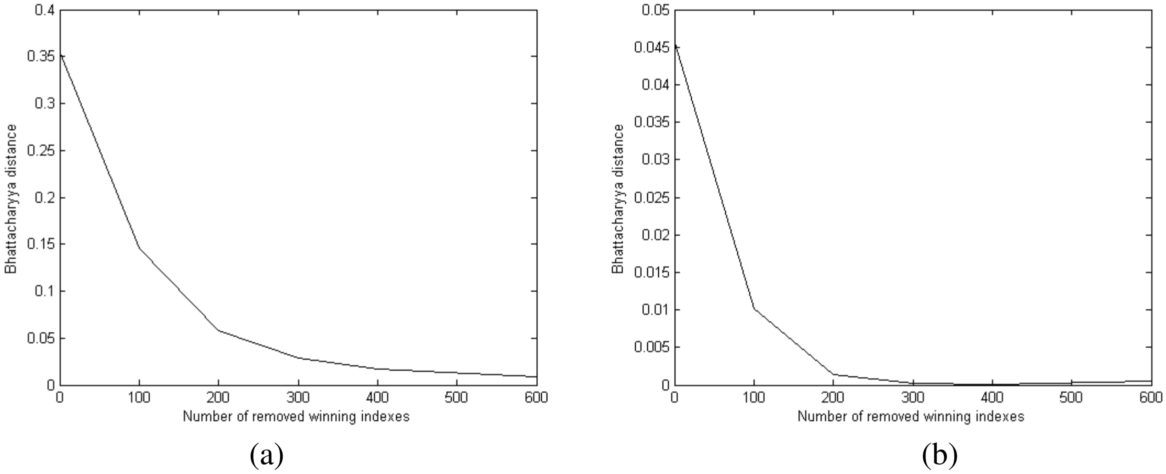


Figure 7.6 (a) The Bhattacharyya distances between general and virtual twin imposter distributions and (b) the Bhattacharyya distances between general and side imposter distributions in Figure 7.5(a)-(f).

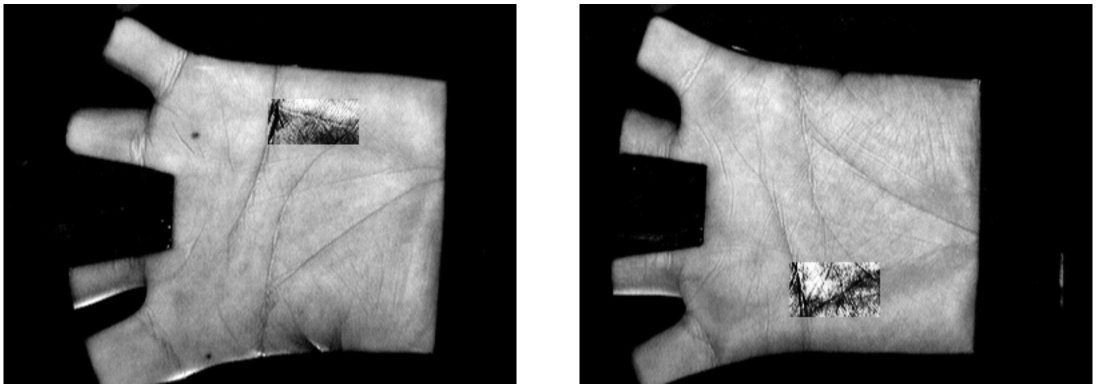
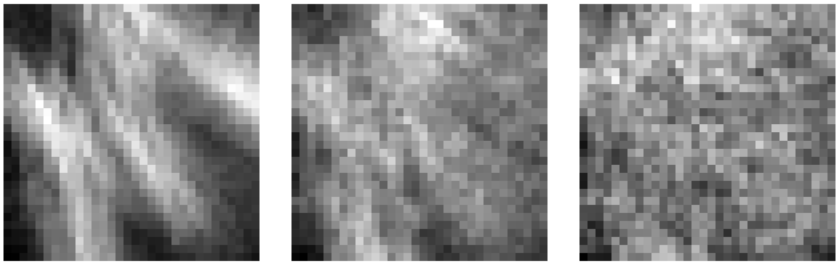


Figure 7.7 Illustration of genetically related weak lines.



(a) (b) (c)

Figure 7.8 The distributions of the most (a) 100, (b) 200 and (c) 300 significant winning indexes

7.4 Conclusion

In this chapter, the palmprints generated from the same genetic information including identical twins' palmprints have been systemically examined. This study shows that they can be distinguished by Competitive Code. The quantitative evidence also demonstrates that the three principal lines are genetically dependent. This evidence supports the usage of the principal lines for genetic research. It implies that the previous palmprint recognition algorithm exploiting only the strong lines as features may not be able to distinguish identical twins' palmpints. In this study, it is also pinpointed that some weak lines are also genetically related.

Chapter 8 Secure Palmprint Identification

Biometric authentication systems are widely applied because they offer inherent advantages over classical knowledge-based and token-based personal identification approaches. However, biometric systems are vulnerable to various attacks, such potential attacks must be analyzed before biometric systems are massively deployed in security systems. In this chapter, four security issues, including template re-issuances, also called cancellable biometrics, brute-force attacks, replay attacks and database attacks are addressed for secure palmprint identification. A random orientation random filter bank (ROFB) is used for template re-issuance. A projected multinomial distribution is proposed for studying the probability of successfully using brute-force attacks to break into a palmprint system using Competitive Code. Secret messages are hidden in Competitive Codes to prevent replay and database attacks. This technique can be regarded as template watermarking. A series of analysis is provided to evaluate their security levels.

8.1 Introduction

Although biometric authentication approaches [79] are much more secure than the traditional approaches, they are not invulnerable. For example, they are open to database, replay, and brute-force attacks. Figure 1.3 shows a number of points, Points 1-8, all being vulnerable points as identified by Ratha and his coworkers [95, 145]. Recently, many biometric and security researchers have proposed techniques for preventing and detecting these attacks [14, 26, 95, 97-98, 144-145, 147-149, 151]. Some researchers have employed watermarking and encryption to prevent replay attacks at Points 2, 4, and 7 [98, 148, 151] and have developed anti-spoofing techniques for specific biometrics to prevent attacks at Point 1 [144, 149]. Other researchers have produced analyses of specific attack types vis-à-vis specific biometrics, for example, brute-force attacks at Point 4 of fingerprint systems [26, 95, 145].

Given the commercial potential of palmprint systems as security applications, the wide variety of capture devices that now exist, and the diversity of preprocessing, feature extraction, matching and classification algorithms [Chapter 2] that have been produced in the field, it is certainly the case that any security issues should be systematically addressed prior to their widespread deployment. Palmprint recognition has been studied for many years and the potential attacks have been identified but only limited research work is related to the security of palmprint systems. A group of researchers develop cancellable palmprints [27, 96] for template re-issuance and claim that their approaches can achieve zero equal error rates. However, their results are based on an unrealistic assumption that users never share and lose their token keys. A detailed analysis is given in [6]. In fact, different features require different cancellable transforms [99, 158]. It is difficult to apply directly one cancellable transform from one feature to another feature. The replay attack and brute-force attack at Point 4 and the database attack at Point 6 have not been addressed for palmprint identification systems. Although some security measures have been proposed for other biometrics such as fingerprint, they cannot be directly applied to palmprints since their feature formats are different [97-98].

Cryptography is one possible solution that would allow us to better defend against replay and database attacks. Systems protected by cryptography store and transmit only encrypted templates in databases and through data links. However, cryptography is not suitable for speed-demanding matching, e.g. real-time large-scale identification, since decryption is required before matching. Another potential solution is cancellable biometrics. Cancellable biometrics transform original templates into other domains and perform matching in the transformed domain. Although cancellable biometrics overcome the weakness of cryptography, current cancellable biometrics are still not secure enough for the palmprint identification. For example, attackers can still insert stolen templates at Point 4 and Point 6 for replay and database attacks before systems can cancel the stolen templates and reissue new templates. Furthermore, current cancellable biometrics cannot detect replay and database attacks. In other words, if attackers insert unregistered templates into data links or databases, systems cannot discover the unregistered templates. To solve these problems, I take advantages of cryptography and cancellable biometrics to design a set of security measures to prevent replay and database attacks for secure palmprint identification.

The rest of this chapter is organized as the follows. Section 8.2 presents cancellable Competitive Code. Section 8.3 develops a probabilistic model describing the relationship between false acceptance rates and the number of attacks. Section 8.4 presents the security measures including one time pad and template watermarking. Section 8.5 presents the experimental results and validates the proposed probabilistic model. Section 8.6 evaluates the security levels of the proposed measures. Section 8.7 offers some concluding remarks.

8.2 Cancellable Competitive Code

To avoid attackers to use the line features in original Competitive Code for brute-force attacks, a random orientation field is inserted into the feature extractor to generate noise-like feature codes. Thus, the feature extractor in Eq. 4.3 becomes

$$j = \arg\max_{p} \int\int I(x,y)\psi_R(x,y,\omega,\theta_p+\alpha(x_o,y_o),\kappa)dxdy\,, \tag{8.1}$$

where $\alpha(x_0, y_0) \in \{0, \pi/6, 2\pi/6, 3\pi/6, 4\pi/6, 5\pi/6\}$ is a random field. The six filters form a random orientation filter bank (ROFB). This random field would not degrade the original performance of Competitive Code since it does not affect the angular distance between two winning indexes. However, this random field does not generate a non-invertible cancellable biometrics. It is noted that non-invertible cancellable biometrics tends to reduce accuracy [157]. Figure 8.1 shows an original Competitive Code and two corresponding cancellable Competitive Codes. It should be mentioned that similar ideas are employed for cancellable iris and for generating cryptographic key from biometrics [99-100].

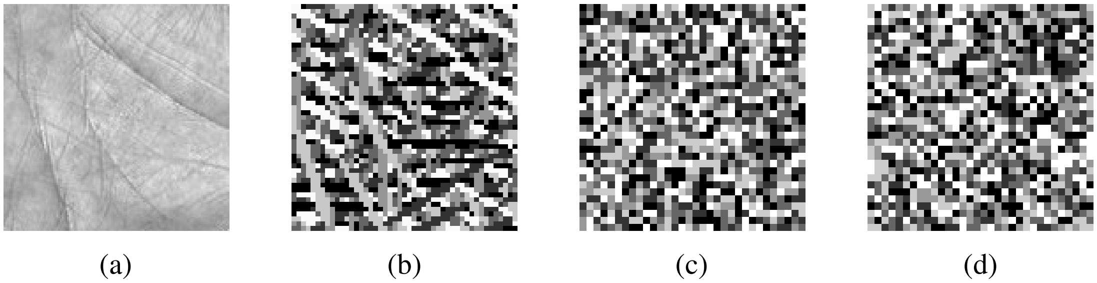


Figure 8.1 Illustration of cancellable Competitive Codes. (a) Preprocessed Image, (b) Original Competitive Code and (c)-(d) Competitive Codes from different random fields.

8.3 A Probabilistic Model for Analyzing Brute-Force Break-ins

The study of brute-force break-ins requires a probabilistic model that describes the relationship between the number of attacks and the probability of a false acceptance. Therefore, it is necessary to establish a probabilistic model for the angular distance given in Eq. 4.6. To simplify the model, it is assumed that all preprocessed palmprint images are clear and devoid of non-palmprint pixels. This will allow us to neglect the masks and the normalization terms. Using either the integer representation or the bitwise representation of Competitive Code, this analysis would have the same result, but for purposes of presentation it is more convenient to use the integer representation. Thus, the angular distance given in Eq. 4.4 is considered for the following analysis.

Let $W=[w_o, w_1, w_2, w_3]$ be a random vector where w_i is the number of $A(P_{x,y}, Q_{x,y})=i$ in Eq. 4.4 and let p_i be the probability of $A(P_{x,y}, Q_{x,y})=i$. As a result, the angular distance described in Eq. 4.4 can be rewritten as $A_f(P,Q)=WK^T$, where $K=[0, 1, 2, 3]$. It is assumed that p_i is stationary and $A(P_{x,y}, Q_{x,y})$ is independent. By stationary I mean that p_i does not depend on the position *(x, y)*. Using these assumptions, it is inferred that *W* follows multinomial distribution i.e.

$$f(w_0, w_1, w_2, w_3)=\frac{n!}{w_0!w_1!w_2!w_3!}p_0^{w_0}p_1^{w_1}p_2^{w_2}p_3^{w_3}, \tag{8.2}$$

where *n* is equal to 1024, the size of the Competitive Codes. Thus, the probability density function of the angular distance is

$$\Pr(A_f(P,Q)=t)=\sum_{W\ni WK^T=t} f(w_o, w_1, w_2, w_3). \tag{8.3}$$

This distribution can be regarded as a projected multinomial distribution.

Let $\Pr(A_f(P,Q)<t)=F(t)$ and therefore $\Pr(A_f(P,Q)\geq t)=1-F(t)$. The probability of the final angular distance A_F being greater than the threshold *t* is

$$\Pr(A_F(P,Q)\geq t)=(1-F(t))^m, \tag{8.4}$$

where m, the number of translated matchings is 25. If z independent comparisons are made, the probability of all the final angular distances being greater than or equal to t is

$$\Pr(A_F(P_i, Q_i) \geq t \mid \forall i = 1,..., z) = (1 - F(t))^{mz}, \tag{8.5}$$

where P_i and Q_i represent different Competitive Codes. Consequently, the probability of at least one of the final angular distances being shorter than t is

$$\Pr(\min_i(A_F(P_i, Q_i)) < t) = 1 - (1 - F(t))^{mz}. \tag{8.6}$$

We can now analyze brute-force attacks against the palmprint system using Eq. 8.6. For verification, each attackers' template, Z_i is compared only with the templates associated with a particular user. It is assumed that each user only has one template, Q, in the database and to attack the system the hackers submit z templates. The probability of a false acceptance for verification is

$$\Pr(\min_i A_F(Z_i, Q) < t) = 1 - (1 - F(t))^{mz}, \tag{8.7}$$

the same as in Eq. 8.6.

For identification, each submitted templates, Z_i as a brute-force attack is compared with all the templates in the database. Let the templates in the database be Q_j where j=1,.., b. As in the previous discussion, the number of templates for the brute-force attack is z. Therefore, the probability of false acceptance occurring in an identification system with b templates in the database is

$$\Pr(\min_{i,j} A_F(Z_i, Q_j) < t) = 1 - (1 - F(t))^{mzb}. \tag{8.8}$$

Eq. 8.7 for verification and Eq. 8.8 for identification each share the same form so, for simplicity of presentation, in the following experiments only verification is considered.

8.4 Security Measures

8.4.1 Replay Attacks

Although cryptography is not suitable for protecting the whole system, it is good for defending against replay attacks at Point 4 since only several templates are required to encrypt and to decrypt in each identification process. To protect this data link, a one time pad, which in cryptography is regarded as perfect secrecy is applied. Random bits, R with the same size as Competitive Code, P^B (bitwise representation of a Competitive Code, P) are generated. Mathematically, the encryption process is $P^B \otimes R$ and the decryption process is $P^B \otimes R \otimes R$. It should be remembered that the random bits R are used one time only.

8.4.2 Database Attacks

When carrying out database attacks, attackers will either insert unregistered templates or modify the templates in the database. So that the system can detect these attacks, the templates are embedded secret messages. If each user provides only one template in the database, a part of winning indexes is changed to specific values and this acts as the secret message. A secret code is defined as (x_i, y_i, v_i), where i=1,…,m and m is the length of the secret message. The original winning indexes at the position, (x_i, y_i) are replaced by v_i i.e. $P(x_i, y_i) = v_i$. It is important to note that different messages are embedded in different templates. Otherwise, attackers can uncover the messages easily.

If each user has more than one template in the database, the correlation between templates from the same user can be used to extract the messages. For example, two different messages are hidden in two nearly identical templates. Let the templates with messages be P_{s1} and P_{s2} and their error map is defined as $e(x, y) = A(P_{s1}(x, y), P_{s2}(x, y))$, the angular distance between two winning indexes. Since the two templates are nearly identical, the messages have high probability in the positions where $e(x, y) \neq 0$. If two templates are very different and the same message is hidden on them, the messages have to be in the positions where $e(x, y) = 0$. Therefore, we need a more complex scheme to hide the secret messages. The secret messages are constituted by two parts, S_{ID} =(x_{IDi} , y_{IDi}, v_{IDi}) and S_T =(x_{Ti} , y_{Ti}, v_{Ti}). All the templates from the same palm are embedded the same S_{ID}. However, their S_Ts are different. To hide S_{ID}, one of template P_0 is selected for aligning other templates and the vertical and horizontal translations (h_j, v_j) between P_0 and P_j are computed. To take into account the translation, I set $P_j(x_{IDi} + h_j, y_{IDi} + v_j) = v_{IDi}$ for S_{ID} while I set $P_j(x_{Ti}, y_{Tj}) = v_{Ti}$ for S_T.

To further hide the correlation between the templates from the same palm, random bits R_c are used to encrypt the c^{th} templates of all users, i.e. $P^B \otimes R_c$. Although R_c is not a one-time pad, not perfect secrecy, it raises the level of difficulty involved in using the correlation to carry out database attacks. Some may immediately think that we need to decrypt the templates for matching. It should be noted that decrypting input Competitive Code, Q^B by R_c i.e. $Q^B \otimes R_c$ and decrypting templates in the database are equivalent activities and the number of R_c which is equal to number of templates per user is limited. Therefore, the computation cost would not be drastically increased. Secret messages are also hidden in input Competitive Codes to detect replay attacks at Point 4.

8.5 Experimental Results

8.5.1 Model Validation and Experimental Results for Brute-Force Attacks

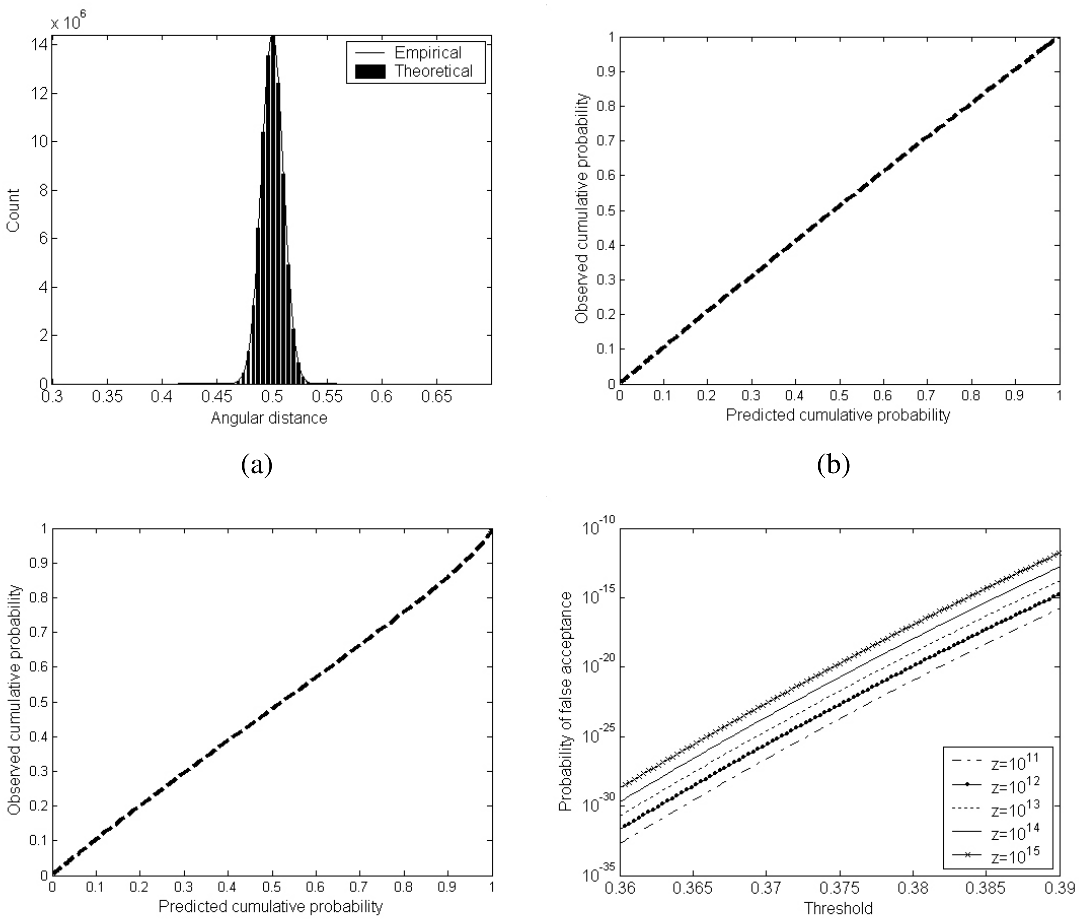
The use of the probabilistic model to analyze brute-force attacks requires us to make some assumptions when obtaining the model parameters p_i. It is assumed that the winning indexes of Competitive Code Q follow independent uniform distributions. In other words, $\Pr(Q_{x,y} = v) = 1/6$, for all v=0, 1, 2, 3, 4, and 5. This assumption holds since the random field is formed by independent uniform distributions. However, this probabilistic model does not require any assumptions as to the winning indexes of the artificial Competitive Codes, Z_i. Let c_v be the probability of the winning index of Z_i being equal to v.

Using Table 4.1, we can infer that $p_o = \sum_{v=0}^{5} c_v / 6 = 1/6$, $p_1 = 2\sum_{v=0}^{5} c_v / 6 = 1/3$, $p_2 = 2\sum_{v=0}^{5} c_v / 6 = 1/3$ and $p_3 = \sum_{v=0}^{5} c_v / 6 = 1/6$.

Now that all the model parameters are determined, a simulation is run to validate the proposed model. 11,074 palmprint images from 568 different palms are collected for this simulation. First of all, 100 different random fields are used to compute the Competitive Codes. Then, uniform distribution is used to generate 100 artificial Competitive Codes to attack each Competitive Code. Each artificial Competitive Code is matched with the true Competitive Code as a brute-force attack. The distribution of the winning indexes of true Competitive Code is listed in Table 8.1 demonstrating that the winning index follows uniform distribution. The probabilities, p_0, p_1, p_2, and p_3 at different positions are listed in Figure 8.2, where black represents probability zero while white represents probability one. Figure 8.2 demonstrates that the stationary assumption for p_i is held. The empirical distribution and the proposed theoretical distribution of non-translated matchings are given in Figure 8.3(a). Figure 8.3(b) and (c) are the plot of the predicted cumulative probability against the observed cumulative probability from non-translated matchings and translated matchings, respectively. Figure 8.3(a)-(c) demonstrate the predictive power of the proposed model.

Now the proposed model is used to estimate the probability of successful break-ins. Figure 8.3(d) plots the probability of false acceptance against different thresholds. Only the threshold between 0.36 and 0.39 is shown since Competitive Code generally operates in this range. Assume that Competitive Code can make 1 million comparisons, 10 times faster than the current implementations. The corresponding computation times for $z=10^{11}$, 10^{12}, 10^{13}, 10^{14} and 10^{15} are 1.16 days, 11.5 days, 115 days, 3.17 years and 31.7 years, respectively. The computation times and the probabilities of false acceptances demonstrate that it is impossible to use brute-force to break into the system based on Competitive Code.

Figure 8.2 (a)-(d), the estimated p_0-p_3 at different positions, respectively, where black represents probability zero while white represents probability one



(a) (b)

(c) (d)

Figure 8.3 Model validation and predications (a) empirical and theoretical distributions from non-translated matchings, (b) a plot of predicted cumulative probability against observed cumulative probability for non-translated matchings, (c) a plot of predicted cumulative probability against observed cumulative probability for translated matching and (d) the probability of false acceptances against different thresholds.

Table 8.1 Distribution of winning indexes

Winning index	0	1	2	3	4	5
Probability	0.166	0.167	0.168	0.166	0.166	0.167

8.5.2 Security Trade-off

Although longer messages provide greater security, they can also degrade the recognition performance. In this section, the accuracy of Competitive Code with message lengths of, 0, 8, 16, 24 and 32 winning indexes are examined. 11,074 palmprint images collected from 284 subjects are collected for this experiment. The detailed information about this dataset can be found at Chapter 3.

To reliably estimate the accuracy of Competitive Code with different messages, each of the palmprint images was compared only with all of the palmprint images in the database from different occasions. A matching is considered as a genuine matching if two palmprints are from the same palm. Otherwise, it is considered as an imposter matching. The number of genuine and imposter matchings in each experiment are 54,081 and 30,603,388, respectively. The genuine and imposter distributions are estimated by the genuine and imposter matchings and their corresponding receiver operating characteristic (ROC) curves are given in Figure 8.4. Figure 8.4 illustrates that the degradation of performance is not serious even when the message length is 32 winning indexes.

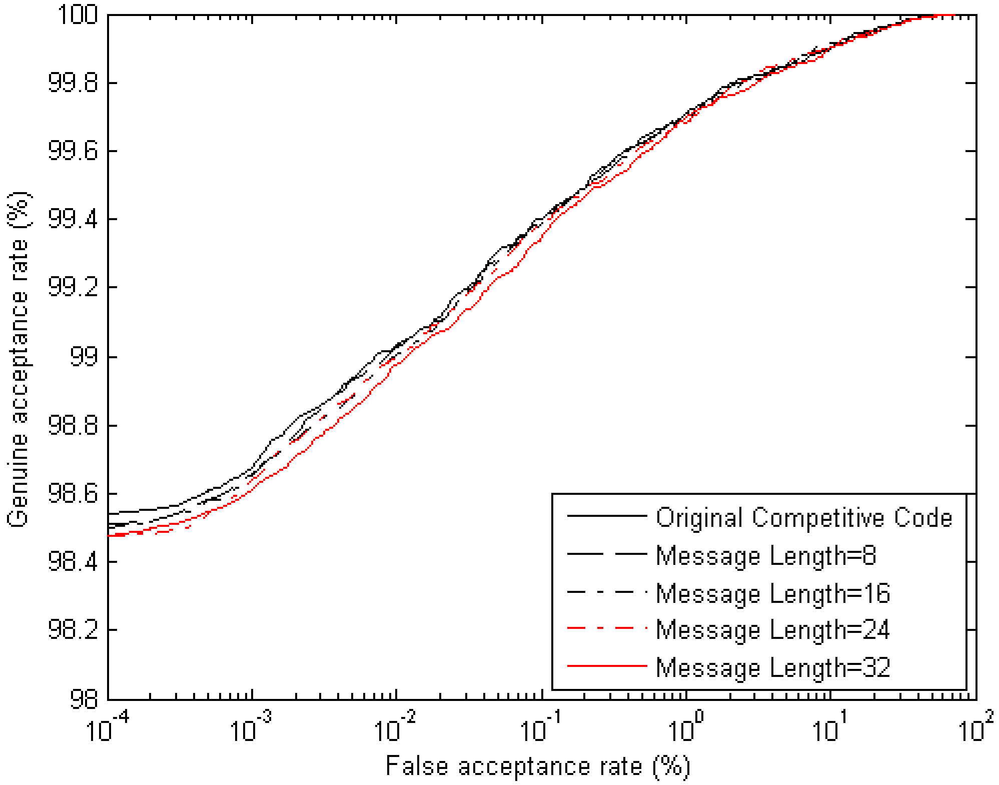


Figure 8.4 Comparison of Competitive Codes with different lengths of messages.(color figure)

8.6 Analysis of the Security Measures

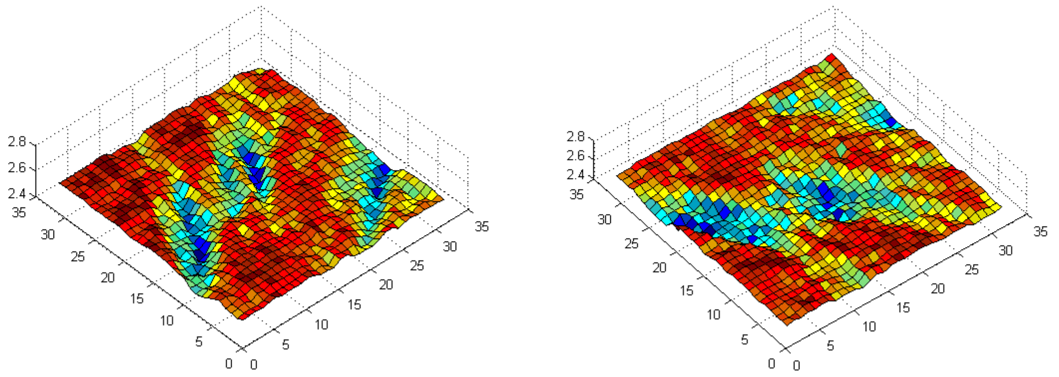
In this section, the effectiveness of the security measures is analyzed. Firstly, the number of effective cancellable templates is estimated. Then, probability models are developed to compute the probability of break-ins using database attacks. Replay attacks at Point 4 are not considered in this analysis since one time pad is regarded as perfect secrecy.

8.6.1 Ability of Template Re-issuance

The ROFB described in Section 8.2 can generate $6^{1024} \approx 10^{797}$ different Competitive Codes for one single image. Although this number is extremely large, not all the templates can be used for re-issuances. The angular distance between compromised and reissued templates can be shorter than the decision threshold. Since the random fields are independent, matching compromised templates and reissued templates is equivalent to match two random synthetic templates. Therefore, the projected multinomial distribution developed for analyzing brute-fore attacks can be applied to estimate the probability of the angular distance between compromised and reissued templates being shorter than a threshold *T* [5]. According to this probabilistic model, when the threshold is 0.39, which is generally used in the system, 10^{15} templates can be reissued and corresponding probability of using compromised templates to break in is 10^{-12} [5]. These numbers demonstrate that the probability of a successfully break-in into the system using compromised templates is extremely low and the number of effective cancellable templates is numerous.

8.6.2 Analysis of Database and Replay Attacks

To perform database attacks, attackers firstly need to estimate the random field and the random bits R_c to adjust their Competitive Codes from unregistered users, which can be generated based on my publications. Then, they either insert the adjusted templates directly into the database or combine the unregistered template with registered templates to form a new template and insert it into the database. So far, I do not have any effective approaches to estimate the random field and R_c. If the entropies of winning indexes are low, attackers are easy to estimate them. The entropy at point (x, y) is defined as $E(x, y) = \sum_{i=0}^{5} -\Pr(P_{x,y} = i)\log(\Pr(P_{x,y} = i))$. The log is base 2 in the following experiments. Figure 8.5(a) and (b) show the entropies of winning indexes of left and right palms, estimated by 1,200 images from 150 the left and the right palms, respectively. The lowest entropies of both left and right palms are 2.4. Comparing the maximum entropy bound, $-\log(1/6) = 2.6$, they are still high.



(a) (b)

Figure 8.5 The entropies of winning indexes. (a) left palm and (b) right palm

In the following analyses, I study the probabilities of an insert template and a registered template having the same secret messages assuming that attackers can perfectly estimate the random field and the random bits R_c. If attackers replace a registered template with an unregistered template directly, the probability of the two templates has the same message is 6^{-m}, where m is length of the message. For m=8, 16, 24 and 32, the corresponding probabilities are 5.9×10^{-7}, 3.5×10^{-13}, 2.1×10^{-19} and 1.3×10^{-25}, respectively.

Attackers may use a more intelligent approach to increase the probability of successful break-ins. They can select some winning indexes from a registered template and the others from an unregistered template to form a new template for database attacks. Let us consider a simple case first. Each user has only one template in the database. Let the number of winning indexes selected from the registered template be M and the number of message codes being selected be u. In other words, the number of winning indexes from the unregistered template is 1024-M. If u is considered as a random variable, the probability of the M winning indexes containing r secret codes can be computed by hypergeometric distribution, whose probability density function is defined as,

$$h(r) = \frac{{}_{m}C_{r}\ {}_{1024-m}C_{M-r}}{{}_{1024}C_{M}}. \tag{8.9}$$

Therefore, the probability of the combined template and the registered template have the same secret message is $\sum_{r=0}^{m} h(r)/6^{m-r}$. Table 8.2 lists the numerical values.

Table 8.2 The probability of combined and registered templates having the same secret message in the case that system stores only one template.

Number of winning indexes in combined template is drawn from registered templates	Message length, *m*			
	8	16	24	32
256	3.8×10^{-4}	1.4×10^{-7}	4.7×10^{-11}	1.5×10^{-14}
512	1.3×10^{-2}	1.7×10^{-4}	2.1×10^{-6}	2.5×10^{-8}

If each user has more than one template, attackers can make use of the correlation between the templates of the same user to further increase the probability of successful break-ins. First of all, they can align all templates based on one of the templates and compute the error map defined as $e(x,y)=\sum_{i=1} A(P_{s0}(x,y), P_{si}(x+h_i, y+v_i))$. The locations of the message S_{ID} are in the zeros of error map and between $2<x\leq 30$ and $2<y\leq 30$ because of the translations. Let the number of zeros between $2<x\leq 30$ and $2<y\leq 30$ be v_1. If attackers select all the corresponding winning indexes from P_{s0}, the combined templates have to include $S_{ID.}$ Some secret codes of S_T are also included in this selection process. Let the number of secret codes of S_T being selected is v_2. The sizes of S_{ID} and S_T both are set to *m*/2 in this analysis. Thus, the number of secret codes that were not selected is *m/2*- v_2. Since attackers have selected v_1 winning indexes, the number of selection for the rest of unselected secret codes is *M*-v_1 and the number of rest of winning indexes in P_{s0} is 1024- v_1. The probability of *r* secret codes being selected from the rest of winning indexes is

$$g(r)=\frac{{}_{m/2-v_2}C_r \; {}_{1024-v_1-m/2+v_2}C_{M-v_1-r}}{{}_{1024-v_1}C_{M-v_1}}. \qquad (8.10)$$

Thus, the probability of the combined template and registered template having the same message code is $\sum_{r=0}^{m/2-v_2} g(r)/6^{m/2-v_2-r}$. In addition to the parameters *M* and *m*, the other two parameters, v_1 and v_2 are needed to compute this probability.

In the following experiments, the system stores three templates for each palm to estimate v_1 and v_2. All the templates are from palmprints collected the first session. The experiment is repeated 10 times by randomly selecting palmprints from the first session to construct the registered templates. In total, 5,680 sets of v_1 and v_2 are obtained. Table 8.3(a) and (b) report the average and maximum probabilities of combined templates and registered templates having the same message, respectively. Table 8.2 and Table 8.3 demonstrate that attackers can make use of the correlation between templates to increase the probability of successful break-ins. However, the probabilities are still very low if the message length is longer than or equal to 32.

We should remember that database attacks on biometric systems are in fact different from cipher attacks. For cipher attacks, attackers have encrypted data and try to recover the original data, such as text.

Generally speaking, they can perform unlimited attempts and know the correctness of decrypted data. However, attackers of biometric systems have only very limited number attempts available to them in which to carry out database attacks since they need to insert the combined templates into databases in order to examine them for correctness.

Table 8.3 The average (a) and maximum (b) probability of combined and registered templates having the same secret message in the case that system stores only three templates.

(a)

Number of winning indexes in combined template is drawn from registered templates	Message length, *m*			
	8	16	24	32
256	2.8×10^{-4}	1.2×10^{-6}	5.8×10^{-9}	4.2×10^{-11}
512	2.9×10^{-2}	1.0×10^{-3}	4.6×10^{-5}	1.9×10^{-6}

(b)

Number of winning indexes in combined template is drawn from registered templates	Message length, *m*			
	8	16	24	32
256	5.0×10^{-2}	4.8×10^{-4}	3.3×10^{-6}	1.4×10^{-7}
512	3.7×10^{-1}	5.7×10^{-2}	3.1×10^{-3}	1.9×10^{-4}

8.7 Conclusion

In this chapter, I employ random orientation field bank for template re-issuance, one time pad for defending replay attacks and secret messages for detecting replay and database attacks and develop a projected multinomial distribution, which describes the relationship between the probability of false acceptance and the number of attacks for analyzing brute-force attacks. This study demonstrates that random orientation field bank can re-issue numerous templates and secret messages can detect database and replay attacks effectively if the message length is longer than or equal to 32. The experimental results also show that the presence of messages in the templates results in a low degradation of accuracy. This experiment demonstrates once again the trade-off described in Figure 1.2. This study also shows that when the system threshold is set to lower than 0.39, it is computationally infeasible to break into the palmprint system using brute-force attacks.

Chapter 9 Conclusion and Future Directions

9.1 Summary

In this thesis, I first modify PalmCode by using multiple elliptical Gabor filters, a fusion rule, and a dynamical threshold to develop a palmprint identification algorithm called Fusion Code. Experimental results demonstrate that Fusion Code performs better than PalmCode. Although palmprints and irises are different biometrics, PalmCode, Fusion Code and IrisCode are all based on phase features. I have found that orientation fields of palmprints are powerful features that enhance the performance of coding methods for palmprint identification. Using these features, I develop a new coding method called Competitive Code. A Gabor filter bank and a winner-take-all rule are used to estimate the orientation fields, and a bitwise angular distance is developed for high-speed matching. A comparison demonstrates that the Competitive Code performs better than other palmprint recognition methods.

Although many coding methods have been proposed based on IrisCode for iris and palmprint identification, we lack an analysis of IrisCode. To better understand IrisCode and to uncover the theory of coding methods, I have analyzed IrisCode. This analysis reveals that

- the locus of Gabor phase is an ellipse;
- under suitable parameterization, the locus is a circle;
- IrisCode is a clustering process with four prototypes;
- angular distance can be represented by bitwise hamming distance;
- Gabor filter can be regarded as a steerable filter;
- IrisCode and other coding methods are under the same framework and
- the theoretical evidence of the binomial imposter distribution of IrisCode is incomplete.

Based on this analysis, a precise phase representation inheriting all the properties of IrisCode has been developed for iris identification. I also provide a note about fair comparison of IrisCode.

In addition to developing new coding methods and analyzing IrisCode, I have systemically studied genetically identical palmprints. This study shows that palmprints have enough distinctive features for classifying identical twins' palmprints, although principal lines and some wrinkles are genetically dependent.

For protecting and analyzing the security of a palmprint identification system based on Competitive Code, I propose projected multinomial distribution, random orientation field, one-time pad and template watermarking for investigating brute-force attacks, reissuing new templates, and defending against replay and database attacks. The projected multinomial distribution indicates that it is computationally infeasible to break into a palmprint identification system based on Competitive Code by brute-force attacks. On the other hand, the projected multinomial distribution points out that the random orientation field can re-issue more than enough effective templates. Using hypergeometric distribution to model replay and database attacks, the analysis demonstrates that the probability of successfully breaking into the system is extremely low if the length of secret messages hidden in the templates is longer than 32 winning indexes.

9.2 Contribution of this Thesis

At the end of this thesis, a summary of contribution of thesis is provided, for evaluation only.

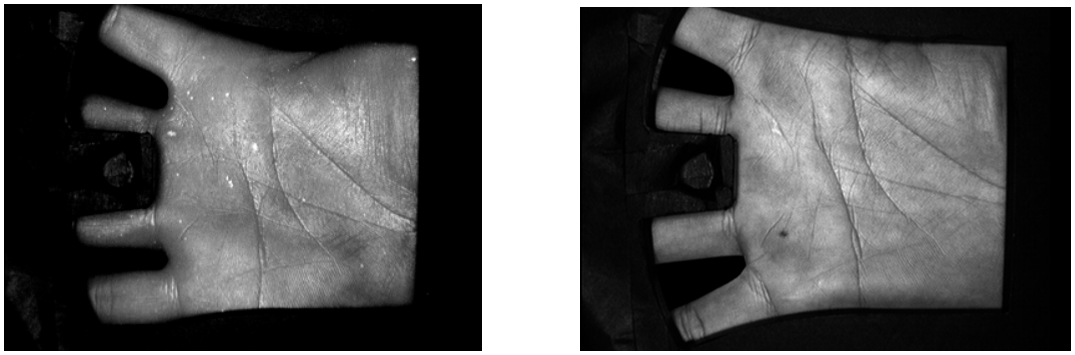
- I use fusion rule and multiple elliptical Gabor filters to improve PalmCode.
- I identify orientation field of palmprints as power features for personal identification.
- Using orientation field as features, I develop Competitive Code, which is constituted by a novel coding scheme and a novel bitwise angular distance. Competitive Code is the first palmprint identification algorithm completely employing the orientation field of palmprints as features.
- I provide a detailed analysis of IrisCode, which first appeared thirteen years ago. This is the first and only document that provides a detailed analysis of this method. This analysis is the core contribution of this thesis.
- Making use of the analysis of IrisCode, a framework is provided to link up all the coding methods for palmprint and iris identification.
- Using Competitive Code to examine genetically identical palmprints, I show that they have enough distinctive features for classifying identical twins. The genetically dependent features are also uncovered. This is the first systematic study of genetically identical palmprints for personal identification.
- I develop a projected multinomial distribution for analyzing brute-force break-ins.
- I use random orientation field for template re-issuance.
- I propose template watermarking for defending against replay and database attacks. This is first document to consider replay and database attacks against palmprint systems.

9.3 Future Work and Directions

This thesis has covered various issues in palmprint identification, including accuracy, matching speed, template re-issuance, brute-force attacks, replay attacks, database attacks and genetically dependent features in palmprints. It has also presented a detailed theoretical analysis of IrisCode and a framework to link up the existing coding methods for palmprints and irises. On the base of this thesis, there are several potential directions:

- Accuracy is always an important objective for biometric systems. Further enhancing the accuracy of the proposed algorithms in this thesis is not difficult if we sacrifice matching speed. I can increase the precision of the orientations in Competitive Code or increase number of sample points in Competitive Code. I also can combine different biometrics such fingers and hand geometry for fusion. The challenge is to enhance accuracy and maintain the matching speed simultaneously. In the coming future, I will make use of the masks of Competitive Code to achieve this goal.
- Coding methods are in fact clustering processes. Their performance highly depends on the quality of alignment algorithms. Even though the palmprint alignment algorithm [7, 164] developed by me has been widely employed by other researchers, further improving this algorithm is necessary.
- Some palmprint researchers report extremely high accuracy based on very simple methods such as PCA. However, the reported accuracies highly depend on evaluation schemes [35]. Open and standard evaluation schemes should be established.
- In Chapter 5, I have pinpointed that the imposter distribution of IrisCode does not theoretically follow a binomial distribution. It is valuable to study the theoretical imposter distributions of IrisCode and other coding methods.
- The coding methods have been extensively applied to irises and palmprints. It is possible to apply or modify the coding methods for other biometrics such as fingerprint and face.
- In Chapter 7, I have used Competitive Code to identify the genetically dependent features. In dermatoglyphic society, genetically dependent features are identified manually. In fact, we can use a computational approach to speed up the process.
- Some basic issues of using palmprints for personal identification are in fact not well addressed. For instance, it is known that ridges in palmprints are, like fingerprints, stable over an entire lifetime, but the stability of principal lines and wrinkles has not been systemically investigated.
- In Chapter 8, I have addressed various security issues, including template re-issuance, brute-force attacks, replay attacks and database attacks for the palmprint identification system. However, current palmprint systems are possible to be spoofed by fake palmprints. Figure 9.1(a) shows a fake palmprint, which is the first fake palmprint reported in the literature and Figure 9.1 (b) shows the corresponding genuine palmprint. It is essential to develop next generation palmprint sensors and algorithms for liveness detection.
- In addition to developing new algorithms and sensors for defending against the attacks, analyzing and evaluating vulnerabilities of palmprint systems are also necessary [65].

In addition to these further directions, biometric researchers are in fact facing some open problems. Most biometric systems are examined only on zero effort attacks (general false acceptance rates). Do they still survive if experts attack them? Furthermore, how can we objectively evaluate and compare the security levels of biometric systems? It should be recognized that successfully breaking into biometric systems depends on money, time and knowledge. Biometric researchers are facing a dilemma. As we publish papers to disclose our findings and algorithms for the sake of distributing knowledge to our society and the next generation, we are also providing the information that potential attackers require breaking into our systems.



(a) (b)

Figure 9.1 (a) A fake palmprint and (b) the corresponding genuine palmprint.

Appendix 1 Ma el al. method based on the operator "≥" and "<" and hamming distance

The appendix shows that the coding method in [108] is based on the operators "≥" and "<" and hamming distance.

Ma et al [108] first apply a quadratic spline wavelet to decompose the one dimensional iris signals into several scales but only two scales are used in the their feature extraction process. Ma et al record the locations, *d*, and types of all the local extremum points (minimum and maximum points) as features. To obtain stable feature points, they remove pairs of adjacent points having an amplitude difference smaller than a predetermined threshold. Since there are only two types of extremum points and since two adjacent points of a maximum point must be minimum points, just one pointer, p_i, will suffice to denote all the type information in scale *i*. The pointer, p_i, is set to 1 if the first extremum point is a minimum point; otherwise p_i is set to –1. Finally, for each decomposed signal S_i, the pointers and locations of feature points are stored as the following form

$$f_i = \{d_1, d_2, ...d_i, ...d_m; d_{m+1}, d_{m+2}, ..., d_{m+n}; p_1, p_2\}\,. \tag{A1}$$

To exploit XOR operations for effective matching, as in IrisCode, the original features in each scale are transformed into a binary feature vector of a fixed length, *L*. Figure A1 illustrates this process, which is called feature transform. If p_j is –1, the first d_1-1 components in the binary feature vector are set to 0; otherwise they are set to 1. Then, all the components in the binary feature vector corresponding to maximum and minimum points are set to 0 and 1, respectively. All the other components between d_i and d_{i+1} are set to 0 if d_i corresponds to a maximum point; otherwise, they are set to 1.

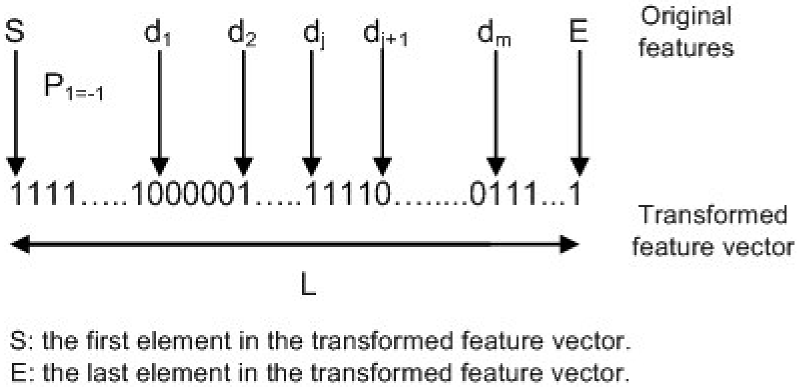

S
d1
d2
dj
dj+1
dm
E
Original features
P1=-1
1111.....1000001.....11110........0111...1
Transformed feature vector
L
S: the first element in the transformed feature vector.
E: the last element in the transformed feature vector.

Figure. A1 Illustration of Ma et al's feature transform

This coding scheme is more complicated than other coding schemes but is very similar to the standard coding scheme defined in Eqs. 5.22-5.23. To demonstrate the relationship between the coding scheme of Ma et al and other coding schemes, a process is applied to S_i, the wavelet transformed signal, to obtain a signal v_i for coding based on "≥" and "<". Each extremum point in v_i has a corresponding extremum point in S_i with the same type, location and amplitude but the amplitude difference between each pair of adjacent extremum points in v_i has to be larger than the predetermined threshold. A simple way to obtain

v_i is to interpolate the extremum points in Eq. 5.25. Since the maximum and minimum points in Eq. A1 correspond to the zero-crossings in dv_i / dx, their coding scheme can be rewritten as follows.

If $dv_i / dx \mid x = x_j$>0, then Bj=1;

otherwise Bj=0.

Therefore, the Ma et al coding scheme can be rewritten based on the operators "≥" and "<" and can be considered as standard coding scheme.

Appendix 2 Condition of Circular Locus of Gabor Phase

Eq. 5.16 demonstrates that the locus of $\vec{Z}(\varphi)$ is an ellipse with respect to φ. In fact, the locus can be further constrained. Under suitable parameterization, it is a circle. Mathematically, $\lim_{k\to\infty}(\|M_R\|^2 - \|M_I\|^2) = 0$, where $k = \omega\beta$.

Considering $\|M_R\|^2 - \|M_I\|^2$

$$= \int\int\left(e^{-\frac{\rho^2}{\alpha^2}} e^{-\frac{\phi^2}{\beta^2}} \cos(\omega\phi)\right)^2 d\rho d\phi - \int\int\left(e^{-\frac{\rho^2}{\alpha^2}} e^{-\frac{\phi^2}{\beta^2}} \sin(\omega\phi)\right)^2 d\rho d\phi$$

$$= \int\int e^{-\frac{2\rho^2}{\alpha^2}} e^{-\frac{2\phi^2}{\beta^2}} \left(\cos^2(\omega\phi) - \sin^2(\omega\phi)\right) d\rho d\phi$$

$$= \int e^{-\frac{2\rho^2}{\alpha^2}} d\rho \int e^{-\frac{2\phi^2}{\beta^2}} (\cos^2(\omega\phi) - \sin^2(\omega\phi)) d\phi$$

$$= \frac{\alpha\sqrt{2\pi}}{2} \int e^{-\frac{2\phi^2}{\beta^2}} (\cos^2(\omega\phi) - \sin^2(\omega\phi)) d\phi$$

Let $\gamma = \frac{k}{\beta}\phi$. Thus

$$= \frac{\alpha\beta\sqrt{2\pi}}{2k} \int e^{-\frac{2\gamma^2}{k^2}} (\cos^2(\gamma) - \sin^2(\gamma)) d\gamma$$

$$= \frac{\alpha\beta\sqrt{2\pi}}{2k} \int e^{-\frac{2\gamma^2}{k^2}} \cos(2\gamma) d\gamma$$

Let $2\gamma = \tau$

$$= \frac{\alpha\beta\sqrt{2\pi}}{4k} \int e^{-\frac{\tau^2}{2k^2}} \cos(\tau) d\tau$$

$$= \frac{\alpha\beta\sqrt{2\pi}}{4k} \sqrt{2\pi} k e^{-\frac{k^2}{2}}$$

$$= \frac{1}{2}\alpha\beta\pi e^{-\frac{k^2}{2}}$$

Since α, β, and k are greater than zeros, $\|M_R\|^2 - \|M_I\|^2$ is always greater than zero. However, $\lim_{k\to\infty}\left(\|M_R\|^2 - \|M_I\|^2\right) = \lim_{k\to\infty}\frac{1}{2}\alpha\beta\pi e^{-\frac{k^2}{2}} = 0$

Appendix 3 Equivalent relationship between the bitwise hamming distance and the angular distance

This appendix shows the equivalent relationship between the bitwise hamming distance and the angular distance in Eq. 5.17.

Let two winning indexes be *j-1*, and *j-1+k*, where $1 \le j \le j+k < 2n$. Their angular distance is $\min(k,\, 2n-k)$. Using the coding scheme given in , the winning indexes are represented by j^{th} and $j+k^{\text{th}}$ column vectors of matrix *A*. I would like to prove

$$\sum_{i=1}^{n} a_{i,j} \otimes a_{i,j+k} = \min(k,\, 2n-k).$$

Since all $a_{i,j}$ are either zero or one, $\sum_{i=1}^{n} a_{i,j} \otimes a_{i,j+h} = \sum_{i=1}^{n} \left| a_{i,j} - a_{i,j+k} \right|$

Case 1:

If $j \le n$ and $j+k \le n$

From the definition of *A*, we know $\sum_{i=1}^{n} \left| a_{i,j} - a_{i,j+k} \right| = k$

Case 2:

If $j > n$ and $j+k > n$

As in Case 1, we know $\sum_{i=1}^{n} \left| a_{i,j} - a_{i,j+k} \right| = k$

Case 3:

If $j \le n$ and $j+k > n$ and $k \le n$

Consider $a_{i,j}=1$ and $a_{i,j+k}=1$ (C1)

From the definition of *A*, we have $1 \le i < j$ and $j+k-n \le i \le n$

Then, $j+k-n \le i < j$

The number of *i* satisfying condition (C1) is $\max(0,\, j-(j+k-n))$. (C2)

Since $k \le n$, $\max(0,\, j-(j+k-n)) = n-k$

Consider $a_{i,j}=0$ and $a_{i,j+k}=0$ (C3)

From the definition of *A*, we have $i \ge j$ and $i < j+k-n$

Then $j \le i < j+k-n$

The number of *i* satisfying condition (C3) is $\max(0,\ j+k-n-j)$ (C4)

Since $k \le n$, $\max(0, k-n)$ =0

Thus, $\sum_{i=1}^{n} \left| a_{i,j} - a_{i,j+k} \right| = n-(n-k) = k$

Case 4:

If $j \le n$ and $j+k > n$ and $k > n$

Consider $a_{i,j}=1$ and $a_{i,j+k}=1$ (C5)

From (C2), number of *i* satisfying (C5) is $\max(0,\ j-(j+k-n))$

Since $k > n$, $\max(0, n-k) = 0$

Consider $a_{i,j}=0$ and $a_{i,j+k}=0,$ (C6)

From (C4), number of *i* satisfying (C6) is $\max(0,\ j+k-n-j)$

Since *k*>*n*, $\max(0, k-n) = k-n$

Thus, $\sum_{i=1}^{n} \left| a_{i,j} - a_{i,j+k} \right| = n-(k-n) = 2n-k$

Thus, $\sum_{i=1}^{n} a_{i,j} \otimes a_{i,j+k} = k$ for Cases, 1-3 and $\sum_{i=1}^{n} a_{i,j} \otimes a_{i,j+k} = 2n-k$ for Case 4. Since $2n-k \ge k$ for Cases 1-3 and $2n-k < k$ for Case 4, $\sum_{i=1}^{n} a_{i,j} \otimes a_{i,j+k} = \min(k, 2n-k)$.

Appendix 4 Effective Computation of the Phase When the Locus of $\vec{Z}(\varphi)$ is a Circle

This appendix shows that when the locus of $\vec{Z}(\varphi)$ is a circle, $\arg\max\limits_{\varphi}\int\limits_{\rho}\int\limits_{\phi}\rho IZ(\varphi)d\rho d\phi / \|\rho I\|\|Z(\varphi)\| = \varphi_1$ or φ_2, where $\varphi_1 = \tan^{-1}\left(\int\limits_{\rho}\int\limits_{\phi}\rho IM_I d\rho d\phi \Big/ \int\limits_{\rho}\int\limits_{\phi}\rho IM_R d\rho d\phi\right)$, $\varphi_1 \in [0,\pi)$ and $\varphi_2 = \varphi_1 + \pi$.

Since the locus of $\vec{Z}(\varphi)$ is a circle, $\arg\max\limits_{\varphi}\int\limits_{\rho}\int\limits_{\phi}\rho IZ(\varphi)d\rho d\phi / \|\rho I\|\|Z(\varphi)\|$ can be rewritten as $\arg\max\limits_{\varphi}\int\limits_{\rho}\int\limits_{\phi}\rho IZ(\varphi)d\rho d\phi$. Using Eq. 18, $\int\limits_{\rho}\int\limits_{\phi}\rho IZ(\varphi)d\rho d\phi = \cos(\varphi)C_R + \sin(\varphi)C_I$, where $C_I = \int\limits_{\rho}\int\limits_{\phi}\rho IM_I d\rho d\phi$ and $C_R = \int\limits_{\rho}\int\limits_{\phi}\rho IM_R d\rho d\phi$. Considering $\dfrac{d\cos(\varphi)C_R + \sin(\varphi)C_I}{d\varphi} = -\sin(\varphi)C_R + \cos(\varphi)C_I$ and setting $-\sin(\varphi)C_R + \cos(\varphi)C_I = 0$, we have $\varphi = \tan^{-1}(C_I/C_R)$. Since $\cos(\varphi)C_R + \sin(\varphi)C_I$ is a continuous periodic function, one of the φ_i corresponds to a maximum and the other corresponds to a minimum

Using the second order derivative, we can demonstrate that the φ_i corresponding to the maximum satisfies the following inequalities. If $\cos(\varphi_i) \neq 0$, $-C_R/\cos(\varphi_i) < 0$. If $\cos(\varphi_i) = 0$, $-\sin(\varphi_i)C_I < 0$.

References:

[1] W.K. Kong and D. Zhang, “Palmprint texture analysis based on low-resolution images for personal authentication”, *in Proceedings of 16th International Conference on Pattern Recognition*, vol. 3, pp. 807-810, 2002.

[2] A. Kong, D. Zhang and G. Lu, “A study of identical twins palmprint for personal verification”, *Pattern Recognition*, vol. 39, no, 11, pp. 2149-2156, 2006.

[3] A.W.K. Kong and D. Zhang, “Competitive coding scheme for palmprint verification”, *in Proceedings of International Conference on Pattern Recognition*, vol. 1, pp. 520-523, 2004.

[4] A. Kong, D. Zhang and M. Kamel, “Palmprint identification using feature-level fusion”, *Pattern Recognition*, vol. 39, no. 3, pp. 478-487, 2006.

[5] A. Kong, D. Zhang and M. Kamel, “A analysis of brute-force break-ins of a palmprint verification system”, *IEEE Transactions on Systems, Man and Cybernetics, Part B*, vol. 36, no. 5, pp. 1201-1205, 2006.

[6] A. Kong, K.H. Cheung, D. Zhang, M. Kamel and J. You, “An analysis of Biohashing and its variants”, *Pattern Recognition*, vol. 39, no. 7, pp. 1359-1368, 2006.

[7] D. Zhang, W.K. Kong, J. You and M. Wong, “On-line palmprint identification”, *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 25, no. 9, pp. 1041-1050, 2003.

[8] T. Connie, A.T.B. Jin, M.G.K. Ong and D.N.C. Ling, “An automated palmprint recognition system”, *Image and Vision Computing*, vol. 23, no. 5, pp. 501-515, 2005.

[9] C.C. Han, “A hand-based personal authentication using a coarse-to-fine strategy”, *Image and Vision Computing*, vol. 22, no. 11, pp. 909-918, 2004.

[10] C.C. Han, H.L. Cheng, C.L. Lin and K.C. Fan, “Personal authentication using palm-print features”, *Pattern Recognition*, vol. 36, no. 2, pp. 371-381, 2003.

[11] Y.H. Pang, T. Connie, A. Jin and D. Ling, “Palmprint authentication with Zernike moment invariants”, *in Proceedings of the 3rd IEEE International Symposium on Signal Processing and Information Technology*, pp. 199-202, 2003.

[12] X. Wu, D. Zhang and K. Wang, “Fisherpalms based palmprint recognition”, *Pattern Recognition Letters*, vol. 24, no, 15, pp. 2829-2838, 2003.

[13] G. Lu, D. Zhang and K. Wang, “Palmprint recognition using eigenpalms features”, *Pattern Recognition Letters*, vol. 24, no. 9, pp. 1463-1467, 2003.

[14] L. Zhang and D. Zhang, “Characterization of palmprints by wavelet signatures via directional context modeling”, *IEEE Transactions on Systems, Man and Cybernetics, Part B*, vol. 34, no. 3, pp. 1335-1347, 2004.

[15] J. You, W.K. Kong, D. Zhang and K.H. Cheung, "On hierarchical palmprint coding with multiple features for personal identification in large databases", *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 14, no. 2, pp. 234-243, 2004.

[16] W. Li, D. Zhang and Z. Xu, "Palmprint identification by Fourier transform", *International Journal of Pattern Recognition and Artificial Intelligence*, vol. 16, no. 4, pp. 417-432, 2002.

[17] S. Ribaric, D. Ribaric and N. Pavesic, "Multimodal biometric user-identification system for network-based applications", *IEE Proceedings, Vision, Image and Signal Processing*, vol. 150, no. 6, pp. 409-416, 2003.

[18] X.Y. Jing and D. Zhang, "A face and palmprint recognition approach based on discriminant DCT feature extraction", *IEEE Transactions on Systems, Man, and Cybernetics — Part B: Cybernetics*, vol. 34, no. 6, pp. 2405-2415, 2004.

[19] S. Ribaric and I. Fratric, "A biometric identification system based on Eigenpalm and Eigenfinger features", *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 27, no. 11, pp. 1698-1709, 2005.

[20] S. Ribaric, I. Fratric and K. Kis, "A biometric verification system based on the fusion of palmprint and face features", *in Proceeding of the 4th International Symposium on Image, Signal and Signal Processing and Analysis,* pp. 15-17, 2005.

[21] A. Kumar and D. Zhang, "Personal authentication using multiple palmprint representation", *Pattern Recognition*, vol. 38, no. 10, pp. 1695-1704, 2005.

[22] X. Wu, D. Zhang, K. Wang and B. Huang, "Palmprint classification using principal lines", *Pattern Recognition*, vol. 37, no. 10, pp. 1987-1998, 2004.

[23] J. L. Wayman, "Technical testing and evaluation of biometric identification devices", in Biometrics Personal identification in Networked Society, edited by A.K. Jain, R. Bolle and S. Pankanti, Kluwer Academic Publisher 1999.

[24] NEC Automated Palmprint Identification System http://www.necmalaysia.com.my/Solutions/PID/products/ppi.html

[25] J.G. Daugman, "High confidence visual recognition of persons by a test of statistical independence", *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 15, no. 11, pp. 1148-1161, 1993.

[26] N.K. Ratha, J.H. Connell and R.M. Bolle, "Biometrics break-ins and band-aids", *Pattern Recognition Letters*, vol. 24, pp, 2105-2113, 2003.

[27] T. Connie, A. Teoh, M. Goh and D. Ngo, "PalmHashing: a novel approach for cancelable biometrics", *Information Processing Letters*, vol. 93, no. 1, pp. 1-5, 2005.

[28] A.B.J. Teoh, D.C.L Ngo and A. Goh, "BioHashing: two factor authentication featuring fingerprint data and tokenised random number", *Pattern Recognition*, vol. 37, pp. 2245-2255, 2004.

[29] C. Poon, D.C.M. Wong and H.C. Shen, "Personal identification and verification: fusion of palmprint representations", *in Proceedings of International Conference on Biometric Authentication*, pp. 782-788, 2004.

[30] L.S. Penrose, "Fingerprints and palmistry", *The Lancet*, vol. 301, no 7814, pp. 1239-1242, 1973.

[31] F. Li, M.K.H. Leung and X. Yu, "Palmprint identification using Hausdorff distance*", in Proceedings of International Workshop on Biomedical Circuits and Systems*, pp. S3/3-S5-8, 2004.

[32] A. Okatan, C. Akpolat and G. Albayrak, "Palmprint verification by using cosine vector", *IJSIT Lecture Note of International Conference on Intelligent Knowledge Systems*, vol. 1, no. 1, pp. 111-113, 2004.

[33] G. Lu, K. Wang and D. Zhang "Wavelet based feature extraction for palmprint identification", *in Proceeding of Second International Conference on Image and Graphics*, pp. 780-784, 2002.

[34] X. Wu, K. Wang and D. Zhang, "Line feature extraction and matching in palmprint", *in Proceeding of the Second International Conference on Image and Graphics*, pp. 583-590, 2002.

[35] K.H. Cheung, A. Kong, D. Zhang, M. Kamel and J. You, "Does EigenPalm work? A system and evaluation perspective", *in Proceedings of International Conference on Pattern Recognition*, vol. 4, pp. 445-448, 2006.

[36] M. Cannon, M. Byrne, D. Cotter, P. Sham, C. Larkin and E. O'Callaghan, "Further evidence for anomalies in the hand-prints of patients with schizophrenia: a study of secondary creases", *Schizophrenia Research*, vol. 13, pp. 179-184, 1994.

[37] M. Gibbons, S. Yoon, S.H. Cha and C. Tappert, "Evaluation of biometric identification in open systems", *Audio- and Video-Based Biometric Person Authentication* (*AVBPA 2005*) *LNCS 3546*, pp. 823-831, 2005.

[38] A. Kong, D. Zhang and M. Kamel, "An anatomy of IrisCode for precise phase representation", *in Proceedings of International Conference on Pattern Recognition*, vol. 4, pp. 429-432, 2006.

[39] Q. Li, Z. Qiu and D. Sun, "Feature-level fusion of hand biometrics for personal verification based on Kernel PCA", *International Conference on Biometrics*, pp. 744-750, 2006.

[40] D. Sun, Q. Li, T. Liu, B. He and Z. Qu, "A secure multimodal biometric verification scheme", *International Workshop on Biometric Recognition Systems*, pp. 233-240, 2005.

[41] C. Poon, D.C.M. Wong and H.C. Shen, "A new method in locating and segmenting palmprint into region-of-interest", *in Proceedings of the 17th International Conference on Pattern Recognition*, vol. 4, pp. 533-536, 2004.

[42] G.M. Lu, K.Q. Wang and D. Zhang, "Wavelet based independent component analysis for palmprint identification", *in Proceedings of International Conference on Machine Learning and Cybernetics*, vol. 6, pp. 3547-3550, 2004.

[43] P. Hennings and B.V.K.V. Kumar, "Palmprint recognition using correlation filter classifiers", *Conference Record of the 38th Asilomar Conference on Signal, Systems and Computers*, vol. 1, pp. 567-571, 2004.

[44] X. Wu, K. Wang and D. Zhang, "Fuzzy direction element energy feature (FDEEF) based palmprint identification", *in Proceedings of International Conference on Pattern Recognition*, vol. 1, pp. 95-98, 2002.

[45] X. Wu, K. Wang and D. Zhang, "Palmprint recognition using directional energy feature", *in Proceedings of International Conference on Pattern Recognition*, vol. 4, pp. 475-478, 2004.

[46] Q. Dai, N. Bi, D. Huang, D. Zhang and F. Li, "M-band wavelets applications to palmprint recognition based on texture features", *in Proceedings Conference on Image Processing*, vol. 2, pp. 893-896, 2004.

[47] K. Dong, G. Feng and D. Hu, "Digital curvelet transform for palmprint recognition", *Lecture Notes in Computer Science*, Springer, vol. 3338, pp. 639-645, 2004.

[48] X. Wu, K. Wang and D. Zhang, "HMMs based palmprint identification", *Lecture Notes in Computer Science*, Springer, vol. 3072, pp. 775-781, 2004.

[49] Y. Li, K. Wang and D. Zhang, "Palmprint recognition based on translation invariant Zernike moments and modular neural network", *Lecture Notes in Computer Science*, Springer, vol. 3497, pp. 177-182, 2005.

[50] A. Kumar, D.C.M. Wong, H.C. Shen and A.K. Jain, "Personal verification using palmprint and hand geometry biometric," *Lecture Notes in Computer Science*, Springer, pp. 668-678, 2003.

[51] X. Wu, K. Wang and D. Zhang, "Wavelet based palmprint recognition", *in Proceeding of the First International Conference on Machine Learning and Cybernetics*, vol. 3, pp. 1253-1257, 2002.

[52] W.W. Boles and S.Y.T. Chu, "Personal identification using images of the human palms", *in Proceedings of IEEE Region 10 Annual Conference, Speech and Image Technologies for Computing and Telecommunications*, vol. 1, pp. 295-298, 1997.

[53] M. Rafael Diaz, C.M. Travieso, J.B. Alonso and M.A. Ferrer, "Biometric system based in the feature of hand palm", *in Proceedings of 38th Annual International Carnahan Conference on Security Technology*, pp. 136-139, 2004.

[54] J.S. Noh and K.H. Rhee, "Palmprint identification algorithm using Hu invariant moments and Otsu binarization", *in Proceeding of Fourth Annual ACIS International Conference on Computer and Information Science*, pp. 94-99, 2005.

[55] J. Doi and M. Yamanaka, "Personal authentication using feature points on finger and palmar creases" *in Proceedings of 32nd Applied Imagery Patten Recognition Workshop*, pp. 282-287, 2003.

[56] Z. Sun, T. Tan, Y. Wang and S.Z. Li, "Ordinal palmprint representation for personal identification", *in Proceeding of Computer Vision and Pattern Recognition*, vol. 1, pp 279-284, 2005.

[57] X. Wu, D. Zhang and K. Wang, "Fusion of phase and orientation information for palmprint authentication", *Pattern Analysis and Applications*, vol. 9, no. 2-3, pp. 103-111, 2006.

[58] S.Y. Kung, S.H. Lin and M. Fang, "A neural network approach to face/palm recognition" *in Proceedings of IEEE Workshop Neural Networks for Signal Processing*, pp. 323-332, 1995.

[59] P.A Recobos Rodrigues and J.D. Landa Silva, "Biometric identification by dermatoglyphics", *in Proceedings of International Conference on Image Processing*, vol. 1, pp. 319-322, 1996.

[60] X. Wu, K. Wang and D. Zhang, "A novel approach of palm-line extraction", *in Proceeding of the Third International Conference on Image and Graphics*, pp. 230-233, 2004.

[61] A. Kumar and D. Zhang, "Integrating shape and texture for hand verification", *in Proceedings of Third International Conference on Image and Graphics*, pp. 222-225, 2004.

[62] A. Kumar and D. Zhang, "Integrating palmprint with face for user authentication", *in Proceedings of Multi Modal User Authentication Workshop*, pp. 107-112, 2003.

[63] A. Kumar and D. Zhang, "Palmprint authentication using multiple classifiers", *in Proceedings of SPIE Symposium on Defence and Security- Biometric Technology for Human Identification*, pp. 20-29, 2004.

[64] A. Kumar and H.C. Shen, "Palmprint identification using PalmCodes"*, in Proceedings of 3rd International Conference on Image and Graphics*, pp. 258-261, 2004.

[65] International Biometric Group: Biometrics Vulnerability and Penetration Testing http://www.biometricgroup.com/biometrics%20vulnerability%20testing.html

[66] G. Feng, K. Dong, D. Hu and D. Zhang, "When face are combined with palmprints: a novel biometric fusion strategy", *Lecture Notes in Computer Science*, Springer, vol. 3072, pp. 701-707, 2004.

[67] L. Shang, D.S. Huang, J.X. Du and C.H. Zheng, "Palmprint recognition using FastICA algorithm and radial basis probabilistic neural network", *Neurocomputing*, vol. 69, no. 13-15, pp. 1782-1786, 2006.

[68] The National Fragile X Foundation http://www.nfxf.org/html/checklist.htm

[69] N. Duta, A.K. Jain and K.V. Mardia, "Matching of palmprints", *Pattern Recognition Letters*, vol. 23, no. 4, pp. 477-485, 2002.

[70] W. Shu and D. Zhang, “Automated personal identification by palmprint”, *Optical Engineering*, vol. 38, no. 8, pp. 2359-2362, 1998.

[71] W. Zuo, K. Wang and D. Zhang, “Bi-directional PCA with assembled matrix distance metric”, *in Proceeding of IEEE International Conference on Image Processing*, vol. 2, pp. 958-961, 2005.

[72] W. Zuo, K. Wang and D. Zhang, “Assembled matrix distance metric for 2DPCA-based face and palmprint recognition”, *in Proceeding of International Conference on Machine Learning and Cybernetics*, vol. 8, pp. 4870-4875, 2005.

[73] G. Feng, D. Hu, D. Zhang and Z. Zhou, “An alternative formulation of kernel LPP with application to image recognition”, *Neurocomputing*, vol. 69, no. 13-15, pp. 1733-1738, 2006.

[74] X. Wu, K. Wang and D. Zhang, “Palmprint authentication based on orientation code matching”, *in Proceeding of 5th International Conference on Audio- and Video- Based Biometric Person Authentication*, pp. 555-562, 2005.

[75] A. Kong and D. Zhang, “Feature-level fusion for effective palmprint authentication” *in Proceedings of International Conference on Biometric Authentication*, vol. 1, pp. 520-523, 2004.

[76] A. Kumar, D.C.M. Wong, H.C. Shen and A.K. Jain, “Personal authentication using hand images”, *Pattern Recognition Letters*, vol. 27, no. 13, pp. 1478-1486, 2006.

[77] A. Ross and A.K. Jain, “Information fusion in Biometrics”, *Pattern Recognition Letters*, vol. 24, no. 13, pp. 2115-2125, 2003.

[78] J.S. Chen, Y.S. Moon and H.W. Yeung, “Palmprint authentication using time series”, *in Proceeding of 5th International Conference on Audio- and Video- Based Biometric Person Authentication*, pp. 20-22, 2005.

[79] A. Jain, R. Bolle, and S. Pankanti (eds), Biometrics: Personal Identification in Networked Society. Kluwer Academic Publishers, Boston, 1999.

[80] NIST report to the United States Congress. Summary of NIST Standards for Biometric Accuracy, Temper, Resistance, and Interoperability. November, 13, 2000.

[81] D. Zhang and W. Shu, “Two novel characteristics in palmprint verification: datum point invariance and line feature matching”, *Pattern Recognition*, vol. 32, no. 4, pp. 691-702, 1999.

[82] R. Sanchez-Reillo, C. Sanchez-Avilla and A. Gonzalez-Marcos, “Biometric identification through hand geometry measurements”, *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 22, no. 10, pp. 1168-1171, 2000.

[83] A.K. Jain, S. Prabhakar, L. Hong, and S. Pankanti, “Filterbank-based fingerprint matching”, *IEEE Transactions on Image Processing*, vol. 9, no. 5, pp. 846-859, 2000.

[84] S. Ben-Yacoub, Y. Abdeljaoued, and E. Mayoraz, “Fusion of face and speech data for person identity verification”, *IEEE Transactions on Neural networks*, vol. 10, no. 5, pp. 1065-1074, 1999.

[85] L. Hong, and A.K. Jain, "Integrating faces and fingerprints for personal identification", *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 20, no. 12, pp. 1295-1307, 1998.

[86] J. Daugman, "How iris recognition works", *IEEE Transactions on Circuits and Systems for Video Technology,* vol. 14, no. 1, pp.21-30, 2004.

[87] R.V. Hogg and E.A. Tanis, Probability and statistical inference, fourth edition, Macmillian, New York, 1993.

[88] K. R. Gabriel, "The distribution of the number of success in a sequence of dependent trials", *Biometrika*, vol. 46, pp. 454-460, 1959.

[89] R. W. Katz, "Computing probabilities associated with the Markov chain model for precipitation", *Journal of Applied Methodology*, vol. 13, pp. 953-954, 1974.

[90] W. Feller, An introduction to probability theory and its applications, John Wiley & Sons Inc, 3rd edition, vol. 1, pp. 321, 1968.

[91] J.O. Kim, W. Lee, J. Hwang, K.S. Baik and C.H. Chung, "Lip print recognition for security systems by multi-resolution architecture", *Future Generation Computer Systems*, vol. 20, no. 2, pp. 295-301, 2004.

[92] A.K. Jain, S.C. Dass and K. Nandakumar, "Soft biometric traits for personal recognition systems", *in Proceedings of International Conference on Biometric Authentication*, pp. 731-738, July, 2004.

[93] S.A. Israel, J.M. Irvine, A. Cheng, M.D. Wiederhold and B.K. Wiederhold, "ECG to identify individuals", *Pattern Recognition*, vol. 38, no. 1, pp. 133-142, 2004.

[94] Y.H. Pang, A. Teoh, D. Ngo and H.F. San, "Palmprint verification with moments", *Journal of Computer Graphics, Visualization and Computer Vision*, vol. 12, no. 2, pp. 325-332, 2004.

[95] N.K. Ratha, J.H. Connell and R.M. Bolle, "Enhancing security and privacy in biometrics-based authentication systems", *IBM Systems Journal*, vol. 40, no. 3, pp. 614-634, 2001.

[96] T. Connie, A. Teoh, M. Goh, "PalmHashing: a novel approach for dual-factor authentication", *Pattern Analysis and Applications*, vol. 7, no. 3, pp. 255-256, 2005.

[97] U. Uludag and A.K. Jain, "Attacks on biometric systems: a case study in fingerprints", *in Proceedings of SPIE-EI 2004, Security, Seganography and Watermarking of Multimedia Contents VI*, pp. 622-633, 2004.

[98] U. Uludag, S. Pankanti, S. Prabhakar and A.K. Jain, "Biometric cryptosystems: issues and challenges", *Proceedings of the IEEE*, vol. 92, no. 6, pp. 948-960, 2004.

[99] M. Braithwaite, U.C. von Seelen, J. Cambier, J. Daugman, R. Class, R. Moore and I. Scott, "Applications-Specific Biometric Template", *IEEE Workshop on Automatic Identification Advanced Technologies*, Tarrytown, NY, March, 14-15, pp. 167-171, 2002.

[100] C. Soutor, D. Roberge, S.A. Stojanov, R. Gilroy and B.V.K. Vijaya Kumar, "Biometric encryption", in ICAS Guide to Cryptography, R.K. Nichols, Ed. New York: McGraw-Hill, 1999.

[101] H. Cummins and C. Midlo, *Finger Prints, Palms and Soles: An Introduction to Dermatoglyphics*, Dover Publications, inc, New York, 1961.

[102] J. Daugman, "Uncertainty relation for resolution in space, spatial frequency and orientation optimized by two-dimensional visual cortical filters", *Journal of the Optical Society of America A*, vol. 2, pp. 1,160-1,169, 1985.

[103] T.S. Lee, "Image representation using 2D Gabor wavelet", *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 18, no. 10, pp. 959-971, 1996.

[104] A. Kong, D. Zhang and M. Kamel, "A Study of brute-force break-ins of a palmprint verification system", *in Proceeding of Audio- and Video-based Biometric Person Authentication*, pp. 447-454, 2005.

[105] A. Kong, D. Zhang and G. Lu, "A study of identical twins palmprints for personal authentication", *in Proceeding of International Conference on Biometrics*, pp. 106-112, 2006.

[106] J. Daugman, "Results from 200 billion iris cross-comparisons", *Technical Report of University of Cambridge, Computer Laboratory*, no. 635, pp. 1-8, 2005.

[107] Tests of the Daugman Iris Recognition Algorithms http://www.cl.cam.ac.uk/users/jgd1000/iristests.pdf

[108] L. Ma, T. Tan, Y. Wang and D. Zhang, "Efficient iris recognition by characterizing key local variations", *IEEE Transactions on Image Processing*, vol. 13, no. 6, pp. 739-750, 2004.

[109] Z. Sun, T. Tan and Y. Wang, "Iris recognition based on non-local comparisons", *Lecture Notes in Computer Science*, Springer, vol. 3338, pp. 491-497, 2004.

[110] E. Krichen, M.A. Mellakh, S. Garcia-Salicetti and B. Dorizzi, "Iris identification using wavelet packets", *in Proceedings of International Conference on Pattern Recognition*, vol. 4, pp. 226-338, 2004.

[111] S.I. Noh, K. Bae, Y. Park and J. Kim, "A novel method to extract features for iris recognition system", *Lecture Notes in Computer Science,* Springer, vol. 2688, pp. 861-868, 2003.

[112] K. Bea, S. Noh and J. Kim, "Iris feature extraction using independent component analysis", *Lecture Notes in Computer Science*, Springer, vol. 2688, pp. 838-844, 2003.

[113] P.F. Zhang, D.S. Li and Q. Wang, "A novel iris recognition method based on feature fusion*", in Proceedings of the Third International Conference on Machine Learning and Cybernetics*, pp. 26-29, 2004.

[114] T. Ea, A. Valentian, F. Rossant, F. Amiel and A. Amara, "Algorithm implementation for IRIS identification", *in Proceeding of 48th Midwest Symposium on Circuits and Systems*, pp. 1207-1210, 2005.

[115] C.H. Park, J.J. Lee, S.K. Oh, Y.C. Song, D.H. Choi and K.H. Park, "Iris feature extraction and matching based on multiscale and directional image representation", *LNCS, Springer*, vol. 2695, pp. 576-583, 2004.

[116] E. Rydgren, T.E.A.F. Amiel, F. Rossant and A. Amara, "Iris features extraction using wavelet packets", *in Proceedings* of *International Conference on Image Processing*, vol. 2, pp. 861-864, 2004.

[117] L. Masek, Recognition of Human Iris Patterns for Biometric Identification, *Bachelor thesis*, The University of Western Australia, 2003.

[118] Institute of Automation, Chinese Academy of Sciences (CASIA) Iris Image Database, http://www.sinobiometrics.com

[119] J. Daugman, "Iris recognition: current stat of the art", *Lecture note of Croucher Advanced Study Institute*, Hong Kong, 6-11 December 2004.

[120] W.K. Kong and D. Zhang, "Detecting eyelash and reflection for accurate iris segmentation", *International Journal of Pattern Recognition and Artificial Intelligence*, vol. 17, no. 6, pp. 1025-1034, 2003.

[121] R. Viveros, K. Balasubramanian and N. Balakrishnan, "Binomial and negative binomial analogues under correlated Bernoulli trials", *The American Statistician*, vol. 48, no. 3, pp. 243-247, 1994.

[122] R.W. Katz, "Comment on: Binomial and negative binomial analogues under correlated Bernoulli trials", *American Statistician*, vol. 49, no. 3, 1995.

[123] R.M. Bolle, S. Pankanti, J.H. Connell and N.K. Ratha, "Iris Individuality: A partial iris model", *in Proceedings of the 17th International Conference on Pattern Recognition*, vol. 2, pp. 927-930, 2004.

[124] J. Daugman, Personal Communication

[125] C. Sanchez-Avila and R. Sanchez-Reillo, "Two different approaches for iris recognition using Gabor filters and multiscale zero-crossing representation", *Pattern Recognition*, vol. 38, pp. 231-240, 2005.

[126] K. Miyazawa, K. Ito, T. Aoki, K. Kobayashi, H. Nakajima, "An efficient iris recognition algorithm using phase-based image matching", *in Proceedings International Conference on Image Processing*, vol. 2, pp. 49-52, 2005.

[127] B. Vijayakumar, C. Xie and J. Thornton, "Iris verification using correlation filters", *in Proceedings of 4th International Conference on Audio-and Video-based Biometric Person Authentication*, pp. 697-705, 2003.

[128] J.L. Wayman, “Degrees of freedom as related to biometric device performance” http://www.engr.sjsu.edu/biometrics/publications_degrees.html

[129] W.T Freeman and E.H Adelson, “The design and use of steerable filters”, *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 13, no. 9, pp. 891-906, 1991.

[130] M. Vatsa, R. Single and P. Gupta, “Comparison of iris recognition algorithms”, *in Proceedings of International Conference on Intelligent Sensing and Information Processing*, pp. 354-358, 2004.

[131] H. Proenca and L.A. Alexandre, “UBIRIS: A noisy iris image database”, *in Proceedings of 13th International Conference on Image Analysis and Processing*, pp. 970-977, 2005.

[132] J. J. Nora and F. C. Fraser, Medical genetics: principles and practice, Philadelphia: Lea & Febiger, 4th ed., 1994.

[133] P.J. Phillips, A. Martin, C.L Wilson, M. Przybocki, “An introduction to evaluating biometric systems,” *Computer*, vol. 33, no. 2, pp. 56-63, 2000.

[134] A.K. Jain, S. Prabhakar and S. Pankanti, “On the similarity of identical twin fingerprint,” *Pattern Recognition*, vol. 35, no. 11 pp. 2653-2663, 2002.

[135] C. Simon and I. Goldstein, “A new scientific method of identification,” *New York state journal of medicine*, vol. 35, no. 18, pp. 901-906, 1935.

[136] L. Flom and A. Safir, U.S. Patent No. 4641349, U.S. Government Printing Office, Washington, DC, 1987.

[137] J. Daugman and C. Downing, “Epigenetic randomness, complexity and singularity of human iris patterns,” *Proceedings of the Royal Society, B*, vol. 268, pp. 1737-1740, 2001.

[138] “Large scale evaluation of automatic speaker verification technology: dialogues spotlight technology report,” The Centre for Communication Interface Research at The University of Edinburgh, May 2000, Available at http://www.nuance.com/assets/pdf/ccirexecsum.pdf.

[139] K. Kodate, R. Inaba, E. Watanabe and T. Kamiya, “Facial recognition by a compact parallel optical correlator,” *Measurement Science and Technology*, vol. 13, pp. 1756-1766, 2002.

[140] C.C. Chibelushi, F. Deravi and J.S.D. Mason, “Adaptive classifier integration for robust pattern recognition,” *IEEE Transactions on Systems, Man and Cybernetics, Part B*, vol. 29, no. 6, pp. 902-907, 1999.

[141] A. M. Bronstein, M. M. Bronstein and R. Kimmel, “Three-dimensional face recognition”, *International Journal of Computer Vision*, vol. 64, no. 1, pp. 5-30, 2005.

[142] Medline Plus, Medical Encyclopedia http://www.nlm.nih.gov/medlineplus/ency/article/003290.htm

[143] A. Milton, *Dermatoglyphic Analysis as a Diagnostic Tool*, National Foundation-March of Dimes, New York, 1966.

[144] J. Daugman, "Recognizing persons by their iris patterns", *in Biometrics: Personal Identification in Networked Society*, edited by A.K. Jain, R. Bolle and S. Pankanti, Amsterdam: Kluwer, pp. 103-121, 1999.

[145] R.M. Bolle, J.H. Connell and N.K. Ratha, "Biometric perils and patches", *Pattern Recognition*, vol. 35, pp. 2727-2738, 2002.

[146] T. Matsumoto, H. Matsumoto, K. Yamada, and S. Hoshino, "Impact of artificial gummy fingers on fingerprint systems", *in Proceedings of SPIE*, vol. 4677, pp. 275-289, San Jose, USA, Feb, 2002.

[147] L. O'Gorman, "Comparing passwords, tokens, biometrics for user authentication", *Proceedings of the IEEE*, vol. 91, no. 12, pp. 2021-2040, 2003.

[148] A.K. Jain and U. Uludag, "Hiding biometric data", *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 25, no. 11, pp. 1494-1498, 2003.

[149] R. Derakhshani, S.A.C. Schuckers, L.A. Hornak and L. O'Gorman, "Determination of vitality from a non-invasive biomedical measurement for use in fingerprint scanners", *Pattern Recognition*, vol. 36, pp. 383-396, 2003.

[150] A. Adler, "Sample images can be independently restored from face recognition templates", *in Proceedings of Canadian Conference on Electrical and Computer Engineering*, Montreal, Canada, pp. 1163-1166 2003.

[151] M. Vatsa, R. Singh, P. Mitra and A. Noore, "Comparing robustness of watermarking algorithms on biometric data", *in Proceedings of the Workshop on Biometric Challenges from Theory to Practice*, pp. 5-8, 2004.

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

[153] The real cost of credit card fraud http://www.creditbloggers.com/2006/07/the_real_cost_o.html

[154] http://www.mandylionlabs.com/PRCCalc/PRCCalc.htm

[155] How many identity theft victims are there? What IS the impact on Victims. http://www.privacyrights.org/ar/idtheftsurveys.htm

[156] S. Prabhakar, S. Pankanti and A.K. Jain, "Biometric recognition security and privacy concerns", *IEEE Security & Privacy Magazine*, vol. 1, no. 2, pp. 33-42, 2003

[157] K.H. Cheung, A. Kong, D. Zhang, M. Kamel, J. You and T.H.W. Lam, "An analysis on accuracy of Cancellable biometrics on Biohashing", *in Proceeding of the 9th International Conference on Knowledge-based intelligent information and engineering system*, pp. 1168-1172, 2005.

[158] M. Savvides, B.V.K. Vijaya Kumar and P.K. Khosla, "Cancellable biometric filters for face recognition", *in Proceedings of the 17th International Conference on Pattern Recognition*, vol. 3, pp. 922-925, 2004.

[159] J.H. Van Deemter and J.M.H. Du Buf, "Simultaneous detection of lines and edges using compound Gabor filters," *International Journal of Pattern Recognition and Artificial Intelligence*, vol. 14, no. 6, pp. 757-777, 2000.

[160] S. Haykin, Neural Networks, A comprehensive Foundation, Prentice Hall International, Inc, New Jersey, 1999.

[161] A.K. Jain, M.N. Murthy and P.J. Flynn, "Data Clustering: A Review", *ACM Computing Review*, vol. 31, no 3, pp. 264-323, 1999.

[162] A.E. Eiben and J.E. Smith, Introduction to Evolutionary Computing, Springer. 2003.

[163] J. Thornton, M. Savvides and B.K. Vijayakumar, "Robust iris recognition using advanced correlation techniques", *the Second International Conference on Image Analysis and Recognition*, pp. 1098-1105, 2005.

[164] A.W.K. Kong, Using texture analysis on biometric technology for personal identification, MPhil Thesis, The Hong Kong Polytechnic University, 2002.

[165] "Biometrics Boom", *IEEE Spectrum*, vol. 41, no. 3, pp. 13-13, 2004.