Automatic Personal Identification Using Fingerprints

By

Lin Hong

A Dissertation

Submitted to
Michigan State University
in partial fulfillment of the requirements
for the degree of

Doctor of Philosophy

Department of Computer Science

June 25, 1998

Abstract

Automatic Personal Identification Using Fingerprints

By

Lin Hong

An accurate automatic personal identification is critical in a wide range of application domains such as national ID card, electronic commerce, and automated banking. Biometrics, which refers to automatic identification of a person based on her physiological or behavioral characteristics, is inherently more reliable and more capable in differentiating between an authorized person and a fraudulent imposter than traditional methods such as passwords and PIN numbers. Automatic fingerprint identification is one of the most reliable biometric technology. In this thesis, our objective is to design a fingerprint-based biometric system which is capable of achieving a fully automatic "positive personal identification" with a high level of confidence. We have identified and explored the following issues: (*i*) feature extraction - finding representative features from an input image for the purpose of fingerprint matching, (*ii*) image enhancement - improving the clarity of ridge structures of fingerprint images to facilitate automatic extraction of features or for visual inspection, (*iii*) minutiae matching - determining whether two sets of features (minutiae patterns) are

extracted from the same finger, (*iv*) integration of multiple biometrics - improving the performance of a biometric system by combining several biometrics (*e.g.* fingerprint, face, speech, *etc.*), and (*v*) fingerprint classification - assigning a fingerprint into one of several pre-specified categories according to its pattern formation. We have designed two prototype biometric systems: (*i*) a verification system which uses only fingerprints to authenticate the identity claimed by a user, and (*ii*) an integrated identification system which combines face recognition and fingerprint verification to make a personal identification. Our systems have been evaluated extensively on a large number of fingerprint images captured with the traditional inked method and more recent inkless optical scanners. Experimental results show that our systems perform very well on these data sets.

To **Yonghong**

Acknowledgments

I would like to acknowledge all the people who have assisted me during four years of my graduate study at Michigan State University. I am most grateful to my advisor, Dr. Anil Jain, for both his professional and personal advice, guidance, and help. He has provided me with numerous valuable ideas, insights, encouragement, and comments. He has also been very understanding and supportive. I am very fortunate to have him as my advisor. I would like to thank Dr. Sharath Pankanti of IBM T. J. Watson Research Center, NY for his many insightful discussions, steady encouragement, and numerous professional and personal help. Dr. John Weng has always been available to me to discuss ideas and concepts in computer vision and learning. I am very grateful to Dr. Weng for his numerous suggestions, discussions, and insights. Thanks to Dr. V. Mandrekar for his willingness to make time for our discussions. He has given me a number of suggestions and ideas on how to use statistics in fingerprint identification. Thanks to Dr. Eric Torng for his willingness to serve on my committee. He has helped me to appreciate the beauty underlying the complexity of theory of computing. Besides the committee members, I would like to thank our PRIP Lab. Director, Dr. George Stockman, who has been very supportive

over the years. I would also like to thank Dr. Ruud Bolle of IBM T. J. Watson Research Center, NY for his support during the past three years.

My fellow students and colleagues in the MSU PRIP lab have provided help and moral support throughout my stay here. I would like to thank them for their interest and concern, especially Paul Albee, Vera Bakic, Jinlong Chen, Shaoyun Chen, Scott Connell, Yuntao Cui, Chitra Dorai, Nico Duta, Mario Figueredo, WeyShuan Hwang, Kalle Karu, Yatin Kulkarni, Gongjun Li, Jinhui Liu, Karissa Miller, Aniati Murni, Salil Prabhakar, Nalini Ratha, Arun Ross, Jarle Strand, Dan Swets, Aditya Vailaya, and Bin Yu. My special thanks go to Salil Prabhakar and Arun Ross for proofreading the draft of this dissertation.

I would also like to thank Cathy Davison, Mary Gebbia, Lora Mae Higbee, Donna London, and Linda Moore for their administrative assistance.

My sincere thanks go to all the members of my family. I am most thankful for my wife's unfailing support, understanding, and love. Without her, I cannot imagine that I would have finished my Ph.D. study. I am also very grateful to my parents for their never-fading love, care, understanding, and encouragement. I could not have accomplished anything without their love and support. I owe my life and every achievement to them.

Finally, I would like to acknowledge Professor Zhisheng You of Sichuan University, China, my former advisor, for his support and encouragement.

Table of Contents

List of Figures

List of Tables

Chapter 1

Introduction

Personal identification is to associate a particular individual with an identity. It plays a critical role in our society, in which questions related to the identity of individuals such as *"Is this the person who he or she claims to be?", "Has this applicant been here before?", "Should this individual be given access to our system?", "Does this employee have authorization to perform this transaction?", etc.* are asked millions of times every day by hundreds of thousands of organizations in financial services, health care, electronic commerce, telecommunication, government, *etc.* With the rapid evolution of information technology, people are becoming even more and more electronically connected. As a result, the ability to achieve highly accurate *automatic* personal identification is becoming more critical [75, 35, 45, 97, 106, 109, 154].

Traditionally, two major types of automatic personal identification approaches have been widely used: (*i*) *knowledge-based* and (*ii*) *token-based* [97, 106, 109]. Token-based approaches use "something that you have" to make a personal identification. Individuals are identified by demonstrating that they are in possession of certain

token, such as passport, driver's license, ID card, credit card, and keys. Knowledge-based approaches use "something that you know" to make a personal identification. Individuals are identified by demonstrating that they are in possession of information or knowledge which only they themselves are expected to know such as *password* and *personal identification number* (PIN). The major advantages of these traditional personal identification approaches are that (*i*) they are very simple and (*ii*) they can be easily integrated into different systems with a low cost. However, since these traditional approaches are not based on any inherent attributes of an individual to make a personal identification, they have a number of disadvantages: tokens may be lost, stolen, forgotten, or misplaced; PIN may be forgotten or guessed by the impostors. All of these approaches are also unable to differentiate between an *authorized person* and an *impostor* who fraudulently acquires the "token" or "knowledge" of the authorized person [35, 45, 97, 106, 109]. Therefore, they are *unable* to satisfy the security requirements of our electronically inter-connected information society.

1.1 Biometrics

Biometrics, which refers to identifying an individual based on her physiological or behavioral characteristics (identifiers) [75, 35, 45, 97, 106], relies on "something which you are or you do" to make a *positive* personal identification. It is inherently more reliable and more capable than knowledge-based and token-based techniques in differentiating between an authorized person and a fraudulent impostor, because the physiological or behavioral characteristics are unique to every person. Also, the per-

son to be identified is required to be physically present at the point-of-identification. Biometrics provides a solution for the security requirements of our electronically interconnected information society and has the potential to become the dominant automatic personal identification in the near future [35, 45, 106, 109, 97, 75].

1.1.1 Biometric System

A *biometric system* is essentially a pattern recognition system which make a personal identification by determining the authenticity of a specific physiological or behavioral characteristic possessed by the user. The block diagram of a generic biometric system is depicted in Figure 1.1. Logically, it can be divided into two modules: (*i*) *enrollment module* and (*ii*) *identification module*. The enrollment module is responsible for enrolling individuals into the biometric system. During the enrollment phase, the biometric characteristic of an individual is first scanned by a biometric reader to produce a raw digital representation of the characteristic. In order to facilitate matching, the raw digital representation is usually further processed by a feature extractor to generate a compact but expressive representation, called a *template*. Depending on the application, the template may be stored in the central database of the biometric system or be recorded on a *magnetic card* or *smart card* issued to the individual. The identification module is responsible for identifying individuals at the point-of-access. During the operation phase, the biometric reader captures the characteristic of the individual to be identified and converts it to a digital format, which is further processed by the feature extractor to produce the same representation. The resulting

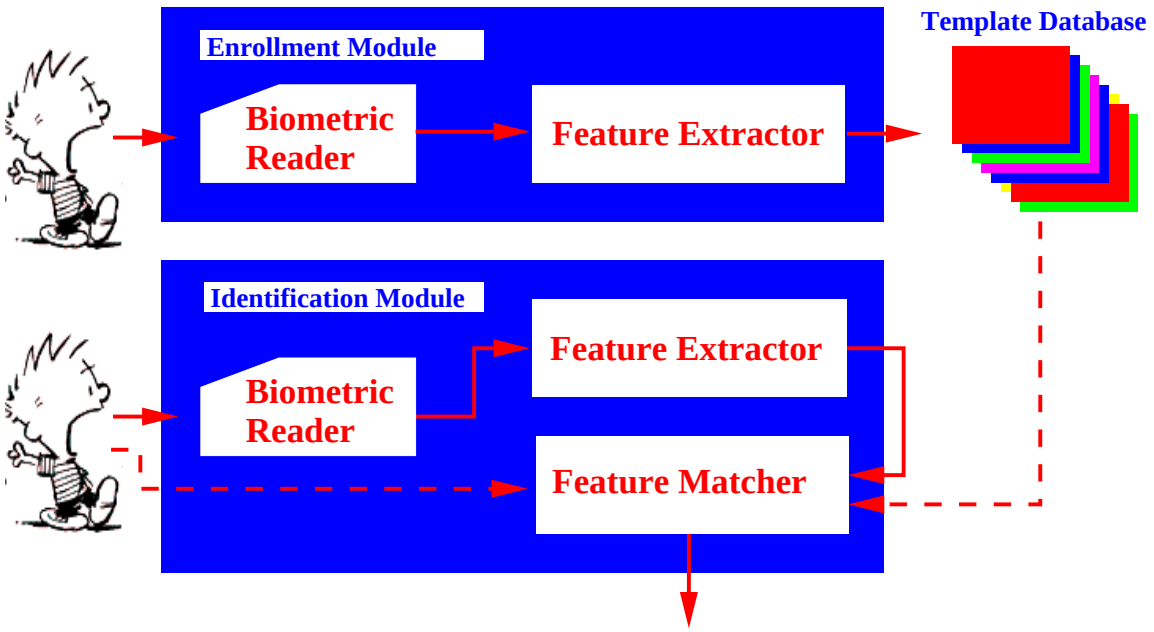


Figure 1.1: A generic biometric system.

representation is fed to the feature matcher which compares it against the template(s) to establish the identity.

1.1.2 Requirements of Biometric Identifiers

Any human physiological or behavioral characteristic can be used as a *biometric characteristic or identifier* to make a personal identification as long as it satisfies the following requirements [35, 106]: (*i*) *universality*, which means that each person should have the characteristic, (*ii*) *uniqueness*, which indicates that no two persons should be the same in terms of the characteristic, (*iii*) *permanence*, which means that the characteristic should not be changeable, and (*iv*) *collectability*, which indicates that the characteristic can be measured quantitatively. However, in practice, a biometric characteristic that satisfies all the above requirements may not always be feasible for a practical biometric system. In a practical biometric system, there

are a number of other issues which should be considered, including [35, 106] (*i*) *performance*, which refers to the achievable identification accuracy, speed, robustness, the resource requirements to achieve the desired identification accuracy and speed, as well as operational or environmental factors that affect the identification accuracy and speed, (*ii*) *acceptability*, which indicates the extent to which people are willing to accept a particular biometric identifier in their daily life, and (*iii*) *circumvention*, which reflects how easy it is to fool the system by fraudulent methods. A practical biometric system should be able to (*i*) achieve an acceptable identification accuracy and speed with a reasonable resource requirements, (*ii*) not be harmful to the subjects and be accepted by the intended population, and (*iii*) be sufficiently robust to various fraudulent methods.

1.1.3 Operational Mode

An important issue in designing a practical biometric system is to determine how an individual is identified. Depending on the application context, a biometric system may be either a *verification (authentication)* system or an *identification* system [106]. A verification system authenticates a person's identity by comparing the captured biometric characteristic with her own biometric template(s) pre-stored in the system. It conducts one-to-one comparison to determine whether the identity claimed by the individual is true or not. In a verification (authentication) system, an individual desired to be identified submits a claim to an identity to the system usually via a magnetic stripe card, login name, smart card, *etc.*, and the system either rejects or

accepts the submitted claim of identity (*Am I whom I claim I am?*). An identification system recognizes an individual by searching the entire template database for a match. It conducts one-to-many comparisons to establish the identity of the individual. In an identification system, the system establishes a subject's identity (or fails if the subject is not enrolled in the system database) without the subject having to claim an identity (*Who am I?*).

Depending on the application domain, a biometric system could be either (*i*) an *online* system or (*ii*) an *offline* system. An online system requires that a verification/identification be performed quickly and an immediate response is imposed. On the other hand, an offline system usually does not require that a verification/identification be performed immediately and a relatively long response delay is allowed.

1.1.4 Performance

Due to intraclass variations present in any biometric characteristic, the identity established by a biometric system is not an absolute "yes" or "no" answer about the identity; instead it is an answer with a certain *confidence level*. Generally, the identity established by a biometric system is either a *genuine* type or an *impostor* type, which can be represented by two statistical distributions, called genuine distribution and impostor distribution, respectively. For each of type of identity, there are two possible outcomes, *true* or *false*. Therefore, there are a total of four possible outcomes: (*i*) a genuine individual is accepted, (*ii*) a genuine individual is rejected, (*iii*)

an impostor is rejected, and (*iv*) an impostor is accepted. Outcomes (*i*) and (*iii*) are correct whereas (*ii*) and (*iv*) are incorrect. The confidence associated with the identity established by the biometric system may be determined by the two error rates, (*i*) *false acceptance rate* (FAR) and (*ii*) *false reject rate* (FRR), which are characterized by the genuine distribution and the impostor distribution, respectively. The false acceptance rate is defined as the probability that an impostor is accepted as a genuine individual and the false reject rate is defined as the probability that a genuine individual is rejected as an impostor. Clearly, FAR and FRR are dual of each other. A smaller FRR usually leads to a larger FAR while a smaller FAR usually implies a larger FRR. Generally, the capability of a biometric system in performing automatic personal identification is specified in terms of FAR [106]. A FAR of zero means that no impostor is accepted as a genuine individual.

An identification system is essentially a *database retrieval system*. In addition to the confidence level of the established identity, two other important accuracy measures, which characterize the retrieval accuracy of a database retrieval system, need to be provided to indicate the capability of the system: (*i*) *precision* and (*ii*) *recall*. Precision is defined as the ratio of genuine records in the template database retrieved by the identification system and the total number of templates retrieved. Recall is defined as the ratio of the genuine records in the template database retrieved by the identification system and the total number of genuine records in the template database.

In addition to accuracy, verification/identification speed constitutes the next important performance measure. In a verification system, since only one-to-one compar-

isons are performed, the speed performance is mainly characterized by the response time of the verification (and feature extraction) algorithm or more precisely by the computational complexity of the algorithm. It is usually easy to meet the speed requirement of a verification system. However, in an identification system, especially for a system which consists of millions of templates, a large number of comparisons need to be performed to identify an individual. The speed performance involves a number of aspects, including response time, throughput, computational complexity, and scalability.

1.2 Applications

Biometrics is a rapidly evolving technology which has been widely used in forensics such as criminal identification and prison security, and has a very strong potential to be widely adopted in a broad range of civilian applications. These applications may be divided into the following two groups: (*i*) applications such as banking, electronic commence, and access control in which biometrics will replace or enforce the current token-based or knowledge-based techniques and (*ii*) applications such as welfare and immigration in which neither the token-based nor the knowledge-based techniques are currently being used.

Electronic commence and electronic banking are one of the most important and emerging application areas of biometrics due to the rapid progress in electronic transactions. These applications include electronic fund transfers, ATM security, check cashing, credit card security, smart cards security, online transactions, *etc.* Cur-

rently, there are several large biometric security projects in these areas under development including credit card security (MasterCard) and smart card security (IBM and American Express). A variety of biometric technologies are now competing to demonstrate their utility in these application areas. *The market of physical access control* is currently dominated by token-based technology. However, it is predicted that, with the progress in biometric technology, market share will increasingly shift to biometric techniques. *Information system/computer network security* such as user authentication and access to databases via remote login is another important potential application area for biometrics. It is expected that more and more information systems/computer networks will be secured with biometrics with the rapid expansion of Internet. With the introduction of biometrics, *government benefits distribution program* such as welfare disbursement programs [98] will experience substantial savings in deterring multiple claimants. In addition, *customs and immigration initiatives* such as *INS* Passenger Accelerated Service System (INSPASS) which permits faster immigration procedures based on hand geometry [17] will greatly increase the operational efficiency. Biometrics-based *national ID systems* provide a unique ID to the citizens and integrate different government services [106]. Biometrics-based *voter and driver registration* provides registration facilities for voters and drivers. Biometrics-based *time/attendance monitoring systems* can be used to prevent any abuses of the current token-based/manual systems [86].

1.3 Biometric Technologies

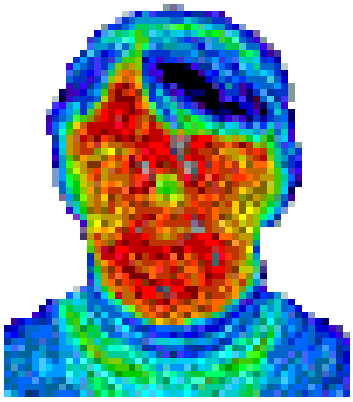
A biometric characteristic could be either (*i*) *a physiological characteristic* or *a behavioral characteristic*. A physiological characteristic is an attribute that is innate to us. A behavioral characteristic captures something that we do. In terms of identification accuracy, generally, it is believed that a physiological biometric characteristic is more reliable than a behavioral biometric characteristic, since a physiological biometric characteristic tends to have smaller intraclass variation than a behavioral biometric characteristic [106].

Currently, there are mainly nine different biometric techniques that are either widely used or are under intensive investigation, including *face, fingerprint, hand geometry, hand vein, iris, retinal pattern, signature, voice-print, and facial thermogram* [6, 35, 45, 75, 97, 43, 106, 143, 50, 70, 155, 105, 153]. Face, fingerprint, hand geometry, hand vein, iris, facial thermogram, and retinal pattern are *physiological* biometrics. Signature and voice-print are *behavioral* biometrics. Examples of these nine different biometric characteristics are shown in Figure 1.2. All these biometric techniques, to a certain extent, satisfy the requirements mentioned in section 1.1.2 and have been used in practical systems [35, 45, 43, 106] or have the potential to become a valid biometric technique [106]. We will briefly review these biometric technologies.

1.3.1 Face

Facial images are probably the most common biometric characteristic used by humans to make a personal identification. Face recognition is one of the most active

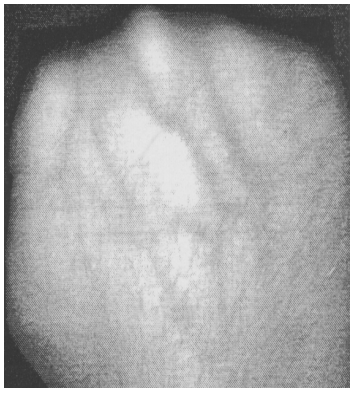
face



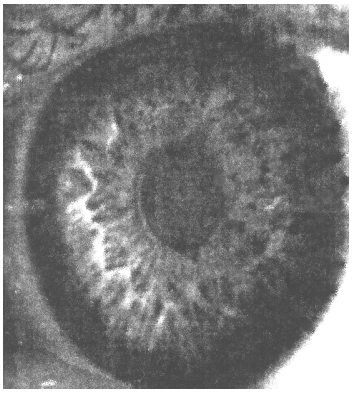
facial thermogram

fingerprint

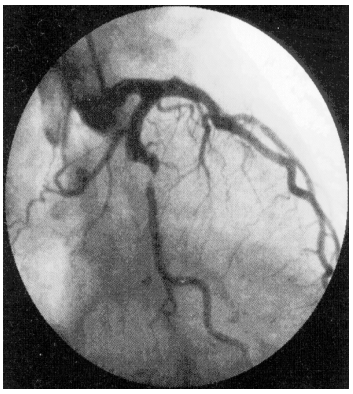
hand geometry



hand vein

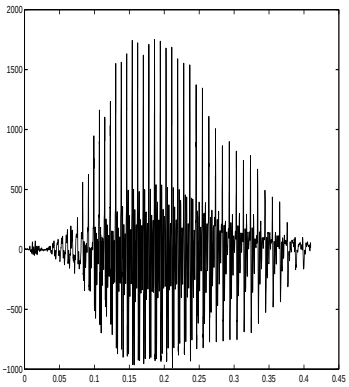


iris



retinal scan

signature



voice print

Figure 1.2: Examples of different biometric characteristics.

area of research with applications ranging from static, controlled mug shot verification to dynamic, uncontrolled face identification in a cluttered background [31]. In the context of automatic personal identification, face recognition usually refers to static, controlled full frontal portrait recognition [31]. By static we mean that the facial portraits used by the face recognition system are still facial images (intensity or range). By controlled we mean that the type of background, illumination, resolution of the acquisition devices and the distance between the acquisition devices and faces, *etc.* are essentially fixed during the image acquisition process. Clearly, in such a controlled situation, the segmentation task is relatively simple and the intraclass variations are small. Face recognition is a non-intrusive technique. People generally do not have any problem in accepting face as a biometric characteristic. Theoretically, it has the potential to become the most friendly and acceptable way to make personal identification [6, 106, 144]. During the past 25 years, a substantial amount of research effort has been devoted to face recognition [31, 145, 155]. In the early 1970's, face recognition was mainly based on measured facial attributes such as eyes, eyebrows, nose, lips, chin shape, *etc.* [31]. Due to lack of computational resources and reliable feature extraction algorithms, only a very limited number of tests were conducted and the recognition performance of these systems was far from desirable [31]. After the dormant early 1980's, there was a resurgence in face recognition research in the late 1980's and early 1990's. In addition to continuing efforts on attribute-based techniques [31], a number of new face recognition techniques were proposed, including principle component analysis (PCA) [144], linear discriminant analysis (LDA) [139], singular value decomposition (SVD) [65], local feature analy-

Figure 1.3: Multiple Personalities: all the people in this image are the same person (*The New York Times Magazine, September 1, 1996/section 6, pages 48-49*).

sis [6], and a variety of neural network-based techniques [145]. The performance of these approaches is impressive. It was concluded that "face recognition algorithms were developed and were sufficiently mature that they could be ported to real-time experimental/demonstration system" [114]. A number of face recognition systems are available on the market, such as TrueFace [99] and Faceit [148]. The performance of these systems is reasonable.

Although humans depend heavily on facial images and attributes to identify individuals, it is widely known that humans utilize a large amount of contextual information in performing face recognition [35]. Without the contextual information, it is questionable whether the face itself is sufficiently effective to make a personal identification with a high level of confidence. For example, without any other information about the faces in figure 1.3, it will be very difficult for both humans and machine

vision systems to conclude that they are all of the same person (*The New York Times Magazine, September 1, 1996/section 6, pp. 48-49*). Since face recognition is supposed to be the most user-friendly biometric technology, a face recognition system should not impose any annoying controlled restrictions on how the facial images are acquired. This requires that the system should be able to automatically (*i*) detect whether there exists a face in the acquired image, (*ii*) locate the face if there is one, and (*iii*) recognize the face from a general viewpoint. These issues highlight some of the difficulties in face recognition [6, 144].

1.3.2 Face Thermogram

The underlying vascular system in the human face produces a unique facial signature when heat passes through the facial tissue and is emitted from the skin [143]. Such facial signatures can be captured using an infrared camera, which are usually called face thermogram. It is believed that a face thermogram is unique to each individual. They are not vulnerable to disguises. Even plastic surgery, which does not reroute the flow of blood through the veins, cannot change the formation of the face thermogram of an individual. Also, face thermograms are independent of ambient light. An infrared camera can capture the face thermogram in low light or in the absence of any light, which greatly reduces the restrictions on how face thermograms are acquired. Clearly, face thermogram is a non-intrusive biometric technique. Identity can be verified without contact, without full camera view, and without the cooperation of subjects. It is claimed that face thermogram-based recognition is superior to face

recognition using CCD cameras [143]. Although it may be true that face thermograms are unique to each individual, it has not been proven that face thermograms are sufficiently discriminative. Face thermograms depend heavily on a number of factors such as the emotion of the subjects, the body temperature, *etc.* Like face recognition, face thermogram recognition is view-dependent. Finally, face thermogram has not been shown to be a permanent biometric characteristic.

1.3.3 Fingerprints

A fingerprint is the pattern of ridges and furrows on the surface of a fingertip. It is formed by the accumulation of dead, cornified cells that constantly slough as scales from the exposed surface [103]. It's formation is determined in the fetal period [103]. Extensive studies have been conducted on fingerprints and fingerprint identification [40, 54, 108, 85, 103, 106]. The biological properties of fingerprints are well understood. Humans have used fingerprints for personal identification for centuries and the validity of fingerprint identification has been well-established. In fact, fingerprint technology is so common in personal identification that it has almost become the synonym of biometrics [45]. A major problem with fingerprint technology is its acceptability by a typical user, because fingerprints have traditionally been used for criminal investigations and police work. People may feel uncomfortable in using fingerprints in civilian applications. Another problem with fingerprint technology is that automatic fingerprint identification generally requires a large amount of computational resources. For more details on fingerprint matching, see section 2.2.

1.3.4 Hand Geometry

A variety of hand geometries including the shape of the hand and lengths and widths of the fingers, *etc.* can be used as biometric characteristics. Hand geometry-based biometric systems have been installed at over 4,000 locations around the world, including the Colombian legislature and the San Francisco International Airport [45, 106]. The technique is very simple and cheap. The accuracy of a hand geometry-based biometric system is quite reasonable. Operational environmental factors generally have very limited negative effects on the identification accuracy. It does not appear to be a problem for people to accept this technology. A main disadvantage of this technique is its low discriminative capability – it is very difficult for a hand geometry-based biometric system to achieve a very high identification accuracy especially for a large population. The physical size of a hand geometry-based system is large, which may restrict it from certain applications such as laptop computers. Among the nine biometric techniques illustrated in Figure 1.2, hand geometry is the least circumventive biometric characteristic. It is usually not very difficult to fool a hand geometry-based biometric system. In addition, hand geometry is not a permanent biometric characteristic [106].

A variant of hand geometry technique, *finger geometry* technique, which relies on a number of geometrical invariants of fingers such as the 3D shape of a finger has recently been investigated. It is claimed that finger geometry is more accurate in personal identification than hand geometry. However, such a conclusion needs some further justification.

1.3.5 Hand Vein

Hand veins provide a very robust and repeatable pattern that can be used as a biometric characteristic to make a personal identification [106]. Digitized images of hand vein patterns can be easily captured with an infra-red camera. Hand vein patterns are unique to each individual. They are separated from external environment, so it is easy to segment it from background. It is very difficult to change the formation of the hand vein pattern of an individual by surgery. Thus, hand vein based technique is very efficient in circumventing fraudulent attempts. A hand vein-based biometric system has the potential to achieve a reasonable identification accuracy and people are normally willing to accept it. However, there is no hand vein-based biometric system available that is able to demonstrate its superior capability in conducting automatic personal identification. Like hand geometry, it might be very difficult for a hand vein-based biometric system to achieve a very high identification accuracy. The physical size of a hand vein-based system is large. Again, hand vein is not a permanent biometric feature, especially for people in developing age group.

1.3.6 Iris

The texture formation of iris in a human eye depends on the initial conditions of the embryonic mesoderm from which it develops [43, 106]. It is unique for each individual and it never changes during the person's life time. Iris is inherently isolated from external environment and can not be modified surgically [43]. All of these properties make iris one of the most secure biometric characteristic in deterring impostors [106,

153]. The technique is simple but very efficient in performing automatic personal identification. It has the potential to become a major biometric technique in the future. Currently, a few iris scan-based biometric systems are available in the market such as the IriScan developed by IriScan, Inc. which is claimed to be able to achieve a very high identification accuracy with a very limited amount of computational resources [43, 106]. The major problem with iris scan is that it is still not accepted as a proven technology and its validity has not yet been well established [106]. Iris imaging needs to project a bean of light on the iris. It appears that people may not feel comfortable in accepting iris scan technique in their daily life, since people are usually very protective of their eyes. In addition, the sensor needs to be placed at a certain distance from the eye to capture the visual texture information and to register the iris images, which is another annoying restriction. Finally, in order to capture an iris image that is suitable for identification, a relatively expensive iris scanner needs to be used.

1.3.7 Retinal Pattern

The retinal veins in human eyes form very stable and repeatable patterns, called retinal patterns. They are unique to each person. Digital images of retinal patterns can be acquired by projecting a low-intensity beam of light on the eyeball. Retinal patterns are isolated from the external environment which is a very good property in deterring impostors. In fact, retinal scan is currently believed to be the most secure biometric technique. A large number of retinal scan-based biometric systems

have been installed in several highly secure environments. Their validity has been well established by these operational installations. For example, it has been reported that one type of retinal scan-based biometric system, the EyeDentify, has never let in any impostor so far [106]. The major problem for a retinal pattern-based biometric system is that most people do not feel comfortable in using such a system and it usually needs a high degree of cooperation from the subjects, since retinal image capture requires peeping into an eye-piece and focusing on a predetermined spot in the visual field so that a predetermined part of the retinal veins could be scanned. The cost of a retinal scanner is high. Again, retinal pattern is not a permanent biometric characteristic [106].

1.3.8 Signature

Each person has a unique style of handwriting. Signature is a kind of "fingerprint" that can be used to make a personal identification [105, 106]. There are two approaches to signature verification: (*i*) static and (*ii*) dynamic. Static signature verification uses only the geometric features of a signature. Dynamic signature verification uses both the static geometric features and the dynamic features such as acceleration, velocity, and trajectory profiles of the signature. An inherent advantage of a signature-based biometric system is that the signature has been established as an acceptable form of personal identification method. Another advantage of signature is that it is impossible for an impostor to obtain the dynamics information from a written signature. The identification accuracy of signature-based biometric systems

is reasonable. For example, a false acceptance rate of 0.58% and false reject rate of 2.1% were *claimed* by a commercial system [106]. However, due to large intraclass variations of signature, it is very difficult for both static and dynamic signature-based systems to reach a very high identification accuracy.

1.3.9 Voice Print

The vocal characteristics of humans are totally determined by the vocal tract, mouth, nasal cavities, and the other speech processing mechanisms of human body [70, 106]. They are unique to each person and are usually called voice-prints. Voice-print verification could be either a text-dependent verification or a text-independent verification. A text-dependent verification authenticates the identity of an individual based on a fixed predetermined phrase. A text-independent verification verifies the identity of a speaker independent of the phrase, which is more difficult than a text-dependent verification. Extensive studies have been conducted on voice print techniques. Currently, there are a number of voice print-based biometric systems available in the market, including SpeakEZ (by T-NETIX), Tespar (by Domain Dynamics), VoiceKey (by International Electronic, Inc.), BHS-1024 (by Technologia Systems), and Veritel (by Veritel). They can achieve a reasonable identification accuracy. Generally, people are willing to accept a voice-print based biometric system. The main problem with voice-print technique is that voice-prints may not be sufficiently unique to permit an identification of an individual from a large population. Voice-prints are sensitive to a number of factors such as background noise as well as the emotional and physical

Biometrics	Universality	Uniqueness	Permanence	Collectability	Performance	Acceptability	Circumvention
Face	high	low	medium	high	low	high	low
Fingerprint	medium	high	high	medium	high	medium	high
Hand Geometry	medium	medium	medium	high	medium	medium	medium
Hand Vein	medium	medium	medium	medium	medium	medium	high
Iris	high	high	high	medium	high	low	high
Retinal Scan	high	high	medium	low	high	low	high
Signature	low	low	low	high	low	high	low
Voice Print	medium	low	low	medium	low	high	low
F.Thermogram	high	high	low	high	medium	high	high

Table 1.1: Comparison of Biometric Technologies.

state of the speaker. It is very difficult for a voice-print based system to achieve an accuracy comparable to fingerprint-based or retinal pattern-based biometric systems. In addition, some people seem to be extraordinarily skilled in mimicking others voice.

1.3.10 Other Biometric Techniques

Besides the techniques mentioned above, a number of other biometric techniques have been investigated or are currently under study, including *ear shape*, *gait*, *keystroke dynamics*, *body odor*, *lip shape*, *DNA*, *etc.* Although each of these techniques has its own advantages, so far, none of them can achieve an accuracy that is comparable to the nine different techniques mentioned above or can be conducted fully automatically. In fact, they do not have a strong potential to become a valid biometric technique to be used widely in the near future.

1.3.11 Comparison of Biometric Technologies

Each of the biometric technique reviewed above has its own advantages and disadvantages. A brief comparison of these nine biometric techniques along seven factors is provided in Table 1.1. The applicability of a specific biometric technique depends

heavily on the application domain. No single technique can outperform all the others in all operational environments [106]. In this sense, each biometric technique is *admissible*. For example, it is well known that both the fingerprint technique and the iris scan technique perform much better than the voice print technique in terms of accuracy and speed. However, in a telephone account security application, the voice print technique is preferred, because it can be integrated seamlessly into the current telephone system.

It is important to point out that most of the biometric techniques reviewed in the previous section are not acceptable (in a court of law) as indisputable evidence of identity. In fact, the only legally acceptable, readily automated, and mature biometric technique so far is the automatic fingerprint identification technique which has been used and accepted in forensics since the early 1970's [85]. Although, signature is also a legally acceptable biometrics, it ranks a distant second to fingerprints due to issues involved with accuracy, forgery, and behavioral variability.

1.4 Problem Definition

In this thesis, our objective is to design a biometric system which is capable of achieving a *fully automatic personal identification* with a high level of confidence using mainly fingerprints. The research problem may be stated as follows: Design a fingerprint-based automatic personal identification system that can (*i*) authenticate whether the identity claimed by an individual is true or not (Am I who I claim I am?) or (*ii*) establish the identity of individuals that are enrolled in the system (Who am

I?). The advantages of using fingerprints are as follows: (*i*) fingerprint identification is one of the most reliable personal identification technique, (*ii*) its validity has long been established and justified, and (*iii*) it is the most commonly used biometric technique which has the potential to stay as a dominant biometric technique in the future [106]. We have identified and investigated the following four issues in this thesis.

(*i*) *fingerprint matching* - determining whether two fingerprints are impressions of the same finger (Figure 1.4). Fingerprint matching constitutes the fundamental capability of a fully automatic fingerprint-based biometric system. The two most important problems in fingerprint matching are: (*i*) *how to derive an efficient representation that is able to capture the individuality of each fingerprint* and (*ii*) *how to match two representations to find the similarity between them.* To derive a representation is to extract a set of features from an input image. To match two representations is to determine a confidence value with which we can claim that two sets of features extracted from two fingerprints are from the same finger. In order to design an efficient fingerprint matching algorithm, it is necessary that a concise but *sufficient* fingerprint representation be derived from the input digital fingerprint images and an *effective* matching algorithm be designed to determine whether two sets of derived representations are of the same finger. By sufficient we mean that a representation should contain enough class-specific (individual) information about the digital fingerprints, while by effective we mean that the matching performance based on the given representation should be high enough to make a confident personal identification.

(*ii*) *fingerprint image enhancement* - improving the quality of an input fingerprint

(a) (b) (c) (d)

Figure 1.4: Fingerprint matching: (a) and (b) are two impressions from the same finger; (c) and (d) are two impressions from different fingers.

image to make it more suitable for the feature extraction algorithm (Figure 1.4). The performance of a feature extraction algorithm relies heavily on the quality of input fingerprint images. In practice, due to variations in impression conditions, ridge configuration, skin conditions (aberrant formations of epidermal ridges of fingerprints, postnatal marks, occupational marks), acquisition devices, and non-cooperative attitude of subjects, *etc.*, a significant percentage of acquired fingerprint images (approximately 10% according to our experience) is of poor quality. The ridge structures in poor-quality fingerprint images are not always well-defined and hence they can not be correctly detected. This leads to a significant number of spurious features as well as missing features, which greatly degrades the performance of fingerprint matching. In order to ensure that the performance of the feature extraction algorithm will be robust with respect to the quality of input digital fingerprint images, an enhancement algorithm which can improve the clarity of the ridge structures is necessary.

(*iii*) *Integration of multiple biometric indicators* - combining multiple biometric indicators to improve the performance and applicability of a biometric system. Different application domains impose different operational and performance requirements

(a) (b)

Figure 1.5: Fingerprint image enhancement: (a) corrupted image; (b) enhanced image.

on a biometric system. By and large, the personal identification systems based solely on fingerprints are able to satisfy these requirements. However, since fingerprint matching is computationally demanding, it is impractical to require that an automatic personal identification system based solely on fingerprints is able to establish the identity of an individual by searching through a huge fingerprint database in "real-time." Integration of multiple clues has been shown to be very effective in improving the performance of pattern recognition systems [20]. In addition, although a necessary requirement for a biometric characteristic is that each individual possess it, it is not necessary that a particular biometric characteristic of a specific individual is suitable for an automatic system. By using multiple biometric characteristics, the system will be applicable on a larger target population. We have explored the design of a biometric system which combines multiple biometric indicators (face and fingerprint) to overcome some of the limitations of a fingerprint-based system (Figure 1.6).

(*iv*) *fingerprint classification* - assigning fingerprints into a number of categories

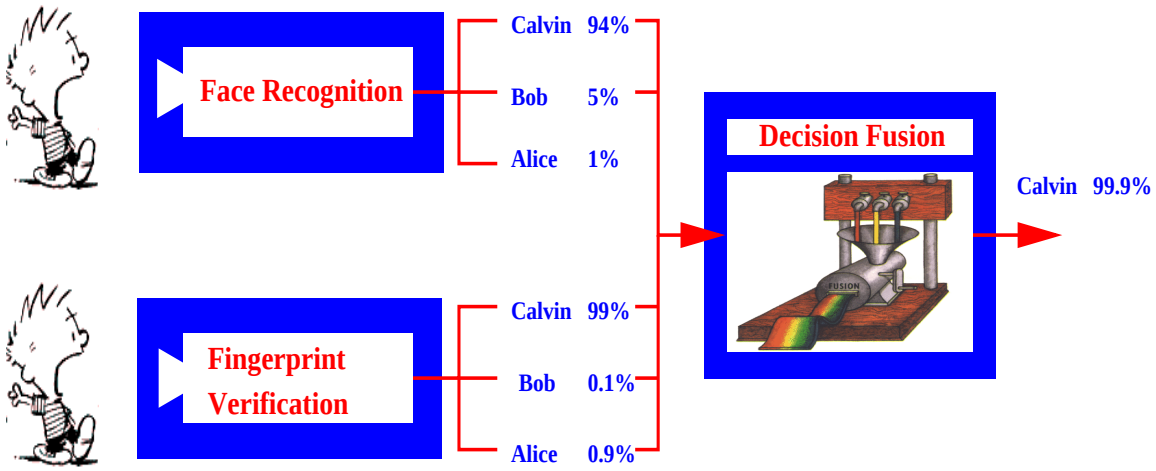


Figure 1.6: Integration of face recognition and fingerprint verification.

based on the global ridge and furrow configurations (Figure 1.4). Fingerprint classification provides important information about the global pattern configuration of fingerprints and, thus, plays an important role in fingerprint matching. In fact, if two fingerprints are not in the same category, then it is certain that the two fingerprints are not from the same finger. Fingerprint classification consistently assigns fingerprints into categories according to the global pattern configurations, which essentially provides an indexing mechanism for a fingerprint database. Since automatic fingerprint matching is a computationally demanding task, this indexing mechanism can greatly reduce the computational complexity in searching for a match in a fingerprint database, especially a large fingerprint database.

1.5 Overview of the Thesis

The rest of the thesis is organized as follows. Chapter 2 introduces the history and methodology of fingerprint identification. Chapter 3 presents the design of our au-

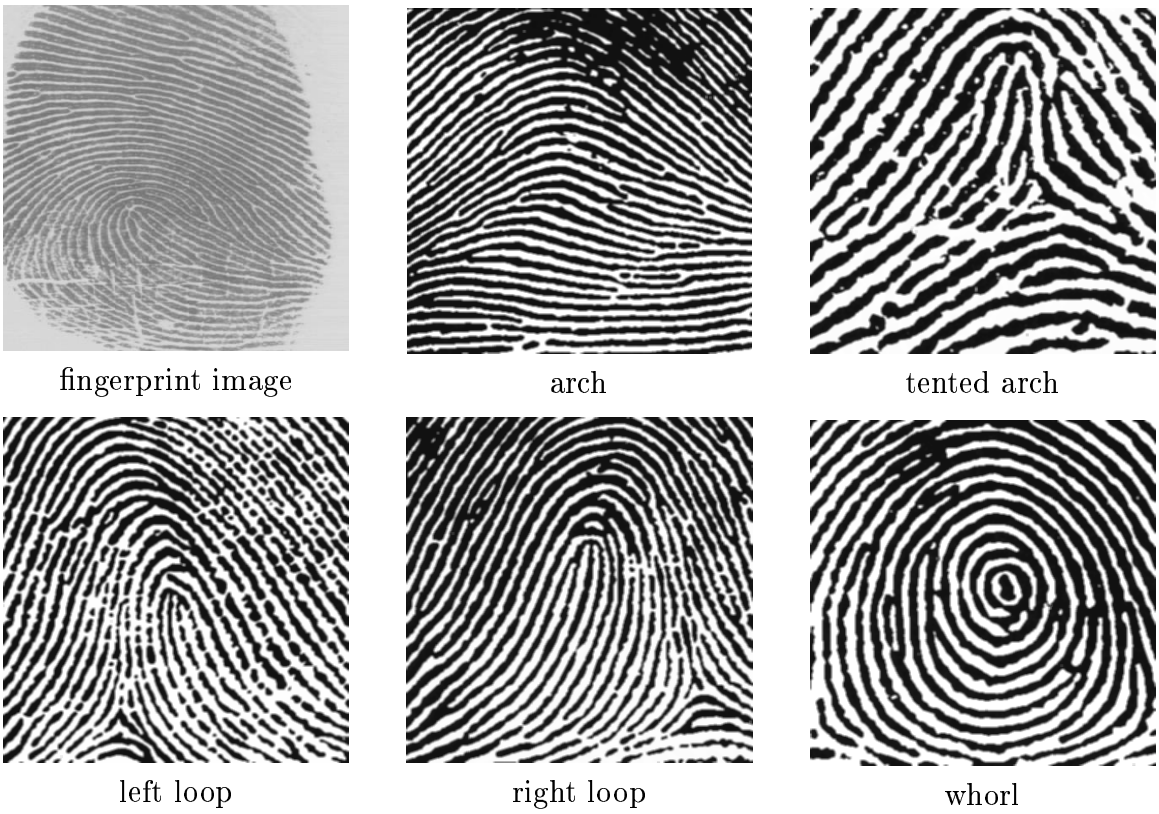


Figure 1.7: A fingerprint image and five major fingerprint classes.

tomatic personal identification system and related issues. Chapter 4 discusses fingerprint feature extraction and presents an improved minutiae extraction algorithm. Chapter 5 emphasizes the need for fingerprint enhancement and proposes a novel fingerprint image enhancement algorithm. Chapter 6 presents our minutiae matching algorithm. Chapter 7 addresses the decision fusion scheme which integrates faces and fingerprints to achieve a better performance. Chapter 8 discusses the problem of fingerprint image classification and presents a novel fingerprint image classification algorithm. Chapter 9 discusses the problem of performance evaluation for biometric systems. Also, experimental results of our system on several data sets are reported. Finally, chapter 10 contains a summary of our research, discusses the limitations of our current algorithms, and gives a list of problems which should be explored by other

researchers.

1.6 Summary

Accurate automatic personal identification is critical in a wide range of application domains such as national ID card, electronic commerce, and automated banking. Biometrics, which refers to automatic identification of a person based on her physiological or behavioral characteristics, is inherently more reliable and more capable in differentiating between an authorized person and a fraudulent impostor than traditional methods such as passwords and PIN numbers. Automatic fingerprint identification is one of the most reliable biometric technology among the nine different major biometric techniques which are either currently available or are under investigation. The objective of our research is to design a biometric system which is capable of achieving fully automatic "personal identification" with a high level of confidence using mainly fingerprints.

Chapter 2

Fingerprint Identification

In the context of fingerprint identification, *fingerprints* or simply *prints* are generally used to refer to the impressions of human fingers. In this thesis, fingerprints, prints, and fingerprint impressions are used synonymously to indicate the impressions of fingertips. Operationally, fingerprint identification can be decomposed into the following three fundamental tasks [103]: (*i*) *fingerprint acquisition*, (*ii*) *fingerprint classification*, and (*iii*) *fingerprint matching*. Fingerprints are acquired from fingertips or impressions of the ridges and furrows. Fingerprint classification assigns a fingerprint into a certain category according to its global ridge and furrow configuration. Fingerprint matching determines whether two fingerprints are from the same finger. Fingerprint identification is one of the most reliable and valid personal identification method, which has been in use for a long time [40, 54, 108, 85, 103, 106]. Automatic fingerprint identification has been studied since the early 1970's and a significant progress has been made. A large number of automatic fingerprint identification systems for both forensic applications and civilian applications are installed

worldwide. However, fully automatic fingerprint identification is still a challenging problem [85, 106].

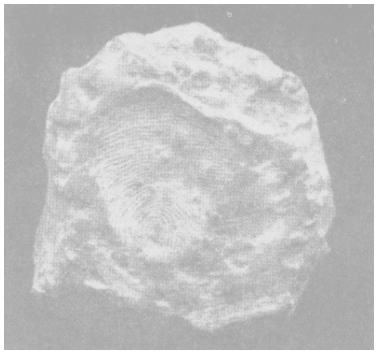
2.1 History of Fingerprint Identification

Humans have used fingerprints for a very long period of time [40, 54, 108, 85, 103, 27, 38, 112, 124, 109, 135]. Human fingerprints have been discovered on a large number of archaeological artifacts and historical items (refer to Figure 2.1 for some examples). Although these archaeological artifacts and historical items provide sufficient evidence to show that ancient people were aware of the individuality of fingerprints, such awareness does not appear to have any scientific basis [85, 103].

It was not until the late 16th century that the modern scientific fingerprint technique was first initiated [40, 54, 108, 85, 103]. In 1864, English plant morphologist, Nehemiah Grew, published the first scientific paper reporting his systematic study on the ridge, furrow, and pore structure in fingerprints (Figure 2.2) [85]. Since then, a large number of researchers have invested huge amounts of effort on fingerprint studies. In 1788, a detailed description of the anatomical formations of fingerprints was made by Mayer [103] in which a number of fingerprint ridge characteristics were identified and characterized (Figure 2.3). Starting in 1809, Thomas Bewick began to use his fingerprint as his trademark (Figure 2.4), which is believed to be one of the most important milestones of the scientific study of fingerprint identification [103]. Purkinje, in 1823, proposed the first fingerprint classification scheme which classified fingerprints into nine categories according to the ridge configurations (Figure 2.5) [103].

Neolithic Carvings (Gavrinis Island) [103]

Standing Stone (Goat Island, 2,000 B.C.) [85]



A Chinese clay seal (300 B.C.) [85]

An impression on a Palestinian lamp (400 A.D.) [103]

Figure 2.1: Examples of archaeological fingerprint carvings and historic fingerprint impressions; although the impressions on the Neolithic carvings and the Goat Island standing stones might not be used to indicate the identity, there is sufficient evidence to suggest that the Chinese clay seal and the impressions on the Palestinian lamp were used to indicate the identity of the providers.

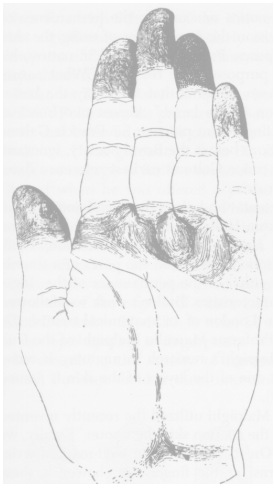


Figure 2.2: Dermatoglyphics drawn by Grew [103].

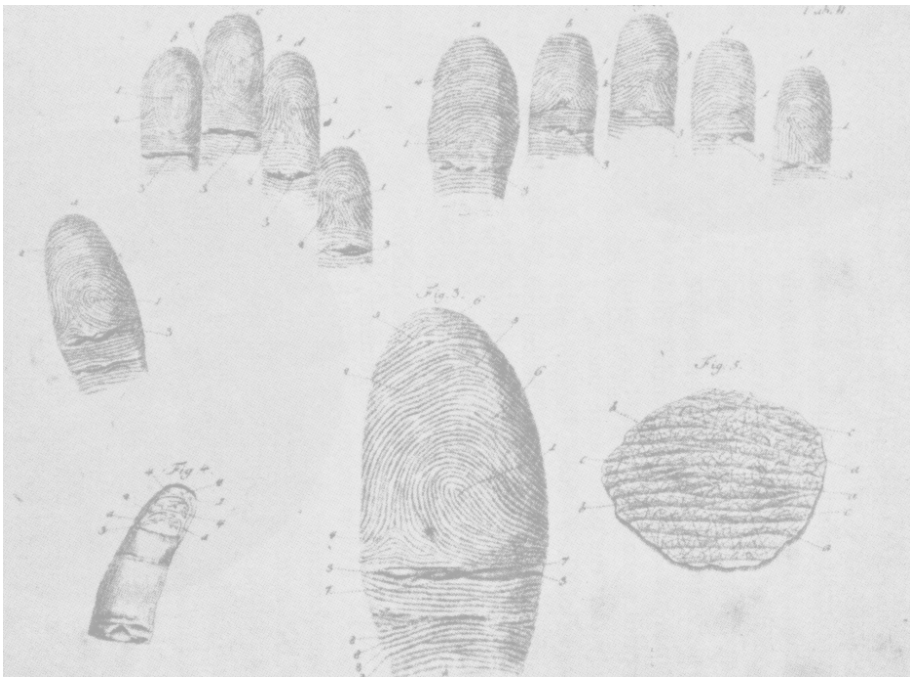


Figure 2.3: Mayer's drawings of fingerprints [40].

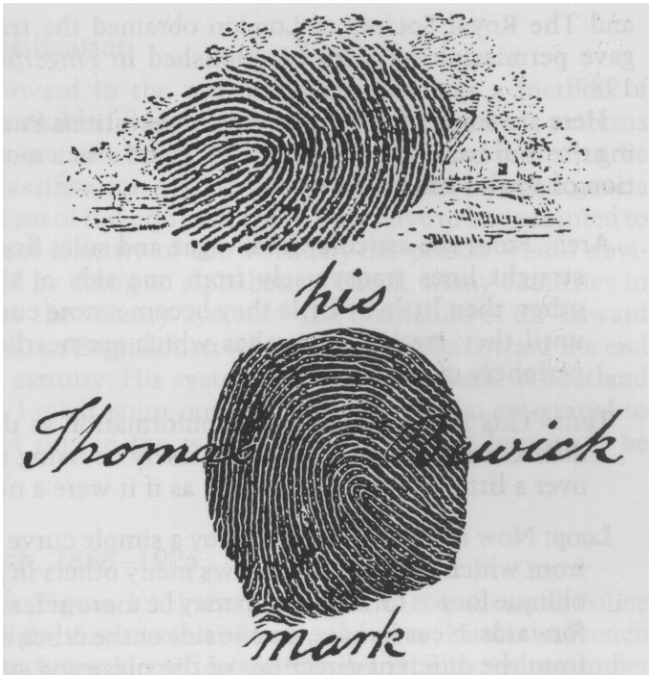


Figure 2.4: Trademarks of Thomas Bewick [85].

Figure 2.5: The nine patterns illustrated in Purkinje's thesis [103].

Henry Fauld, in 1880, first scientifically suggested the individuality of fingerprints based on his own observation. At the same time, Herschel asserted that he had practiced fingerprint identification for about 20 years [85, 103]. This discovery established the foundation of modern fingerprint identification. In the late $19th$ century, Sir Francis Galton conducted an extensive study on fingerprints [54]. He introduced the minutiae features for single fingerprint classification in 1888. An important advance in fingerprint identification was made in 1899 by Edward Henry, who (actually his two Indian assistants) established the well known "Henry system" of fingerprint classification [85, 106]. By the early 20th century, the formations of fingerprints were well understood. The biological principles of fingerprints are now well established [103] and are summarized below:

- *Individual epdiermal ridges and furrows have different characteristics for different fingerprints.*

- *The configuration types are individually variable, but they vary within limits which allow for a systematic classification.*

- *The configurations and minute details of individual ridges and furrows are permanent and unchanging.*

The first principle constitutes the foundation of fingerprint identification and the second principle constitutes the foundation for fingerprint classification.

In the early $20th$ century, fingerprint identification was formally accepted as a valid personal identification method and became a standard routine in forensics [85]. Fingerprint identification agencies were setup worldwide and criminal fingerprint databases were established [85]. Various fingerprint identification techniques, including latent fingerprint acquisition, fingerprint classification, and fingerprint matching were developed. For example, the FBI fingerprint identification division was setup in 1924 with a database of 810,000 fingerprints [108].

With the rapid expansion of fingerprint identification in forensics, operational fingerprint databases became so huge that manual fingerprint identification became infeasible. For example, the total number of fingerprints in the FBI fingerprint database now stands at 70 million from its original number of 810,000. With thousands of identification requests being received daily, even a team of more than 1,300 fingerprint experts were not able to provide timely responses to these requests [85]. Starting in the early 1960's, FBI, Home Office in the UK, and Paris Police Department began to

invest a large amount of effort to develop automatic fingerprint identification systems (AFIS) [85, 108]. Based on the observations of how human fingerprint experts perform fingerprint identification, three major problems in designing AFISs were identified and investigated: (*i*) digital fingerprint acquisition, (*ii*) local ridge characteristic extraction, and (*iii*) ridge characteristic pattern matching. Their efforts were so successful that a large number of commercial AFIS are currently installed and in operation in law enforcement agencies worldwide. These systems have greatly improved the operational productivity of these agencies and reduced the cost of hiring and training human fingerprint experts.

Recently, due to the rising demand in our increasing electronically inter-connected society for automatic personal identification and the success of various AFIS installations in forensics, automatic fingerprint identification technology has rapidly grown beyond forensic applications into civilian applications [106]. In fact, fingerprint based biometric systems are so popular that they have almost become the synonym of biometric systems [45].

2.2 Fingerprint Acquisition

Depending on whether the acquisition process is *online* or *offline*, a fingerprint may be either (*i*) an *inked fingerprint* or (*ii*) a *live-scan fingerprint*.

Inked fingerprint is a term which is used to indicate that the fingerprint image is obtained from an impression of the finger on an intermediate media such as paper. Generally, inked fingerprint is obtained using the *rolled method*, called *rolled*

(a) (b) (c)

Figure 2.6: Comparison of different fingerprint impressions: (a) a rolled fingerprint (from NIST 4 database); (b) a live-scan fingerprint (captured with a scanner manufactured by Digital Biometrics); (c) a latent fingerprint.

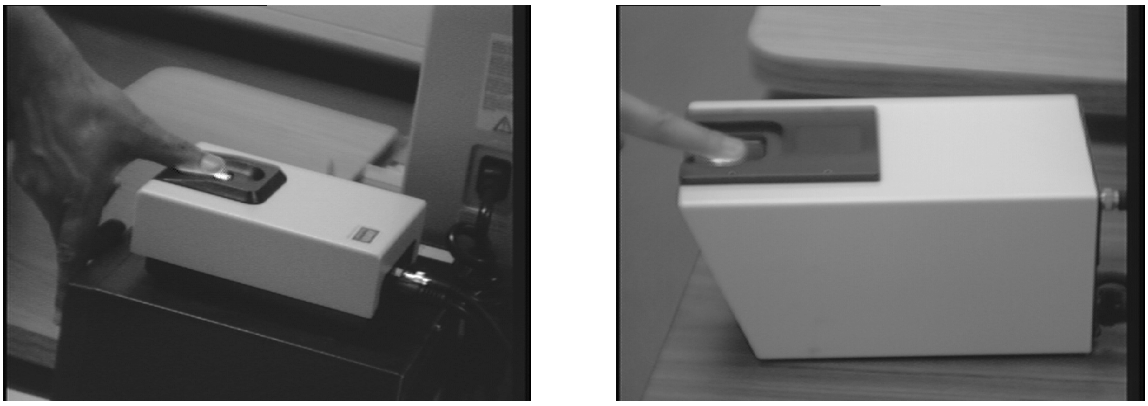
inked fingerprint. An example of a rolled inked fingerprint is shown in Figure 2.6 (a). Typically, the first step in capturing a rolled impression of a fingerprint is to place a few dabs of ink on a slab and rolling it out smoothly with a roller until the slab is covered with a thin, even layer of ink. Then the finger is rolled from one side of the nail to the other side over the inked slab which inks the ridge patterns on top of the finger completely. After that, the finger is rolled on a piece of white paper so that the inked impression of the ridge pattern of the finger appears on the white paper. Rolled inked fingerprints impressed on paper can be electronically scanned into digital rolled fingerprints using optical scanners or video cameras. So far, rolled acquisition method remains the most popular acquisition technique. In fact, it has been essentially a standard technique for fingerprint acquisition for more than a hundred years [108, 103]. Rolled inked fingerprints tend to have a larger area of valid ridges and furrows, but have large deformations due to the inherent nature of the rolled acquisition process. Direct feedback is not available to the subject to control the acquisition process which, in turn, may result in difficulties in quality control.

Acquisition of rolled fingerprints is cumbersome and slow. In the context of an automatic personal identification system, it is both infeasible and socially unacceptable to use the rolled inked method to acquires fingerprints in the operational phase although it may be feasible to use the rolled inked method in the enrollment phase[1].

In forensics, a special kind of inked fingerprints, called *latent fingerprints*, is of great interest. Constant perspiration exudation of sweat pores on fingerprint ridges and intermittent contact of finger with other parts of human body and various objects leave a film of moisture and/or grease on the surface of fingers. In touching an object, the film of moisture and/or grease may be transferred to the object and leave an impression of the ridges thereon. This type of fingerprints is called latent fingerprint. Latent fingerprints are very important in forensics. Actually, a major task in forensic fingerprinting application is searching and reliably recording latent fingerprints [85, 103], which is beyond the scope of this thesis. An example of a latent fingerprint is shown in Figure 2.6 (c).

The live-scan fingerprint is a collective term for a fingerprint image directly obtained from the finger without the intermediate step of getting an impression on a paper. A number of sensing mechanisms can be used to sense the ridge and furrows of the finger impressions, including (*i*) optical frustrated total internal reflection (FTIR) [3, 60, 137, 61, 77], (*ii*) ultrasonic total internal reflection [126], (*iii*) optical total internal reflection of edge-lit holograms [2, 53, 129], (*iv*) thermal sensing of the temperature differential (across the ridges and valleys) [83, 39], (*v*) sensing of differential capacitance [93, 130, 146], and non-contact 3D scanning [88]. Scanners based

[1]For example, Master Card relies on inked impressions for *enrollment*.



(a) (b)

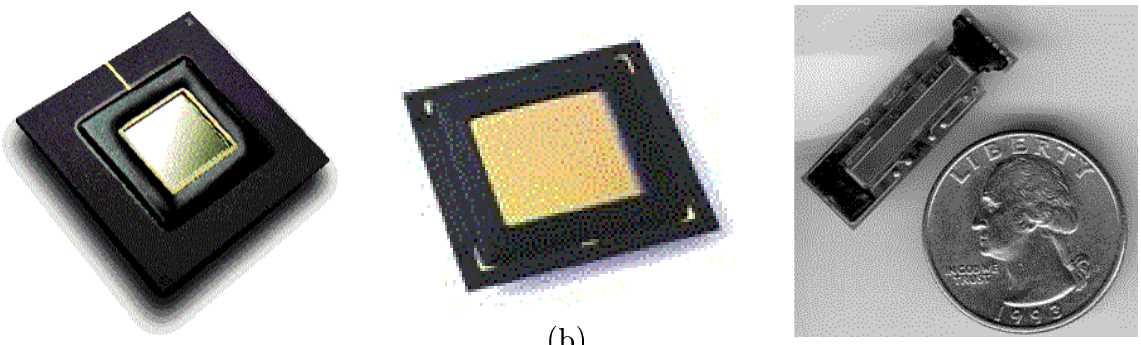
Figure 2.7: FTIR fingerprint scanners: (a) manufactured by *Identix*; (b) manufactured by *Digital Biometrics*.

on these physical processes can be used to acquire the impressions, called live-scan fingerprints, of human fingers directly. These acquisition methods eliminate the intermediate digitization process of inked impressions and makes it possible to build on-line systems. Depending on the clarity of ridge structures of scanned fingers and acquisition conditions, acquired live-scan fingerprints vary in quality. However, since there exists a direct feedback on such type of devices, it is relatively easier to control the quality of acquired fingerprints.

A live-scan fingerprint is usually obtained using the *dab method*, in which a finger is impressed on the acquisition surface of a device without rolling[2]. A dab live-scan fingerprint only captures the ridges and furrows that are in contact with the acquisition surface. Therefore, it tends to have a smaller area of valid ridges and furrows and smaller deformations than a rolled fingerprint.

The most popular technology to obtain a live-scan fingerprint image is based on

[2]It is also possible to capture a rolled live-scan fingerprint. Some vendors are trying to develop such fingerprint scanners.



(a) (b) (b)

Figure 2.8: Solid state fingerprint chips: (a) differential capacitance fingerprint chip manufactured by *Harris* [130]; (b) differential capacitance fingerprint chip manufactured by *Veridicom* [146]; (c) thermal fingerprint chip manufactured by Thomson CSF [39].

optical frustrated total internal reflection (FTIR) concept [60, 125]. When a finger is placed on one side of a glass platen (prism), ridges of the finger are in contact with the platen, while the valleys of the finger are not in contact with the platen. The rest of the imaging system essentially consists of an assembly of an LED light source and a CCD placed on the other side of the glass platen. The laser light source illuminates the glass at a certain angle and the camera is placed such that it can capture the laser light reflected from the glass. The light which is incident on the plate at the glass surface touched by the ridges is randomly scattered while the light incident at the glass surface corresponding to valleys suffers total internal reflection, resulting in a corresponding fingerprint image on the imaging plane of the CCD. An example of live-scan fingerprint is shown in Figure 2.6 (b). Figure 2.7 shows the two FTIR fingerprint scanners used in our prototype systems.

Optical scanners are too large to be readily integrated in a number of applications such as laptop security, cellular phone security, and notebook security. Recently, a number of different types of compact solid state fingerprint chips have become avail-

able. The quality of the images acquired using these solid state chips is comparable to the quality of images acquired using optical scanners. These solid state chips can be manufactured with a very low cost if manufactured in a large quantity. Figure 2.8 shows the three different types of solid state fingerprint chips which are commercially available.

2.3 Fingerprint Classification

Global patterns of ridges and furrows in the central region of fingerprints form special configurations, which have a certain amount of intraclass variability. But these variations are sufficiently small which allows for a systematic classification of fingerprints.

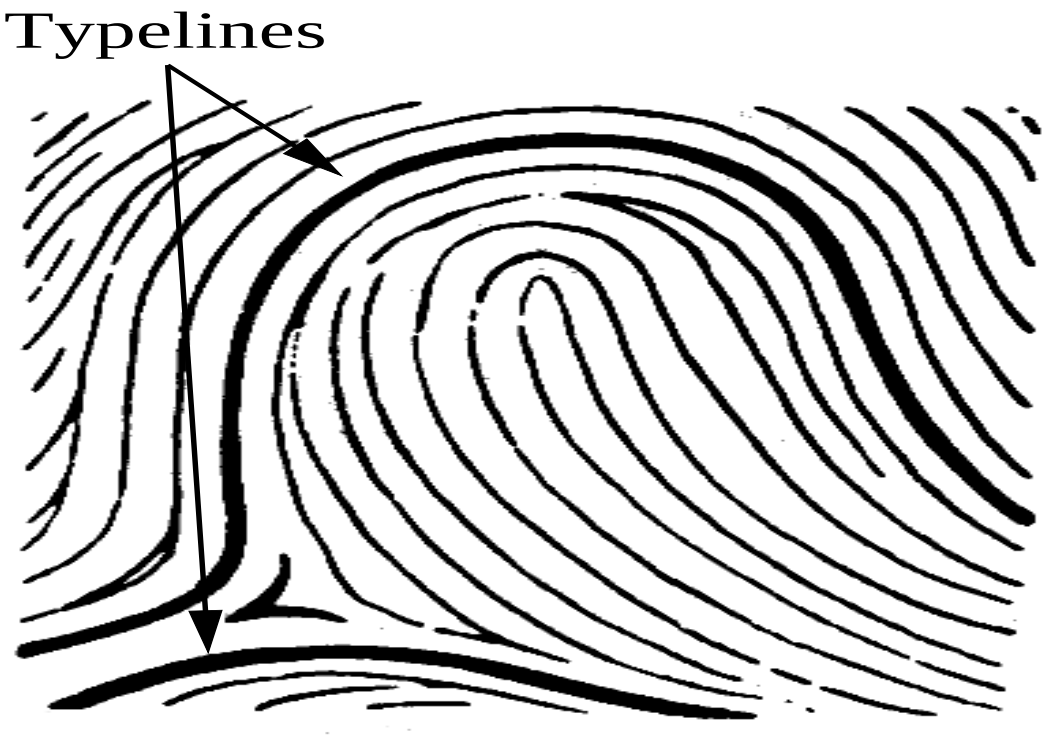


Figure 2.9: Pattern area and typelines.

In regard to fingerprint classification, only a portion of a fingerprint, called *pattern area* is of interest [108]. The pattern area of a fingerprint consists of those ridges encircled by *typelines* which is defined as the two innermost ridges that form a divergence tending to encircle or encompass the central portion of a fingerprint (Figure 2.9 shows an example of pattern area and typelines) [108]. The pattern areas of loop or whorl types of fingerprints contain two types of *singular points*, (*i*) *delta* and (*ii*) *core*. The delta, sometimes called the outer terminus, is defined as the point of ridge at or in front of and nearest to the center of the divergence of the typelines. It may be a ridge dot, a short ridge, the forking point of a bifurcated ridge, ending ridge, or the point on the ridge running in front of the divergence nearest to the center between the innermost diverging ridges. Examples of delta configurations are shown in Figure 2.10. The core, sometimes called the inner terminus, is defined as the specific point located on or within the innermost *sufficiently curved* ridges. Due to large variations in the formations of curved ridges, the rules for the selection of the core are very complicated. Figure 2.11 shows several examples of core configurations. Another important concept in both fingerprint classification and fingerprint matching is *ridge count*, which may be roughly defined as the number of ridges that touch or cross an imaginary line drawn between the core and delta. Due to the high complexity of ridge configurations, a precise definition of ridge count is difficult. Three simple ridge counting examples are shown in Figure 2.12. In this thesis, we extend the definition of ridge count to be the number of ridges that touch or cross an imaginary line drawn between a given pair of minutiae.

With the above definitions, fingerprint categories can be described as follows. A

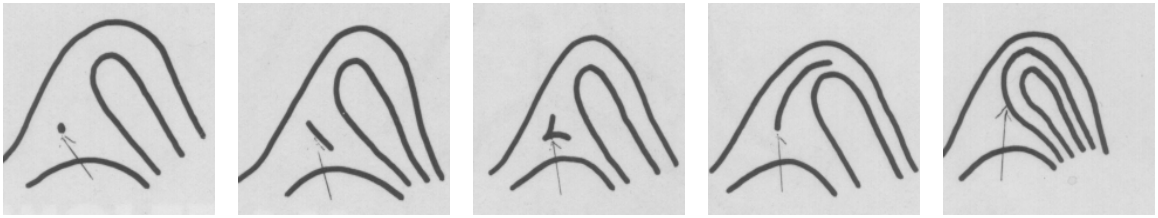


Figure 2.10: Examples of delta configuration [103].

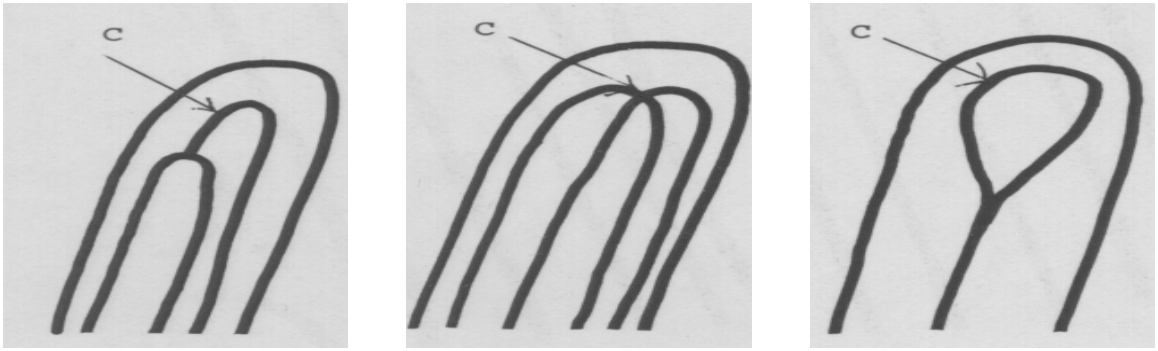


Figure 2.11: Examples of core configuration [103].

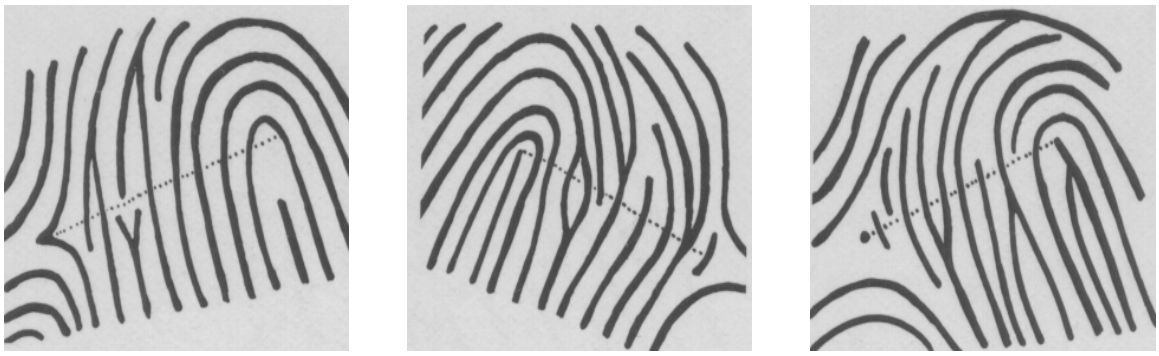


Figure 2.12: Ridge counting [103].

loop is the type of fingerprint in which "one or more of the ridges enter on either side, recurve, touch or pass an imaginary line drawn from the delta to the core, and terminate or tend to terminate on or toward the same side from which such ridge or ridges entered" [108]. There are three essential ingredients for classifying a fingerprint into a loop: (*i*) at least one sufficiently recurve ridge, (*ii*) a delta, and (*iii*) nonzero ridge count. Loops may be further divided into *lunar* loop and *radial* loop depending on the orientation tendency and fingers. About 60-65% of human fingerprints belong to this category [108, 103].

A whorl is that type of fingerprint in which "at least two deltas are present with a recurve in front of each" [108]. This definition, though very general, captures the essence of the category. Whorls may be further divided into four sub-categories: (*i*) plain whorl, (*ii*) central pocket loop, (*iii*) double loop, and (*iv*) accidental. About 30-35% of human fingerprints belong to this category [108, 103].

Arch is a special type of fingerprint configuration. Less than 5% of all fingerprints are arches [108, 103]. Arch may be divided into two sub-categories: (*i*) plain arch and (*ii*) tented arch. A plain arch is that type of fingerprint in which ridges enter one side and flow or tend to flow out the other with a rise of wave in the center [108]. In a tented arch, most of the ridges enter one side and flow or tend to flow out the other with a rise of wave in the center and the rest of the ridges form a definite angle, or up-thrush [108].

Fingerprint classification still remains a very difficult problem for both human experts and automatic systems [108, 103]. On the one hand, only a limited number of major fingerprint categories have been identified and the distribution of fingerprints

(a) (b) (c)

Figure 2.13: Examples of fingerprints that are difficult to classify; (a) tented arch; (b) a loop; (c) a whorl; it seems that all the fingerprints shown here should be in the loop category.

into these categories is not uniform. On the other hand, as we mentioned above, there exists a large variation in fingerprint configurations. The definition of each fingerprint category is both complex and vague. A human inspector needs a long period of experience to reach a satisfactory performance in performing fingerprint classification. In fact, fingerprint classification is more like an art than a science, since the long period of experience can only be gained by practice [103]. Figure 2.13 shows examples of fingerprints that are difficult to classify.

2.4 Fingerprint Matching

Although fingerprint category information and other global pattern configurations such as the number and positions of core and delta and ridge count may indicate, to a certain extent, the individuality of fingerprints, the uniqueness of a fingerprint is exclusively determined by the local ridge characteristics and their relationships. Fingerprint matching depends on the comparison of local ridge characteristics and

their relationships to determine the individuality of fingerprints. A total of one hundred and fifty different local ridge characteristics, called minute details, have been identified [103]. These local ridge characteristics are not evenly distributed. Most of them depend heavily on the impression conditions and quality of fingerprints and are rarely observed in fingerprints. The two most prominent ridge characteristics, called minutiae, are (*i*) *ridge ending* and (*ii*) *ridge bifurcation*. A ridge ending is defined as the ridge point where a ridge ends abruptly. A ridge bifurcation is defined as the ridge point where a ridge forks or diverges into branch ridges. Minutiae in fingerprints are generally stable and robust to the fingerprint impression conditions. Normally, they can be easily identified. Examples of minutiae are shown in Figure 2.14. For a given fingerprint, a minutia can be characterized by its type, its x and y coordinates, and its direction whose definition is also shown in Figure 2.14.

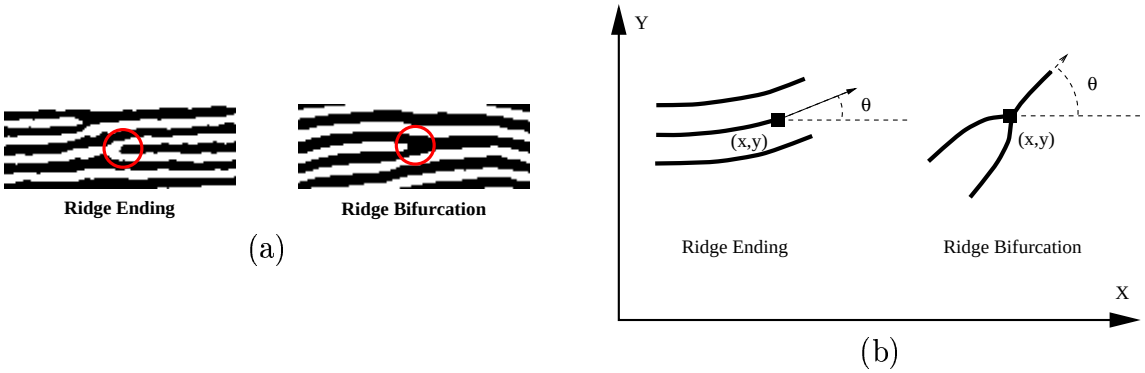


Figure 2.14: Minutiae; (a) example of minutiae; (b) characterization of minutiae.

If two fingerprints belong to the same category and have a sufficient number of minute details that are identical, then it can be concluded confidently that they are from the same finger. Generally, in order to determine that two fingerprints are from the same finger, four factors must be evaluated: (*i*) general pattern configuration

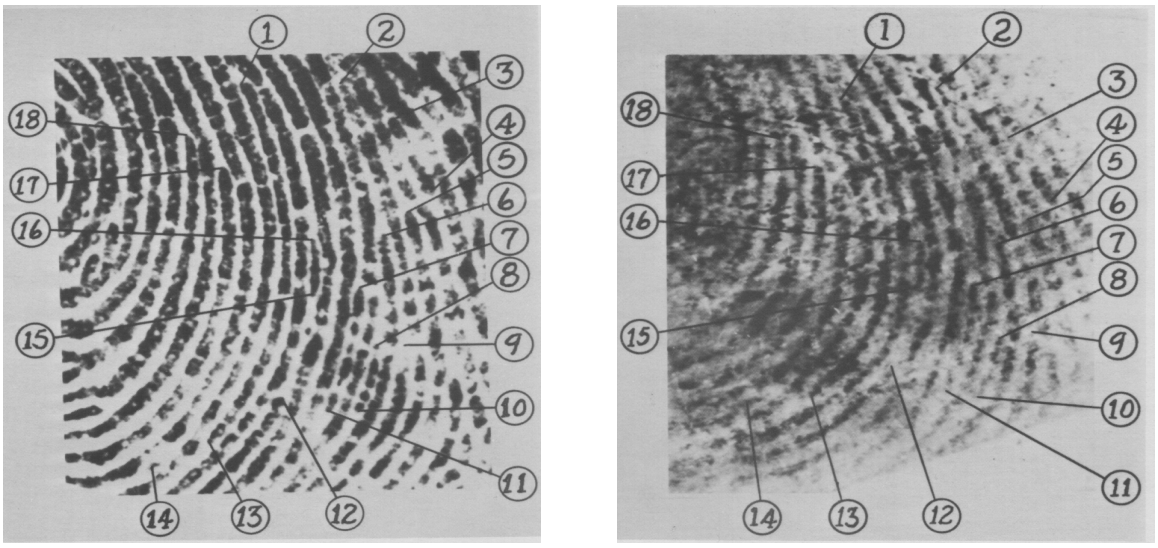


Figure 2.15: Fingerprint matching result in which 18 identical minute details are identified [103].

agreement which means that two fingerprints must be of the same pattern configuration, (*ii*) qualitative concordance which requires that the corresponding minute details must be identical, (*iii*) quantitative factor which specifies that at least a certain number (a minimum of 12 according to the forensic guidelines in the United States) of corresponding minute details must be found, and (*iv*) relationship of minute details which specifies that the corresponding minute details must be identically interrelated. In practice, complex identification protocols have been defined for fingerprint matching. These protocols are carefully designed based on the knowledge of fingerprint experts. A detailed flow chart is available to guide fingerprint examiners in performing fingerprint matching.

Although various protocols for fingerprint matching may be different in the concept definition and decision making process, the major steps in the associated flow charts are essentially the same. Typically, a fingerprint matching process is executed with an iterative three-stage process. First of all, two fingerprints to be matched

are compared to determine whether they are similar to each other in global pattern configuration. If the two fingerprints are totally different in terms of global pattern configuration, it is impossible that these two fingerprint are from the same finger. Next, a pattern alignment process is conducted in which several salient feature points are first located from the fingerprints and, then, an approximate alignment of the fingerprints is performed. Finally, a matching process is conducted in which corresponding minute details are evaluated in the valid fingerprint pattern areas and a decision is made based on the identified corresponding pairs and pattern configuration. Due to variations in fingerprint quality, impression deformation, fingerprint ridge configuration, and skin conditions, several steps in fingerprint matching protocols can not be clearly and precisely defined. Fingerprint examiners must depend heavily on their experience to make decisions. For example, even the most prominent minute details, minutiae, can not be identified easily. Some ridge bifurcations may be inevitably identified as ridge endings if the fingerprint impression pressure is too low. Therefore, although fingerprint matching is practiced daily by thousands of operational fingerprint experts around the world, fingerprint matching is still an art instead of a science. Experience plays a key role in manual fingerprint matching. Figure 2.15 shows an example of fingerprint matching result in which 18 corresponding minute details have been identified.

Chapter 3

System Design

The design of a biometric system can be characterized at two different levels: (*i*) system level and (*ii*) algorithm level.

3.1 System Level Design

The major issues at the system level design include which biometric characteristics should be used, which operational mode should be used, how to acquire a raw digital representation of the biometric characteristic, the system architecture, and other issues such as ergonomics, physical size, power supply, weight, cost, administrative and maintenance costs, and environmental influence.

The selection of a biometric characteristic is mainly determined by the practical requirements, especially performance requirements. The practical performance requirement is very much application related. On the one extreme, from the view point of system accuracy, a low false reject rate may be the primary objective. For example,

in some forensic applications such as criminal identification, it is the false reject rate that is a major concern and not the false acceptance rate: *i.e.*, we do not want to miss a criminal even at the risk of examining a large number of potential matches identified by the biometric system. In forensic applications, it is the human expert that will make the final decision anyway. On the other extreme, the false reject rate may be the most important factor in a highly secure access control application, where the primary objective is deterring impostors although we are concerned with the possible inconvenience to the legitimate users due to a high false reject rate. In between these two extremes are several civilian applications, where both false acceptance rate and false reject rate need to be considered. For example, in applications like ATM card verification, false reject rate is more important than the false acceptance rate – a false acceptance means a loss of several hundred dollars while a high false reject rate may irritate the customers. Obviously, even in civilian applications, a high false acceptance rate is not desirable since the main advantage of automation is defeated if human experts are involved in examining a long list of false positives. Ideally, we would like to have a reliable binary output - Is the subject in the system database or not? On the other hand, a high false reject rate is also not desirable since that would let in an impostor easily. But the risks involved in civilian applications are not as severe as in a criminal or security system. Figure 3.1 graphically depicts the situation discussed above.

Different biometric characteristics possess different discrimination capability in terms of system accuracy. At the one extreme, we have biometric characteristics such as face and dynamic signature that are inherently better at accepting genuine indi-

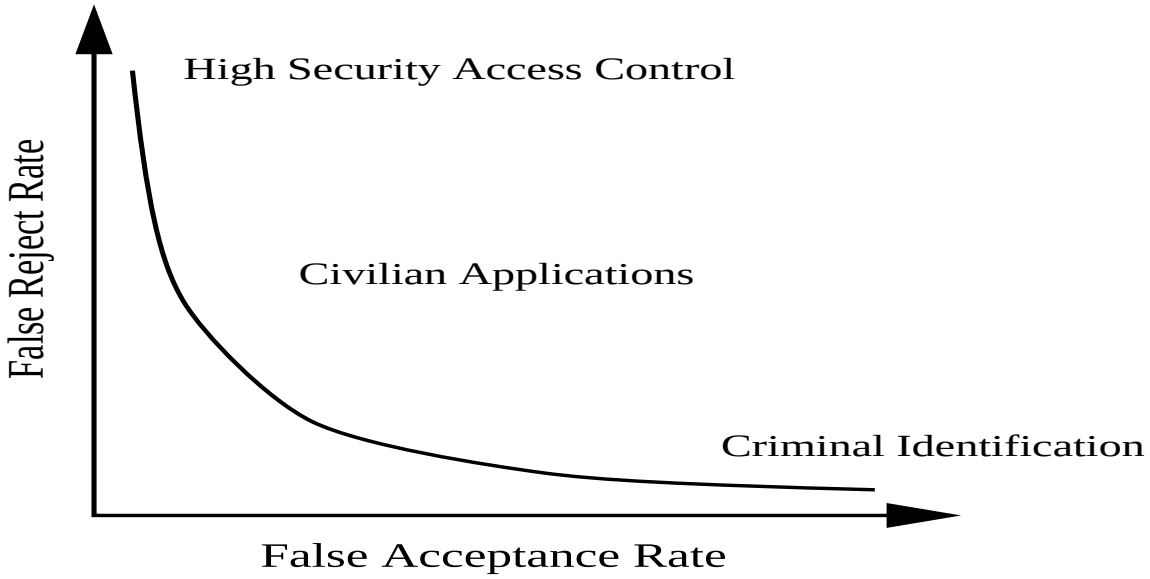


Figure 3.1: Different applications have different requirements for the FAR and FRR.

viduals, but do not perform well in deterring impostors. For example, an individual may be easily mistaken due to changes in makeup, hair style, lighting conditions, background, *etc*. At the other extreme, we have the biometric characteristics such as retinal scans, fingerprints, and iris that are better at preventing impostors but are less efficient in identifying genuine individuals. Somewhere in between these two extremes are those biometric characteristics such as hand geometry and hand vein which perform about the same in deterring impostors and accepting genuine individuals [106]. Generally, there is no rule of thumb to indicate which biometric technique should be used for a given application. A realistic design strategy is to examine what are the system requirements, assess which technique is suitable for the given application, and then tune the biometric system to satisfy the practical performance requirements.

As mentioned in Chapter 1, a biometric system may operate either in (i) verification mode or (ii) identification mode. It is more difficult to design an identification

system than to design a verification system [106]. For a verification system, the major challenge is the system accuracy. It is usually not very difficult to meet the response time requirement in a verification system, because only one-to-one comparison is conducted. However, for an identification system, both accuracy and speed are critical. An identification system needs to explore the entire template database to establish an identity. Thus, more requirements are imposed on the feature extractor and, especially, the feature matcher. Inherently, some biometric approaches are more feasible for operating in the identification mode than the others. For example, the individuality of fingerprints is mainly determined by the local minute details and their relationship. To automatically match a pair of unregistered fingerprints is computationally expensive. A linear search of the entire template database even for a small size database is not acceptable. Fingerprint classification may provide a partial solution to index a fingerprint database. However, it is doubtful that an one-finger-indexing mechanism based solely on the global fingerprint configuration, which is difficult to be registered in a fully automatic personal identification application, can reach the desirable accuracy[1]. Therefore, although a significant progress has been made in fingerprint identification, it is still not practical to conduct a real-time search even on a relatively small size fingerprint database of several thousand images without dedicated hardware matchers and an efficient indexing mechanism. On the other hand, it is feasible to design a face recognition system operating in the identification mode, because (*i*) face comparison is a relatively less expensive operation,

[1] An indexing mechanism based on multiple fingers such as the Henry System which is based on ten-finger classification is widely used in AFIS [85]. However, using all ten fingers to make a personal identification may not always be acceptable in a civilian application.

and (*ii*) efficient indexing techniques are available and the recognition performance is admissible [139].

Choosing the operational mode for a biometric system depends mainly on practical requirements. Generally, an identification system is more desirable than a verification system. However, an identification system usually requires more resources and, thus, costs more. In addition, an identification system may not always be technically practical. For example, an ATM card identification system needs to connect all the ATM machines around the world, establish a template database of millions of records, and it should be able to search through millions of template records in real time for each access authentication.

Data acquisition is one of the critical processes in a biometric system. The quality of the acquired data determines the performance of the entire system. However, the selection of a data acquisition device depends on practical requirements such as availability, cost, and size. There is no rule of thumb to determine which device should be used. In this thesis, we concentrate on online applications. Thus, we want a device which is able to acquire the fingerprint images directly from human fingers.

The biometric system architecture depends on the application. Logically, as mentioned in Chapter 1, a biometric system mainly consists of two modules: enrollment module and identification module. Each module consists of a number of sequential feed-forward submodules, which accept inputs from the previous submodule(s) and produce intermediate results which are, in turn, treated as inputs to the next submodule(s). The design of these submodules depends on the biometrics being used. It is tightly related to the algorithm level design.

3.2 Algorithm Level Design

Given the system level specifications and the practical requirements, the major tasks in algorithm level design are: (*i*) *feature extraction* and (*ii*) *matching*. Feature extraction is responsible for extraction of representative features from the raw input data. Matching is responsible for determining whether two sets of representative features are extracted from the same source. The algorithm level design also consists of other modules such as database management, quality control, encryption, and user interface, which are beyond the scope of this thesis.

The fingerprint representation (features) constitutes the essence of algorithm level design and determines almost all aspects of the recognition mechanism. A representation should have the following two properties: (*i*) *saliency* and (*ii*) *suitability*. Saliency means that a representation should contain enough class-specific (individual) information about the input data. Suitability means that the representation can be easily extracted, stored in a compact fashion, and is useful for matching. Saliency and suitability properties are not highly correlated. A salient representation is not necessarily a suitable representation. There is no general representation scheme that is suitable for all biometrics.

A matching algorithm is generally based on a similarity function to determine whether two sets of features are from the same source. For a given representation, deriving a similarity function is a very difficult problem because of intraclass and interclass variations. Typically, there is no systematic way to derive a similarity function.

For automatic fingerprint identification, it is well known that the acquired image has redundancy and tends to have large intraclass variations. Therefore, the fingerprint image itself is not a desirable representation. Currently, the two major representation schemes for automatic fingerprint identification are: (*i*) *image-based representation* and (*ii*) *feature-based representation*. The image-based representation assumes that the individuality of fingerprints may be exclusively determined in the spatial or the frequency domain. For example, a fingerprint can be represented by its Fourier spectrum. Due to orientation-specific flow pattern of ridges in the fingerprints, a concise representation may be obtained using the Fourier spectrum. The image-based representations usually require that the input image be registered. In practice, registering an input image is as difficult as matching itself. Therefore, although several image-based fingerprint representations have been proposed in the literature [52, 51, 9, 1, 7, 73, 74], the validity of these representations is still far from established.

The feature-based representation originates from the fact that if a pair of fingerprints belong to the same category (*e.g.*, arch, loop, whorl, *etc.*) and share a sufficiently large number of significant local ridge characteristics then it can be concluded confidently that they are from the same finger. Each fingerprint has a small number of significant local ridge characteristics. So, a compact and efficient fingerprint representation can be obtained. In addition, feature-based representation is widely accepted by the automatic fingerprint identification community. Its validity has been proven by the large number of automatic fingerprint identification systems in practical operation, which use this representation.

There are a large number of feature-based methods which utilize different types of minute details, including minutiae, singular points, orientation field, ridge counts, ridge pores, and ridges [87, 5, 12, 13, 15, 30, 47, 49, 57, 59, 69, 116, 123, 134, 71, 85, 142, 152, 121, 147, 62, 66, 25, 151]. In this thesis, we focus on a minutiae-based method, in which each fingerprint is represented by a minutiae pattern and matching is accomplished by determining the number of corresponding minutiae between the two patterns. There are two major tasks in minutiae-based matching: (*i*) *minutiae extraction* and (*ii*) *minutiae matching*. The objective of minutiae extraction is to extract the *minutiae* from input fingerprint images. Minutiae matching determines whether an extracted minutiae pattern and a stored template pattern are from the same finger or not.

3.3 Verification System

We have designed a prototype verification system which uses only fingerprints in identity authentication - conducts only one-to-one comparison to authenticate whether the identity claimed by an individual is true or not. It is designed for applications such as ATM card security, smart card security, information system security, and access control. The architecture of the prototype verification system is shown in Figure 3.2. Logically, the system consists of four major components: (*i*) user interface, (*ii*) system database, (*iii*) enrollment module, and (*iv*) verification module. The user interface provides a mechanism for a user to indicate her identity and present her fingerprints to the system. Depending on the application, the user interface can be

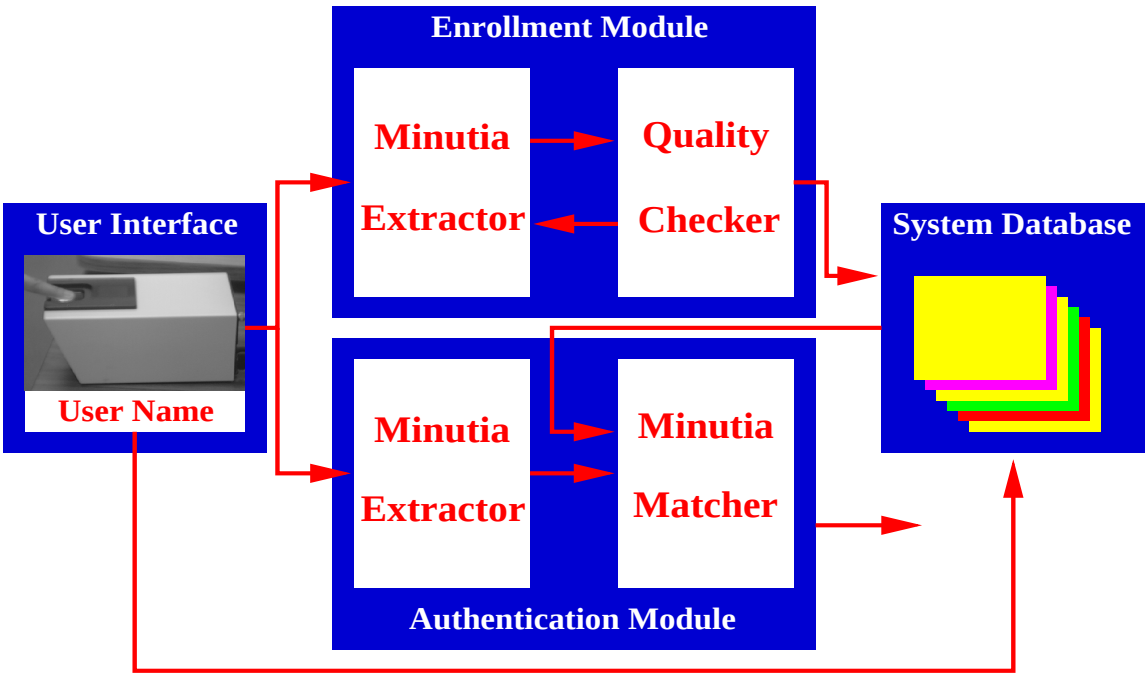


Figure 3.2: Architecture of the prototype automatic identity verification system.

designed to fit the practical requirements. In the prototype verification system, FTIR fingerprint scanners are used to acquire the live-scan fingerprint images. Figure 3.3 shows the graphical user interface (GUI) of our prototype verification system.

The system database consists of a collection of records, each of which corresponds to an authorized individual that has access to the system. Each record contains the following fields which are used for authentication purpose: (*i*) the profile of the individual and (*ii*) fingerprint templates of the individual. Depending on the application, the system database may be either a physical database that resides in the system or a virtual database with the record of each individual being carried on the magnetic card issued to the individual. For example, in information system security, a database that resides in the system may be used to store the record of each individual. At the point-of-access, the individual indicates her identity by entering her user name and

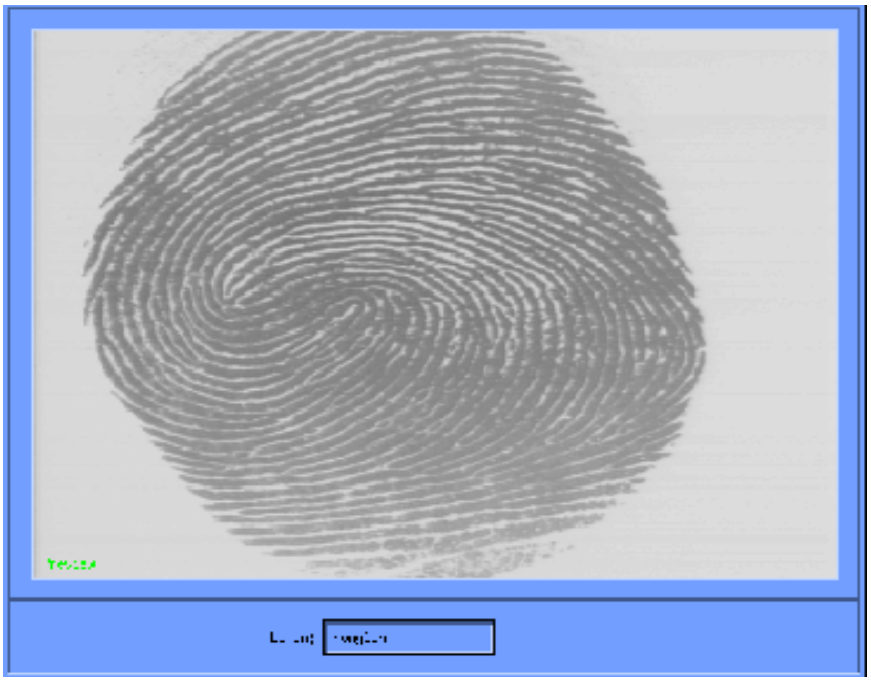


Figure 3.3: Graphical user interface of the automatic identity verification system.

the system retrieves the corresponding record from the database for authentication. In ATM card authentication, it may not be practical to have a database which stores all the records, since a large template database may become the point of failure. Generally, it is more efficient to store records on magnetic cards and let each individual keep her own magnetic card. At the point-of-access, the individual presents her magnetic card to indicate her identity and to provide the system her biometric template(s). In this case, the template database is only a virtual database and there is no physical database in the system. Both the number of templates and the quality of the templates for each individual are important design parameters of the verification system. On the one hand, the larger the number of templates and better the quality of the templates, the better the expected accuracy of the verification system. On the other hand, the larger the number of templates stored for each individual, the more

resources are required. There is obviously a trade-off.

The task of enrollment module is to enroll the profile of each individual and her fingerprint(s) into the system database. In our system, a compact but expressive biometric template which is generated by a feature extractor is used instead of the raw digital representation of the biometric characteristic. When the fingerprint images and the profile of an individual to be enrolled are fed to the enrollment module, a minutiae extraction algorithm is first applied to the fingerprint images and the minutiae patterns which are the valid representation of fingerprints are extracted. A quality checking algorithm is used to ensure that the records in the system database only consist of fingerprints of good quality, in which a significant number (default value is 25) of genuine minutiae may be detected. This is important, because there is no point in using a fingerprint with only a very small number of genuine minutiae as a template to make an authentication. If a fingerprint image is of poor quality, it is enhanced to improve the clarity of ridge/valley structures and mask out all the regions where minutiae cannot be reliably recovered. The enhanced fingerprint image is fed to the minutiae extractor again.

The task of verification module is to authenticate the identity of the individual who intends to access the system. The individual to be authenticated indicates her identity and places her finger on the fingerprint scanner; a digital image of her fingerprint is captured; minutiae pattern is extracted from the captured fingerprint image and fed to a minutiae matching algorithm which matches it against the individual's minutiae templates stored in the system database to authenticate whether the identity claimed by the individual is correct or not.

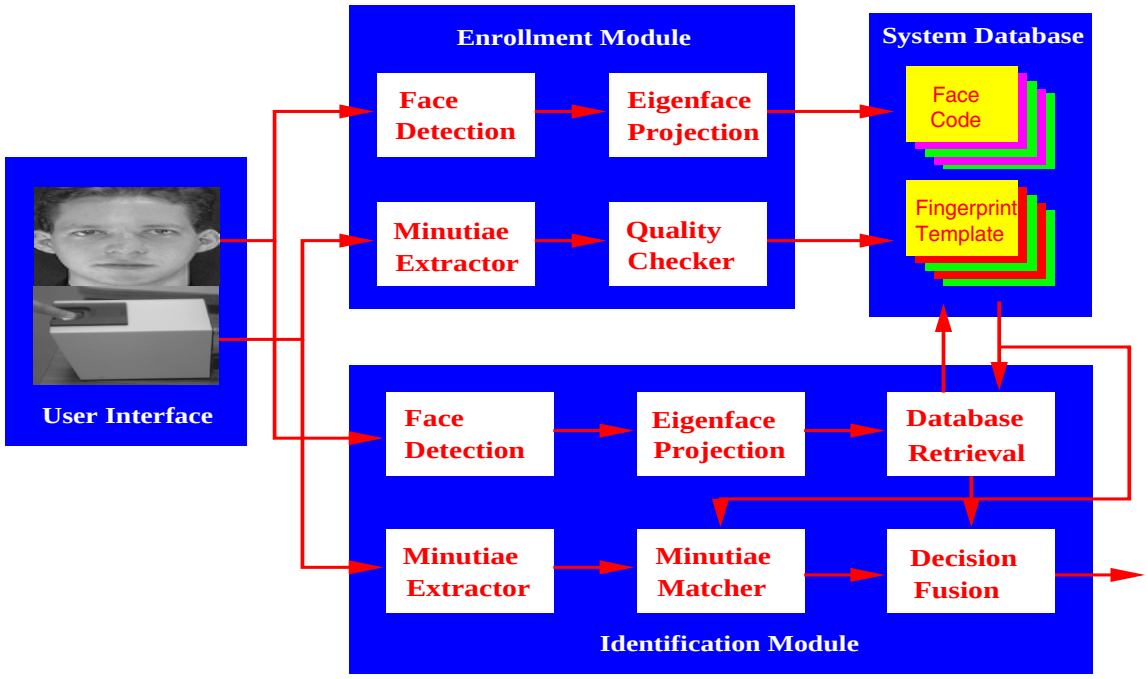


Figure 3.4: System architecture of the prototype integrated biometric identification system.

3.4 Identification System

The identification system we propose is mainly intended for the information system security and access control. The goal is to design an identification system that works in a limited environment such as an intranet environment in medium or small enterprises. In our identification system, we integrate multiple biometric characteristics (fingerprint and face) to improve the performance.

An important aspect that needs to be specified about an identification system is whether it is intended for conducting a fully automatic personal identification or not. An automatic fingerprint identification system (AFIS) is generally not deemed as a fully automatic system, since the candidates retrieved by the system usually need to be further examined by human experts to reach a final decision. However,

for certain applications such as information system security, it is not feasible to let human experts make the final decision. Instead, the system has to be fully automatic - the answer to a query needs to be either "yes" or "no". In this thesis, we focus on a fully automatic personal identification.

We have developed a prototype integrated biometric system that uses both facial and fingerprint information in conducting personal identification. The architecture of the system is shown in Figure 3.4. The system, in fact, consists of two subsystems, a face recognition subsystem and a fingerprint verification subsystem, which are integrated by a decision fusion module. The face recognition subsystem is responsible for retrieving the top n possible matches of a query from the template database, where n is usually a small number ($n = 5$ in our design). The fingerprint verification subsystem is responsible for matching the fingerprints of the top n possible matches of the query and providing the corresponding fingerprint matching scores. The decision fusion module integrates the results from the face recognition and the results from the fingerprint verification to establish the final decision.

Like the prototype verification system, logically, the prototype identification system also consists of four components: (*i*) user interface, (*ii*) system database, (*iii*) enrollment module, and (*iv*) identification module. The user interface is responsible for acquiring facial and fingerprint images of the users who intend to access the system. The system database stores the template records of all the individuals that have access to the system. Unlike the verification system, the system database in the identification system is always a physical database containing all the template records of the individuals who are enrolled in the system.

3.5 Difficult Problems

While a significant progress has been made in automatic fingerprint identification, there are still a number of research issues which need to be addressed to improve system performance. Some of these problems are listed below:

- *Robust live-scan fingerprint scanner*

  The quality of acquired fingerprint images is critical to the performance of an automatic fingerprint identification system. It is desirable to have a more advanced live-scan fingerprint scanner that is able to tolerate different types of skins, cuts and bruises on the finger, and dryness of the impressed finger.

- *Fingerprint feature extraction*

  In practice, a significant percentage of acquired fingerprint images is of poor quality. The performance of the feature extraction algorithms reported in the literature on different types of poor quality fingerprint images is still far from desirable. To design a feature extraction algorithm that is robust to different types of image degradations is a challenge. Examples of fingerprint images that are of very poor quality are shown in Figure 3.5. Figure 3.6 shows the extracted minutiae obtained by our algorithm on a fingerprint image of poor quality due to high humidity of the impressed finger.

- *Fingerprint enhancement*

  Fingerprint enhancement can be used to recover the genuine ridge structures from the corrupted images. However, to design a fingerprint enhancement al-

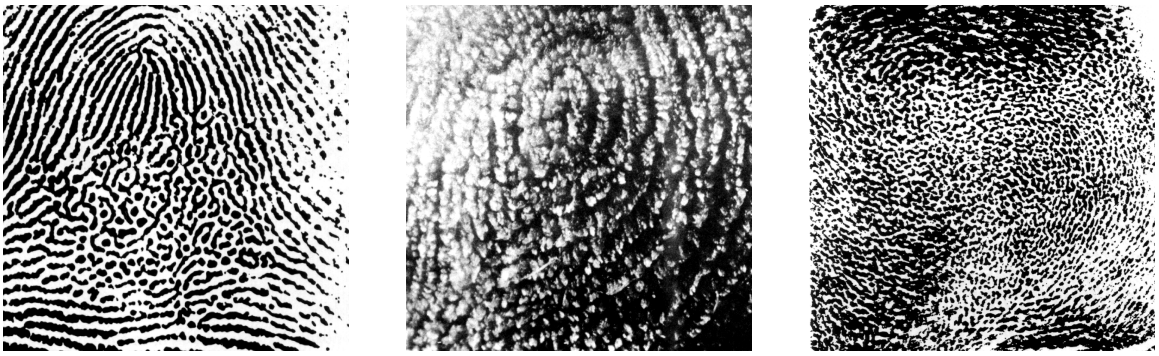


Figure 3.5: Fingerprint images of very poor quality.

Figure 3.6: Minutiae extraction from a poor quality image; white: correct minutiae; red: spurious minutiae; green: missing minutiae.

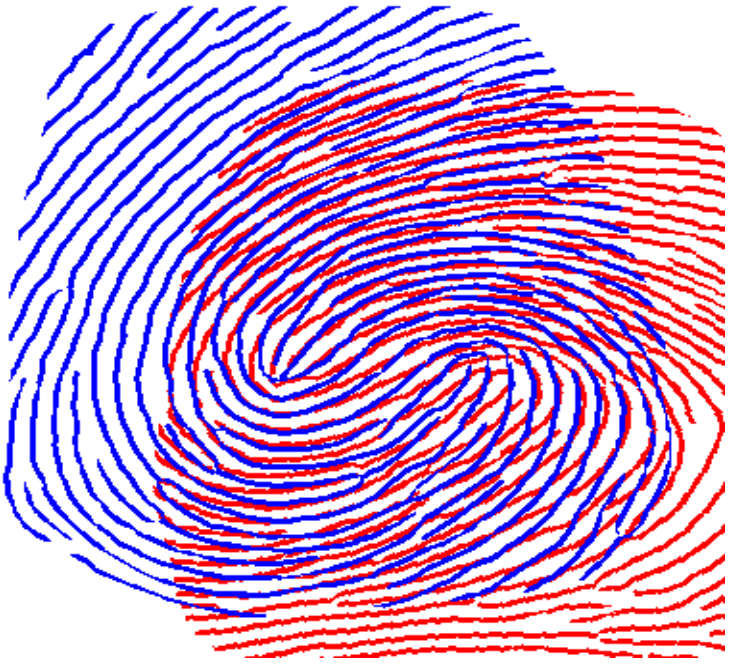


Figure 3.7: Fingerprint impression deformation.

gorithm that is able to handle all types of noise sources is very difficult.

- *Minutiae matching*

  The performance of minutiae matching algorithms depends heavily on the reliability of minutiae and external alignment. To design a minutiae matching that is able to handle different situations such as a large percentage of spurious and missing minutiae and impression deformations is still a very difficult problem. Figure 3.7 shows an example of an impression deformation.

- *Fingerprint classification*

  Although a number of automatic fingerprint classification methods have been proposed and some of them are used in operational AFISs, fingerprint classification still remains one of the most difficult problem for both humans and machines. Currently, the fingerprint classification framework is mainly intended

for human experts which may not be optimal for an automatic system.

- *Fingerprint compression*

  Without a good fingerprint compression scheme, storing hundreds of millions of fingerprints is too expensive. A wavelet-based method which has been proposed as the standard for fingerprint compression can compress a fingerprint image by a factor of 10 to 25 [32]. An algorithm that can reach even higher compression ratio is an important research topic.

- *Computational complexity of matching*

  Computational complexity is a very important issue in automatic fingerprint identification. It is a practical requirement that all verifications should be performed in "real time" for all online applications. However, to achieve both high accuracy and high speed poses another difficulty.

- *Integration of multiple biometric characteristics*

  An integration scheme that fuses multiple cues can be used to reach a desired performance that can not be reached using only a single biometric technique.

- *Performance evaluation*

  In designing a biometric system, an important issue is the performance assessment of the system: how to evaluate the performance of a given biometric system or how to verify that a deployed biometric system satisfies certain performance specifications? Unfortunately, the performance evaluation problem is far from well established.

The above difficulties are not isolated, instead they are highly correlated with one another. For example, if a perfect fingerprint scanner is available, which can acquire a very clear fingerprint image even though the impressed finger does not have clear ridge structures[2], then even a very simple minutiae extraction algorithm can locate all the minutiae without any errors. In turn, the minutiae matching can be simplified greatly.

[2]This is possible in theory by scanning the internal layers of friction skin.

Chapter 4

Minutiae Extraction

Minutiae extraction is to extract representative features, called minutiae, from the input fingerprint images. For automatic fingerprint matching, a salient and suitable representation of the input fingerprint images is critical. Generally, this representation should have the following properties [120]: (*i*) *retain the discriminating power of raw digital fingerprint images,* (*ii*) *compactness,* (*iii*) *amenable to matching algorithms,* (*iv*) *robust to noise and distortions, and* (*v*) *easy to compute.* The first property requires that a representation should be able to retain the individuality of fingerprints such that the identity can be reliablely established based solely on the representation. The second property insists that the representation should not contain information besides the individuality of the fingerprints. The third property postulates that the representation should be suitable for a matching algorithm. Clearly, the representation should be sufficiently robust to the quality of fingerprint images, which is specified in the fourth property. Finally, the representation should not be computationally demanding.

The pattern of the minute details of a fingerprint forms a valid representation of the fingerprint. It is compact, amenable to matching algorithms, robust to noise and distortions, and easy to compute. However, as indicated in chapter 2, most of the 150 types of minute details in fingerprint images are not stable and can not be reliably identified. In an automatic fingerprint matching, only the two most prominent types of minute details are used for their stability and robustness: (*i*) *ridge ending* and (*ii*) *ridge bifurcation*. In addition, since various data acquisition conditions such as impression pressure can easily change one type of minutiae into the other, we do not make any distinction between these two types of minutiae in our system. So, each minutiae is completely characterized by the following parameters: (*i*) x-coordinate, (*ii*) y-coordinate, and (*iii*) orientation (refer to Figure 2.14 for their definition). Typically, in a live-scan fingerprint image of good quality, there are about 50-100 minutiae.

A good minutiae extraction algorithm should be both reliable and efficient. Reliability means that the minutiae extraction algorithm should (*i*) not create spurious minutiae, (*ii*) not miss genuine minutiae, and (*iii*) be precise in minutiae position localization and minutiae orientation computation. Reliable extraction of minutiae from fingerprint images is a difficult task. When the quality of fingerprint images is good, the ridges and furrows in a fingerprint, which alternate and flow in a locally constant direction, are well-defined and are clearly differentiated from one another. In such situations, ridge endings and ridge bifurcations which are essentially anomalies of ridges can be easily identified and be precisely located from the binary ridges. Examples of good quality live-scan fingerprint images are shown in Figure 4.1. However, in

Figure 4.1: Examples of good quality live-scan fingerprint images, which were captured using a fingerprint scanner manufactured by Digital Biometrics.

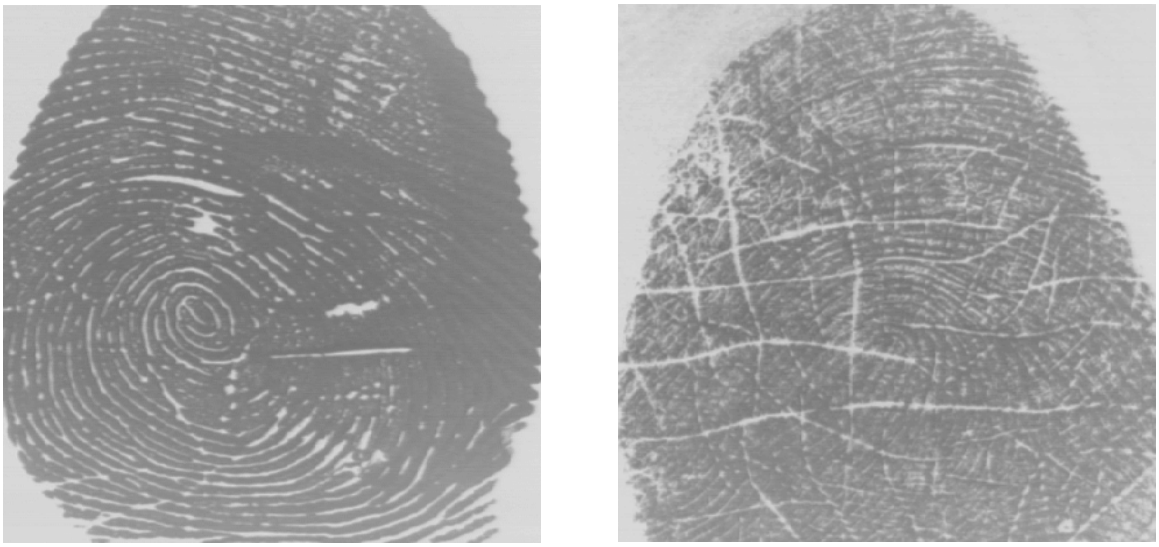


Figure 4.2: Examples of poor quality live-scan fingerprint images, which were captured using a fingerprint scanner manufactured by Digital Biometrics.

practice, a significant percentage of acquired fingerprint images (approximately 10%) is of poor quality. The ridge structures in such kind of fingerprint images are not always well-defined and hence they can not be correctly extracted. Thus, a significant number of spurious minutiae may be created, a large percent of genuine minutiae may be ignored, and large errors in their localization (position and orientation) may be introduced. Examples of fingerprint images of very poor quality, in which ridge structures are completely corrupted, are shown in Figure 3.5. Figure 4.2 shows examples of poor quality live-scan images. The minutiae extraction result on a poor quality fingerprint image is shown in Figure 3.6. Depending on the quality, a poor fingerprint image can be either rejected or enhanced prior to the minutiae extraction. A very poor fingerprint image in which ridge structures are corrupted completely should be rejected, while a poor fingerprint image in which ridge structures are still visible should be enhanced before minutiae extraction. A good minutiae extraction algorithm should be able to tolerate, to a limited extent, the corrupted ridge structures and degrade gracefully with the image quality.

It is critical that a minutiae extraction algorithm is able to operate in "real-time" in an online application such as ATM card security, smart card security, and access control. However, there is a trade-off between speed and reliability. In order for a minutiae extraction algorithm to be fast in speed, only simple operations which may not be robust to image quality can be allowed. On the other hand, in order for the minutiae extraction algorithm to be robust, complex operations which are usually computationally demanding are needed. A practical design strategy is to select a set of operations that are efficient in both speed and reliability.

4.1 Related Work

Extensive studies have been conducted on minutiae extraction [26, 12, 85, 100, 28, 29, 95, 94, 18, 120, 91, 115, 67, 140, 84, 48, 111, 107, 36, 122]. In the following, we will briefly review the well-known techniques used for minutiae extraction.

As indicated in the previous section, a critical task in minutiae extraction is ridge extraction. Ridge extraction is essentially a segmentation operation which separates ridges from the background (furrows). A global thresholding method is not able to correctly separate the ridges from the background [85]. An adaptive thresholding technique is necessary. One of the earliest attempts at minutiae extraction is the FBI minutiae reader which is a typical two-stage algorithm [12, 122, 85]. The algorithm adaptively binarizes the input fingerprint images using a "composite" approach and extracts the minutiae from the binarized ridges. The "composite" approach is essentially a local thresholding method based on the local ridge direction estimated by a "slit comparison" formula. This algorithm has been used as a standard algorithm for minutiae extraction in AFISs [85]. The performance of this algorithm is reasonable if the quality of the input fingerprint images is good. Moayer and Fu's algorithm [100] applies the Laplacian operator and dynamic thresholding iteratively to the gray-level input fingerprint images to extract ridges. Chatterjee *et al.* [28, 29] proposed a fuzzy approach which first enhances the input fingerprint images and then utilizes an adaptive thresholding method which preserves the same number of 1 and 0 pixels in each neighborhood to extract the ridges from the enhanced images. Although the above algorithms use the adaptive thresholding technique in ridge extraction, the perfor-

mance of these algorithms is good only when the quality of the input fingerprint images is reasonable. The adaptive thresholding techniques they employ do not fully exploit the important information residing in the local ridge orientation.

Fingerprints are flow-like patterns. Therefore the local ridge orientation provides very important information about the ridge structures. By incorporating local ridge orientation efficiently, the performance of minutiae extraction algorithm can be greatly improved. Mehtre [95, 94] proposed an algorithm using the directional image. The directional image, which represents the local ridge direction in a 16×16 neighborhood, is first computed and a set of eight 7×7 convolution masks is applied to the gray-level input fingerprint images to improve the quality of ridge structures. Then, the ridges are extracted by applying a locally adaptive thresholding method and a thinning operation is applied to the ridges. Finally, the minutiae are obtained based on the computation of the *connection number*. A post-processing stage based on a set of heuristics is used to eliminate the spurious minutiae. Although this algorithm established the basic principle of incorporating local ridge orientation in ridge extraction, its performance is not very impressive due to the inefficient utilization of the local ridge orientation. Botha and Coetzee [18] extract edges from the gray-level input fingerprint images using the Marr-Hildreth edge operator and compute the ridge orientation in a local neighborhood. Then, they binarize the gray-level input fingerprint images using a segmentation algorithm which conducts local thresholding based on the estimated local ridge orientation and extracted edges. Finally, they apply a thinning algorithm to the smoothed binary image and extract the minutiae from the thinned ridges. Again, due to the inefficient way that they utilize the local

ridge orientation, the performance of their algorithm is not impressive.

Ratha *et al.* [120] proposed a minutiae extraction algorithm in which the flow direction of the ridges is computed by viewing the fingerprint image as a directional textured image. A waveform projection-based algorithm is used for ridge extraction, the thinned skeleton of the extracted ridges is smoothed using morphological filters, minutiae are extracted from the skeleton ridges, and a postprocessing step is applied to delete spurious minutiae. Since the waveform projection and directional smoothing operation used in the algorithm are very effective in suppressing small amounts of noise, this algorithm performs very well. Maio and Maltoni [91] extract minutiae directly from gray-level fingerprint images. The algorithm is essentially a gray-level ridge tracer which extracts ridges by sequentially following each gray-level ridge until it reaches a ridge ending or a ridge bifurcation. Although they claim that the algorithm does not binarize the gray-level fingerprint image directly when conducting minutiae extraction, binarization is still conducted implicitly by the gray-level ridge tracer. The robustness of this algorithm with respect to image quality is questionable, due to the fact that the gray-level ridge tracer may behave unpredictably when ridges and furrows are not well defined.

In practice, the occurrences of minutiae in a fingerprint image follow certain rules. Therefore, a number of heuristics can be used to correct the minutiae errors. Xiao and Raafat [115] describe a method to identify and eliminate spurious minutiae using the structural information of minutiae. For each minutiae, statistics of ridge width and ridge attributes such as ridge length, ridge direction and minutiae direction are used to decide the spurious minutiae. Szekely and Szekely [140] proposed a minutiae

extraction algorithm based on the computation of the directional image divergence. Hung [67] enhanced binary fingerprint images by equalizing the ridge widths. Directional enhancement of ridges is done after estimating the local direction in a small window. The enhancement process has two steps: (*i*) direction-oriented ridge shrinking, followed by (*ii*) direction-oriented ridge expanding. In addition, methods for detecting bridges and breaks were also implemented.

Besides the attempts mentioned above, a number of alternative approaches have also been investigated [84, 48, 111, 26]. For example, Engeler *et al.* [48] introduced a neural network-based minutiae extraction algorithm, in which a multilayer perceptron is used to extract ridges from gray-level fingerprint images. The input to the multilayer perceptron is a set of Gabor filter responses in a local neighborhood. Unfortunately, the performances of these approaches have not been established.

In summary, a good minutiae extraction algorithm should efficiently incorporate local ridge orientation in ridge extraction. Directional ridge enhancement should be employed before the ridge extraction operation. However, it should also be kept in mind that directional smoothing is usually a computationally expensive operation. Postprocessing is a very important step for minutiae extraction, which can eliminate a significant number of errors.

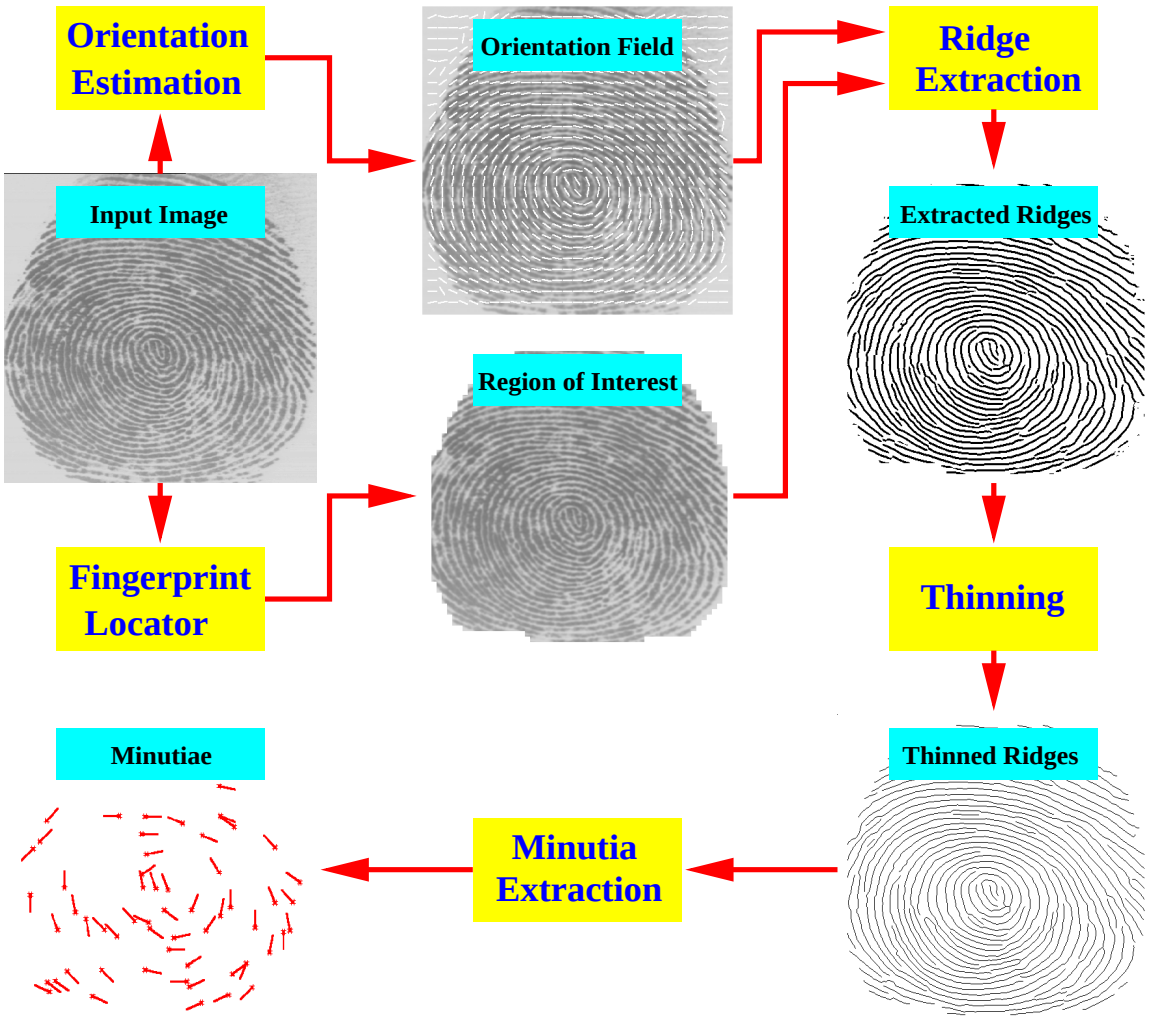


Figure 4.3: Flowchart of the minutiae extraction algorithm

4.2 Minutiae Extraction Algorithm

We have developed a minutiae extraction algorithm which is an improved version of the technique described in [120]. Our algorithm employs a more reliable and more efficient way to conduct adaptive ridge extraction than the original algorithm. Experimental results demonstrate that this algorithm not only performs very well, but it is also fast. The overall flowchart of this algorithm is depicted in Figure 4.3. It mainly consists of three stages: (*i*) orientation field estimation, (*ii*) ridge extraction, and (*iii*) minutiae extraction and postprocessing. First, for an input image, the local ridge orientation is estimated and the region of interest is located. Then, ridges are extracted from the input image, refined to get rid of the small speckles and holes, and thinned to obtain 8-connected single pixel wide ridges. Finally, minutiae are extracted from the thinned ridges and refined using some heuristics. In the following subsections, we will describe in detail our minutiae extraction algorithm. In our description, we assume that the resolution of input fingerprint images is 500 dpi, which is the recommended resolution for automatic fingerprint identification by the FBI [85].

4.2.1 Definitions

In order to introduce our minutiae extraction algorithm, a list of notations and some basic definitions are given below.

A *gray-level fingerprint image*, $\mathcal{I}$, is defined as a $N \times N$ matrix, where $\mathcal{I}(i, j)$ represents the intensity of the pixel at the ith row and jth column.

An *orientation field/image*, $\mathcal{O}$, is defined as an $N \times N$ image, where $\mathcal{O}(i,j)$ represents the *local ridge orientation* at pixel (i,j). Local ridge orientation is usually specified for a region (block) rather than at every pixel; an image is divided into a set of $w \times w$ non-overlapping blocks and a single local ridge orientation is defined for each block. Note that in a fingerprint image, the ridges oriented at 0^o and the ridges oriented at 180^o in a local neighborhood can not be differentiated from each other.

A *ridge map*, $\mathcal{R}$, is an $N \times N$ binary image, where $\mathcal{R}(i,j) = 1$ indicates that pixel (i,j) is a ridge pixel and $\mathcal{R}(i,j) = 0$ indicates that pixel (i,j) is not a ridge pixel. A ridge in a ridge map is an 8-connected component. A *thinned* ridge has a width of 1 pixel and a *thinned* ridge map, $\mathcal{TR}$, consists of thinned ridges.

4.2.2 Orientation Field Estimation

The orientation field of a fingerprint image represents an intrinsic nature of the fingerprint image and defines invariant coordinates for ridges and furrows around each local neighborhood, which plays a very important role in fingerprint image analysis. By viewing a fingerprint image as an oriented texture, a number of methods have been proposed to estimate the orientation field of fingerprint images [80, 118, 79, 26]. We have developed an iterated least mean square orientation estimation algorithm. The main steps of the algorithm are as follows:

1. *Divide the input fingerprint image into blocks of size $w \times w$. For 500 dpi images, the initial value of w is 16.*

2. *Compute the gradients $\partial_x(i,j)$ and $\partial_y(i,j)$ at each pixel, (i,j). Depending on the computational requirement, the gradient operator may vary from the simple* Sobel *operator to the more complex* Marr-Hildreth *operator [92].*

3. *Estimate the local orientation of each block centered at pixel* (i, j) *using the following equations [118]:*

$$\mathcal{V}_x(i,j) = \sum_{u=i-\frac{w}{2}}^{i+\frac{w}{2}} \sum_{v=j-\frac{w}{2}}^{j+\frac{w}{2}} 2\partial_x(u,v)\partial_y(u,v), \tag{4.1}$$

$$\mathcal{V}_y(i,j) = \sum_{u=i-\frac{w}{2}}^{i+\frac{w}{2}} \sum_{v=j-\frac{w}{2}}^{j+\frac{w}{2}} (\partial_x^2(u,v) - \partial_y^2(u,v)), \tag{4.2}$$

$$\theta(i,j) = \frac{1}{2} tan^{-1}(\frac{\mathcal{V}_y(i,j)}{\mathcal{V}_x(i,j)}), \tag{4.3}$$

where $\theta(i, j)$ *is the least square estimate of the local ridge orientation at the block centered at pixel* (i, j)*. Mathematically, it represents the direction that is orthogonal to the dominant direction of the* Fourier spectrum *of the* $w \times w$ *window.*

4. *Due to the presence of noise, corrupted ridge and valley structures, minutiae, etc. in the input image, the estimated local ridge orientation,* $\theta(i, j)$*, may not always be correct. Since local ridge orientation varies slowly in a local neighborhood where no singular points appear, a low-pass filter can be used to modify the incorrect local ridge orientation. In order to perform the low-pass filtering, the orientation image needs to be converted into a* continuous vector field, *which is defined as follows:*

$$\Phi_x(i,j) = \cos(2\theta(i,j)), \quad and \tag{4.4}$$

$$\Phi_y(i,j) = \sin(2\theta(i,j)), \tag{4.5}$$

where δ_x *and* δ_y*, are the* x *and* y *components of the vector field, respectively. With the resulting vector field, the low-pass filtering can then be performed as follows:*

$$\Phi'_x(i,j) = \sum_{u=-w_\Phi/2}^{w_\Phi/2} \sum_{v=-w_\Phi/2}^{w_\Phi/2} h(u,v)\Phi_x(i-uw, j-vw) \quad and \tag{4.6}$$

$$\Phi'_y(i,j) = \sum_{u=-w_\Phi/2}^{w_\Phi/2} \sum_{v=-w_\Phi/2}^{w_\Phi/2} h(u,v)\Phi_y(i-uw, j-vw), \tag{4.7}$$

where h *is a 2-dimensional low-pass filter with unit integral and* $w_\Phi \times w_\Phi$ *specifies the size of the filter. Note that the smoothing operation is performed at the block level. The default size of the filter is* 5×5*.*

5. *Compute the local ridge orientation at* (i, j) *using*

$$\mathcal{O}(i,j) = \frac{1}{2}\tan^{-1}(\frac{\Phi'_y(i,j)}{\Phi'_x(i,j)}). \tag{4.8}$$

6. *Compute the* consistency level *of the orientation field in the local neighborhood of a block* (i, j) *with the following formula:*

$$\mathcal{C}(i,j) = \frac{1}{n}\sqrt{\sum_{(i',j')\in D} |\mathcal{O}(i',j') - \mathcal{O}(i,j)|^2}, \tag{4.9}$$

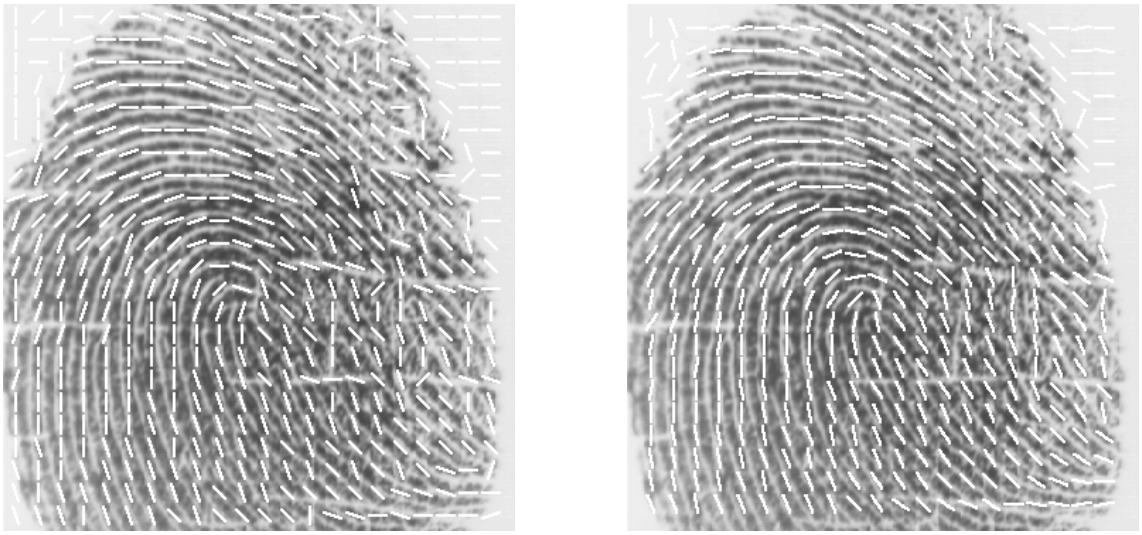
$$|\mathcal{O}(i',j') - \mathcal{O}(i,j)| = \begin{cases} d & \text{if } d < 180, \\ d - 180 & \text{otherwise}, \end{cases} \tag{4.10}$$

$$d = (\mathcal{O}(i',j') - \mathcal{O}(i,j) + 360) \; mod \; 360, \tag{4.11}$$

where D *represents a local neighborhood around the block* (i, j) *(the default size of* D *is* 5×5*);* n *is the number of blocks within* D*;* $\mathcal{O}(i', j')$ *and* $\mathcal{O}(i, j)$ *are local ridge orientations for blocks* (i', j') *and* (i, j)*, respectively.*

7. *If* $\mathcal{C}(i, j)$ *is above a certain threshold* T_c*, then the local orientations in this block are re-estimated at a lower resolution level until* $\mathcal{C}(i, j)$ *is below a certain level.*

With this algorithm, a fairly smooth orientation field estimate can be obtained. Figure 4.4 shows the orientation field of a fingerprint image estimated with our algorithm.



(a) Method proposed in [118] (b) Iterated method

Figure 4.4: Comparison of orientation fields estimated by the method proposed in [118] and our method; $w \times w = 16 \times 16$ and $w_\Phi \times w_\Phi = 5 \times 5$.

After the orientation field of an input fingerprint image is estimated, a region of interest localization algorithm which is based on the local *certainty level* of the

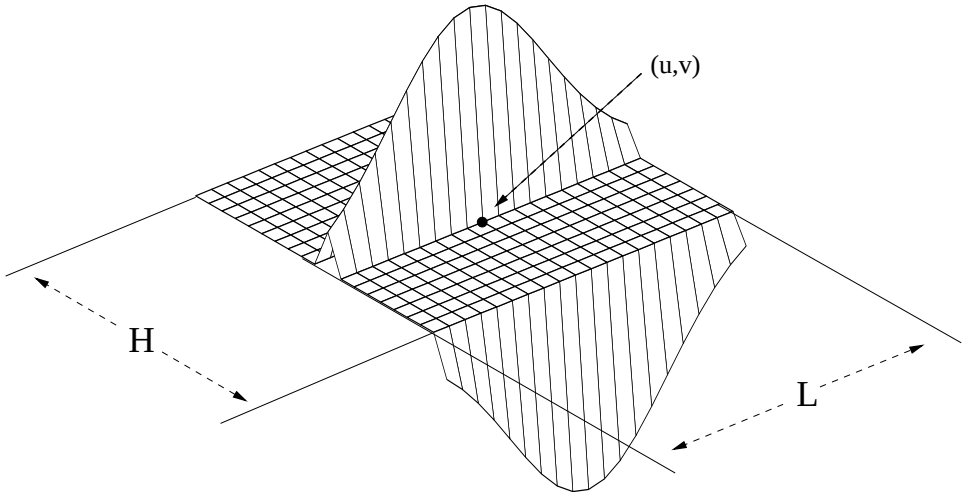


Figure 4.5: Ridge filter, $h_t(i,j;u,v)$.

orientation field is used to locate the region of interest within the input image. The certainty level of the orientation field in block (i,j) is defined as follows:

$$\mathcal{E}(i,j) = \sqrt{\frac{1}{w \times w} \frac{(\mathcal{V}_x(i,j)^2 + \mathcal{V}_y(i,j)^2)}{\mathcal{V}_e(i,j)}}, \tag{4.12}$$

$$\mathcal{V}_e(i,j) = \sum_{u=i-\frac{w}{2}}^{i+\frac{w}{2}} \sum_{v=j-\frac{w}{2}}^{j+\frac{w}{2}} (\partial_x^2(u,v) + \partial_y^2(u,v)). \tag{4.13}$$

For each block, if its certainty level of the orientation field is below a threshold T_l, then all the pixels in this block are marked as a *background pixel*. The main reason that we use $\mathcal{E}(i,j)$ to locate the region of interest within the input image is that (i) $\mathcal{E}(i,j)$ is actually a by-product of the estimated local ridge orientation, so it is efficient to compute $\mathcal{E}(i,j)$; (ii) $\mathcal{E}(i,j)$ performs reasonably well in detecting the region of interest. In our region of interest localization algorithm, we assume that only one fingerprint is present in the image.

4.2.3 Ridge Detection

An important property of the ridges is that the gray-level values on ridges attain their local maxima along a direction that is orthogonal to the local ridge orientation and the gray-level values of furrows attain their local minima along the same direction. Locally, ridges and furrows run parallel to one another forming a two-dimensional sine wave. Therefore, pixels can be identified to be ridge pixels in a local neighborhood based on this property. In our minutiae detection algorithm, a fingerprint image is first convolved with two masks, $h_t(i,j;u,v)$ and $h_b(i,j;u,v)$, of size $L \times H$ (on an average 11×7), respectively. The two masks, $h_t(i,j;u,v)$ and $h_b(i,j;u,v)$, are essentially the same except that one is rotated by 180^o with respect to the other (see Figure 4.5):

$$h_t(i,j;u,v) = \begin{cases} -\frac{1}{\sqrt{2\pi}\delta}e^{-\frac{u^2}{\delta^2}}, & \text{if } u = (v\cot(\mathcal{O}(i,j)) - \frac{H}{2\cos(\mathcal{O}(i,j))}), v \in \Omega \\ \frac{1}{\sqrt{2\pi}\delta}e^{-\frac{u^2}{\delta^2}}, & \text{if } u = (v\cot(\mathcal{O}(i,j))), v \in \Omega \\ 0, & \text{otherwise,} \end{cases} \tag{4.14}$$

$$h_b(i,j;u,v) = \begin{cases} -\frac{1}{\sqrt{2\pi}\delta}e^{-\frac{u^2}{\delta^2}}, & \text{if } u = (v\cot(\mathcal{O}(i,j)) + \frac{H}{2\cos(\mathcal{O}(i,j))}), v \in \Omega \\ \frac{1}{\sqrt{2\pi}\delta}e^{-\frac{u^2}{\delta^2}}, & \text{if } u = (v\cot(\mathcal{O}(i,j))), v \in \Omega \\ 0, & \text{otherwise,} \end{cases} \tag{4.15}$$

$$\Omega = \left[-\left| \frac{L\sin(\mathcal{O}(i,j))}{2} \right|, \left| \frac{L\sin(\mathcal{O}(i,j))}{2} \right| \right], \tag{4.16}$$

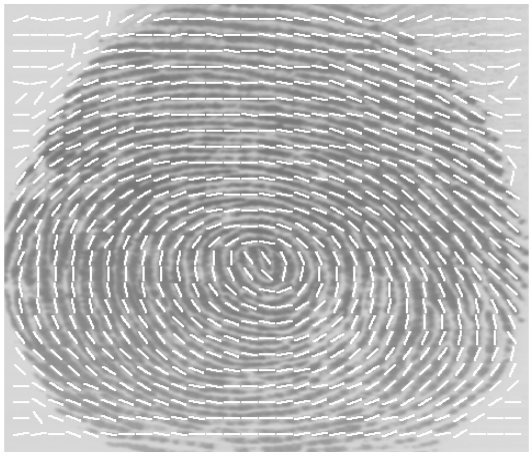
where $\mathcal{O}(i,j)$ represents the local ridge direction at pixel (i,j). These two masks are capable of accentuating the local maximum gray-level values along a direction that is orthogonal to the local ridge orientation. They can also adaptively smooth

the fingerprint images along the local ridge orientation and thus enhance the ridges. The smoothing effect depends on the value of δ. The larger the value of δ, the more robust are the filters to noise but more sensitive they are to highly curved ridges. In order to speedup the algorithm, the value of δ is set to a very large number such that each of the filter is actually degenerated into a filter with all the coefficients being equal. Pixel (i, j) is labeled as a ridge pixel ($\mathcal{R}(i, j) = 1$) if *both* the gray level values at pixel (i, j) of the convolved images are larger than a certain threshold T_{ridge}. By adapting the mask width to the width of the local ridge, this algorithm can efficiently locate the ridges. However, due to the presence of noise, breaks, smudges, *etc.* in the input fingerprint image, the resulting binary ridge map often contains holes and speckles. Therefore, a hole and speckle removal algorithm needs to be applied before ridge thinning. Our implementation of the hole and speckle removal algorithm uses a connected component algorithm to compute the number of pixels within each ridge and each hole and removes those connected components with number of pixels being less than a threshold, $T_{component}$ (the default value is 50). After the small speckles and holes are removed, a thinning algorithm generates the thinned ridges with each ridge being 8-connected and single pixel in width.

4.2.4 Minutiae Detection

Minutiae detection is a trivial task when an ideal thinned ridge map is available. If $(\mathcal{TR}(i, j) = 1)$ (a ridge pixel) and $(\sum_{u=-1}^{1} \sum_{v=-1}^{1} \mathcal{TR}(i + u, j + v) = 2)$, then pixel (i, j) is a ridge ending. If $(\mathcal{TR}(i, j) = 1)$ and $((\sum_{u=-1}^{1} \sum_{v=-1}^{1} \mathcal{TR}(i + u, j + v) > 3)$,

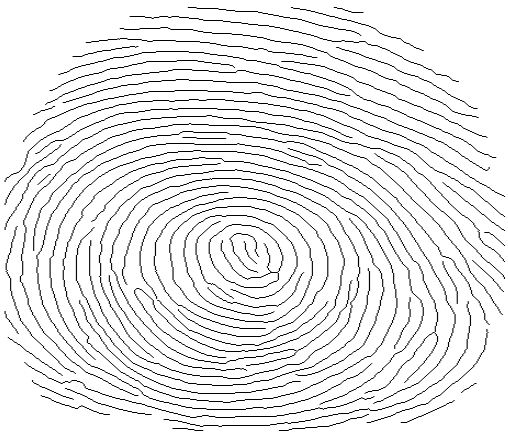
(a) input image



(b) orientation field

(c) fingerprint region

(d) ridge map

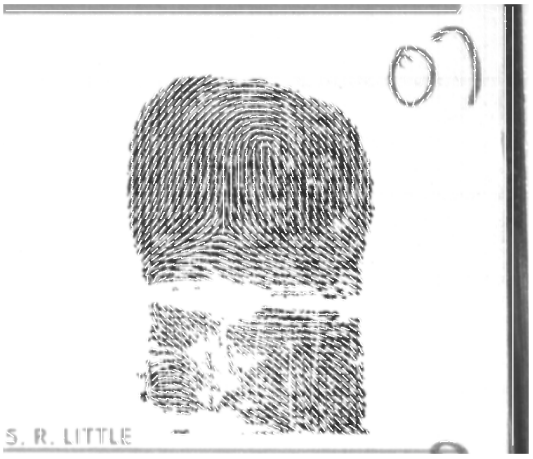


(e) thinned ridge map

(f) extracted minutiae

Figure 4.6: Results of our minutiae extraction algorithm on a live-scan fingerprint image (512×512); (a) input image; (b) orientation field superimposed on the input image; (c) fingerprint region; (d) extracted ridges (e) thinned ridge map; (f) extracted minutiae and their orientations superimposed on the input image.

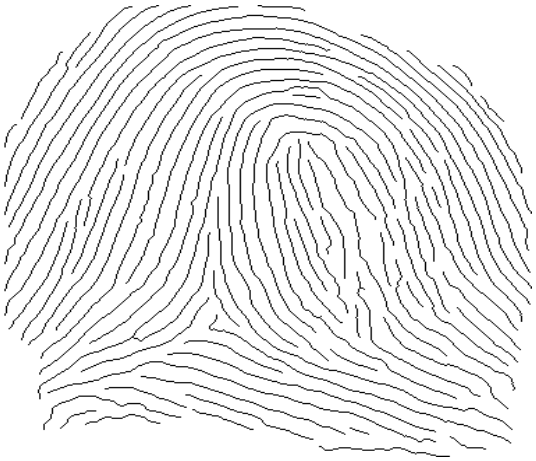
(a) input image



(b) orientation field

(c) fingerprint region

(d) ridge map



(e) thinned ridge map

(f) extracted minutiae

Figure 4.7: Results of our minutiae extraction algorithm on a rolled image from NIST 9 database (832 × 768); (a) input image; (b) orientation field superimposed on the input image; (c) fingerprint region; (d) extracted ridges (e) thinned ridge map; (f) extracted minutiae and their orientations superimposed on the input image.

then pixel (i, j) is a ridge bifurcation. However, the presence of undesired spikes and breaks in a thinned ridge map may lead to many spurious minutiae being detected. Therefore, before the minutiae detection, a smoothing procedure is applied to remove spikes and to join broken ridges. Our ridge smoothing algorithm uses the following heuristics (Figure 4.8):

- If the angle formed by a branch and the trunk ridge is larger than T_{lower} $(= 70°)$ and less than T_{upper} $(= 110°)$ and the length of the branch is less than T_{branch} $(= 20$ pixels), then the branch is removed.

- If a break in a ridge is shorter than T_{break} $(= 15$ pixels) and no other ridges pass through it, then the break is connected.

The parameters controlling the behavior of ridge smoothing heuristic are presently set to large values to ensure that all the genuine minutiae are detected. Although, it is possible that the ridge smoothing algorithm may occasionally annihilate genuine minutiae, by and large, it deletes the spurious minutiae generated due to poor quality of images, artifacts introduced during image processing, and fingerprint creases.

For each detected minutiae, the following parameters are recorded: (i) x-coordinate, (ii) y-coordinate, (iii) orientation which is defined as the local ridge orientation of the associated ridge, and (iv) the associated ridge segment. The recorded ridges are represented as one-dimensional discrete signals which are normalized by a preset length parameter which is approximately equal to the average inter-ridge distance (presently computed manually once for the given imaging setup). About 10 locations on each ridge are sampled per minutiae. The entire representation for a fin-

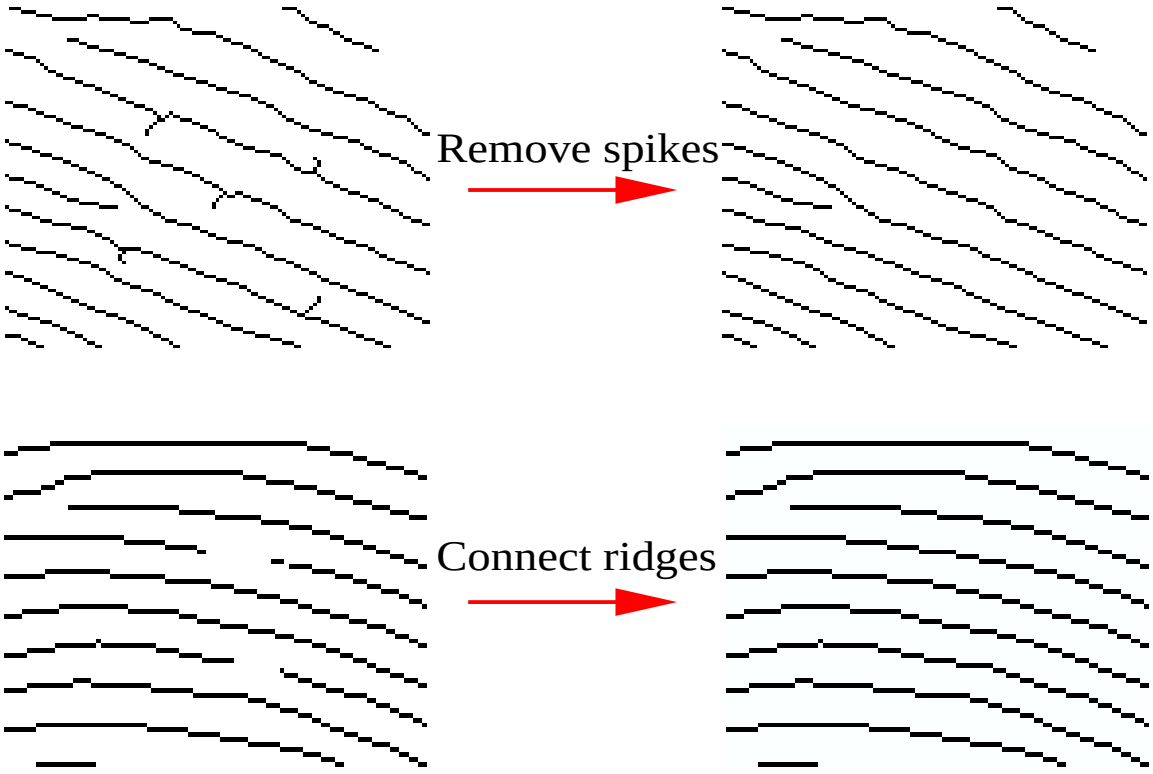


Figure 4.8: Examples of postprocessing heuristics.

ger when stored in a compressed format takes, on an average, about 800 bytes. These recorded ridges are used for alignment in the minutiae matching stage. Figure 4.6 shows the results of our minutiae extraction algorithm on a live-scan fingerprint image captured with an FTIR optical fingerprint scanner and Figure 4.7 shows the results of our minutiae extraction algorithm on a rolled fingerprint image from the NIST 9 fingerprint database.

4.3 Summary

Minutiae extraction finds representative features, called minutiae, from the input fingerprint images. A minutiae extraction algorithm should be reliable as well as

computationally efficient. A poor fingerprint image can be either rejected or enhanced prior to the minutiae extraction. A good minutiae extraction algorithm should be able to tolerate, to a limited extent, the corrupted ridge structures and degrade gracefully with the image quality. We have developed a minutiae extraction algorithm which is both fast and reliable in minutiae extraction. The new orientation field estimation algorithm results in a smoother orientation field which greatly improves the performance of the minutiae extraction. The adaptive ridge finder is capable of tolerating, to a certain extent, low ridge contrast and various sources of noise in fingerprint images such as short breaks and small smudges. The postprocessing step further refines the extracted minutiae.

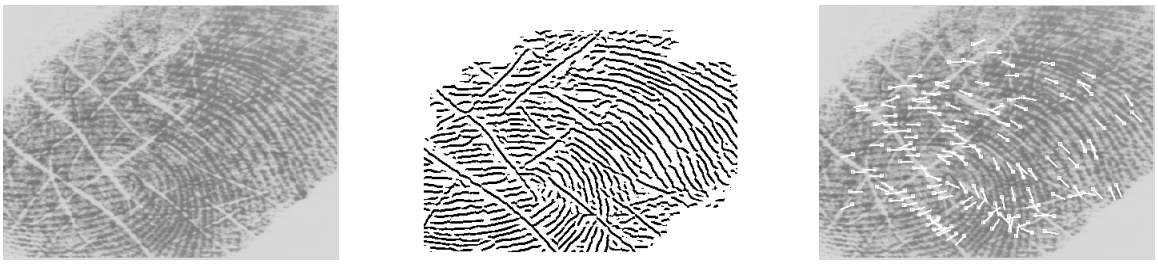
Chapter 5

Fingerprint Enhancement

The performance of currently available minutiae extraction algorithms depends heavily on the quality of input images. In an ideal fingerprint image, ridges can be easily detected and minutiae can be precisely located from the thinned ridges. However, in practice, due to the factors mentioned early, a significant percentage of acquired fingerprint images (approximately 10%) is of poor quality. The ridge structures in poor-quality fingerprint images are not always well-defined and hence they can not be correctly detected. This leads to the following problems: (*i*) a significant number of spurious minutiae may be created, (*ii*) a large percentage of genuine minutiae may be ignored, and (*iii*) large errors in their localization (position and orientation) may be introduced. Figures 5.1 and 5.2 show typical examples of applying our minutiae extraction algorithm to live-scan fingerprint images of both good and poor quality. We can see that the performance of the minutiae extraction algorithm on the poor quality image is far from desirable; a significant number of spurious minutiae are created and a large percentage of genuine minutiae are ignored by the algorithm.

(a) (b) (c)

Figure 5.1: Results of applying a minutiae extraction algorithm to a fingerprint image of good quality; (a) input image; (b) extracted ridge map; (c) extracted minutiae superimposed on the input fingerprint image.

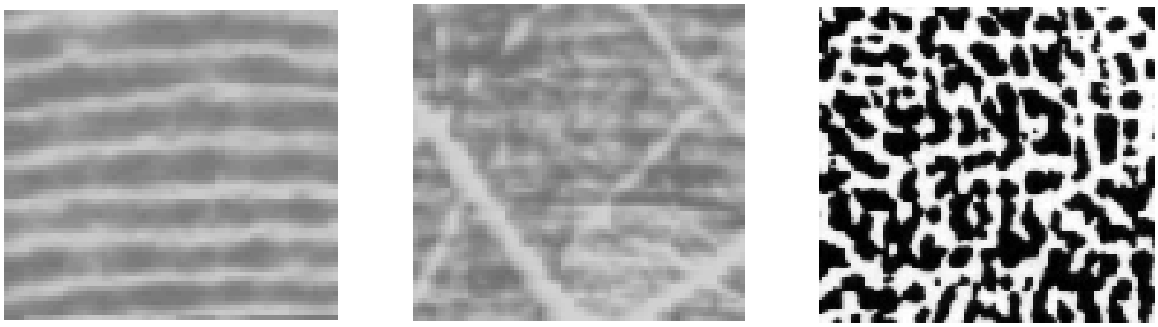


(a) (b) (c)

Figure 5.2: Results of applying a minutiae extraction algorithm to a fingerprint image of poor quality; (a) input image; (b) extracted ridge map; (c) extracted minutiae superimposed on the input fingerprint image.

In order to ensure that the performance of the minutiae extraction algorithm will be robust with respect to the quality of input fingerprint images, an enhancement algorithm which can improve the clarity of the ridge structures of input fingerprint images is, thus, necessary.

Ideally, the ridge structures in a fingerprint image are well-defined. Each ridge is separated by two parallel narrow furrows, each furrow is separated by two parallel narrow ridges; and minutiae are anomalies of ridges, *i.e.*, ridge endings and ridge bifurcations. When a fingerprint image is corrupted, such well-defined ridge structures are no longer visible. However, despite the existence of such noise, a fingerprint expert



(a) (b) (c)

Figure 5.3: Fingerprint regions; (a) well-defined region; (b) recoverable corrupted region; (c) unrecoverable corrupted region.

is often able to correctly identify the minutiae by using various visual clues such as local ridge orientation, ridge continuity, and ridge tendency. It is possible to develop an enhancement algorithm that can exploit these visual clues to improve the clarity of ridge structure in fingerprint images, which, in turn, will improve the performance of the minutiae extraction algorithm.

Generally, for a given fingerprint image, the region of interest can be divided into the following three categories (Figure 5.3):

- *Well-defined region*, where ridges and valleys are clearly differentiated from one another such that a minutiae extraction algorithm is able to operate reasonably.

- *Recoverable corrupted region*, where ridges and valleys are corrupted by a small amount of creases, smudges, *etc.* But, they are still visible and the neighboring regions provide sufficient information about the true ridge and valley structures.

- *Unrecoverable corrupted region*, where ridges and valleys are corrupted by such a severe amount of noise and distortion that no ridges and valleys are visible and the neighboring regions do not provide sufficient information about the true

ridge and valley structures either.

We refer to the first two categories of fingerprint regions as *recoverable* and the last category as *unrecoverable*. It is impossible to recover the original ridge structures in the unrecoverable regions, since no ridges and furrows are present at all within these regions. Any effort to improve the quality of the fingerprint image in these regions is futile. Therefore, the goal of a reasonable enhancement algorithm is to improve the clarity of ridge structures of fingerprint images in recoverable regions and to mask out the unrecoverable regions. In addition, since the objective of a fingerprint enhancement algorithm is to improve the clarity of ridge structures of input fingerprint images to facilitate the extraction of ridges and minutiae, a fingerprint enhancement algorithm should not result in any spurious ridge structures. This is very important, because spurious ridge structure may change the individuality of input fingerprints.

Fingerprint enhancement can be conducted on either (*i*) *binary images* or (*ii*) *gray-level images*. The parallel property of ridges provides a number of simple heuristics to differentiate the spurious ridge configurations from the true ridge configurations in binary images [67]. However, after applying a ridge extraction algorithm on the original gray-level images, information about the true ridge structures is often lost depending on the performance of the ridge extraction algorithm. Enhancement based on binary images has its inherent limitations.

A number of algorithms have been proposed to enhance grey level fingerprint images [14, 41, 67, 110, 76, 36, 96, 131, 132, 81]. Most of these techniques take advantage of the information about the local ridge structures and are capable of adaptively im-

proving the quality of input fingerprint images [41, 110, 76, 36, 96, 131, 81]. They usually assume that the local ridge orientation can be reliably estimated from input fingerprint images. In practice, this assumption is mainly valid for fingerprint images of good quality. For fingerprint image of poor quality, such an assumption is really not true, due to the existence of noise, creases, smudges, and holes; figure 5.4 shows some examples of estimated orientation field of fingerprint images of poor quality. Therefore, a good fingerprint enhancement algorithm should not assume that local ridge orientation can be easily obtained. Instead, it should focus a significant amount of effort on reliable computation of orientation field.

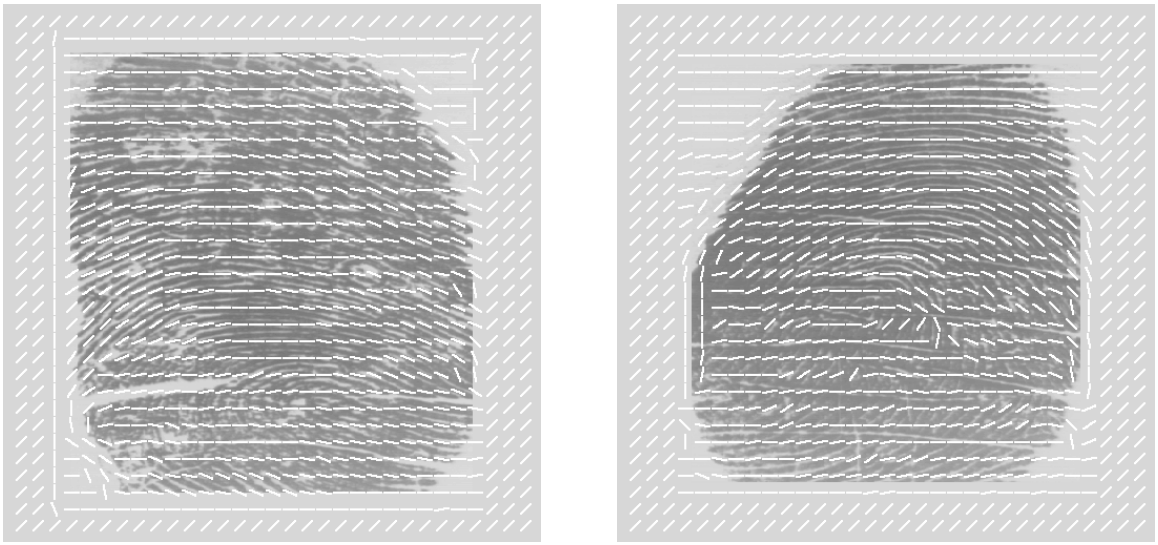


Figure 5.4: Estimated orientation fields of fingerprint images of poor quality.

We have developed a fingerprint enhancement algorithm, which improves the clarity of ridge structures in recoverable regions and make them suitable for minutiae extraction algorithms. Our algorithm also identifies all the corrupted regions in which it does not have the capability of recovering the true ridge structures and labels them as unrecoverable regions. The overview of the algorithm is shown in Figure 5.5. It

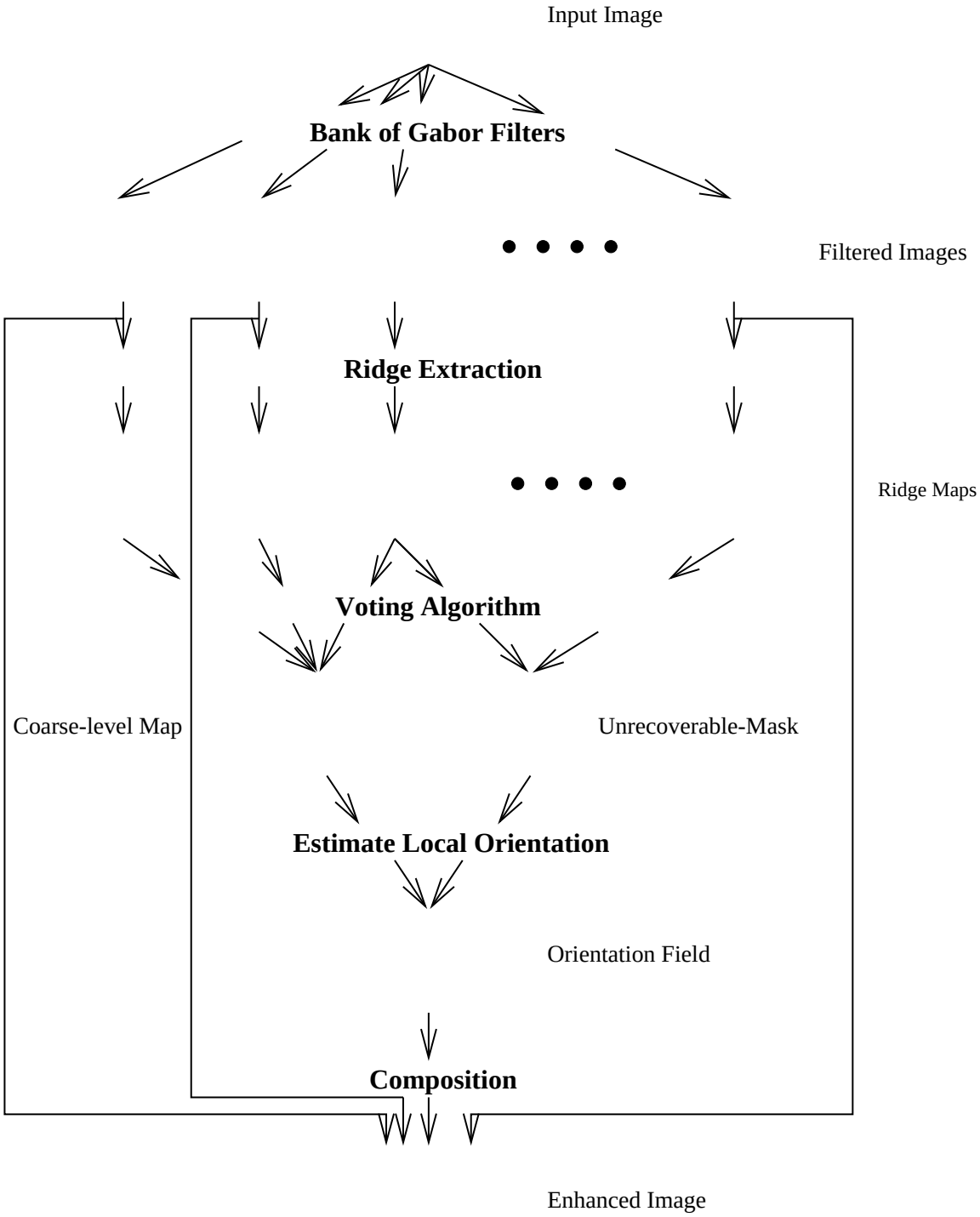


Figure 5.5: An overview of the fingerprint enhancement algorithm.

consists of two main stages: (*i*) orientation field estimation, and (*ii*) enhancement. Instead of estimating the orientation field directly from the input fingerprint image, we estimate it from the filtered images in which noise that is orthogonal to the dominant ridge orientation is greatly attenuated. Because our algorithm can obtain a reliable estimate of the orientation field, a better performance can thus be achieved in the enhancement stage. Its main steps are described as follows:

1. *A bank of even-symmetric Gabor filters is applied to an input fingerprint image and a set of filtered images is produced.*

2. *A ridge extraction algorithm is applied to each of the filtered images and the corresponding ridge map is obtained.*

3. *From the extracted ridge maps of filtered images, a voting algorithm is used to generate a coarse-level ridge map and unrecoverable-region mask. The generated coarse-level ridge map is used for orientation field estimation.*

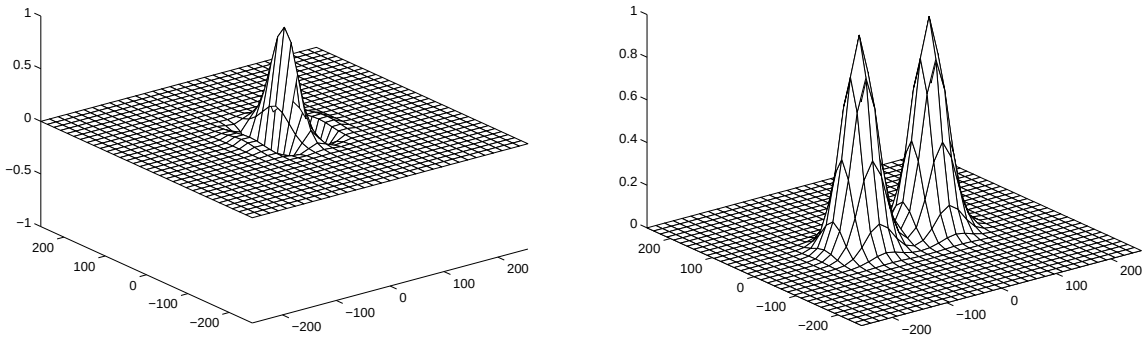
4. *An orientation estimation algorithm is applied to the generated coarse-level ridge map, and the local orientation at each pixel is obtained.*

5. *From the computed orientation field and filtered images, an enhanced image is obtained.*

5.1 Filtering of Fingerprint Image

In a small local neighborhood, the ridges and furrows in a fingerprint image approximately form a two-dimensional sine wave along the local ridge orientation. Thus, the ridges and valleys in a small local neighborhood have well-defined local frequency and local orientation properties. A set of bandpass filters can remove the undesired noise and preserve the true ridge structures [76, 110, 36, 131]. Gabor filters have both

frequency-selective and orientation-selective properties and have optimal joint resolution in both spatial and frequency domains [42, 72]. Therefore, it is beneficial to use Gabor filters as bandpass filters to remove the noise and preserve true ridge/furrow structures.



(a) (b)

Figure 5.6: An even-symmetric Gabor filter: (a) Gabor filter tuned to 60 cycles/width and 0^o orientation; (b) corresponding MTF.

The even-symmetric Gabor filter has the following general form [72]:

$$h(x,y) = exp\left\{-\frac{1}{2}\left[\frac{x^2}{\delta_x^2} + \frac{y^2}{\delta_y^2}\right]\right\}\cos(2\pi u_0 x), \tag{5.1}$$

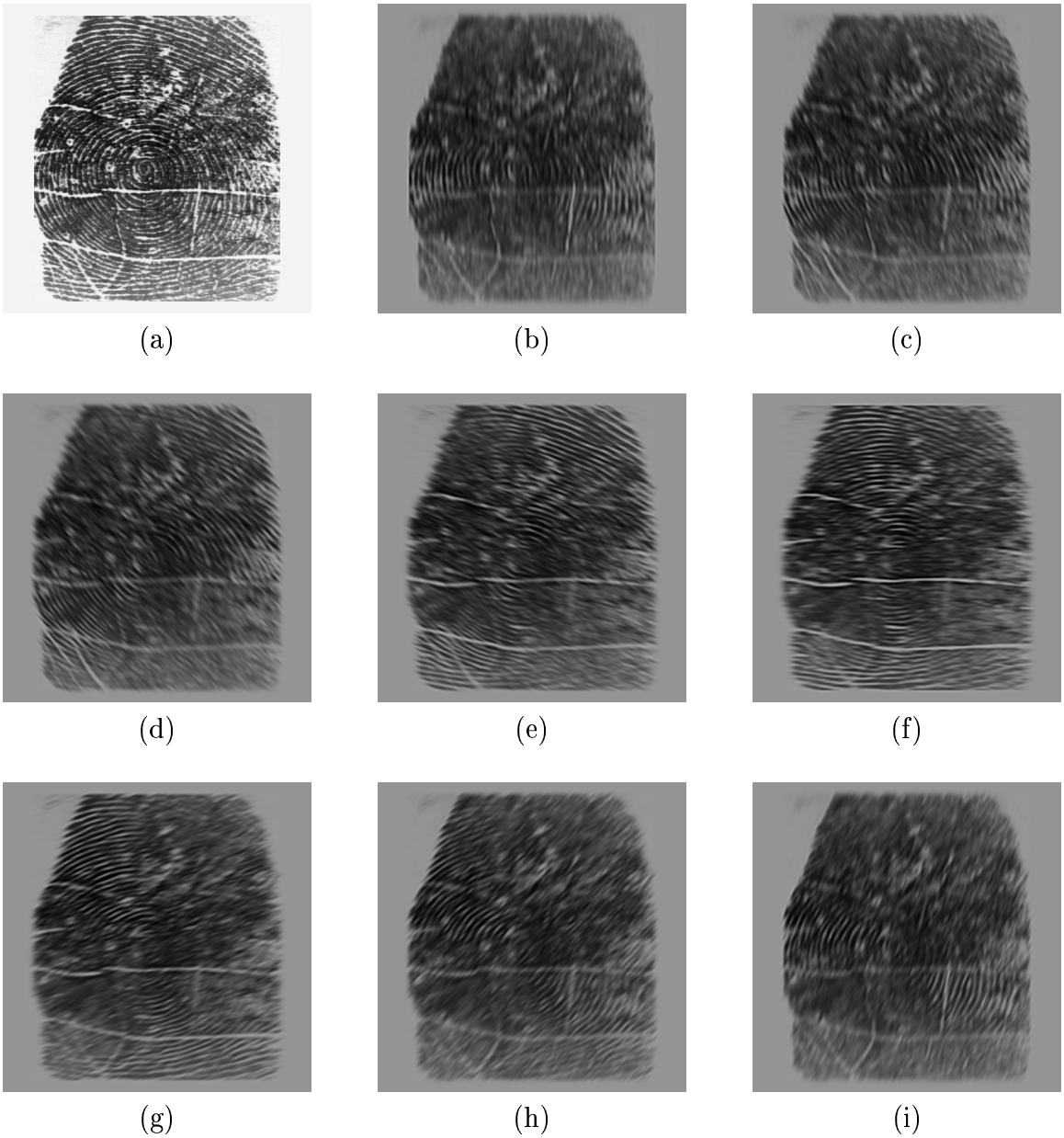
where u_0 is the frequency of a sinusoidal plane wave along the x-axis, and δ_x and δ_y are the space constants of the Gaussian envelope along x and y axes, respectively. Gabor filters with arbitrary orientation can be obtained via a rotation of the $x-y$ coordinate system. The modulation transfer function (MTF) of a Gabor filter can be represented as

$$H(u,v) = 2\pi\delta_x\delta_y\left(exp\left\{-\frac{1}{2}\left[\frac{(u-u_0)^2}{\delta_u^2} + \frac{v^2}{\delta_v^2}\right]\right\} + exp\left\{-\frac{1}{2}\left[\frac{(u-u_0)^2}{\delta_u^2} + \frac{v^2}{\delta_v^2}\right]\right\}\right) \tag{5.2}$$

where $\delta_u = 1/2\pi\delta_x$ and $\delta_v = 1/2\pi\delta_y$. Figure 5.6 shows an even-symmetric Gabor filter and its MTF.

An important issue in applying Gabor filters is the selection of filter parameters. We have observed that in a 500 dpi fingerprint image, the ridge frequency is generally around 60 cycles per image width (height). Therefore, in our fingerprint enhancement algorithm, the central frequency, u_0, is selected as 60 cycles/width (height). The radial bandwidth is selected as 2.5 octaves. Eight values of central orientation θ_0 are used: 0^o, 22.5^o, 45^o, 67.5^o, 90^o, $112,5^o$, 135^o, 157.5^o. The orientation bandwidth is selected as 35^o. For a given input fingerprint image, these 8 Gabor filters are applied to obtain 8 filtered images. To obtain a filtered image, a FFT is first performed on the input fingerprint image. Then the corresponding Gabor filters with tuned radial and orientation frequencies are applied to the frequency image and an inverse FFT is performed to obtain the filtered image. Figures 5.7(b)-(i) show the eight filtered images for the fingerprint image shown in Figure 5.7(a).

The filtered image corresponding to a given Gabor filter mainly preserves the ridges and valleys that are of the same direction as the filter direction. A channel selection algorithm is needed to combine the filtered images to generate an enhanced image. Ideally, a Bayesian evidence integration scheme which is based on the *difference* of the orthogonal channel contribution can be used to select channel(s) corresponding to each block. However, in order to ensure that such an evidence integration scheme is robust to noise, evidence should be collected from a relatively large local neighborhood. Computationally, this approach is very expensive. Therefore, in our algorithm, a simplifying scheme, which is based on the binary ridge maps of the fil-



(a) (b) (c)

(d) (e) (f)

(g) (h) (i)

Figure 5.7: Examples of filtered images for a 512×512 fingerprint image: (a) input image; (b-i) filtered images with Gabor filters tuned to 60 cycles/width and orientations of 0^o, 22.5^o, 45^o, 67.5^o, 90^o, 112.5^o, 135^o, 157.5^o, respectively.

tered images, is used. Although the simplifying scheme is not as efficient as a Bayesian evidence integration scheme, it is adequate in selecting the correct channels and is computationally inexpensive.

5.2 Ridge Extraction

For each filtered image, the ridge extraction algorithm which is described in chapter 4 is applied and the corresponding ridge maps are extracted from the filtered images. These ridge maps are not used for minutiae extraction. Instead, they are used to generate a coarse-level ridge map of the input fingerprint image.

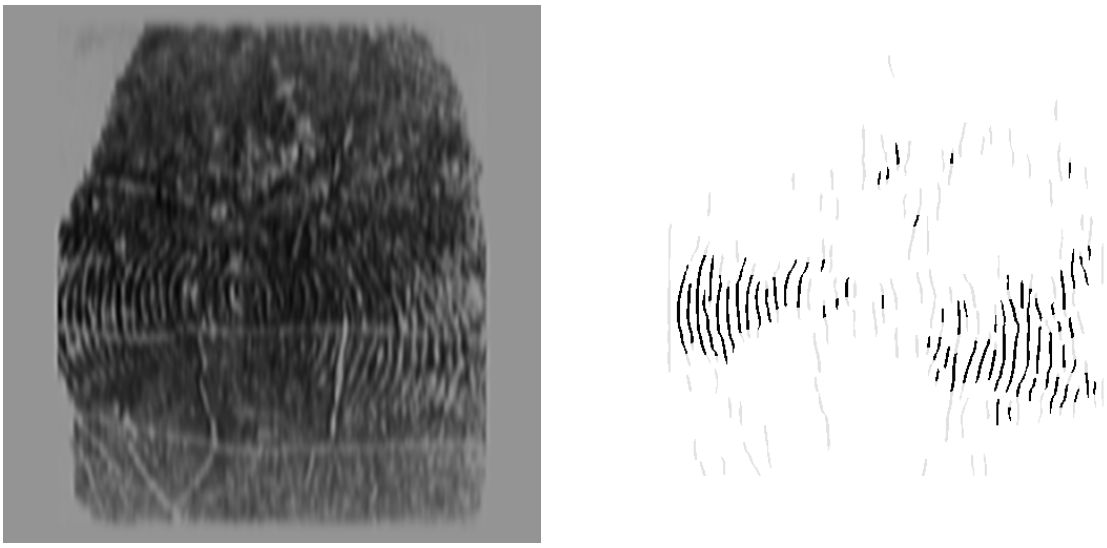
Due to the presence of noise, creases, smudges, *etc.* in the input fingerprint image, the resulting ridge maps of the filtered images often contain a large number of non-ridge pixels being labeled as ridge pixels. A postprocessing step is needed to remove these non-ridge pixels. In our fingerprint enhancement algorithm, we use the following heuristics:

- Compute the area of each connected component appearing in the ridge map. If the area is less than a threshold T_{min} (the default value is 200), then label this connected component as background; otherwise break the connected component into a set of short line segments and go to the next step.

- For each short line segment, if it is between a pair of narrow parallel ridges, then label it as a true ridge; otherwise label it as background.

The motivation behind these two heuristics is based on the fact that a genuine ridge

should (i) be sufficiently long and (ii) is located between a pair of ridges. The algorithm based on these two heuristics can remove the spurious ridges as well as the short genuine ridges and boundary ridges. Figure 5.8 shows an example of how the above algorithm performs on a ridge map extracted from a filtered image.



(a) (b)

Figure 5.8: The extracted ridge map of the 0^o filtered image: (a) the 0^o filtered image; (b) the extracted ridge map from the 0^o filtered image; the dark lines represent the valid ridges; grey lines represent the spurious ridges removed by the postprocessing step.

5.3 Ridge Voting

After the ridge map of each filtered image is obtained, the next step in our fingerprint enhancement algorithm is to generate a coarse-level ridge map and a mask of unrecoverable regions of the input fingerprint image. The coarse-level ridge map consists of the ridges extracted from each filtered images that are consistent with one another. It is used to estimate a reliable orientation field. The only requirement for the gen-

erated coarse-level ridge map is that it should roughly reflect the orientation of the local ridge structures of the input fingerprint image. It is not necessary to impose a requirement that this coarse-level ridge map should be very precise in terms of local ridge structures, since the minutiae will not be extracted from the coarse-level ridge map.

Neighboring ridges in a fingerprint image are usually oriented in the same direction. A filtered image obtained by applying a Gabor filter tuned to a certain direction retains the ridges that are oriented approximately in the same direction as the tuned direction of the Gabor filter. Generally, a ridge is a genuine ridge only if it is in a continuous region of significant size and tends to run parallel to its neighboring ridges. We can use this property as a heuristic to differentiate the genuine ridges from the spurious ridges. In our enhancement algorithm, the coarse-level ridge map and unrecoverable region mask are generated from the ridge maps of filtered images by using the following ridge voting algorithm.

- *Divide each ridge map of filtered images into blocks of size* $W \times W$ *(*8×8 *in our algorithm).*
- *Label each block as foreground (with a value* 1*) if there are enough ridge pixels appearing around the block; otherwise label it as background (with a value* 0*). After this process, a binary block map in which a pixel value of 1 represents the existence of ridges and 0 as non-ridges is obtained for each ridge map of filtered images.*
- *Delete all the connected components (8-connected) in the binary block maps which have an area less than a threshold (16 in our algorithm).*
- *For each block, examine all the eight filtered images and compute the coarse-level ridge map according to the following rules (an intuitive meaning of these rules is shown in Figure 5.9):*

Rule 1. If only one of the eight binary block map at pixel (x, y) *has the value* 1 *and this pixel belongs to a connected component of size* K, $K > T_{block}$,

then the pixel values of the corresponding block in the coarse-level ridge map are duplicated from the associated ridge map. The pixel value of the corresponding recoverable region mask is set to the value 0 *to indicate that this block is recoverable.*

Rule 2. If more than one binary block map at pixel (x, y) *has the value* 1 *and the associated local ridge orientations are not orthogonal to one another, the pixel values of the corresponding block in the coarse-level ridge map are taken as the average values of the associated ridge maps. The pixel values of the corresponding recoverable region mask is set to the value* 0 *to indicate that this block is recoverable.*

Rule 3. If more than one binary block map at pixel (x, y) *has the value* 1*, the associated local ridge orientations may be orthogonal to one another, and only one pixel with the value* 1 *resides in a connected component of size larger than a certain threshold* T_{block}*, then the pixel values of the corresponding block in the coarse-level ridge map are duplicated from the ridge map associated with the largest connected component and the pixel value of the corresponding recoverable region mask is set to the value* 0 *to indicate that this block is recoverable.*

Rule 4. If the above conditions are not satisfied, then the block is assigned a label 1 *to indicate that it is unrecoverable.*

By applying this algorithm to the set of ridge maps of filtered images, a coarse-level ridge map and an unrecoverable region mask are generated. An example of ridge voting is shown in Figure 5.10.

5.4 Enhanced Image

The coarse-level ridge map generated from the ridge maps of the filtered images preserves the local orientation information of the ridge structures of the input fingerprint image. The orientation field of the input fingerprint image can now be estimated from the coarse-level ridge map by ignoring the unrecoverable regions. The orientation field estimated from the coarse-level ridge map is more reliable than the orientation field estimated directly from the original image, because the steps introduced in the previ-

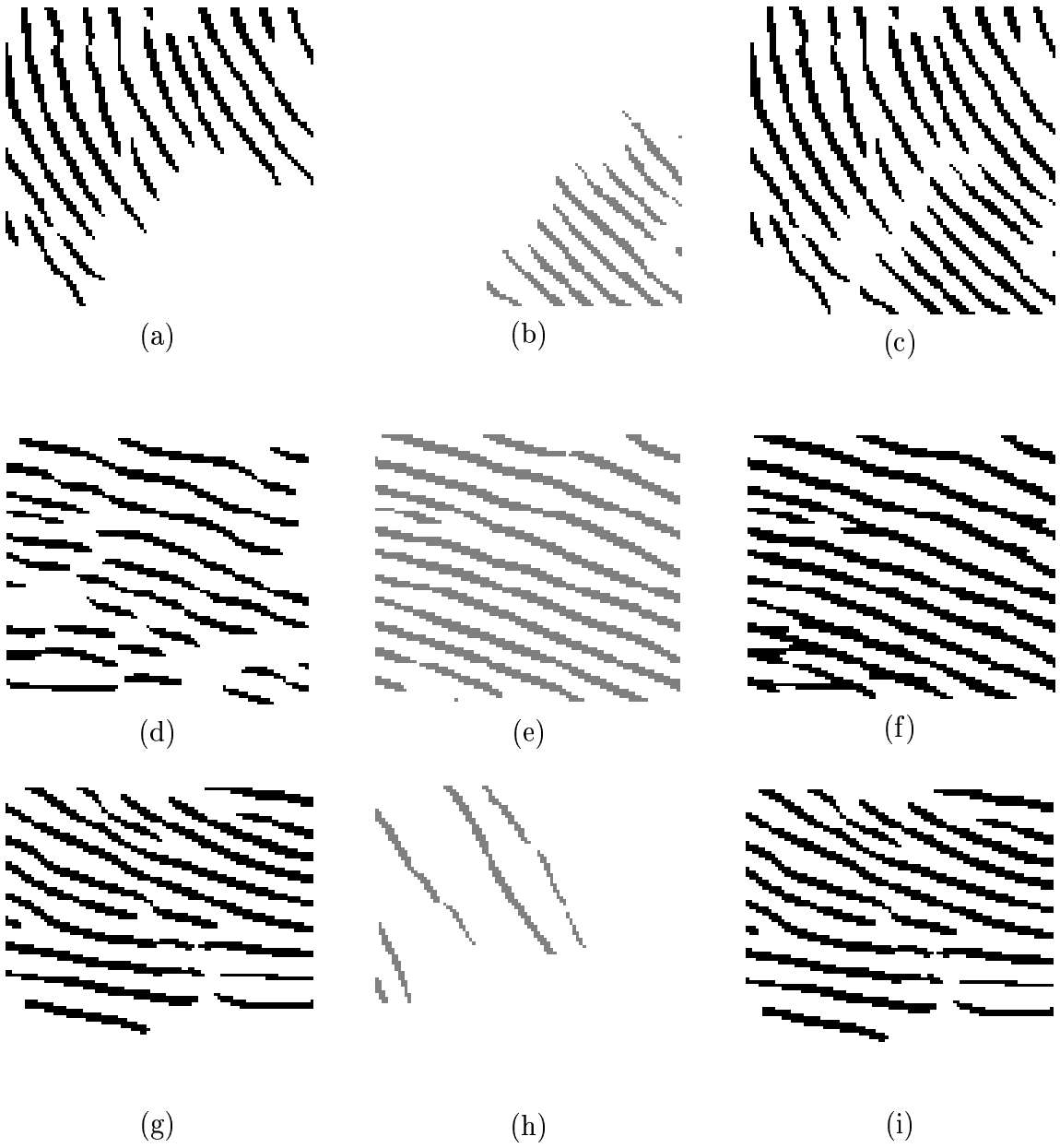


Figure 5.9: Intuitive meaning of the voting algorithm; here for simplicity, we assume that the input image is decomposed into two filtered images; (a)-(c) correspond to rule 1; (d)-(f) correspond to rule 2; (g)-(h) correspond to rule 3; the left two columns show the inputs to the voting algorithm while the third column shows the voting results.

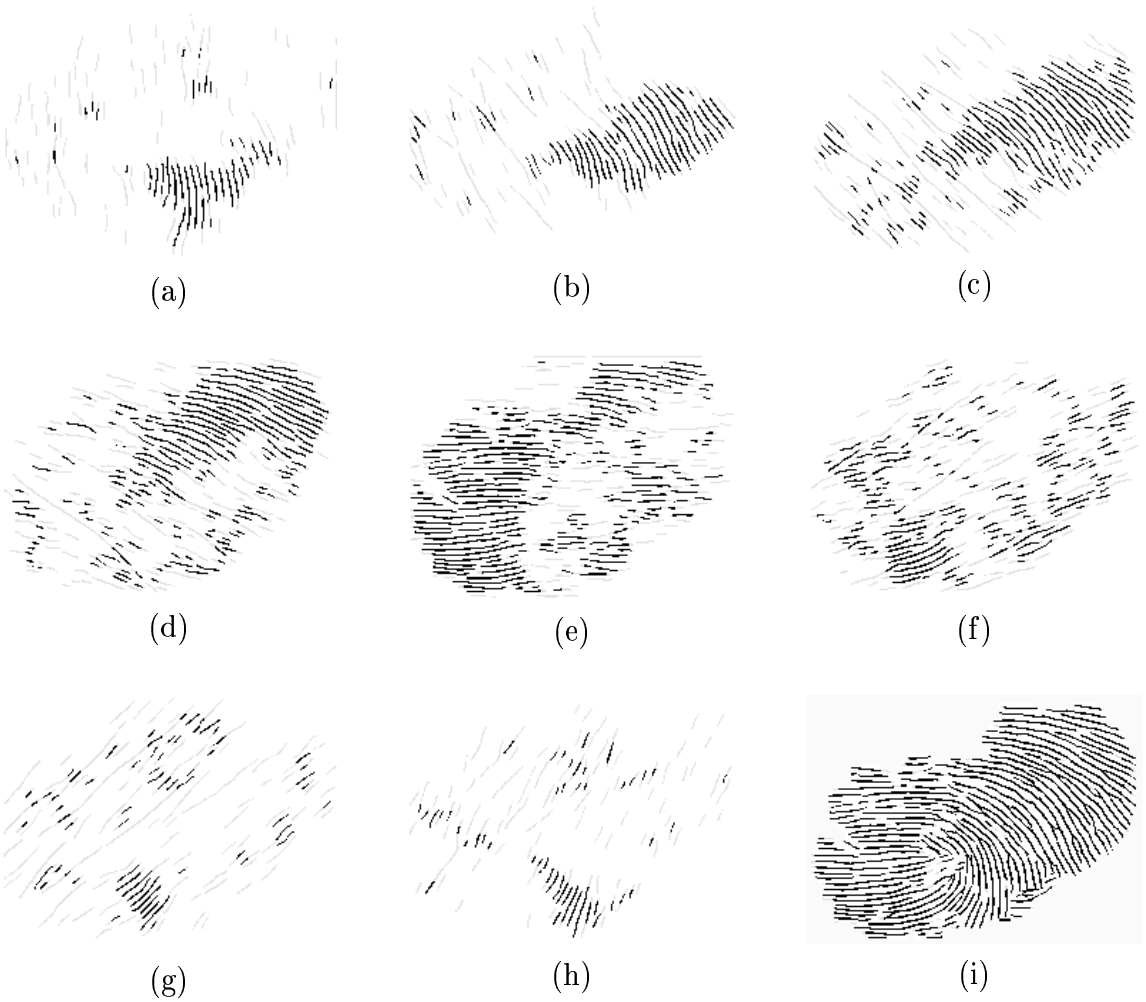


Figure 5.10: An example of ridge voting: (a-h) the ridge maps extracted from filtered images at 0^o, 22.5^o, 45^o, 67.5^o, 90^o, 112.5^o, 135^o, 157.5^o, respectively; (i) the voting result.

ous sections are able to suppress the harmful effect of noise, speckles, creases, holes, *etc.* The orientation estimation algorithm described in chapter 4 is used to compute the orientation field.

After the orientation field is obtained, the fingerprint image can then be adaptively enhanced by using the local orientation information. Let $f_i(x,y)$ (i=0, 1, 2, 3, 4, 5, 6, 7) denote the grey level value at pixel (x,y) of the filtered image corresponding to the orientation θ_i, $\theta_i = i * 22.5^o$. The grey level value at pixel (x,y) of the enhanced image can be interpolated according to the following formula:

$$f_{enh}(x,y) = a(x,y) f_{p(x,y)}(x,y) + (1 - a(x,y)) f_{q(x,y)}(x,y), \tag{5.3}$$

where

$$p(x,y) = \lfloor \frac{\theta(x,y)}{22.5} \rfloor, \tag{5.4}$$

$$q(x,y) = \lceil \frac{\theta(x,y)}{22.5} \rceil \bmod 8, \tag{5.5}$$

$$a(x,y) = \frac{\theta(x,y) - p(x,y)}{22.5}, \tag{5.6}$$

$$\tag{5.7}$$

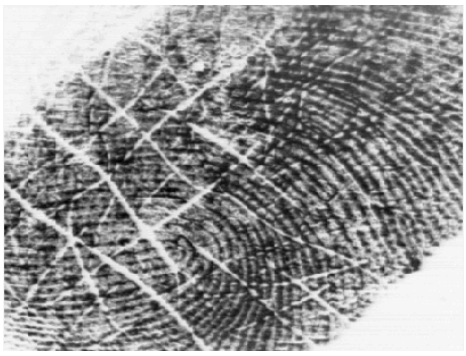
and $\theta(x,y)$ represents the value of the local orientation field at pixel (x,y). The main reason for interpolating the enhanced image directly from the limited number of filtered images is that the filtered images are already available and the above interpolation is computationally efficient. Obviously, the quality of the image obtained from such an interpolation scheme is not as good as the quality of the image ob-

tained by adaptively filtering the original image using the Gabor filters. However, it is sufficient for our minutiae extraction algorithm, which, in fact, has the capability of tolerating the unsmoothed effect of the enhanced images.

5.5 Summary

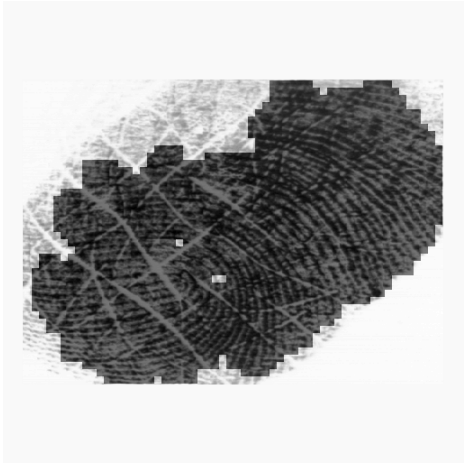
We have introduced our fingerprint enhancement algorithm. This algorithm, unlike other algorithms, concentrates a large amount of effort on a reliable estimation of the orientation field, which plays a critical role in the minutiae extraction algorithm. Our algorithm is capable of obtaining a relatively good estimate of orientation field even if the quality of the input fingerprint image is poor. Our algorithm also identifies the unrecoverable corrupted regions in the fingerprint and masks them out. This is a very important property because such unrecoverable regions do appear in some of the corrupted fingerprint images and they are extremely harmful to minutiae extraction. We note that our algorithm does not perform very well around singular regions where ridges and valleys have relatively high curvature values. It tends to mask these regions as unrecoverable regions. However, because minutiae around singular regions are usually assigned lower weights during matching, such a deficiency is not serious.

The major disadvantage of the current algorithm is that it is relatively slow. It takes approximately 13.8 seconds for our enhancement algorithm to process one 512×512 fingerprint image on a UltraSPARC 1 workstation. Obviously, this is too slow for an online application. Therefore, this algorithm is only used in the enrollment module. We have also proposed a fast enhancement algorithm which is able

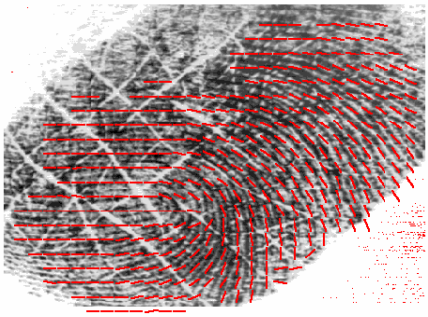


(a)

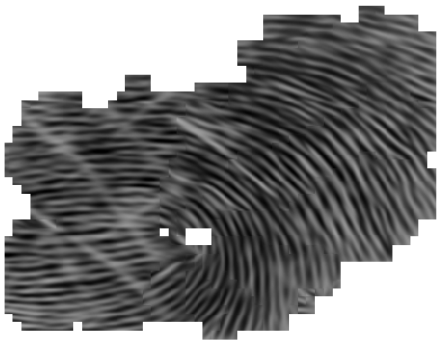
(b)



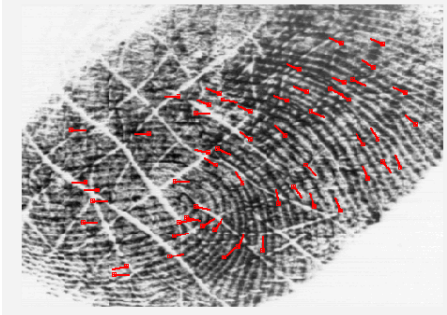
(c)



(d)



(e)



(f)

Figure 5.11: Results of applying the enhancement algorithm to a fingerprint image of poor quality: (a) input image; (b) coarse-level ridge map; (c) unrecoverable-region mask which consists of white pixels; (d) estimated orientation field; (e) enhanced image; (f) minutiae extracted from the enhanced image superimposed on the input image.

to adaptively enhance the ridge and valley structures using Gabor filters controlled by the local ridge orientation and local frequency information [64]. Experimental results have demonstrated that the fast algorithm can improve the performance of minutiae matching. It takes 2.3 seconds for the fast algorithm to enhance a 512×512 image on a UltraSPARC 1 workstation.

Chapter 6

Minutiae Matching

Given two (an input and a template) minutiae patterns, the minutiae matching algorithm determines whether they are from the impressions of the same finger.

6.1 Problem Specification

A minutiae matching problem is essentially a point pattern matching problem. The similarity of two minutiae patterns is determined by the total number (or normalized total number) of corresponding minutiae and the decision is made by comparing the value of the similarity with a pre-specified *threshold*. Formally, it can be stated as follows: Let $P = ((x_1^P, y_1^P, \theta_1^P), ..., (x_M^P, y_M^P, \theta_M^P))$ and $Q = ((x_1^Q, y_1^Q, \theta_1^Q), ..., (x_N^Q, y_N^Q, \theta_N^Q))$ denote the M minutiae in the template and the N minutiae in the input image, respectively. Find the number, M_{pair}, of corresponding pairs between P and Q and compare it against a threshold value $T_{minutiae}$.

In the ideal case, if (i) the correspondence between the template and input is

known, (*ii*) there are no deformations such as translation, rotation and deformations between them, and (*iii*) each minutiae present in a fingerprint image is exactly localized, then minutiae matching is only a trivial task of counting the number of spatially matching pairs between the two fingerprints and comparing it against a pre-specified threshold value.

In practice, determining whether two minutiae patterns extracted from two fingerprint impressions, possibly separated by a long duration of time, are indeed from the same finger, is an extremely difficult problem. The difficulty can be attributed to two primary reasons. First, even though the test and template minutiae patterns are indeed mated pairs, the correspondence between the test and template minutiae patterns is generally not known. Secondly, the imaging system presents a number of peculiar and challenging situations some of which are unique to fingerprint image capture scenario: (*i*) Inconsistent contact: The act of sensing distorts the finger. Based on the pressure and contact of the finger on the glass platen, the three-dimensional shape of the finger gets mapped onto the two-dimensional surface of the glass platen. Typically, this mapping function is uncontrolled and results in different fingerprint images across the impressions. (*ii*) Non-uniform contact: The ridge structure of a finger would be completely captured if ridges of the part of the finger being imaged are in complete optical contact with the glass platen. However, dryness of the skin, skin disease, sweat, dirt, humidity in the air all confound the situation, resulting in a non-ideal contact situation; some parts of the ridges may not come in complete contact with the platen and regions representing some furrows may come in contact with the glass platen. This results in "noisy" low contrast images, leading to either

spurious minutiae or missing minutiae. (*iii*) Irreproducible contact: Manual work, accidents, etc. inflict injuries to the finger, thereby, changing the ridge structure of the finger either permanently or semi-permanently. This may introduce additional spurious minutiae. (*iv*) Feature extraction artifacts: The feature extraction algorithm is imperfect and introduces measurement errors. Various image processing operations might introduce inconsistent biases to perturb the location and orientation estimates of the reported minutiae from their gray scale counterparts. (*v*) The act of sensing itself adds noise to the image. For example, residues are leftover from the previous fingerprint capture. A typical imaging system distorts the image of the object being sensed due to imperfect imaging conditions. In the FTIR sensing scheme, for example, there is a geometric distortion because the image plane is not parallel to the glass platen.

In the light of the operational environments mentioned above, the design of the minutiae matching algorithms needs to establish and characterize a realistic model of the variations among the representations of mated pairs. This model should include the properties of interest listed below:

1. The finger may be placed at different locations on the glass platen resulting in a (global) translation of the minutiae of the test representation from those in the template representation.

2. The finger may be placed in different orientations on the glass platen resulting in a (global) rotation of the minutiae of the test representation from those of the template representation.

3. The finger may exert a different (average) downward normal pressure on the glass platen resulting in a (global) spatial scaling of the minutiae of the test representation from those in the template representation.

4. The finger may exert a different (average) shear force on the glass platen resulting in a (global) shear transformation (characterized by a shear direction and magnitude) of the minutiae of the test representation from those in the template representation.

5. Spurious minutiae may be present in both the template as well as the test representations.

6. Genuine minutiae may be absent in the template or test representations.

7. Minutiae may be locally perturbed from their "true" location and the perturbation may be different for each individual minutiae. (The magnitude of such perturbations, however, is assumed to be small and within a fixed number of pixels.)

8. The individual perturbations among the corresponding minutiae could be relatively large (with respect to ridge spacings) but the perturbations among pairs of the minutiae are spatially linear.

9. The individual perturbations among the corresponding minutiae could be relatively large (with respect to ridge spacings) but the perturbations among pairs of the minutiae are spatially non-linear.

10. Only a (ridge) connectivity preserving transformation could characterize the relationship between the test and template representations.

A minutiae matcher may rely on one or more of these assumptions, resulting in a wide spectrum of behavior. At the one end of the spectrum, we have the "Euclidean" matchers which allow only rigid transformations among the test and template representations. At the other extreme, we have a "topological" matcher which may allow the most general transformations including, say, order reversals[1]. The choice of assumptions often represents matching performance trade-offs. Only a highly constrained system with not too demanding accuracies could get away with restrictive assumptions.

Figure 3.7 illustrates a typical situation of aligned ridge structures of mated pairs. Note that the best alignment in one part of the image may result in a large amount of displacements between the corresponding minutiae in the other regions. In addition, observe that the distortion is non-linear: given distortions at two arbitrary locations on the finger, it is not possible to predict the distortion at all the intervening points on the line joining the two points. In our opinion, a good minutiae matcher needs to accommodate not only global similarity transformations, but also shear transformations, linear and non-linear differential distortions. In our experience, *assumption 10* is too general a model to characterize the impressions of a finger and its inclusion into the matcher design may compromise efficiency and discriminatory power of the matcher. In addition, the minutiae matchers based on such an assumption need to use

[1]Order reversal means that the minutiae in the test representation are in totally different spatial order with respect to their correspondences in the template representation.

connectivity information which is notoriously difficult to extract from the fingerprint images of poor quality.

6.2 Literature Review

A large number of minutiae matchers (point pattern matchers) which are essentially "Euclidean" matchers have been proposed [47, 57, 4, 8, 11, 55, 25, 117, 128, 127, 138, 136, 141, 121, 147]. These matchers assume similarity transformation (*assumptions 1, 2, and 3*) and can tolerate, to a limited extent, both spurious minutiae as well as missing genuine minutiae (*assumptions 5 and 6*). Also, some of them can be modified to tolerate *assumption 7*, *i.e.* to be "elastic" in accommodating a small bounded local perturbation of minutiae. But they are not able to handle large displacements of the minutiae from their true locations. The relaxation approach [117] iteratively adjusts the confidence level of each corresponding pair based on its consistency with other pairs until a certain criterion is satisfied. Although a number of modified versions of this algorithm have been proposed to reduce the matching complexity [141], these algorithms are inherently slow because of their iterative nature and are unable to handle large distortions. The generalized Hough transform-based approach [11, 138, 25] converts point pattern matching to a problem of detecting peaks in the Hough space of transformation parameters. It discretizes the parameter space and accumulates evidence in the discretized space by deriving transformation parameters that relate two point patterns using a substructure or feature matching technique. A hierarchical Hough transform-based algorithm may be used to reduce the size of the accumulator

array by using a multi-resolution approach [25]. However, if there are only a few minutiae points available, it is very difficult to accumulate enough evidence in the Hough transform space for a reliable match. Again, it is difficult for this approach to handle large distortions. Tree-pruning approaches attempt to find the correspondence between a pair of point sets by searching over a tree of possible matches while employing different tree pruning methods such as branch-and-bound to reduce the search space [8]. To efficiently prune the tree of possible matches, this approach tends to impose a number of requirements on the input point sets such as an equal number of points and no outliers. These requirements are difficult to satisfy in practice, especially in a fingerprint identification/verification system. The energy minimization approach to point pattern matching establishes the correspondence between a pair of point sets by defining an energy function based on an initial set of possible correspondences and uses an appropriate optimization technique such as genetic algorithm, neural network, simulated annealing, *etc.* [4, 136, 147, 128, 127, 55] to find a possible suboptimal match. These methods tend to be very slow and are unsuitable for a real-time identification/verification system. They can tolerate only a very limited percentage of spurious and missing minutiae.

There also exist a number of graph-based matchers [134, 116, 69, 66, 62], which are essentially a "topological" type of matchers. They allow general transformations, positional errors, missing minutiae, and spurious minutiae. Since a general graph matching is a NP-complete problem, ridge features such as the position of the core points, ridge counts, inter-minutiae ridge counts and/or external alignment information are widely used to reduce the exponential search problem to a tractable problem.

The performance of these algorithms depends heavily on the availability of the ridge features and external alignment information. In a semi-automatic fingerprint identification system, these algorithms usually perform well, since the minutiae patterns can be aligned and the errors in minutiae and ridge features can be corrected interactively. However, a fully automatic fingerprint matching may not always be able to guarantee the availability of the correct ridge features and external alignment information.

6.3 Alignment-based Algorithm

We have developed an alignment-based matching algorithm, which is simple in theory, efficient in discrimination, and fast in speed. The alignment-based matching algorithm decomposes the minutiae matching into two stages: (i) *alignment stage* and (*ii*) *matching stage*. In the alignment stage, an alignment hypothesis, including translation and rotation between the input and the template is first generated and the input minutiae are aligned with the template minutiae according to the hypothesis. In the matching stage, the input minutiae and the template minutiae are first converted to a string representation in the polar coordinate system and an elastic string matching algorithm is used to evaluate the similarity between the two strings. The hypothesis that results in the largest similarity value is determined as the optimal alignment. The corresponding minutiae pairs are determined based on the optimal alignment. The main steps of our algorithm are as follows:

1. *For each pair of minutiae in P and Q, find the translation and rotation parameters between the ridge associated with input minutiae and the ridge associated with template minutiae and align the two minutiae patterns according to the estimated parameters.*

2. *Convert the template pattern and input pattern into the polar coordinate representations with respect to the corresponding minutiae on which alignment is achieved and represent them as two symbolic strings by concatenating each minutiae in an increasing order of radial angles:*

$$P_p = ((r_1^P, e_1^P, \theta_1^P), ..., (r_M^P, e_M^P, \theta_M^P)) \tag{6.1}$$
$$Q_p = ((r_1^Q, e_1^Q, \theta_1^Q), ..., (r_N^Q, e_N^Q, \theta_N^Q)), \tag{6.2}$$

where r_*, e_*, and θ_* represent the corresponding radius, radial angle, and normalized minutiae orientation with respect to the reference minutiae, respectively.

3. *Match the resulting strings P_p and Q_p with a modified dynamic-programming algorithm described below to find the 'edit distance' between P_p and Q_p.*

4. *Find the minimum edit distance between P_p and Q_p. Use the minimum edit distance to establish the correspondence of the minutiae between P_p and Q_p and compute the total number of corresponding minutiae, M_{PQ}. The matching score, S, is then computed according to*

$$S = \frac{100 M_{PQ} M_{PQ}}{MN}. \tag{6.3}$$

6.4 Alignment Hypothesis

Ideally, two sets of planar point patterns can be aligned completely by only two corresponding point pairs. A true alignment between two point patterns can be obtained by testing all possible corresponding point pairs and selecting the optimal one. However, due to the presence of noise and deformations, the input minutiae cannot always be aligned exactly with respect to those of the templates. In order to accurately recover pose transformations between two point patterns, a relatively large number of corresponding point pairs need to be used. This leads to a prohibitively large number of possible correspondences to be tested. Therefore, an alignment by corresponding point pairs is not practical even though it is feasible.

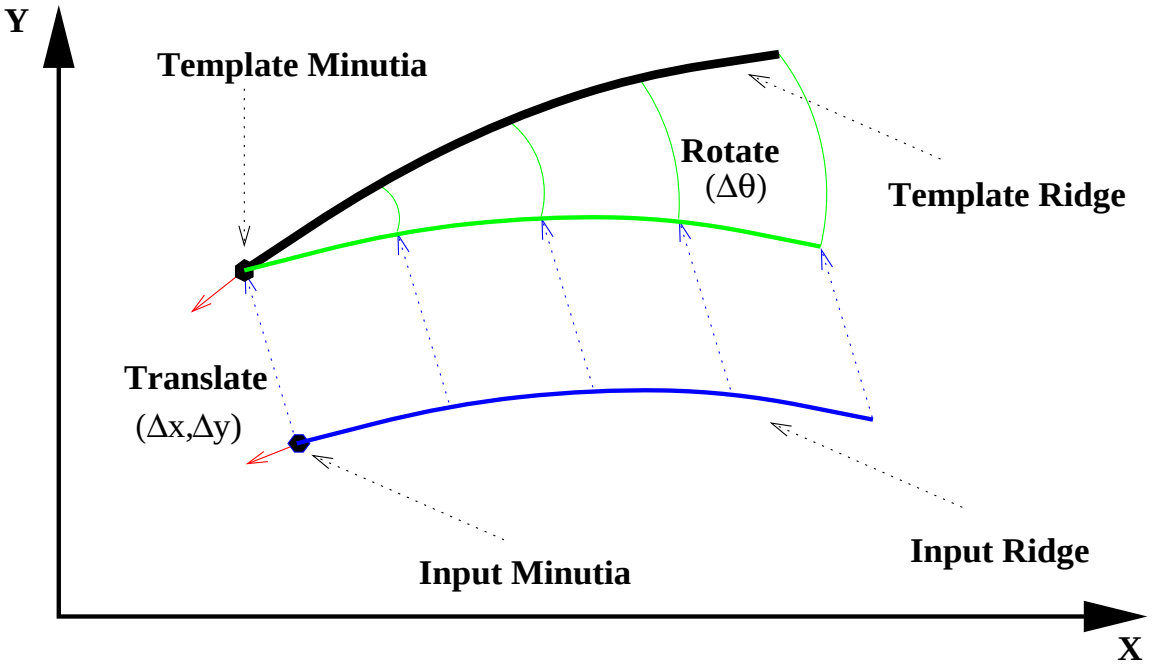


Figure 6.1: Alignment of the input ridge and the template ridge.

It is well known that corresponding curve segments are capable of aligning two point patterns with a high accuracy in the presence of noise and deformations [68]. Each minutiae in a fingerprint is associated with a ridge. Therefore, it is clear that a true alignment can be achieved by aligning corresponding ridges (see Figure 6.1). During the minutiae detection stage, when a minutiae is extracted and recorded, the ridge on which it resides is also recorded. This ridge is represented as a planar curve with its origin coincident with the minutiae and its x-coordinate being in the same direction as the direction of the minutiae. Also, this planar curve is normalized with the average inter-ridge distance. By matching these ridges, the relative pose transformation between the input fingerprint and the template can be accurately estimated. To be specific, let R^d and R^D denote the sets of ridges associated with

the minutiae in the input and the template, respectively. Our alignment algorithm is described as follows:

1. *For each ridge $d \in R^d$, represent it as an one-dimensional discrete signal and match it against each ridge, $D \in R^D$ according to the following formula:*

$$S = \frac{\sum_{i=0}^{L} d_i D_i}{\sqrt{\sum_{i=0}^{L} d_i^2 D_i^2}}, \tag{6.4}$$

where L is the minimal length of the two ridges and d_i and D_i represent the distances from point i on the ridges d and D to the x-axis, respectively. The sampling interval on a ridge is set to the average inter-ridge distance. If the matching score S ($0 \leq S \leq 1$) is larger than a certain threshold T_r (0.8), then go to step 2, otherwise continue to match the next pair of ridges.

2. *Estimate the transformation between the two ridges (Figure 6.1). Generally, a least-square method can be used to estimate the pose transformation. However, in our system, we observe that the following method is capable of achieving the same accuracy with fewer computations. The translation vector $(\Delta x, \Delta y)^T$ between the two corresponding ridges is computed as*

$$\begin{pmatrix} \Delta x \\ \Delta y \end{pmatrix} = \begin{pmatrix} x^d \\ y^d \end{pmatrix} - \begin{pmatrix} x^D \\ y^D \end{pmatrix}, \tag{6.5}$$

where $(x^d, y^d)^T$ and $(x^D, y^D)^T$ are the x and y coordinates of the two minutiae, which are called reference minutiae, associated with the ridges d and D, respectively. The rotation angle $\Delta\theta$ between the two ridges is computed as

$$\Delta\theta = \frac{1}{L}\sum_{i=0}^{L}(\gamma_i - \Gamma_i), \tag{6.6}$$

where L is the minimal length of the two ridges d and D; γ_i and Γ_i are radial angles of the ith *point on the ridge with respect to the reference minutiae associated with the two ridges d and D, respectively. The scaling factor between the input and template images is assumed to be 1.*

3. *Denote the minutiae $(x^d, y^d, \theta^d)^T$, based on which the transformation parameters are estimated, as the reference minutiae. Translate and rotate all the N input minutiae with respect to this reference minutiae, according to the following formula:*

$$\begin{pmatrix} x_i^A \\ y_i^A \\ \theta_i^A \end{pmatrix} = \begin{pmatrix} \Delta x \\ \Delta y \\ \Delta\theta \end{pmatrix} + \begin{pmatrix} \cos\Delta\theta & \sin\Delta\theta & 0 \\ \sin\Delta\theta & -\cos\Delta\theta & 0 \\ 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} x_i - x^d \\ y_i - y^d \\ \theta_i - \theta^d \end{pmatrix}, \tag{6.7}$$

where $(x_i, y_i, \theta_i)^T$, $(i = 1, 2, ..., N)$, *represents an input minutiae and* $(x_i^A, y_i^A, \theta_i^A)^T$ *represents the corresponding aligned minutiae.*

Note that, because the aspect ratio of the pixels in our acquisition devices is not one (non-square pixels), a rectification is performed before the alignment.

6.5 Alignment Hypothesis Evaluation

If two identical point patterns are exactly aligned with each other, then each pair of corresponding points are completely coincident. In such a case, a point pattern matching can be simply achieved by counting the number of overlapping pairs. However, in practice, such a situation is rarely encountered. On the one hand, the error in determining and localizing minutiae hinders the alignment algorithm to recover the relative pose transformation exactly, while on the other hand, our alignment scheme does not model the nonlinear deformation of fingerprints which is an inherent property of fingerprint impressions. With the existence of such a nonlinear deformation, it is impossible to exactly recover the position of each input minutiae with respect to its corresponding minutiae in the template. Therefore, the aligned point pattern matching algorithm needs to be *elastic* which means that it should be capable of tolerating, to some extent, the deformations due to inexact extraction of minutiae positions and nonlinear deformations. Usually, such an elastic matching can be achieved by placing a bounding box around each template minutiae, which specifies all the possible positions of the corresponding input minutiae with respect to the template minutiae, and restricting the corresponding minutiae in the input image to be within this box [121]. This method does not provide a satisfactory performance in practice, because local

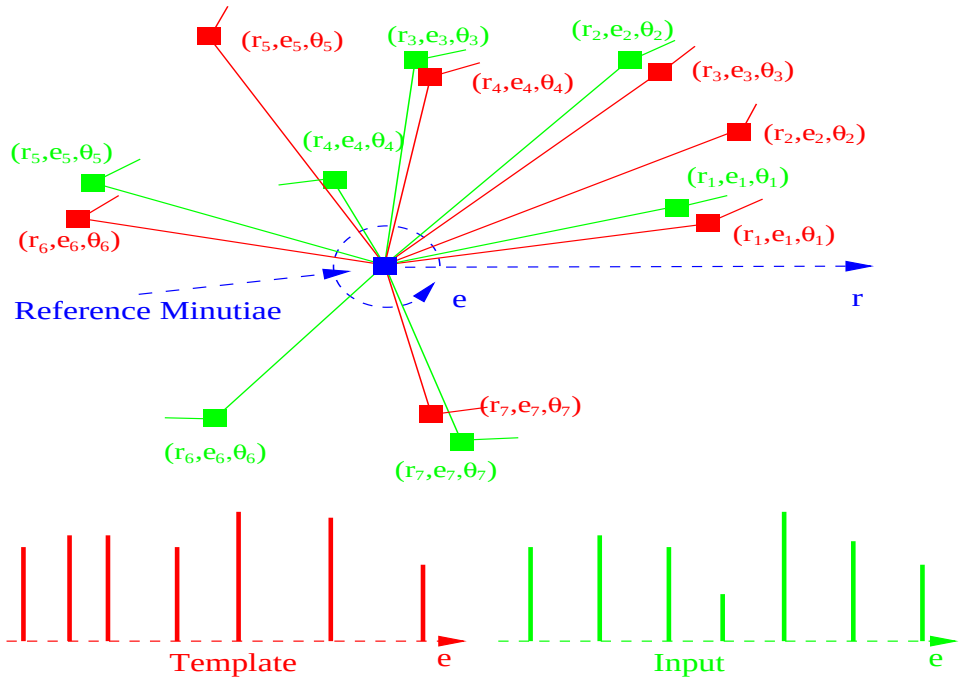


Figure 6.2: The string matching of a pair of point patterns.

deformations may be small while the accumulated global deformations can be quite large. We have proposed an adaptive elastic matching algorithm with the ability to compensate the minutiae localization errors and nonlinear deformations.

Our adaptive elastic matching algorithm consists of two main steps: (*i*) representing minutiae patterns as a *string* in the polar coordinate system and (*ii*) matching the strings with a dynamic programming algorithm to establish the correspondence. Minutiae matching in the polar coordinate system has several advantages. Although the deformation of fingerprints depends on a number of factors such as impression pressure and impression direction, the deformation in a local region is usually consistent and it may become less consistent as one moves further away from the region where the fingerprint patterns are consistent (see Figure 3.7). Consequently, it is easier to represent and manipulate the representations in polar coordinate space (with

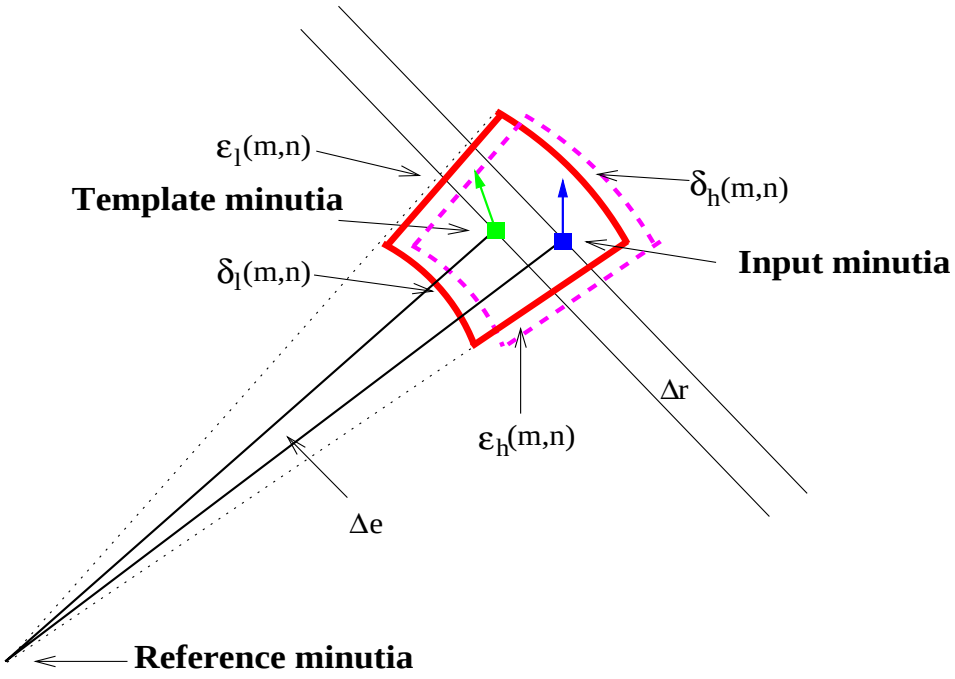


Figure 6.3: Bounding box and its adjustment.

origin at a point of maximal consistency between the reference and aligned test template). At the same time, it is easier to formulate rotation, which constitutes the main part of the alignment error between an input image and a template, in the polar space than in the Cartesian space. The symbolic string generated by concatenating points in an increasing order of radial angle in polar coordinates uniquely represents a point pattern. This reveals that point pattern matching can be achieved with a string matching algorithm.

A number of string matching algorithms have been reported in the literature [37]. Generally, string matching can be thought of as the maximization/minimization of a certain cost function such as the edit distance. Including an elastic term in the cost function of a string matching algorithm can achieve a certain amount of error tolerance. Given two strings P_p and Q_p of lengths M and N, respectively, we define

the "edit distance", $C(M,N)$ recursively as follows:

$$C(m,n) = \begin{cases} 0 & \text{if } m=0 \text{ and } n=0 \\ \min\left\{ \begin{array}{c} C(m-1,n)+\Omega \\ C(m,n-1)+\Omega \\ C(m-1,n-1)+w(m,n) \end{array} \right\} & 0<m\le M \text{ and } 0<n\le N, \end{cases} \tag{6.8}$$

where

$$w(m,n) = \begin{cases} \alpha\left|r_m^P - r_n^Q\right| + \beta\Delta e + \gamma\Delta\theta & \text{if } \left|r_m^P - r_n^Q\right|<\delta,\ \Delta e<\epsilon \text{ and } \Delta\theta<\varrho \\ \Omega & \text{otherwise,} \end{cases} \tag{6.9}$$

$$\Delta e = \begin{cases} a & \text{if } (a=(e_m^P - e_n^Q + 360) \bmod 360) < 180 \\ a-180 & \text{otherwise,} \end{cases} \tag{6.10}$$

$$\Delta\theta = \begin{cases} a & \text{if } (a=(\theta_m^P - \theta_n^Q + 360) \bmod 360) < 180 \\ a-180 & \text{otherwise,} \end{cases} \tag{6.11}$$

α, β, and γ are the weights associated with radius, radial angle, and minutiae direction, respectively; δ, ϵ and ϱ specify the bounding box; and Ω is a pre-specified penalty for a mismatch. Such an edit distance, to some extent, captures the elastic property of string matching. It represents the cost of changing one polygon to the other. The intuitive meaning of the string matching is depicted in Figure 6.2. However, this scheme can only tolerate, but not compensate for, the adverse effect on matching produced by the inexact localization of minutiae and nonlinear deformations. Therefore, an adaptive mechanism is needed. This adaptive mechanism should be able

to track the local nonlinear deformation and inexact alignment and try to alleviate them during the minimization process. However, we do not expect that this adaptive mechanism can handle the "order flip" of minutiae, which, to some extent, can be solved by an exhaustive re-ordering and matching within a local angular window.

In our matching algorithm, the adaptation is achieved by adjusting the bounding box (Figure 6.3) when an inexact match is found. It can be represented as follows:

$$w'(m,n) = \begin{cases} \alpha \left| r_m^P - r_n^Q \right| + \beta \Delta e + \gamma \Delta\theta & \text{if} \begin{cases} \delta_l(m,n) < (r_m^P - r_n^Q) < \delta_h(m,n) \\ \epsilon_l(m,n) < \Delta e < \epsilon_h(m,n) \\ \Delta\theta < \varrho \end{cases} \\ \Omega & \text{otherwise,} \end{cases} \tag{6.12}$$

where

$$\begin{pmatrix} \Delta r_a \\ \Delta e_a \end{pmatrix} = \begin{cases} \begin{pmatrix} r_m^P - r_n^Q \\ \Delta e \end{pmatrix} & \text{if} \begin{cases} \delta_l(m,n) < (r_m^P - r_n^Q) < \delta_h(m,n) \\ \epsilon_l(m,n) < \Delta e < \epsilon_h(m,n) \\ \Delta\theta < \varrho \end{cases} \\ 0 & \text{otherwise,} \end{cases} \tag{6.13}$$

$$\delta_l(m+1,n+1) = \delta_l(m,n) + \eta \Delta r_a, \tag{6.14}$$

$$\delta_h(m+1,n+1) = \delta_h(m,n) + \eta \Delta r_a, \tag{6.15}$$

$$\epsilon_l(m+1,n+1) = \epsilon_l(m,n) + \eta \Delta e_a, \tag{6.16}$$

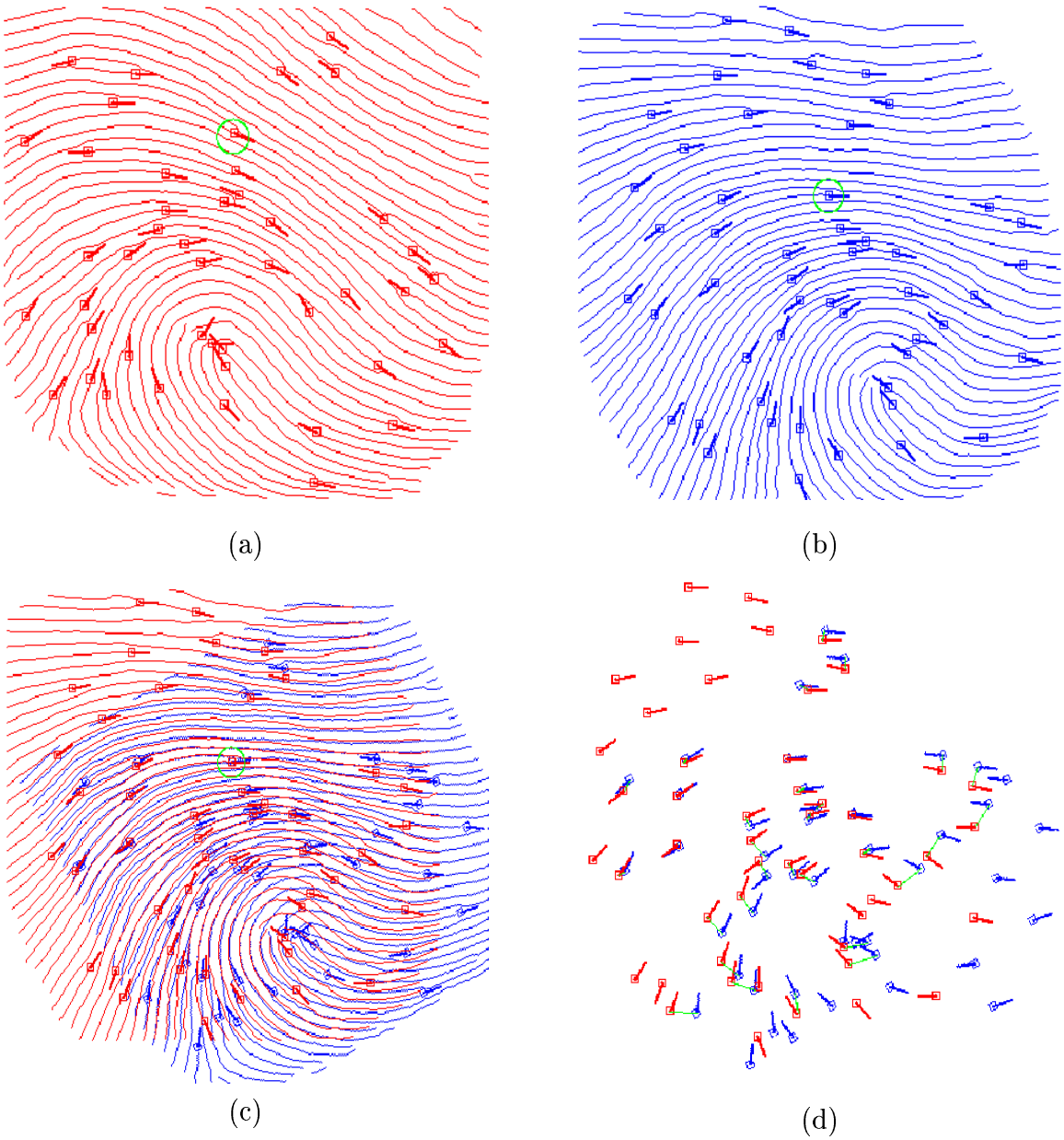
$$\epsilon_h(m+1,n+1) = \epsilon_h(m,n) + \eta \Delta e_a, \tag{6.17}$$

$w'(m,n)$ represents the penalty for matching a pair of minutiae $(r_m^P, e_m^P, \theta_m^P)^T$ and

$(r_n^Q, e_n^Q, \theta_n^Q)^T$, $\delta_l(m,n)$, $\delta_h(m,n)$, $\epsilon_l(m,n)$, and $\epsilon_h(m,n)$ specify the adaptive bounding box in the polar coordinate system (radius and radial angle), and η is the learning rate. This elastic string matching algorithm has a number of parameters which are critical to its performance. We have empirically determined the values of these parameters as follows: $\delta_l(0,0) = -8$; $\delta_h(0,0) = +8$; $\epsilon_l(0,0) = -7.5$; $\epsilon_h(0,0) = +7.5$; $\varrho = 30$; $\alpha = 1.0$; $\beta = 2.0$; $\gamma = 0.1$; $\Omega = 200(\alpha + \beta + \gamma)$; $\eta = 0.5$. The values of $\delta_l(0,0)$, $\delta_h(0,0)$, $\epsilon_l(0,0)$, and $\epsilon_h(0,0)$ depend on the resolution of fingerprint images. Figure 6.4 shows the results of applying the matching algorithm to an input and a template minutiae set pair.

6.6 Summary

Given two minutiae sets, minutiae matching determines whether they are extracted from the fingerprint impressions of the same finger. Minutiae matching is an extremely difficult problem for the following reasons: (*i*) the correspondence between the minutiae sets is not known, (*ii*) presence of relative translation, rotation, and impression deformations between the two minutiae sets, (*iii*) presence of a significant number of spurious minutiae and missing minutiae, and (*iv*) minutiae may not be precisely located. A good minutiae matching algorithm should be able to accommodate these transformations, missing minutiae, spurious minutiae, small position errors, shear transformation, and linear and non-linear differential distortions. We have proposed an alignment-based elastic matching algorithm. This algorithm is



(a) (b) (c) (d)

Figure 6.4: Results of applying the matching algorithm to an input minutiae set and a template; (a) input minutiae set; (b) template minutiae set; (c) alignment result based on the minutiae marked with green circles; (d) matching result where template minutiae and their correspondences are connected by green lines.

capable of finding the correspondences between minutiae without resorting to an exhaustive search. It achieves a good performance in minutiae matching because of its capability to adaptively compensate for the nonlinear deformations and inexact transformations between mated fingerprints.

Chapter 7

Decision Fusion

A biometric system can be based on either a (or one snapshot of a) single biometric characteristic or multiple biometric characteristics (or multiple snapshots of a single biometric characteristic) to make a personal identification. We define a biometric system which uses only a single biometric characteristic as a *unimodal biometric system* and a biometric system which uses multiple biometric characteristics as a *multimodal biometric system.*

7.1 Multimodal Biometrics

A unimodal biometric system is usually more cost-effective than a multimodal biometric system. However, it may not always be applicable in a given domain because of (*i*) unacceptable performance and (*ii*) inability to operate on a large user population. A multimodal biometric system can overcome, to a certain extent, these limitations. First of all, identification using multiple biometrics is essentially a sensor fusion

problem, which utilizes information from multiple sensors to increase fault-tolerance capability, to reduce uncertainty, to reduce noise, and to overcome incompleteness of individual sensors [34, 133]. A multimodal approach can increase the reliability of the decisions made by a biometric system [21, 46, 82, 16]. Although a necessary requirement for a biometric characteristic is that each individual possess it, it is not necessary that a particular biometric characteristic of a specific individual is suitable for an automatic system. By using multiple biometric characteristics, the system will be applicable on a larger target population. Finally, a multimodal biometric system is generally more robust to fraudulent technologies, because it is more difficult to forge multiple biometric characteristics than to forge a single biometric characteristic.

In designing a multimodal biometric system, a number of issues need to be considered: (*i*) what is the main purpose of utilizing multiple biometrics? (*ii*) what is the operational mode? (*iii*) which biometrics should be integrated? and (*iv*) how many biometrics are sufficient? Since the applicable population and system robustness depend mainly on the characteristics of the selected biometrics, the main problem in designing a multimodal biometric system is the integration of individual biometrics to improve the performance of personal identification. Typically, performance refers to (*i*) *accuracy* and (*ii*) *speed*. System accuracy indicates how reliable and confident a biometric system is in differentiating between a genuine individual and an impostor. System speed refers to the time taken by a biometric system in making a personal identification. By properly incorporating those biometrics that are relatively fast, the overall speed of a biometric system can be improved.

A biometric system can operate in either a *verification mode* or an *identification*

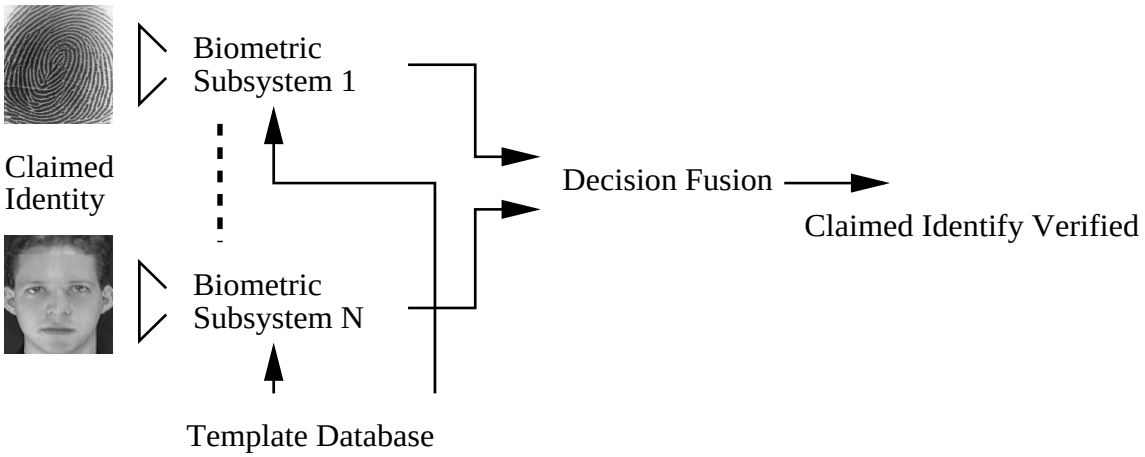


Figure 7.1: A generic multimodal verification system.

mode. The integration schemes for these two modes are very different. Since only a one-to-one comparison is performed in a verification system, multimodal biometrics cannot really improve the verification speed. Therefore, integration of multiple biometrics in a verification system is mainly intended to improve the accuracy of the system. The block diagram of a generic multimodal verification system is shown in Figure 7.1. In a typical identification system, a large number of matchings need to be performed to identify an individual. A biometrics that has a large discriminating power can improve the identification accuracy, while a biometrics that is computationally efficient can improve the identification speed. The block diagram of a generic multimodal identification system is shown in Figure 7.2.

Which biometrics and how many of them should be integrated depend very much on the application domain. It is difficult to establish a systematic procedure to determine which biometrics should be used. Intuitively, the larger the number of integrated biometrics, the higher the system accuracy, but more expensive the system. In this thesis, we mainly concentrate on improving the system performance by integrating

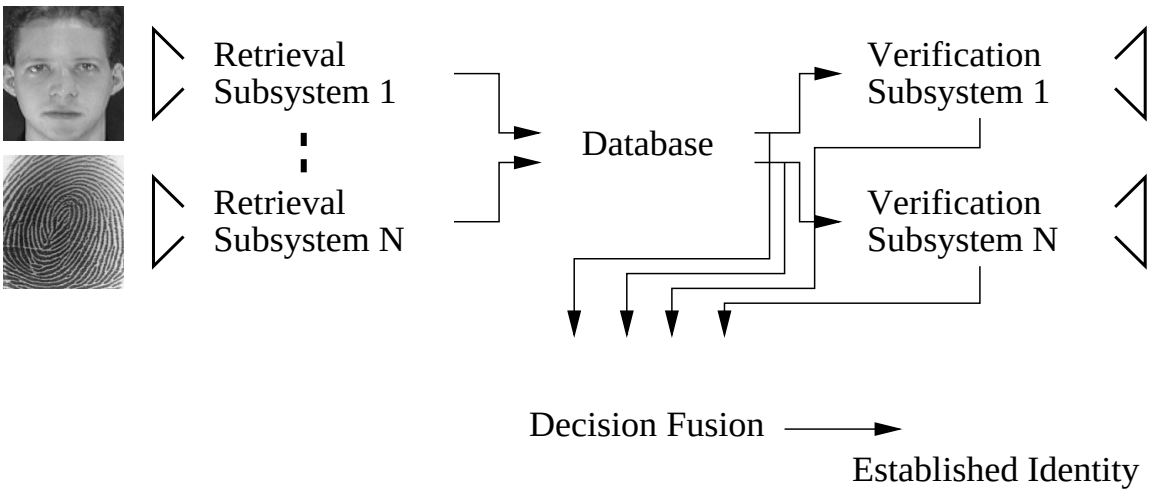


Figure 7.2: A generic multimodal identification system.

two specific biometrics, namely face and fingerprint.

7.1.1 Multimodal Biometrics for Verification

Integration of multiple biometrics for a verification system may be performed in the following scenario: (*i*) integration of multiple snapshots of a single biometrics, for example, several fingerprint images of the same finger in fingerprint verification (Figure 7.3) and (*ii*) integration of a number of different biometrics (Figure 7.4). In this sense, multimodal biometrics is a conventional decision fusion problem - to combine evidence provided by each biometrics to improve the overall decision accuracy. Generally, multiple evidences may be integrated at one of the following three different levels [20]: (*i*) Abstract level; the output from each module is only a set of possible labels without any confidence value associated with the labels; in this case, the simple majority rule may be employed to reach a more reliable decision [156], (*ii*) Rank level; the output from each module is a set of possible labels ranked by decreasing

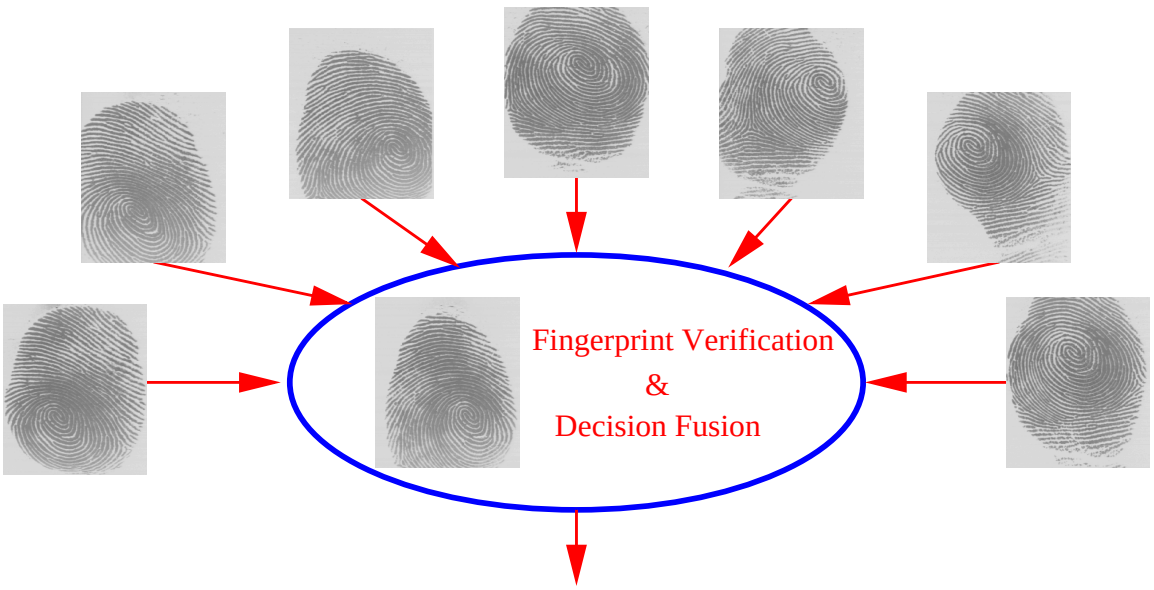


Figure 7.3: Integration of multiple snapshots of a single biometric characteristic.

confidence values, but the confidence values themselves are not specified; (*iii*) Measurement level; the output from each module is a set of possible labels with associated confidence values; in this case, more accurate decisions can be made by integrating different confidence values.

Dieckmann *et al*. [46] have proposed an abstract level fusion scheme: "2-from-3 approach" which integrates face, lip motion, and voice based on the principle that a human uses multiple clues to identify a person. This approach uses a simple voting algorithm to find whether the decision made by each individual classifier is consistent with the other two classifiers. Brunelli and Falavian [21] have proposed two schemes to combine evidence from speaker verification and face recognition. The first scheme is a measurement level scheme in which the outputs of two different speech classifiers and the outputs of three different face classifiers are normalized and combined using geometric average. The second scheme is a hybrid rank/measurement level scheme

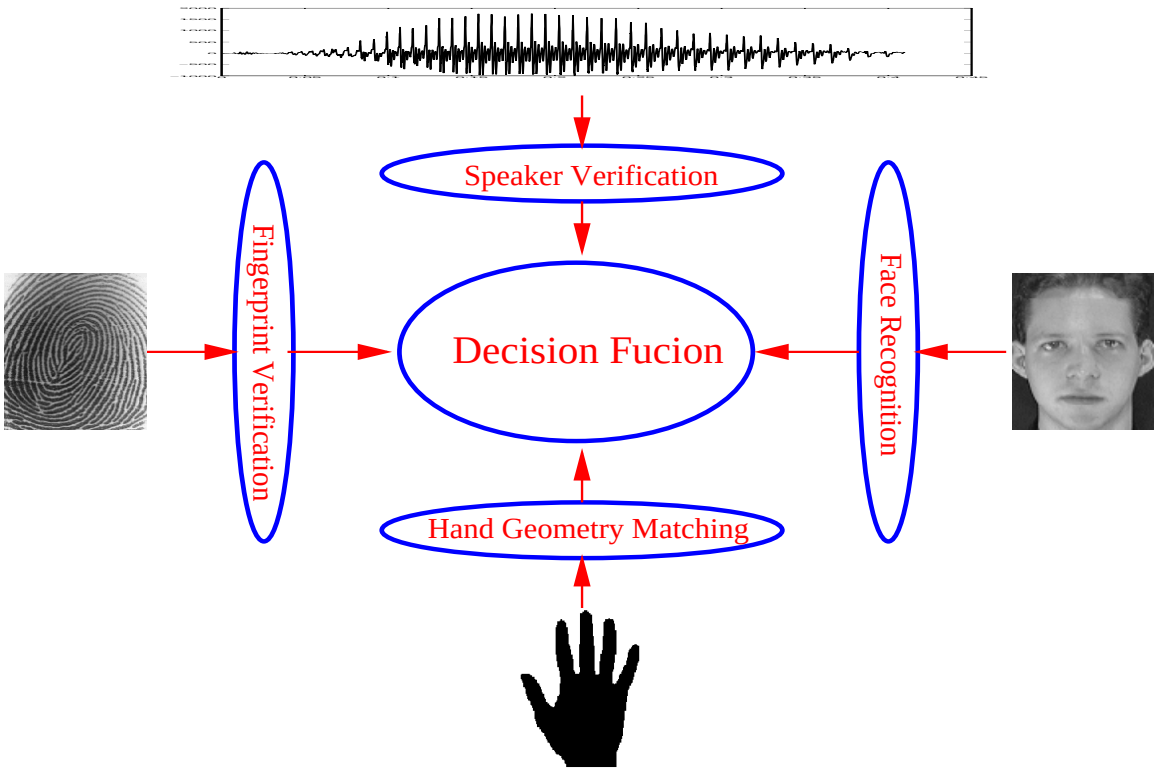


Figure 7.4: Integration of different biometric characteristics.

which uses HyperBF network to combine the outputs of these five classifiers. The authors have demonstrated that the system accuracy can be improved by using these fusion schemes. Kittler *et al.* [82] have demonstrated the efficiency of an integration strategy which fuses multiple snapshots of a single biometrics using a Bayesian framework. In this scheme, the *a posteriori* class probabilities for each individual are estimated and the decision is made based on the average or maximum or median of the *a posteriori* class probabilities for a given set of snapshots. Bigun *et al.* [16] have proposed a Bayesian integration scheme to combine different evidences based on the assumption that the evidences are independent of one another. Their scheme results in an improved recognition accuracy by combining voice and face as well as voice and lip motion. Maes *et al.* [89] have proposed to combine biometric data (*e.g.*, voice)

with non-biometric data (*e.g.*, password).

7.1.2 Multimodal Biometrics for Identification

All the decision fusion schemes mentioned above can be used to improve the identification accuracy in a multimodal identification system. However, since an identification system needs to perform one-to-many comparisons to find a match, the average computational complexity for each comparison should be as low as possible to enable a reasonable response time, especially for a large database. However, the integration schemes mentioned above increase the computational complexity for each comparison. Therefore, it is not practical to directly apply these schemes to an identification system; an integration scheme that is able to improve both the speed and the accuracy should be used. We introduce a multimodal biometrics scheme which integrates two biometrics (in particular, face and fingerprint) which complement each other in terms of identification speed and identification accuracy: a biometric approach (*e.g.*, face recognition) that is suitable for database retrieval is used to index the template database and a biometric approach (*e.g.*, fingerprint verification) that is reliable in deterring impostors is used to ensure the overall system accuracy. In addition, since each biometric approach provides a certain confidence about the identity being established, a decision fusion scheme which exploits all the information at the output of each module can be used to make a more reliable decision. We have designed a integrated biometric system which uses this decision fusion scheme to integrate face recognition and fingerprint verification in making a personal identification [63]. The

system essentially consists of a face recognition subsystem and a fingerprint verification subsystem, which are integrated by a decision fusion module. It operates in three stages: (*i*) the face recognition subsystem retrieves the top n matches of a query face from the template database, (*ii*) the fingerprint verification subsystem verifies the top n possible matches and provides the corresponding fingerprint matching scores, and finally (*iii*) the decision fusion module integrates the results from the face recognition and the results from the fingerprint verification to establish the final decision.

7.2 Face Recognition

In the context of personal identification, face recognition refers to static, controlled full frontal portrait recognition [31]. By static we mean that the facial portraits used by the face recognition system are still facial images (intensity or range). By controlled we mean that the type of background, illumination, resolution of the acquisition devices and the distance between the acquisition devices and faces, *etc.* are essentially fixed during the image acquisition process. Obviously, in such a controlled situation, the segmentation task is relatively simple and the intra-class variations are small.

Generally, there are two major tasks in face recognition: (*i*) locating faces in input images and (*ii*) recognizing the located faces. Face location itself continues to be a challenging problem for uncontrolled and cluttered images [31]. Fortunately, in the context of personal identification, the background is controlled or almost controlled, so face location is generally not considered to be a difficult problem for a face-based biometric system. Face recognition from a general view point remains an open prob-

lem because transformations such as position, orientation, and scale and changes in illumination produce substantially large intra-class variations [114]. Again, in the context of personal identification, the variations in acquired face images can be restricted to a certain limit, which enables the current techniques to achieve a desirable performance [31, 114].

In our system, the eigenface approach is used for the following reasons: (*i*) in the context of personal identification, the background, transformations, and illumination can be controlled, (*ii*) eigenface approach has a compact representation - a facial image can be concisely represented by a feature vector with a few elements, (*iii*) it is feasible to index an eigenface-based template database using different indexing techniques such that the retrieval can be conducted efficiently [139], and (*iv*) the eigenface approach is a generalized template matching approach which was demonstrated to be more accurate than the attribute-based approach [21].

The eigenface-based face recognition consists of the following two stages [144]: (*i*) training stage in which a set of training face images are collected; eigenfaces that correspond to the M highest eigenvalues are computed from the training set; and each face is represented as a point in the M-dimensional eigenspace, and (*ii*) operational stage in which each test image is first projected onto the M-dimensional eigenspace; the M-dimensional face representation is then deemed as a feature vector and fed to a classifier to establish the identity of the individual.

A $W \times H$ face image $I(x, y)$ can be represented as a $W \times H$-dimensional feature vector by concatenating the rows of $I(x, y)$ together. Thus, each $W \times H$ face image becomes a point in the $W \times H$-dimensional space. The value of $W \times H$ is typically

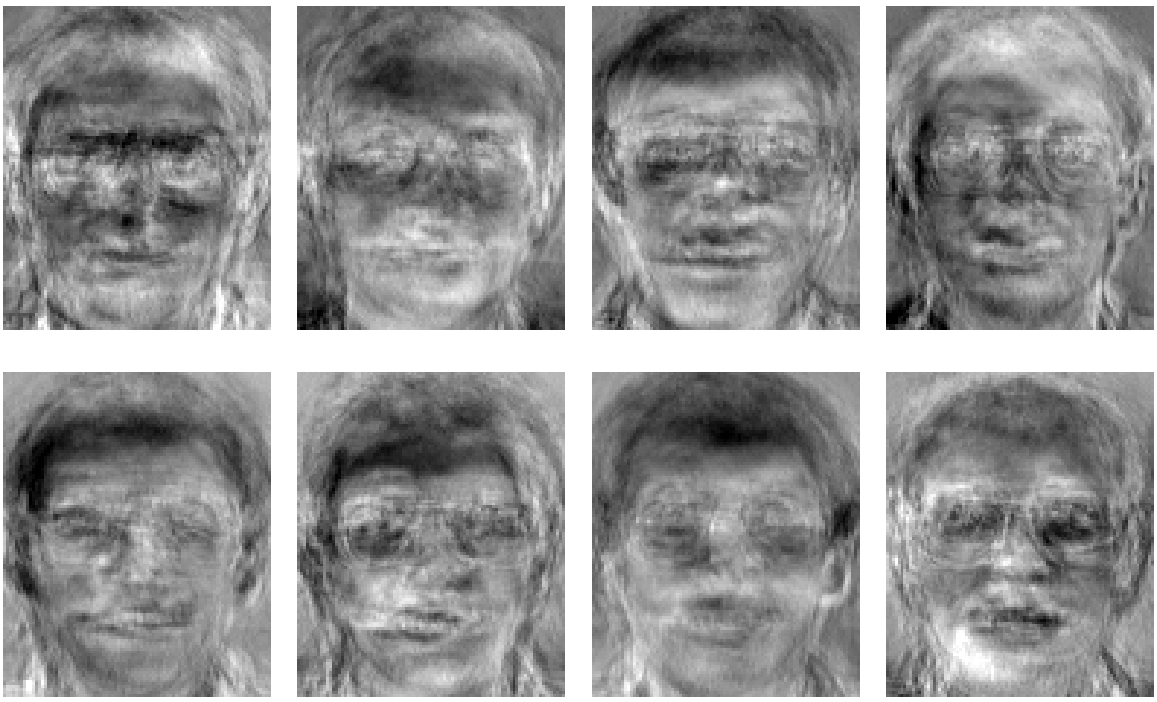


Figure 7.5: First eight eigenfaces obtained from 542 training images of size 92×112; they are listed, from left to right and top to bottom, in decreasing values of the corresponding eigenvalues.

large, on the order of several thousands for even small image sizes. Face images in such a high dimensional space are not randomly distributed. Therefore, it is efficient and beneficial to project them to a lower dimensional subspace using principle component analysis [144]. Let $\Psi_1, \Psi_2, ...\Psi_N$ denote the N $W \times H$-dimensional training vectors with zero-mean. Let the M basis vectors, $\mathbf{u_1}, \mathbf{u_2}, ..., \mathbf{u_M}$ be a set of orthonormal vectors that best describe the distribution of face images in the M-dimensional subspace (eigenspace), $M \leq N$. The kth vector, $\mathbf{u_k}, \mathbf{k = 1, 2, ..., M}$, is computed such that [144]

$$\lambda_k = \frac{1}{N} \sum_{i=1}^{N} (\mathbf{u_k^T \Psi_i})^\mathbf{2} \tag{7.1}$$

is maximum, subject to

$$\mathbf{u_i^T u_j} = \begin{cases} 1, & \text{if i = j} \\ 0, & \text{otherwise.} \end{cases} \tag{7.2}$$

The value λ_k is the kth largest eigenvalue of the covariance matrix Σ which can be estimated using the training samples by

$$\hat{\Sigma} = \frac{1}{N}\sum_{i=1}^{N} \Psi_i \Psi_i^T. \tag{7.3}$$

The vector $\mathbf{u_k}$ is the kth eigenvector of the covariance matrix Σ corresponding to λ_k.

With the M-dimensional eigenspace defined, training vectors, $\Psi_1, \Psi_2, \cdots, \Psi_N$, can be represented as a set of M-dimensional feature vectors, $\Phi_1, \Phi_2, \cdots, \Phi_N$:

$$\Phi_k = \mathbf{u^T \Psi_i},\ \mathbf{i = 1, 2, \cdots, N}, \tag{7.4}$$

where $\mathbf{u} = (\mathbf{u_1}, \mathbf{u_2}, ..., \mathbf{u_M})$. Figure 7.5 shows the first 8 eigenfaces corresponding to the 8 largest eigenvalues.

In the operational phase, a detected face image, Γ, which is normalized to zero mean, is vectorized and projected onto the eigenvectors according to $\Pi = \mathbf{u^T \Gamma}$. With both training samples and test samples being projected onto M-dimensional eigenspace, face recognition can be accomplished by a classifier operating in the eigenspace. In the context of personal identification, only a very limited number of training samples is available for each individual [106]. Thus, a k-nearest neighbor

classifier is typically used, in which the distance, d, called Distance From Feature Space (DFFS) [144] between a template, Φ, and a test pattern, Π, is defined as $\|\Phi - \Pi\|$, where $\| \bullet \|$ means L_2 *norm*.

7.3 Decision Fusion

The decision made by each biometrics has an associated confidence value. A decision fusion scheme should utilize all these confidence values associated with individual decisions to reach a more reliable decision. As we mentioned early, in order to derive a decision fusion scheme, we need to define (i) a confidence measure for each individual biometrics and (ii) a decision fusion criterion. The confidence of a given biometrics may be characterized by its *false acceptance rate* (FAR), which is defined as the probability of an impostor being accepted as a genuine individual. In order to estimate FAR, the *impostor distribution* which is defined as the distribution of similarity between biometric characteristic(s) of different individuals needs to be computed.

7.3.1 Impostor Distribution for Fingerprint Verification

A model that can precisely characterize the impostor distribution of a minutiae matching algorithm is not easy, since (i) the minutiae in a fingerprint are distributed randomly in the region of interest, (ii) the region of interest for each input fingerprint may be different, (iii) each input fingerprint tends to have a different number of minutiae, (iv) there may be a significant number of spurious minutiae and missing minutiae, (v) sensing, sampling, and feature extraction may result in errors in minu-

tiae positions, and (vi) sensed fingerprints may have different distortions. However, it is possible to obtain a general model of the overall impostor distribution by making some simplifying assumptions.

Let us assume that the input fingerprint and the template have already been registered and the region of interest of both the input fingerprint and the template is of the same size, a $W \times W$ (for example, 500×500) region. The $W \times W$ region is tessellated into small cells of size $w \times w$. These cells are assumed to be sufficiently large (for example, 40×40) such that possible deformation and transformation errors are within the specified bound. Therefore, there are a total of $\frac{W}{w} \times \frac{W}{w} (= N_c)$ different cells in the region of interest. Further, assume that each fingerprint has the same number of minutiae, N_m ($\leq N_c$), which are distributed randomly in different cells and each cell contains at most one minutiae. Each minutiae is directed towards one of the D (for example, 8) possible orientations with equal probability. Thus, for a given cell, the probability, P_{empty}, that the cell is empty with no minutiae present is $\frac{N_m}{N_c}$ and the probability, P, that the cell has a minutiae that is directed towards a specific orientation is $\frac{1-P_{empty}}{D}$. A pair of corresponding minutiae between a template and an input is considered to be identical if and only if they are in the cells at the same position and directed in the same direction (see Figure 7.6). With the above simplifying assumptions, the number of corresponding minutiae pairs between any two randomly selected minutiae patterns is a random variable, Y, which has a binomial

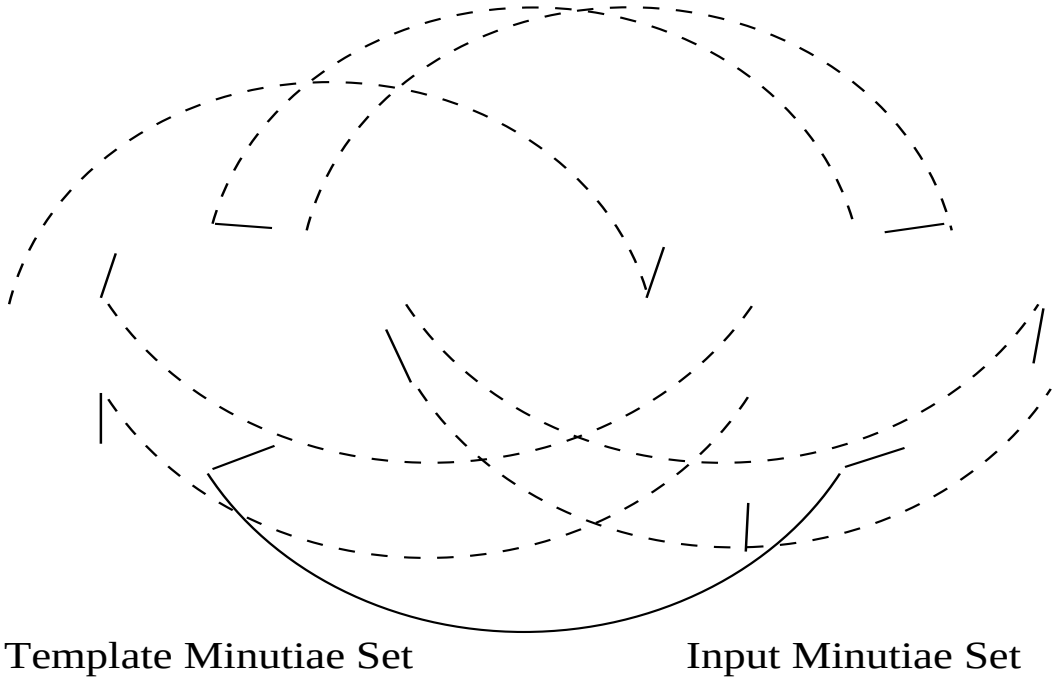


Figure 7.6: Minutiae matching model. A solid line indicates a match and a dashed line indicates a mismatch.

distribution with parameters N_m and P [113]:

$$g(Y) = \frac{N_m!}{Y!(N_m - Y)!} P^Y (1 - P)^{(N_m - Y)}. \tag{7.5}$$

The probability that the number of corresponding minutiae pairs between any two minutiae patterns is less than a given threshold value, y, is

$$G(y) = g(Y < y) = \sum_{k=0}^{y-1} g(k). \tag{7.6}$$

The decision made by the proposed minutiae matching algorithm for an input fingerprint and a template is based on the comparison of the "normalized" number

of corresponding minutiae pairs against a threshold. Therefore, under the assumption that minutiae in the region of interest of fingerprints of different individuals are randomly distributed, the probability that an impostor is accepted is $(1 - G(y_I))$.

7.3.2 Impostor Distribution for Face Recognition

The characterization of impostor distribution for face recognition is more difficult. Due to the relatively low discrimination capability of face recognition, this module needs to keep the top n matches to guarantee that the genuine individual will be identified if he or she is in the database.

Let $\Phi_1, \Phi_2, ..., \Phi_N$ be the N face templates stored in the database. The top n matches, $\Phi_1^r, \Phi_2^r, ..., \Phi_n^r$, are obtained by searching through the entire database, in which N comparisons are conducted explicitly (in the linear search case) or implicitly (in organized search cases such as the tree search). The top n matches are arranged in the increasing order of DFFS (Distance From Feature Space, Section 2) values. The smaller the DFFS value, the more likely it is that the match is correct. Since the relative distances between consecutive DFFSs are invariant to the mean shift of the DFFSs, it is beneficial to use relative instead of absolute DFFS values. The probability that a retrieved top n match is incorrect is different for different ranks. The probability that the first match is incorrect tends to be smaller than the probability that the second match is incorrect, the probability that the second match is incorrect tends to be smaller than the probability that the third match is incorrect, and so on. Thus, the impostor distribution should be a decreasing function of rank order and

it is a function of both the relative DFFS values, Δ, and the rank order, i, which has the following form: $F_i(\Delta)P_{order}(i)$, where $F_i(\Delta)$ represents the probability that the consecutive DFFS between impostors and their claimed individuals at rank i are larger than a value Δ and $P_{order}(i)$ represents the probability that the retrieved match at rank i is an impostor. In practice, $F_i(\Delta)$ and $P_{order}(i)$ need to be estimated from empirical data.

In order to simplify the analysis, we assume that each individual has only one face template in the database. Thus, there are a total of N individuals enrolled in the database and $I_1, I_2, ..., I_N$ are used as identity indicators. Let X^α denote the DFFS between an individual and her own template which is a random variable with density function $f^\alpha(X^\alpha)$ and let $X_1^\beta, X_2^\beta, ..., X_{N-1}^\beta$ denote the DFFSs between an individual and the templates of the other individuals in the database, which are random variables with density functions, $f_1^\beta(X_1^\beta), f_2^\beta(X_2^\beta), ..., f_{N-1}^\beta(X_{N-1}^\beta)$, respectively. Assume that X^α and $X_1^\beta, X_2^\beta, ..., X_{N-1}^\beta$ are statistically independent and $f_1^\beta(X_1^\beta) = f_2^\beta(X_2^\beta) = ...f_{N-1}^\beta(X_{N-1}^\beta) = f^\beta(X^\beta)$. For an individual, Π, who has a template stored in the database, $\{\Phi_1, \Phi_2, ...\Phi_N\}$, the rank, R, of X^α among $X_1^\beta, X_2^\beta, ..., X_{N-1}^\beta$ is a random variable with probability

$$P(R = i) = \frac{(N-1)!}{i!(N-1-i)!} p^i (1-p)^{(N-1-i)}, \tag{7.7}$$

where

$$p = \int_{-\infty}^{\infty} \int_{-\infty}^{X^\alpha} f^\alpha(X^\alpha) f^\beta(X^\beta) dX^\beta dX^\alpha. \tag{7.8}$$

When $p \ll 1$ and N is sufficiently large, P(R) may be approximated by a Poisson distribution [113],

$$P(R = i) \doteq \frac{e^{(-a)}a^i}{i!}, \tag{7.9}$$

where $a \doteq np$. Obviously, $P(R = i)$ is exactly the probability that matches at rank i are genuine individuals. Therefore,

$$P_{order}(i) = 1 - P(R = i). \tag{7.10}$$

Although the assumption that $X_1^\beta, X_2^\beta, ..., X_{N-1}^\beta$ are *i.i.d.* may not be true in practice, it is still reasonable to use the above parametric form to estimate the probability that retrieved matches at rank i are impostors. Our experimental results support this claim.

Without any loss of generality, we assume that, for a given individual, Π, $X_1^\beta, X_2^\beta, ..., X_{N-1}^\beta$ are arranged in increasing order of values. Define the non-negative distance between the $(i+1)th$ and ith DFFS values as the ith DFFS distance,

$$\Delta_i = X_{i+1}^\beta - X_i^\beta, \quad 1 \leq i < N - 1. \tag{7.11}$$

The distribution, $f_i(\Delta_i)$, of the ith relative distance, Δ_i, is obtained from the joint distribution $w_i(X^\beta, \Delta_i)$ of the ith value, X^β, and the ith relative distance, Δ_i,

$$f_i(\Delta_i) \quad = \quad \int_{-\infty}^{\infty} w_i(X^\beta, \Delta_i) dX^\beta, \tag{7.12}$$

$$w_i(X^\beta, \Delta_i) = CF^\beta(X^\beta)^{i-1}[1 - F^\beta(X^\beta + \Delta_i)]^{N-i} f^\beta(X^\beta) f^\beta(X^\beta + \Delta_i), \quad (7.13)$$

$$C = \frac{(N-1)!}{(i-1)!(N-2-i)!}, \quad (7.14)$$

where $F^\beta(X^\beta) = \int_{-\infty}^{X^\beta} f^\beta(X^\beta) dX^\beta$ [58]. With the distribution, $f_i(\Delta_i)$, of the ith distance defined, the probability that the DFFS of the impostor at rank i is larger than a threshold value, Δ, is

$$F_i(\Delta) = \int_{\Delta}^{\infty} f_i(\Delta_i) d\Delta_i. \quad (7.15)$$

The above equations do not make any assumptions about the distributions of $X_1^\beta, X_2^\beta, ..., X_{N-1}^\beta$ as long as they are $i.i.d.$ The equations also hold even if the mean values of $X_1^\beta, X_2^\beta, ..., X_{N-1}^\beta$ shift. Therefore, it can tolerate, to a certain extent, DFFS variations which is a desirable property. In our system, we assume that $X_1^\beta, X_2^\beta, ..., X_{N-1}^\beta$ are distributed with a Gaussian distribution with unknown mean and variance.

7.3.3 Decision Fusion

The impostor distribution for face recognition and the impostor distribution for fingerprint verification provide confidence measures for each of the top n matches retrieved by the face recognition module. Without a loss of generality, we assume that at most one of the n possible identities established by the face recognition module for a given individual is the genuine identity of the individual. The final decision by integration either rejects all the n possibilities or accepts only one of them as the genuine identity.

In practice, it is usually specified that the FAR of the system should be less than a given value [106]. Therefore, the goal of decision fusion, in essence, is to derive a decision criterion which satisfies the FAR specification.

It is reasonable to assume that the DFFS between two different individuals is statistically independent of the fingerprint matching score between them; facial similarity between two individuals does not imply that they have similar fingerprints, and vice versa. This assumption should not be confused with the situation where an impostor tries to fool the system by counterfeiting the face and/or fingerprints of the genuine individual. Let $F_i(\Delta)P_{order}(i)$ and $G(Y)$ denote the impostor distribution at rank i for face recognition and fingerprint verification modules, respectively. The composite impostor distribution at rank i may be defined as

$$H_i(\Delta, Y) = F_i(\Delta)P_{order}(i)G(Y). \tag{7.16}$$

Let $\{I_1, I_2, ..., I_n\}$ denote the n possible identities established by face recognition, $\{X_1, X_2, ..., X_n\}$ denote the corresponding n DFFSs, $\{Y_1, Y_2, ..., Y_n\}$ denote the corresponding n fingerprint matching scores, and FAR_o denote the specified value of FAR. The final decision, $ID(\Pi)$, for a given individual Π is determined by the following criterion:

$$ID(\Pi) = \begin{cases} I_k, & if \begin{cases} H_k(\Delta_k, Y_k) < FAR_o, \ \ and \\ H_k(\Delta_k, Y_k) = \min\{H_1(\Delta_1, Y_1), ..., H_n(\Delta_n, Y_n)\} \end{cases} \\ impostor, & otherwise, \end{cases} \tag{7.17}$$

where $\Delta_i = X_{i+1} - X_i$. Since $H_i(\Delta, Y)$ defines the probability that an impostor is accepted at rank i with consecutive relative DFFS, Δ, and fingerprint matching score, Y, the above decision criterion satisfies the FAR specification.

Note that the decision criteria in Eq. (7.17) depends on the number of individuals, N, enrolled in the database, since F_i depends on N. However, it does not mean that the distributions of F_is have to be recomputed whenever a new individual is enrolled in the database. In fact, if $N \gg 1$, the distributions of F_is for different values of N are quite similar to one another. On the other hand, the decision criterion still satisfies the FAR specification when N increases, though it may not be able to take full advantage of the information contained in the N comparisons. In practice, an update schema which recomputes the decision criteria whenever the number of added individuals is larger than a pre-specified value can be used to exploit all the available information.

7.4 Summary

A biometric system which is based only on a (or one snapshot of a) single biometric characteristic may not always be able to achieve the desired performance. A *multimodal biometrics* technique, which combines multiple biometrics in making an identification, can be used to overcome the limitations. Integration of multimodal biometrics for an identification system has two goals: (*i*) improve the identification accuracy and (*ii*) improve the identification speed (throughput). We have developed a decision fusion scheme which integrates two different biometrics (face and finger-

print) that complement each other. In this scheme, a biometrics that is suitable for database retrieval is used to index the template database and a biometrics that is reliable in deterring impostors is used to ensure the overall system accuracy. In addition, a decision fusion scheme which exploits all the information in the decisions made by each individual biometrics is used to make a more reliable decision.

The decision fusion schema may be applied to similar scenarios in other domains to provide a better discrimination performance. For example, in image database retrieval, a less reliable but computationally attractive algorithm may be used to retrieve the top n matches; then a more reliable, but computationally more expensive algorithm may be used to verify the top n matches; and finally a decision fusion scheme can be used to reach a more reliable decision.

Chapter 8

Fingerprint Classification

The main advantage of fingerprint classification is that it provides an indexing scheme to facilitate efficient matching in large fingerprint databases; if two fingerprint images are the impressions of the same finger, then they must belong to the same category. Therefore, a query fingerprint needs to be compared only with the database fingerprints of the same category in the fingerprint matching process. Without an effective fingerprint classification scheme or some other indexing scheme, fingerprint identification involves an exhaustive matching of query fingerprint to all the fingerprints in the database, which is computationally demanding [85].

8.1 Automatic Fingerprint Classification

An automatic fingerprint classification algorithm classifies a fingerprint into a number of pre-specified categories according to the features extracted from the fingerprint. Generally, the pre-specified categories could be either the categories defined in Chap-

ter 2 or a "new" set of categories which have not been used by forensic experts, but may be easily recognized automatically.

For automatic fingerprint classification, the following five issues are of great interest: (*i*) number of fingerprint categories, (*ii*) distribution of fingerprints among the categories, (*iii*) consistency of classification scheme, (*iv*) classification accuracy, and (*v*) computational requirements of the classification algorithm. Property (*i*) specifies that a classification algorithm should be able to classify fingerprints into a sufficiently large number of categories. The effectiveness of the indexing mechanism depends on the number of categories; the larger the number of categories, the more efficient the resulting indexing mechanism. Property (*ii*) emphasizes that fingerprints should be distributed uniformly among the categories of interest; it is desirable that each category should contain the same number of fingerprints. The more uniformly the fingerprints are distributed among the categories of interest, the more effective the resulting indexing mechanism. Property (*iii*) suggests that the fingerprints in each category of interest should be similar in terms of global pattern configuration. Property (*iv*) stipulates that classification scheme should be able to reach a desirable classification accuracy. If the classification scheme can not reach a desirable accuracy, then the classification scheme will be useless for the purpose of indexing. Property (*v*) says that the classification should be performed quickly. If the classification is more expensive than conducting a linear search of a fingerprint database, then it is meaningless to use the classification scheme to index the fingerprint database. Ideally, an automatic fingerprint classification algorithm should be able to quickly classify fingerprints into a significant number of categories consistently with a desirable accuracy.

In practice, due to the complex and noisy nature of fingerprint ridge configurations, it is a major challenge to design such a fingerprint classification algorithm.

Due to large variations in the ridge pattern configuration, the definition of the global pattern features used to specify the classification criteria used by fingerprint experts is very complex and vague. When the quality of input images is not very good, it is extremely difficult for an automatic algorithm to reliably extract these "high level" features from the images. Therefore, the performance of the fingerprint classification algorithms based on these global features is far from desirable. On the other hand, the classification criteria used by fingerprint experts is sufficiently vague that it may classify fingerprints with similar pattern configurations into different categories. Figure 8.1 shows examples of fingerprints that appear similar but are classified into different categories according to fingerprint experts.

(a) (b) (c)

Figure 8.1: Examples of fingerprints from different categories; (a) tented arch; (b) loop; (c) whorl; it seems that all the fingerprints shown here should be in the loop category.

As mentioned early, the fingerprint classification is intended for quickly providing an indexing mechanism and to give an indication of general pattern agreement. It is not necessarily required that fingerprints should be classified according to the tradi-

tional classification scheme. Essentially, what is needed is a classification scheme that is able to consistently and correctly classify fingerprints into a number of uniformly distributed categories with large intra-class similarity and small inter-class similarity. A central problem in designing such a classification scheme is to decide what features should be used to classify fingerprints and how categories are defined based on these features. These fingerprint features should be invariant to the translation and rotation of the input fingerprint images and be able to capture the inherent nature of the global fingerprint pattern configuration. Global fingerprint features are mainly derived from the orientation field and global ridge shape. The orientation field of a fingerprint consists of the local ridge orientation tendency in local neighborhoods and forms an abstraction of the local ridge structures. It has been shown that the orientation field is highly structured and can be roughly approximated by a core-delta model [104]. Therefore, singular points and their relationship can be used to derive fingerprint categories.

Previous approaches to fingerprint classification can be roughly divided into two categories: (*i*) statistical approach [23, 22, 24, 10, 18, 56] and (*ii*) structural approach [33, 78, 80, 119, 101, 90, 102]. A statistical approach classifies a fingerprint using feature vectors derived directly from the orientation field of the input images. A structural approach extracts and represents fingerprints using a number of salient fingerprint properties and their relationships. These algorithms perform reasonably well when the input fingerprint images are of good quality. When the quality of the input fingerprint images is poor, the performance of these algorithms degrades rapidly. The major reason for the brittleness of these algorithms is that they do not

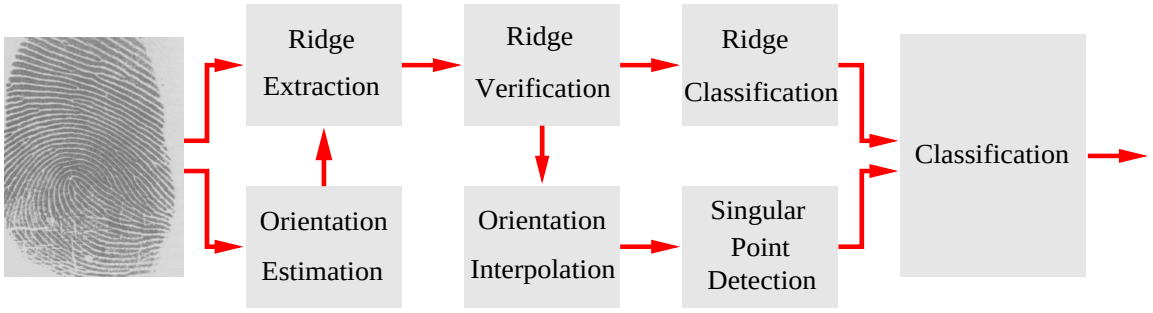


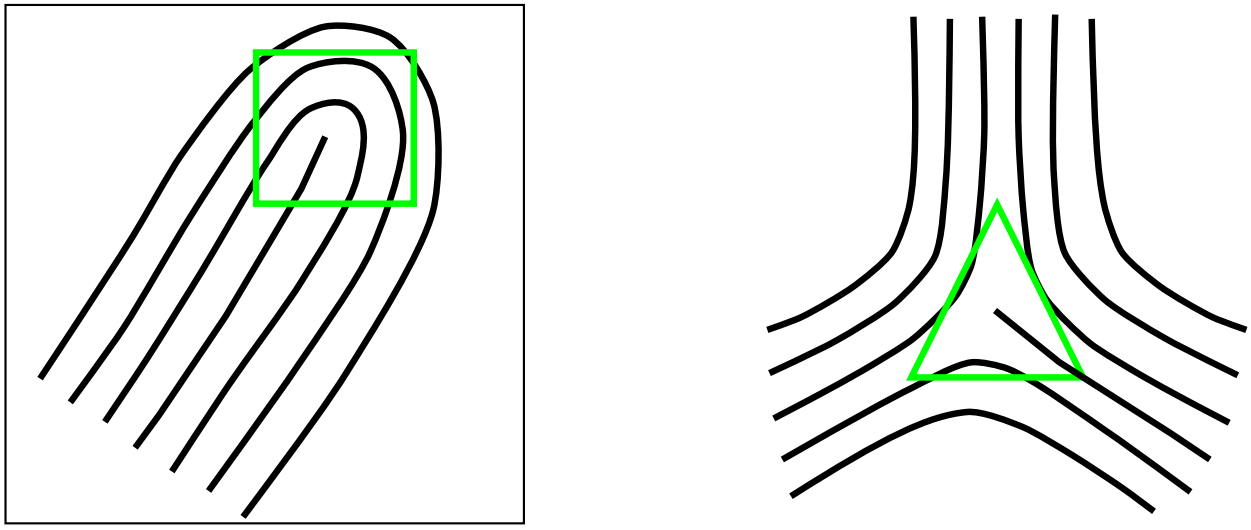
Figure 8.2: The flow-chart of the fingerprint classification algorithm.

utilize robust features.

We have designed two fingerprint classification schemes based on the features mentioned above. One follows the approach to classify fingerprints into five categories *(arch, tented arch, left loop, right loop, and whorl)* according to the extracted cores, deltas, and ridge shape. The other scheme classifies fingerprints into four categories which depends on pattern similarity based on cores and deltas. The main stages of the classification algorithms are depicted in Figure 8.2, which can be divided into two parts: (i) feature extraction and (ii) classification scheme. The feature extraction module extracts two types of features: (i) singular points, and (ii) fingerprint ridges.

8.2 Feature Extraction

For local orientation estimation and ridge extraction, we use the same procedure as used for minutiae extraction algorithm. Our algorithm spends a significant amount of effort to improve the quality of extracted orientation field and ridges, which results in a more robust feature extraction and classification. In order to understand the



core delta

Figure 8.3: Singular points.

algorithm, some additional definitions are given.

8.2.1 Definitions

A *Recurring ridge* is defined as a chain of pixels, $r_1, r_2, ..., r_n$, in a thinned ridge map, where r_1 is the first ridge pixel, r_n is the last ridge pixel, and each pair of consecutive pixels, (r_{i-1}, r_i) is eight connected, which cumulatively turns more than a certain degree when traveling from r_i to r_j, where $1 \leq i < j \leq n$.

A *singular point* is either a core point or a delta point which is characterized by its position and type (see Figure 8.3). A core is defined as a point in the orientation field where the orientation in a small local neighborhood around the point presents semi-circular tendency. A delta is defined as a point in the orientation field where a small local neighborhood around the point forms three sectors and the orientation in each sector presents hyperbolic tendency.

8.3 Ridge Verification

Since ridge structures in poor-quality fingerprint images are not always well-defined, it may lead to: (i) incorrect local ridge orientation estimates and (ii) incorrect extracted ridges. It is very difficult to correctly classify a fingerprint based on the incorrect orientation field and incorrect ridge structure. Therefore, a noise removal algorithm should be applied to obtain more precise orientation estimates and ridges.

A direct way to ensure that orientation field estimation and ridge extraction are robust with respect to the quality of input fingerprint images is to enhance the input images before orientation estimation and ridge extraction. As long as the ridge structures are not corrupted completely, it is possible to develop an enhancement algorithm to improve the clarity of ridge structures in corrupted fingerprint images. However, the extraction of local orientation and ridge structure are intertwined; a correct estimate of either one will result in a correct estimate of the other.

Fingerprint ridges are highly structured both locally and globally. Locally, each ridge runs parallel to the neighboring ridges. Globally, ridges form families of similar types of smooth curves. In an extracted ridge map, ridges in the well-defined regions satisfy the above properties, while in the corrupted regions, they do not have such properties. Therefore, we can use these properties to differentiate the corrupted ridges from the true ridges. The orientation field in the regions where the true ridges appear can be correctly estimated as the tangent direction of the ridges. If the true regions occupy most of the fingerprint image, we can correctly interpolate the local orientation in those regions where true ridges are not identified using these true

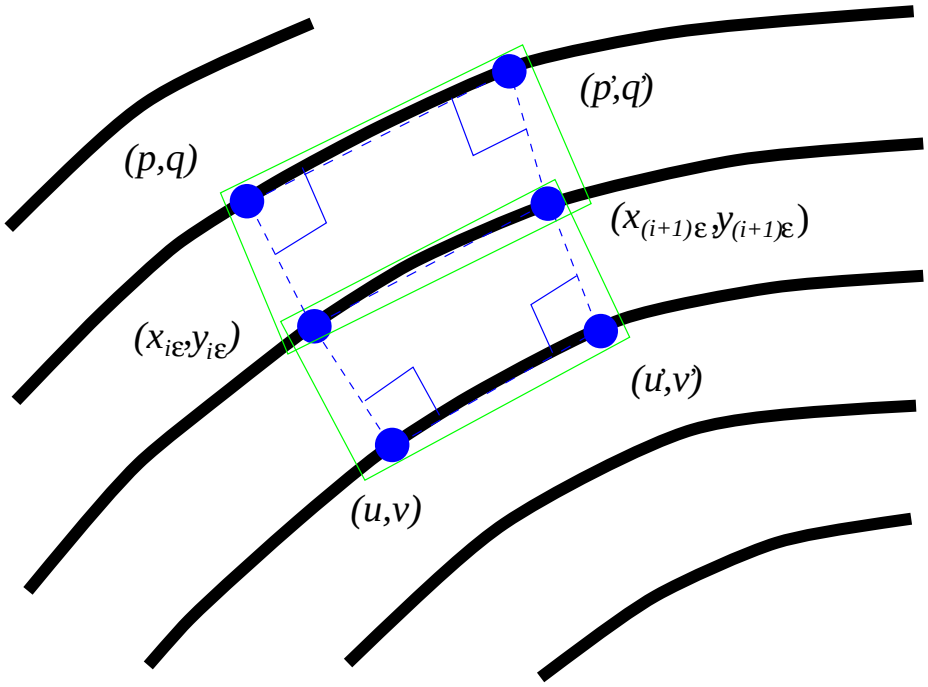


Figure 8.4: Ridge verification.

local orientations. Obviously, if the corrupted regions occupy a significant portion of the fingerprint image, the true local orientation may not be recovered using the interpolation method.

We have developed a ridge verification algorithm which receives as input a thinned ridge map and outputs a refined thinned ridge map, a refined orientation field, and a quality index which indicates the goodness of the input ridge map. Let $\mathcal{R}$, $\mathcal{O}'$, and $\mathcal{R}'$ be the input ridge map, the interpolated orientation field, and the verified ridge map, respectively. The major steps in our ridge verification algorithm are as follows:

1. *Initialize $\mathcal{O}'$, $\mathcal{R}'$, and $\mathcal{A}$ which is a map used to indicate the true regions.*

2. *Delete all ridge pixels in $\mathcal{R}$ which have more than two 8-connected neighboring pixels to ensure that each ridge is a single 8-connected chain.*

3. *Trace and label all the ridges in $\mathcal{R}$.*

4. *For each traced ridge, $r = \{(x_1, y_1), (x_2, y_2)..., (x_n, y_n)\}$, represented in terms of the coordinates of points on it, do the following:*

   - *smooth r.*
   - *let $(x_{i\epsilon}, y_{i\epsilon})$ and $(x_{(i+1)\epsilon}, y_{(i+1)\epsilon})$ denote the starting point and ending point of a segment in r, where ϵ is the length of the segment and $i = 0, \epsilon, 2\epsilon, ..., \lfloor \frac{n-\epsilon}{\epsilon} \rfloor \epsilon$. Find the 2 nearest neighboring ridge points which are on the line that crosses $(x_{i\epsilon}, y_{i\epsilon})$ and perpendicular to the ridge segment on both sided of the segment: (u, v) and (u', v') and the 2 nearest neighboring ridge points which are on the line that crosses $(x_{(i+1)\epsilon}, y_{(i+1)\epsilon})$ and perpendicular to the ridge segment on both sided of the segment: (p, q) and (p', q'). Note that $(x_{i\epsilon}, y_{i\epsilon})$, $(x_{(i+1)\epsilon}, y_{(i+1)\epsilon})$, (p, q), and (u, v) form a quadrilateral at one side of the segment and $(x_{i\epsilon}, y_{i\epsilon})$, $(x_{(i+1)\epsilon}, y_{(i+1)\epsilon})$, (p', q'), and (u', v') form a quadrilateral at the other side of the segment.*
   - *for each quadrilateral, find the minimum rectangle that contains the quadrilateral. Compute the ratio, η, between the area of the quadrilateral and the area of the minimum rectangle. If η is larger than a threshold ($\eta_0 = 0.75$), then label all the pixels inside the quadrilateral as foreground pixels. Otherwise, label them as background pixels.*

5. *Remove in $\mathcal{A}$ all the foreground connected components whose area is less than a threshold ($\omega_0 = 15$).*

6. *Fill in $\mathcal{A}$ all the background connected components whose area is less than a threshold ($\tau_0 = 15$).*

7. *Compute local orientation at all the pixels in $\mathcal{O}'$ where the corresponding pixels in $\mathcal{A}$ are foreground pixels as the orientation of the nearest ridge segment.*

8. *Interpolate the local orientation at all pixels in $\mathcal{O}'$, where the corresponding pixels in $\mathcal{A}$ are background pixels.*

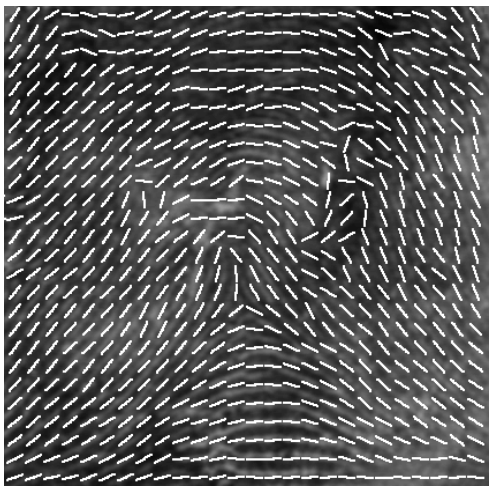
9. *Return the percentage of the area of the foreground regions in $\mathcal{A}$ with respect to total area of $\mathcal{A}$ as the quality index.*

An example of ridge verification is depicted in Figure 8.5, which demonstrates that a better orientation field can be obtained by using our ridge verification algorithm.

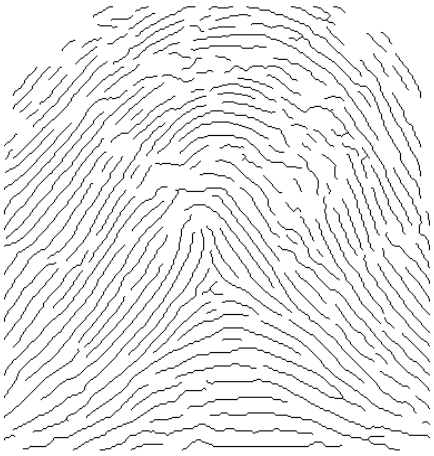
8.3.1 Singular Point Detection

A singular point is defined as the point where the vector field is not continuous, which can be characterized using *Poincare index*. The Poincare index at a given point, (x, y),

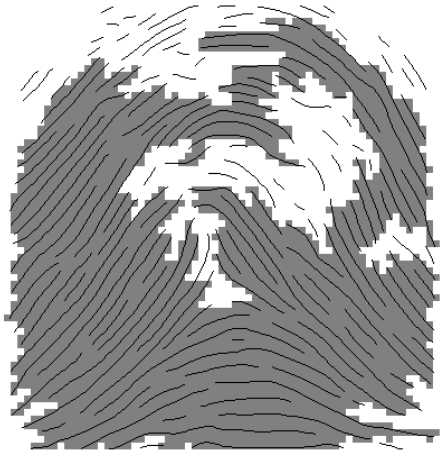
(a)



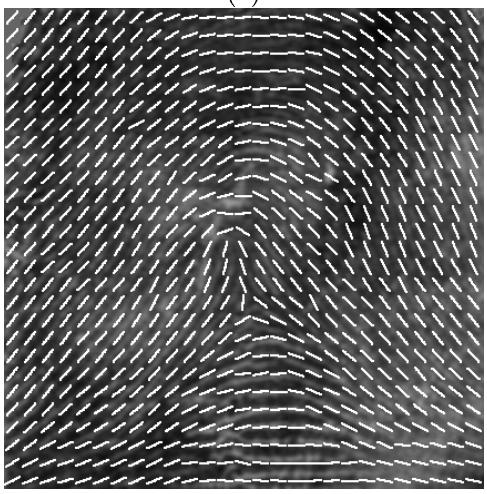
(b)



(c)



(d)



(e)

Figure 8.5: Ridge verification; (a) input image; (b) orientation field; (c) thinned ridge map, (d) verified ridge map, where the verified ridges are marked with gray shade; (e) interpolated orientation field.

in a vector field, $\mathcal{V}$, is defined as follows:

$$Poincare(x,y) = \frac{1}{2\pi}\lim_{\varepsilon\to 0}\left\{\int_0^{2\pi}\frac{\partial}{\partial}\mathcal{V}(x+\varepsilon\cos\theta, y+\varepsilon\sin\theta)d\theta\right\}. \tag{8.1}$$

If the Poincare index at a given point (x,y) is not equal to zero, then the point (x,y) is called a singular point. The Poincare index of a core-shaped singular point in a vector field has a value of 1 and the Poincare index for a delta-shaped singular point has a value of -1.

It has been shown that the orientation field of a fingerprint image can not be unambiguously represented as a vector field [104]. The definition of Poincare index needs to be extended to the orientation field. Let $\mathcal{O}'$ be the orientation field. The Poincare index at a given point (i,j) is defined as follows:

$$Poincare(i,j) = \frac{1}{2\pi}\lim_{\varepsilon\to 0}\left\{\int_0^{2\pi}\frac{\partial}{\partial}\mathcal{O}'(i+\varepsilon\cos\theta, j+\varepsilon\sin\theta)d\theta\right\}, \tag{8.2}$$

where

$$\frac{\partial}{\partial}\mathcal{O}'(i+\varepsilon\cos\theta, j+\varepsilon\sin\theta) = \begin{cases} d\delta, & \text{if } |d\delta| < \pi/2, \\ \pi + d\delta, & \text{if } d\delta \le -\pi/2, \\ \pi - d\delta, & otherwise, \end{cases} \tag{8.3}$$

$$d\delta = \lim_{\nu\to 0}\frac{\mathcal{O}'(i+\varepsilon\cos(\theta+\nu), j+\varepsilon\sin(\theta+\nu)) - \mathcal{O}'(i+\varepsilon\cos\theta, j+\varepsilon\sin\theta)}{\nu}. \tag{8.4}$$

In an orientation field, the Poincare index of a core-shaped singular point has a value of 1/2 and the Poincare index of a delta-shaped singular point has a value of -1/2. In

a digital image, the computation of the Poincare index is implemented by replacing the integration in Eq. (8.2) by a summation of all orientation differences along a closed digital curve. Let $\Psi_x(\cdot)$ and $\Psi_y(\cdot)$ represent the x and y coordinates of a closed digital curve with Ψ pixels. The Poincare index at pixel (i, j) which is enclosed by the digital curve can be computed as follows:

$$Poincare(i,j) = \frac{1}{2\pi}\sum_{k=0}^{\Psi}\Delta(k), \tag{8.5}$$

where

$$\Delta(k) = \begin{cases} d\delta(k), & \text{if } |d\delta(k)| < \pi/2, \\ \pi + d\delta(k), & \text{if } d\delta(k) \leq -\pi/2, \\ \pi - d\delta(k), & otherwise, \end{cases} \tag{8.6}$$

$$d\delta(k) = \mathcal{O}'(\Psi_x((i+1)MOD\Psi), \Psi_y((i+1)MOD\Psi)) - \mathcal{O}'(\Psi_x(i), \Psi_y(i)). \tag{8.7}$$

The size of the closed digital curve is crucial for the performance of a singular point detection algorithm using the Poincare index. If it is too small, then a small perturbation of orientations may result in spurious singular points being detected. On the other hand, if it is too large, then a true pair of core and delta which are close to one another may be ignored because the Poincare index of a digital curve that includes an equal number of cores and deltas is 0. We have developed a singular point detection algorithm which uses a closed square curve with a length of 25 pixels. We have empirically determined that a curve of 25 pixels is a good trade-off between detections and misses of singular points. Let $\mathcal{O}'$ be the interpolated orientation field.

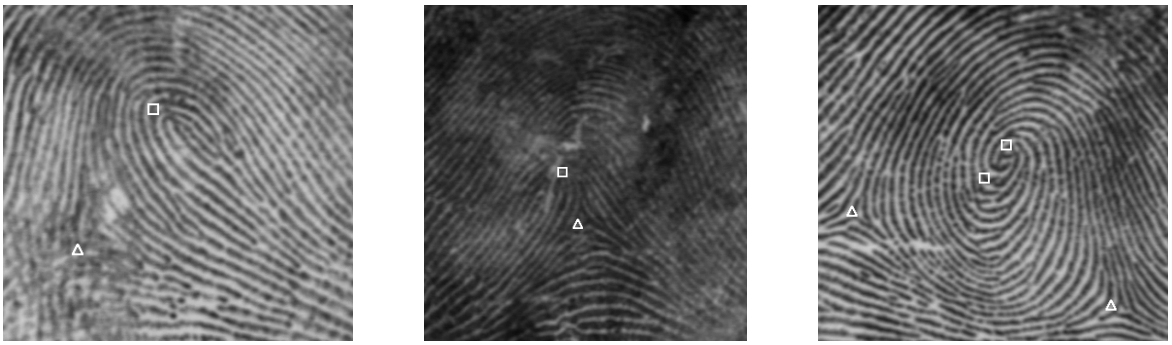


Figure 8.6: Singular point detection: a core is labeled by a rectangle and a delta is labeled by a triangle.

The main steps in our singular point detection algorithm are as follows:

1. *Initialize $\mathcal{A}$, which is a label image used to indicate the singular points.*
2. *For each pixel (i, j) in $\mathcal{O}'$, compute the Poincare index and assign the corresponding pixel in $\mathcal{A}$ a value 1 if the Poincare index is (1/2) and a value 2 if the Poincare index is (-1/2).*
3. *Find each connected component in $\mathcal{A}$ with pixel values 1. If the area of the connected component is larger than 7, a core is detected at the centroid of the connected component. If the area of the connected component is larger than 20, then two cores are detected at the centroid of the connected component.*
4. *Find each connected component in $\mathcal{A}$ with pixel values 2. If the area of the connected component is larger than 7, a delta is detected at the centroid of the connected component.*
5. *If more than two cores or more than two deltas are detected, smooth the orientation field $\mathcal{O}'$ and go back to step 1.*

Although the heuristic that at most two cores and two deltas exist in a fingerprint is not always true, it is rarely observed that a fingerprint has more than two cores and two deltas. Results of applying our singular point detection algorithm on two fingerprint images are shown Figure 8.6.

8.3.2 Recurring Ridges

The global shape of ridges determines the global configuration of fingerprints. Ridges in fingerprints are highly structured. Generally, in the upper region (which can be

roughly defined as the region above the highest core points in loops, tented arches, and whorls and the region above the most curved ridges in arches) of a fingerprint, ridges are a family of uni-modal smooth curve segments. In the bottom region, ridges form a family of relatively flat curves. In the middle region, depending on the fingerprint class, ridges may be of the following types: uni-modal curve segment, recurring segment, circular segments, multi-recurring segments, spiral segments, *etc.* The presence of a particular type of ridges defines the class of a fingerprint. If the ridge type can be accurately determined, then the fingerprint can be correctly classified.

We classify ridges into three categories: (*i*) *non-recurring ridge*, (*ii*) *type-1 recurring ridge*, and (*iii*) *type-2 recurring ridge*. Let $r = \{r_1, r_2, ..., r_n\}$ be a ridge of length n, where r_1 is the first ridge pixel, r_n is the last ridge pixel, and each pair of consecutive pixels is eight connected. Then $r' = \{r_1, r_{2\epsilon}, ..., r_{m\epsilon}\}$, where $m = \lfloor \frac{n-\epsilon}{\epsilon} \rfloor$, is obtained by sampling r at intervals of length ϵ. Define the cumulative orientation of r as:

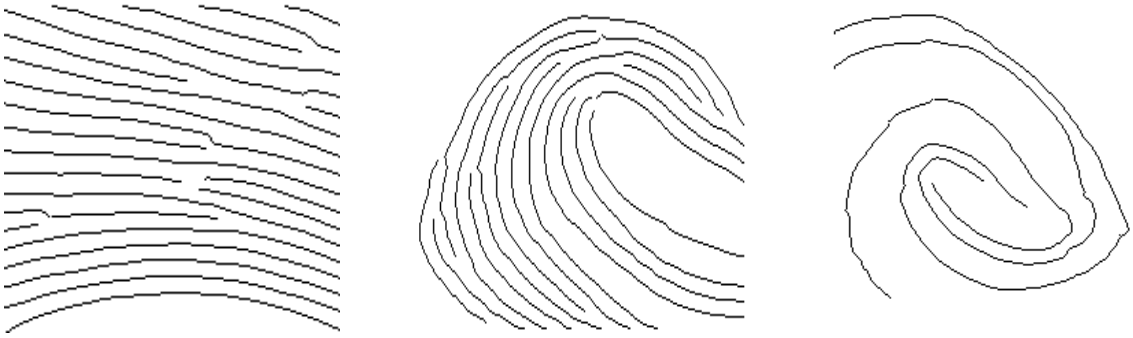
$$
\begin{aligned}
AO(r) &= |\frac{1}{2\pi} \sum_{k=2}^{m-1} \varpi(k)| \\
\varpi(k) &= \begin{cases} \rho(k), & \text{if } |\rho(k)| < \pi, \\ 2\pi + \rho(k), & \text{if } \rho(k) \leq -\pi, \\ 2\pi - \rho(k), & otherwise, \end{cases} \\
\rho(k) &= \vartheta(k) - \vartheta(k-1),
\end{aligned}
$$

where $\vartheta(k)$ represents the angle from $r'(k)$ to $r'(k+1)$. Define any sequence of ridge pixels in r, $\{r_i, r_2, ..., r_j\}$, where $1 \leq i < j \leq n$, a sub-ridge of r. A non-recurring

ridge, r, is a ridge such that the cumulative orientation of any sub-ridge of r is less than a threshold, $T_{non} = 150^o$. A type-1 recurring ridge, r, is a ridge such that the cumulative orientation of any sub-ridge of r is between the two thresholds, T_{non} and $T_{rec} = 270^o$. A type-2 recurring ridge, r, is a ridge such that the cumulative orientation of any sub-ridge of r is larger than a threshold, T_{rec} or a ridge such that there exist multiple disjoint sub-ridges of r, which are type-1 recurring ridges. Obviously, uni-modal ridge segments and flat ridge segments are non-recurring ridges. Circular ridge segments, multi-recurring ridge segments and spiral ridge segments are type-2 recurring ridges.

It is very difficult to correctly extract all the true ridges from an input fingerprint image, especially when the quality of the input fingerprint image is poor. It is essential that a ridge classification algorithm be able to handle the following undesirable situations: (*i*) spurious ridges, (*ii*) broken ridges, and (*iii*) missing ridges. Ridge verification (see Figure 8.5) can be used to remove all the spurious ridges from a ridge map. Broken ridges can be connected based on the information present near the end of broken ridges. However, it is very difficult to recover missing ridges. This needs both high-level structural analysis and local structural analysis of the ridge pattern, which is very difficult to formulate and implement. We have developed a ridge classification algorithm which traces each ridge in the verified ridge map and classifies each ridge into one of the three categories mentioned above (Figure 8.7). The main steps of the algorithm are depicted as follows:

1. *Trace and label all the ridges in* $\mathcal{R}'$.
2. *For each traced ridge,* $r = \{r_1, r_2, ..., r_n\}$, *expand* r *at both ends to generate a new ridge and repeat the expansion operation until the new generated ridge can*



(a) (b) (c)

Figure 8.7: Ridge classification; (a) ridges classified as non-recurring ridges; (b) ridges classified as type-1 recurring ridges; and (c) ridges classified as type-2 recurring ridges.

not be expanded anymore. Ridge expansion is to connect r with a ridge in $\mathcal{R}'$ at the end points to generate a new ridge with the condition that the expanded ridge is consistent with the original ridge at the expansion points.

3. *Compute the AO of the expanded ridge and classify it according to the criteria mentioned above.*

Examples of ridge classification are shown Figure 8.7.

8.4 Classification

Fingerprints can be classified into a number of pre-specified categories based on the features extracted from the orientation field and ridge map.

8.4.1 Classification Scheme I

This classification scheme classifies input fingerprints into *five* categories according to the number of singular points detected, their relative position and presence of type-1 and type-2 recurring ridges. These categories are: (*i*) arch, (*ii*) tented arch, (*iii*) left loop, (*iii*) right loop, and (*v*) whorl. A prototype of each class is shown in Figure 8.8. Let $\mathcal{O}'$ be the interpolated orientation field; N_c and N_d be the number of

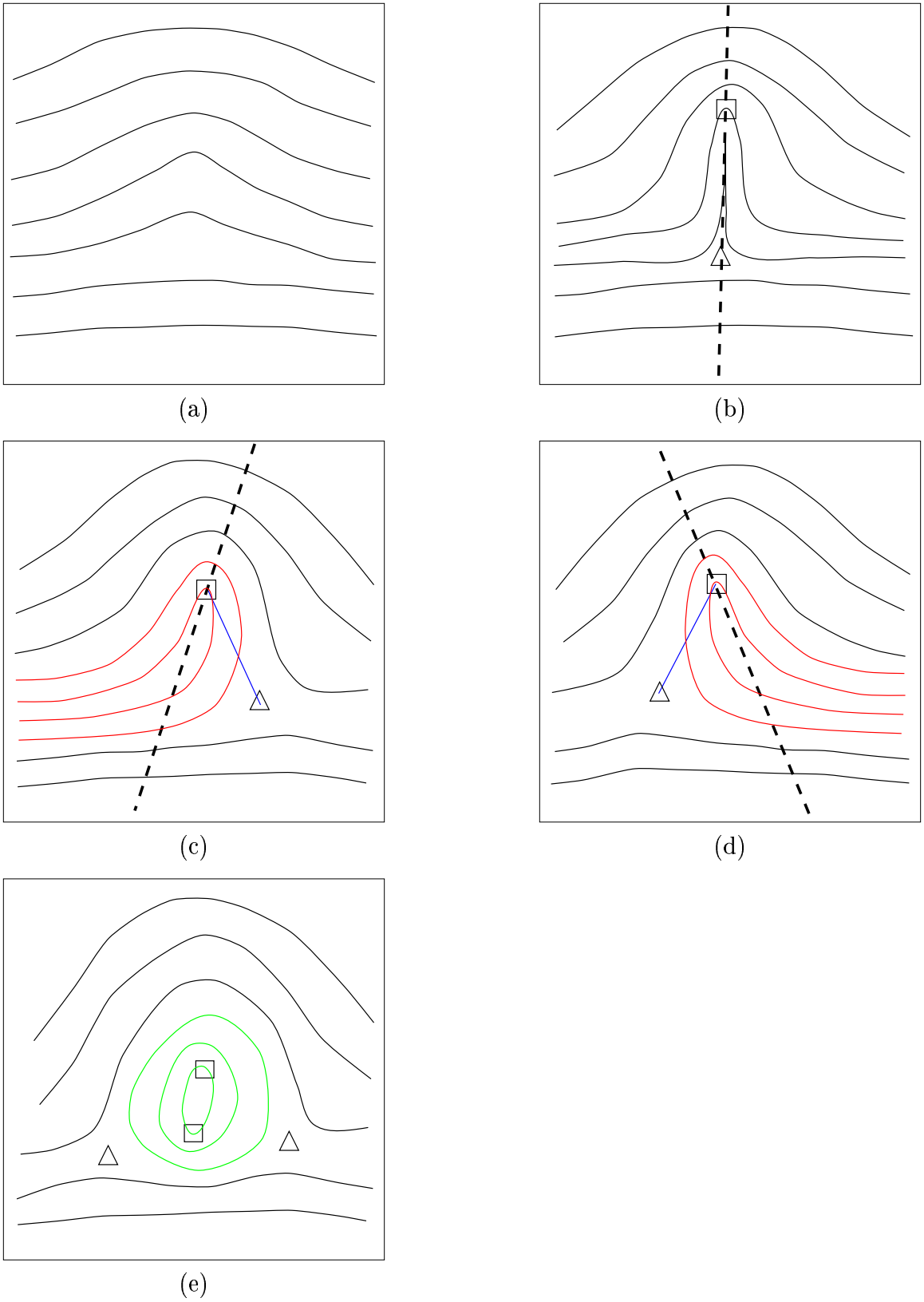


Figure 8.8: Fingerprint class prototypes; (a) arch; (b) tented arch; (c) left loop; (d) right loop; and (e) whorl; the dashed lines in (b), (c), and (d) are the symmetric axes.

cores and deltas detected from $\mathcal{O}'$, respectively; N_1 and N_2 be the number of type-1 recurring ridges and type-2 recurring ridges in $\mathcal{R}'$. The classification criteria used in our algorithm is depicted as follows:

1. *If* $(N_2 > 0)$ *and* $(N_c = 2)$ *and* $(N_d = 2)$, *then a whorl is identified.*
2. *If* $(N_1 = 0)$ *and* $(N_2 = 0)$ *and* $(N_c = 0)$ *and* $(N_d = 0)$, *then an arch is identified.*
3. *If* $(N_1 > 0)$ *and* $(N_2 = 0)$ *and* $(N_c = 1)$ *and* $(N_d = 1)$, *then classify the input using the core and delta assessment algorithm given below.*
4. *If* $(N_2 > T_2)$ *and* $(N_c > 0)$, *then a whorl is identified.*
5. *If* $(N_1 > T_1)$ *and* $(N_2 = 0)$ *and* $(N_c = 1)$ *then classify the input using the core and delta assessment algorithm.*
6. *If* $(N_c = 2)$, *then a whorl is identified.*
7. *If* $(N_c = 1)$ *and* $(N_d = 1)$, *then classify the input using the core and delta assessment algorithm.*
8. *If* $(N_1 > 0)$ *and* $(N_c = 1)$, *then classify the input using the core and delta assessment algorithm.*
9. *If* $(N_c = 0)$ *and* $(N_d = 0)$, *then an arch is identified.*
10. *If none of the above conditions is satisfied, then reject the fingerprint.*

The core and delta assessment algorithm is used to classify an one-core and one-delta fingerprint into one of the following categories: (*i*) left loop, (*ii*) right loop, and (*iii*) tented arch. It is depicted as follows:

1. *Estimate the symmetric axis which crosses the core in its local neighborhood.*
2. *Compute the angle,* α, *between the line segment from the core to the delta and the symmetric axis.*
3. *Compute the average angle difference,* β, *between the local ridge orientation on the line segment from the core to the delta and the orientation of the line segment.*
4. *Count the number of ridges,* γ, *that cross the line segment from the core to the delta.*
5. *If* $(\alpha < 10^o)$ *or* $(\beta < 15^o)$ *and* $(\gamma = 0)$, *then classify the input as a tented arch.*

6. *If the delta is on the right side of the axis, then classify the input as a left loop.*

7. *If the delta is on the left side of the axis, then classify the input as a right loop.*

8.4.2 Classification Scheme II

This classification scheme classifies fingerprints into *four* categories according to the number of singular points detected and their relative positions. Let N_c and N_d be the number of core and delta points detected from $\mathcal{O}'$, respectively; (C_{x1}, C_{y1}), (C_{x2}, C_{y2}), be the x and y coordinates of the cores, respectively; If $(N_c = 1)$, then $(C_{x2}, C_{y2}) = (-1, -1)$; if $(N_c = 0)$, then $(C_{x1}, C_{y1}) = (C_{x2}, C_{y2}) = (-1, -1)$; Let (D_{x1}, D_{y1}), (D_{x2}, D_{y2}), be the x and y coordinates of the deltas, respectively; if $(N_d = 1)$, then $(D_{x2}, D_{y2}) = (-1, -1)$; If $(N_d = 0)$, then $(D_{x1}, D_{y1}) = (D_{x2}, D_{y2}) = (-1, -1)$. The definition of the four categories is as follows:

1. *If $(N_c = 2)$, then the fingerprint is a class-1 fingerprint.*

2. *If $(N_c = 1)$ and $(N_d = 1)$ and $(d((C_{x1}, C_{y1}), (D_{x1}, D_{y1})) > S)$ and the delta is at the right side of the symmetric axis, where S is a threshold and $d((C_{x1}, C_{y1}), (D_{x1}, D_{y1}))$ is the distance between (C_{x1}, C_{y1}) and (D_{x1}, D_{y1}), then the fingerprint is a class-2 fingerprint.*

3. *If $(N_c = 1)$ and $(N_d = 1)$ and $(d((C_{x1}, C_{y1}), (D_{x1}, D_{y1})) > S)$ and the delta is at the left side of the symmetric axis, where S is a threshold and $d((C_{x1}, C_{y1}), (D_{x1}, D_{y1}))$ is the distance between (C_{x1}, C_{y1}) and (D_{x1}, D_{y1}), then the fingerprint is a class-3 fingerprint.*

4. *If none of the above conditions is satisfied, then the fingerprint is a class-4 fingerprint.*

In this classification scheme, a class-1 fingerprint corresponds to a whorl in the traditional classification scheme. Class-2 fingerprints and class-3 fingerprints are fingerprints of loop, and tented arch with the distance between core and delta being

rather large. Class-4 fingerprints consist arches and loops and tented arches with a small distance between core and delta.

8.5 Summary

Fingerprint classification provides an important indexing mechanism for automatic fingerprint identification. At a first glance, the fingerprint classification problem appears to be rather simple. But, because of large intraclass and small interclass variations in global pattern configuration and due to poor quality of input images, the desired accuracy of 1% error rate at 20% reject rate is very difficult to achieve. We have designed two fingerprint classification algorithms. One classifies input fingerprints into five categories according to the number of singular points detected, their relative positions, and presence of type-1 and type-2 recurring ridges. The other scheme classifies fingerprints into four categories. Since we invest a significant amount of effort in feature extraction so that the features are robust to interclass variations as well as poor quality of input images, the resulting classification algorithms are more robust to image quality.

Chapter 9

Experimental Results

The biometrics community is slow in establishing benchmarks for biometric systems [45]. Although, benchmark results on standard databases in themselves are useful only to a limited extent and may result in excessive tuning of the system parameters to "improve" the system performance[1], they constitute a good starting point for comparing the gross performance characteristics of the systems.

No metric is sufficiently adequate to give a reliable and convincing indication of the identification accuracy of a biometric system. In principle, we can use the false (impostor) acceptance rate (FAR), the false (genuine individual) reject rate (FRR) and the equal error rate (EER)[2] to indicate the identification accuracy of a biometric system [106, 43, 44]. In practice, these performance metrics can only be estimated from empirical data and the estimates of the performance are very data dependent. Therefore, they are meaningful only for a specific database in a

[1]Several additional techniques like data-sequestering [114] and third party benchmarking [19] may also help in obtaining fairer performance results.

[2]Equal error rate is defined as the value where FAR and FRR are equal.

specific test environment. For example, the manufacturer of a particular biometric system claimed that the system had an FRR of 0.3% and an FAR of 0.1%. An independent test by the Sandia National Lab. found that the same system had an FRR of 25% with an unknown FAR [75]! In order to provide a more reliable assessment of a biometric system, some more descriptive performance measures are necessary. A receiver operating curve provides an empirical assessment of the system performance at different operating points which is more informative than FAR and FRR. The statistical metric d' gives an indication of the separation between the genuine distribution and impostor distribution [44]. It is defined as the difference between the means of the genuine distribution and impostor distribution divided by a conjoint measure of their standard deviations [44]:

$$d' = \frac{\|M_{impostor} - M_{genuine}\|}{\sqrt{(SD^2_{impostor} + SD^2_{genuine})/2}}, \tag{9.1}$$

where $M_{genuine}$, $SD_{genuine}$, $M_{impostor}$, and $SD_{impostor}$ are the means and standard deviations of the genuine distribution and impostor distribution, respectively. Like FAR, FRR, and EER, both ROC and d' also depend heavily on test data and test environments. For such performance metrics to be able to precisely generalize to the entire population of interest, the test data should (i) be large enough to represent the population and (ii) contain enough samples from each category of the population [44]. To obtain fair test results, enough samples should be available, the samples should be representative of the population, and adequately represent all the categories (impostors and genuine). Further, irrespective of the performance measure, error bounds

that indicate the confidence of the estimates are valuable for understanding the significance of the test results.

9.1 Test Databases

The MSU fingerprint database contains 10 images (640×480) per finger from 150 individuals for a total of 1,500 fingerprint images, which were captured with a scanner manufactured by Digital Biometrics. When these fingerprint images were captured, no restrictions on the position and orientation of fingers were imposed. The captured fingerprint images vary in quality. Figure 9.1 shows some of the fingerprint images in our database. Approximately 90% of the fingerprint images in our database are of reasonable quality similar to those shown in Figure 9.1, while about 10% of the fingerprint images in our database are not of good quality (Figure 9.2), which are mainly due to large creases and smudges in ridges and dryness of the impressed finger.

A portion of the NIST 9 fingerprint database is also used in our experiments. NIST 9 fingerprint database contains 1,350 mated fingerprint card pairs (image size is 832×768) that approximate a natural distribution of the National Crime and Information Center (NCIC) fingerprint classes (examples of fingerprints in the NIST 9 database are shown in Figure 9.3) [149]. It is divided into multiple volumes. Each volume has 3 CD's. Each CD contains 900 images of card type 1 and 900 images of card type 2. Fingerprints on card type 1 were scanned using a rolled method and fingerprints on card type 2 were scanned using a live-scan method. The fingerprint images in

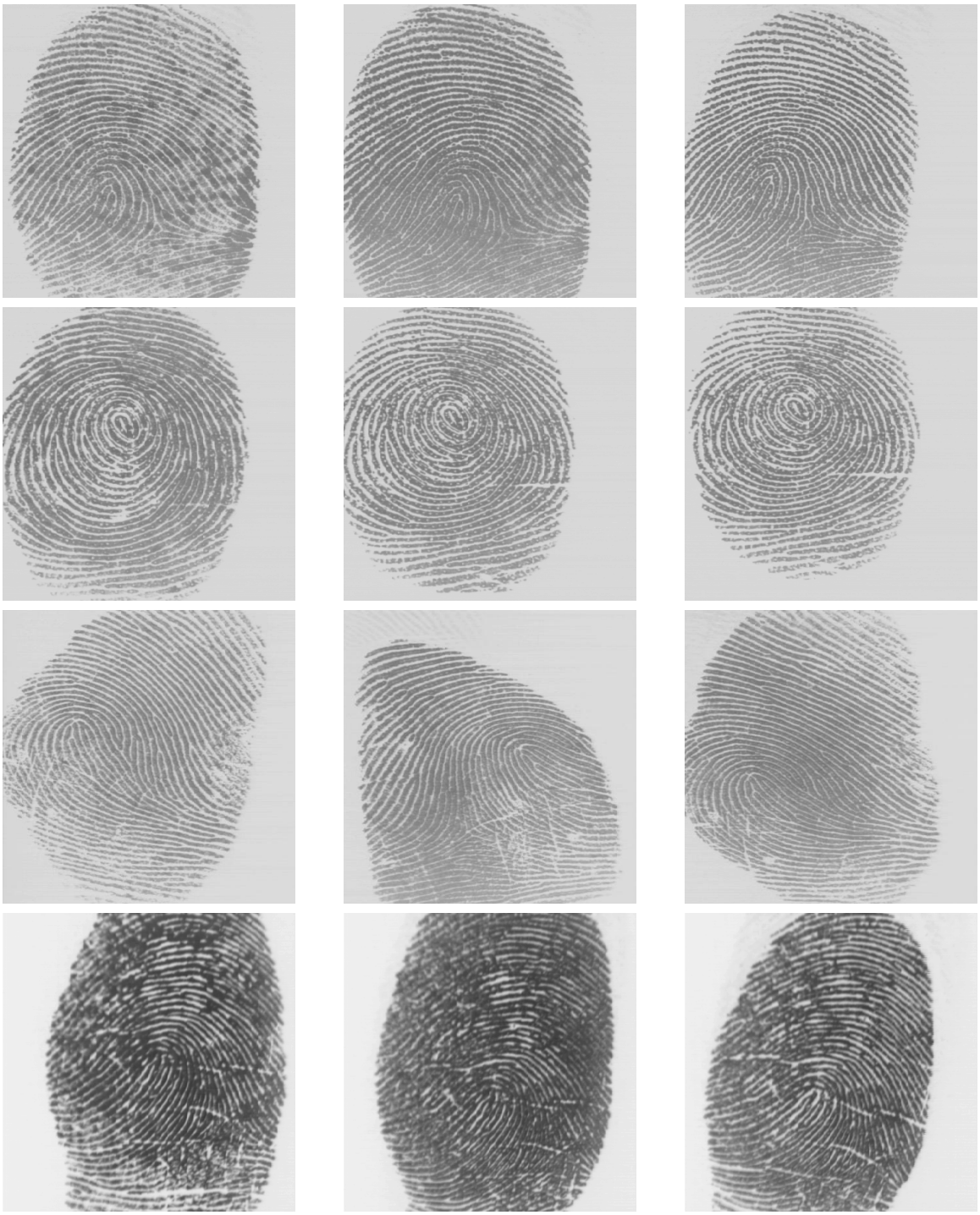


Figure 9.1: Fingerprint images captured with a scanner manufactured by Digital Biometrics; the size of these images is 640×480; images in each row are from the same finger.

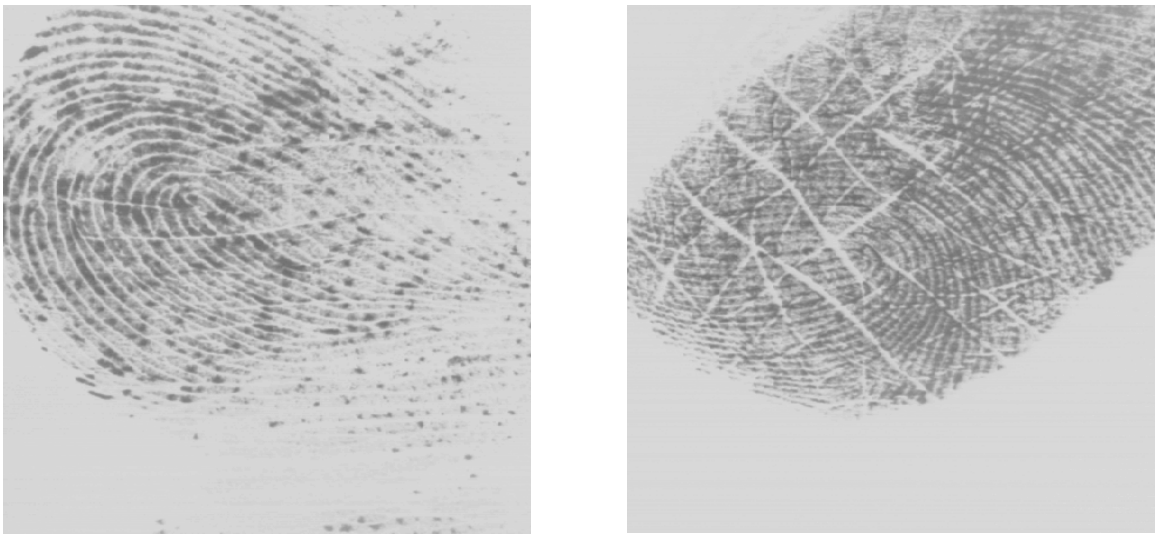


Figure 9.2: Fingerprint images of poor quality.

NIST 9 database are more challenging compared to the live-scan fingerprint images for a number of reasons, including: (*i*) the NIST 9 fingerprints are a combination of dabs and rolled impressions; large discrepancy between the number of minutiae in test and reference template inherently skews the matching score normalization; (*ii*) a large number of NIST 9 images are of much poorer image quality than a typical live-scan fingerprint image; (*iii*) NIST 9 images often contain extraneous objects like handwritten characters and other artifacts common to inked fingerprints. Although only one-half of the fingerprint images in NIST 9 fingerprint database are live-scan images and there exists a large distortion between a rolled fingerprint and a live-scan fingerprint, we can still use this database to generate some statistics and comparative performance numbers for our matching algorithm.

NIST 4 fingerprint database contains 4,000 images (image size is 512×480) taken from 2,000 different fingers, two images per finger. Five fingerprint classes are defined: (*i*) Arch, (*ii*) Tented arch, (*iii*) Left Loop, (*iv*) Right Loop, and (*v*) Whorl.

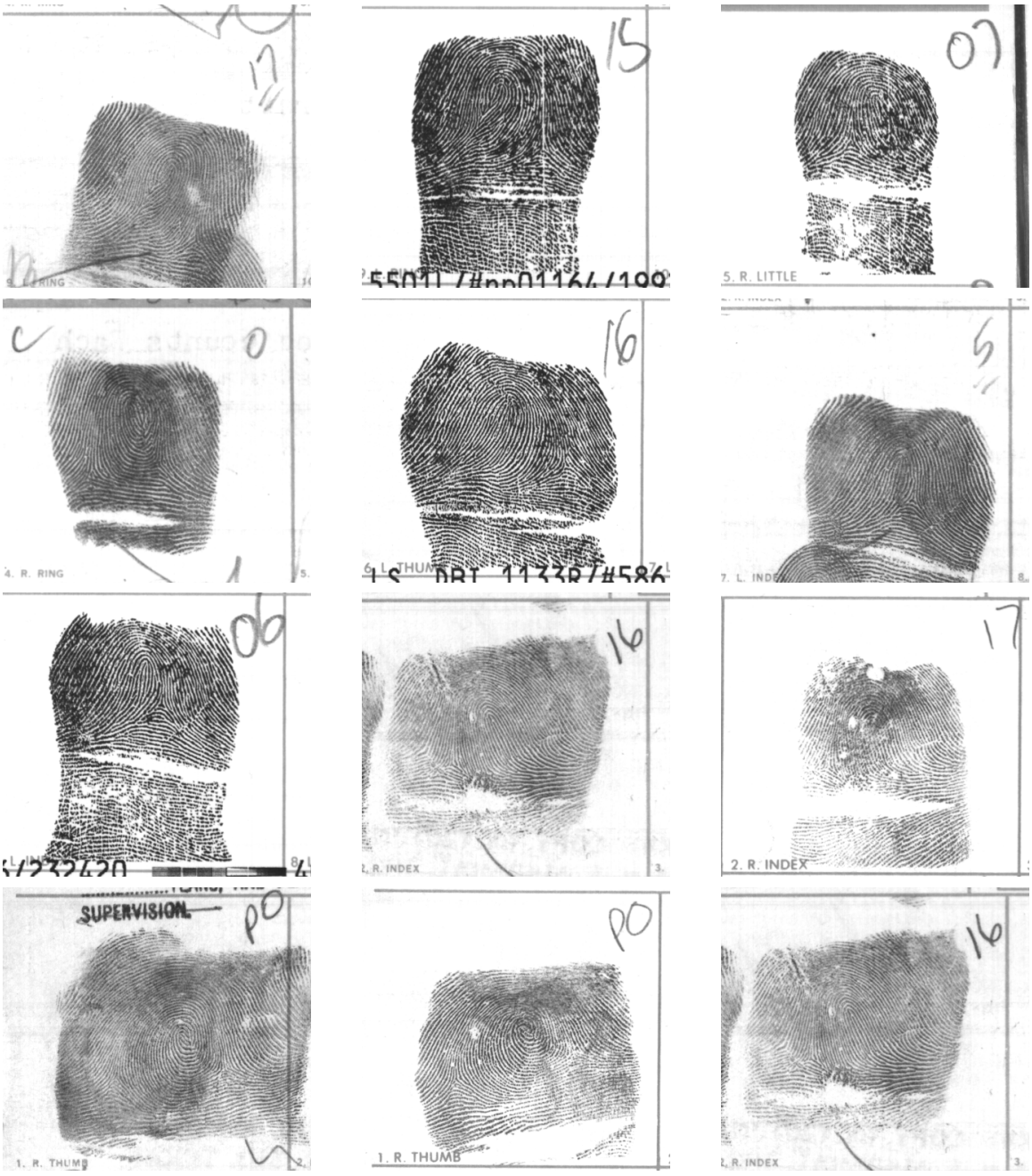


Figure 9.3: Examples of fingerprints in the NIST 9 database; the size of these images is 832×768.

Fingerprints are uniformly distributed among these five classes. NIST 4 database is compiled for evaluating the performance of a fingerprint classification algorithm. Examples of fingerprints in the NIST 4 database are shown in Figure 9.4 [150].

A database from IBM which contains 1,044 live-scan fingerprint images (varying image size) was also used to test the performance of our classification algorithm. Fingerprints in the IBM database are classified manually by fingerprint experts into: (*i*) Arch, (*ii*) Tented arch, (*iii*) Left loop, (*iv*) Right loop, (*v*) Whorl, (*vi*) Twin, and (*vii*) Composite. Examples of fingerprints in the IBM database are shown in Figure 9.5.

A composite public domain face database was used in the performance evaluation of the identification system. The face database contains a total of 1,132 images of 86 individuals (examples of images in this database are shown in Figure 9.6); 400 images of 40 individuals with 10 images per individual are from the Olivetti Research Lab., 300 images of 30 individuals with 10 images per individual are from the University of Bern, and 432 images of 16 individuals with 27 images per individual are from MIT Media Lab. The images were re-sampled from the original sizes to a fixed size of 92×112 and normalized to zero mean.

9.2 Feature Extraction Performance

It is very difficult to independently assess the performance of feature extraction algorithms. Accuracy of the extracted minutiae was subjectively confirmed in two ways. Visual inspection of a large number of typical minutiae extraction results showed that

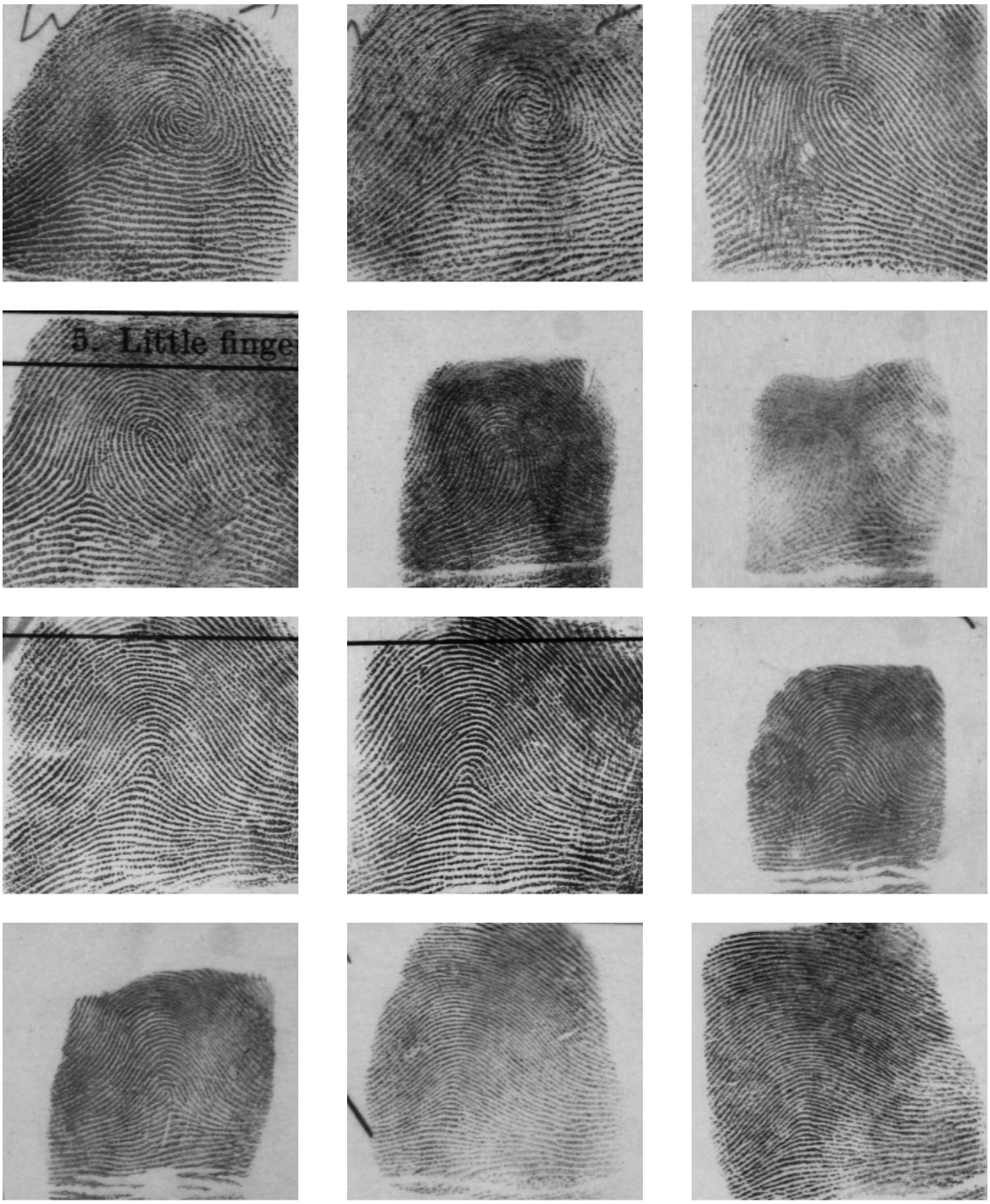


Figure 9.4: Examples of fingerprints in the NIST 4 database; the size of these images is 512×480.

Figure 9.5: Examples of fingerprints in the IBM database.

Figure 9.6: Examples of faces in the composite database.

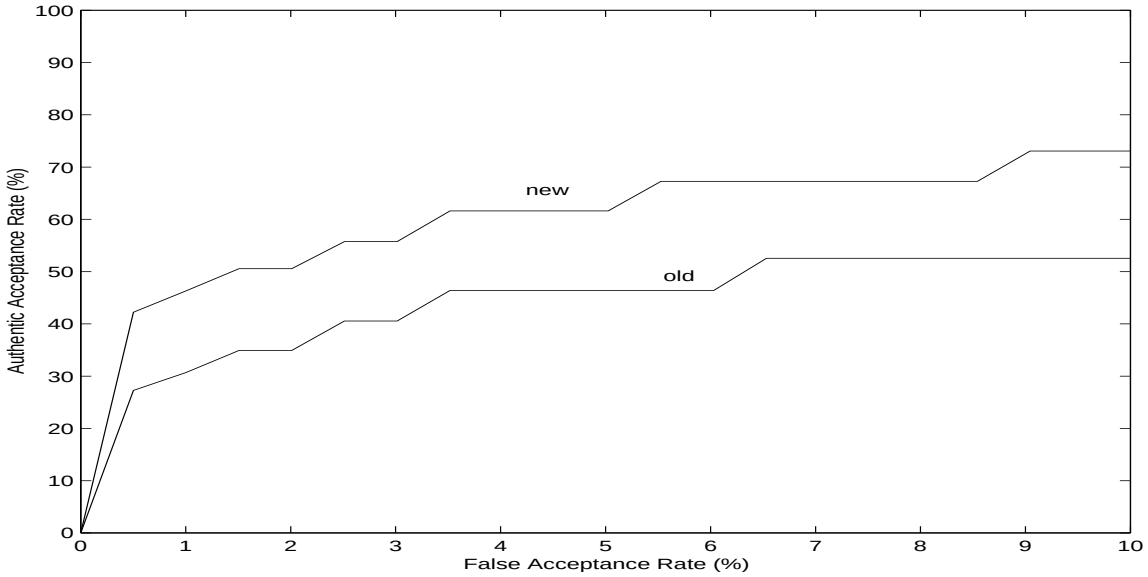


Figure 9.7: Receiver Operating Curves; the vertical axis is (1-FRR); the ROC shows the improvement in verification performance of the new minutiae extraction algorithm in contrast to the algorithm in [120].

our algorithm rarely missed minutiae in reasonable quality fingerprint images.

We have compared the performance of our feature extraction algorithm with that of the feature extraction algorithm in [120]. The premise underlying this experiment is that given an identical matcher, the accuracy of the system indicates the performance of the feature extraction algorithm. We extracted fingerprint representations from a sample set of fingerprint images using our feature extraction algorithm. The verification accuracy was estimated using a Hough-transform based matcher [121] by performing an "all against all" verification test to obtain distributions of match and mismatch scores. The same test was also performed on the features extracted from our previous feature extraction algorithm [120]. The ROCs resulting from these two experiments are shown in Figure 9.7. The CPU time required by our new feature extraction algorithm to process a 640×480 fingerprint image is, on an average, 1.1

seconds on a UltraSPARC workstation whereas the CPU time for the old algorithm is 16.1 seconds. Thus, we can see that the new minutiae extraction algorithm results in a significant improvement in the overall accuracy as well as the speed of the system. Note that the above accuracy improvement is context dependent, *i.e.*, it reveals the accuracy improvement when the Hough-transform based matcher is used. However, since the accuracies of both the matchers depend solely on the correctness of the extracted minutiae, the performance of our matcher should also improve with the new minutiae extraction algorithm.

9.3 Fingerprint Enhancement Performance

The purpose of a fingerprint enhancement algorithm is to improve the quality of input fingerprint images and make them more suitable for the minutiae extraction module. Therefore, the ultimate criterion for evaluating such an enhancement algorithm is the amount of performance improvement when the algorithm is applied to the noisy fingerprint images. In order to evaluate the performance of our fingerprint enhancement algorithm, we have conducted two experiments using the verification system on a subset of the MSU fingerprint database which consists of 700 images of 70 individuals.

In the first experiment, the fingerprint enhancement algorithm was not applied. Each fingerprint image in the database was directly matched against the other fingerprint images in the database. In the second experiment, our fingerprint enhancement algorithm was first applied to each fingerprint image in the database. Then, the ver-

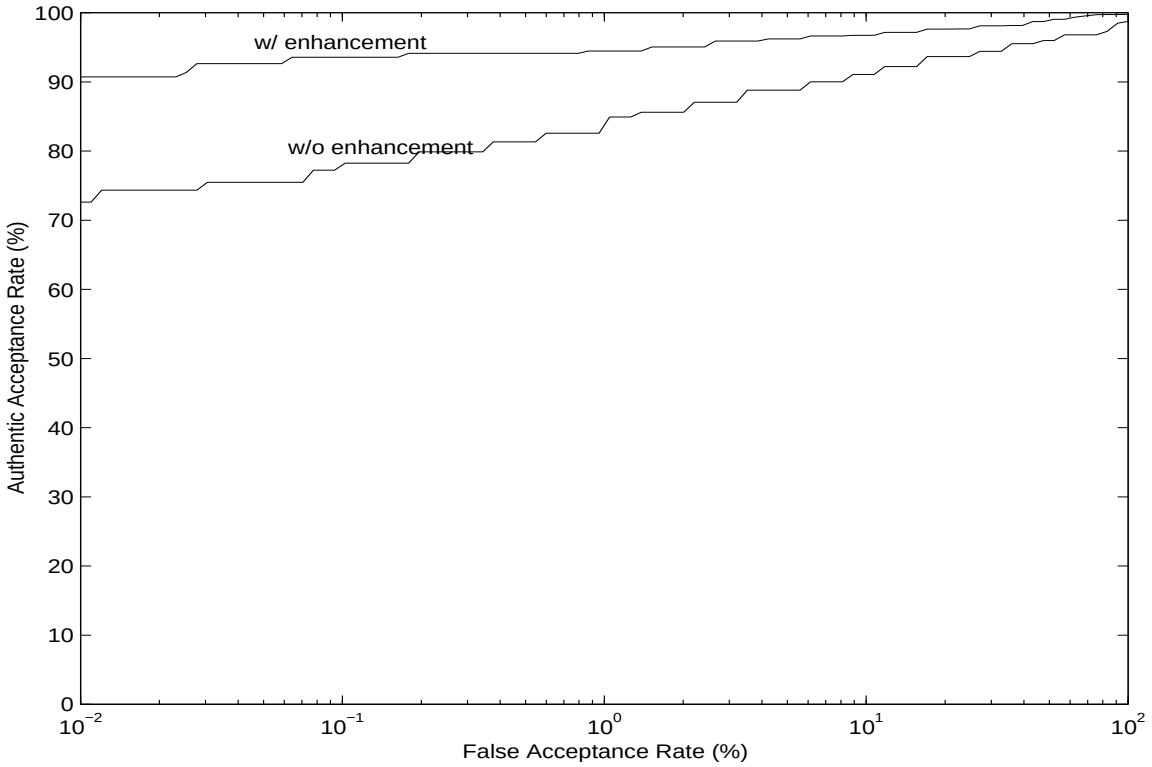


Figure 9.8: Receiver Operating Curves; the ROC shows the improvement in verification performance of the enhancement algorithm.

ification is conducted on the enhanced fingerprint images. The ROCs resulting from these two experiments are shown in Figure 9.8. From these experimental results, we can observe that the performance of the online fingerprint verification system is significantly improved when our fingerprint enhancement algorithm is applied to the input fingerprint images. In particular, using the enhancement algorithm has substantially reduced the reject rate while maintaining essentially the same recognition rate.

9.4 System Performance

We first evaluated the matching scores of correct and incorrect matches and then evaluated the performance of the verification system in conducting identity authentication

and the performance of the identification system in conducting personal identification.

9.4.1 Matching Scores

(a) (b) (c)

Figure 9.9: Fingerprint images from the same finger.

In test 1, each of the first 1,500 fingerprints in the MSU fingerprint database was matched with all the other fingerprints in the database. A matching was labeled correct if the matched fingerprint was from the same finger, and incorrect otherwise. A total of 2,248,500 $(1,500 \times 1,499)$ matchings were performed. The distributions of correct and incorrect matching scores are shown in Figure 9.12(a). In test 2, each of the 900 fingerprints of card type 1 in the NIST 9 (CD No. 1) was matched with all the 900 fingerprints of card type 2. A matching was labeled correct if a matched fingerprint was from the same finger. A total of 810,000 (900×900) matchings were performed on this database (to our knowledge, no comparative results are available on NIST 9 database). The distributions of correct and incorrect matching scores are shown in Figure 9.12(b). Table 9.1 lists the d' values in addition to the mean and standard deviation of correct and incorrect matching scores. The large variance of correct matching scores is mainly due to different number of detected minutiae,

Database	d'	Mean (correct)	Standard Deviation (correct)	Mean (incorrect)	Standard Deviation (incorrect)
MSU	2.26	23.46	13.59	1.56	0.71
NIST-9	2.01	18.76	11.22	2.39	0.83

Table 9.1: d' and mean and standard deviation of the correct and incorrect matching scores.

quality of acquired fingerprint images, and fingerprint distortion. For example, the fingerprint images shown in Figures 9.9(a) and (b) are captured from the same finger. However, only a small region of interest is common to these two fingerprint images (approximately 30%). Obviously, it is very difficult to make a highly confident decision based only on the limited number of minutiae appearing in the region of interest which are common to both the fingerprints. In practice, such a problem can be solved by requiring that each input fingerprint image should have a sufficient amount of overlapping region of interest with its stored template(s). Figure 9.10 shows a mated pair in the MSU database that has a relatively low matching score (8). Figure 9.11 shows a pair of fingerprints from different fingers in the MSU database that has a relatively high matching score (9).

9.4.2 Authentication Test

In test 1, for each individual, we randomly selected 3 fingerprint images which passed the quality checking as the template minutiae patterns for the individual and inserted them into the system database. The major reason why we use 3 fingerprint templates

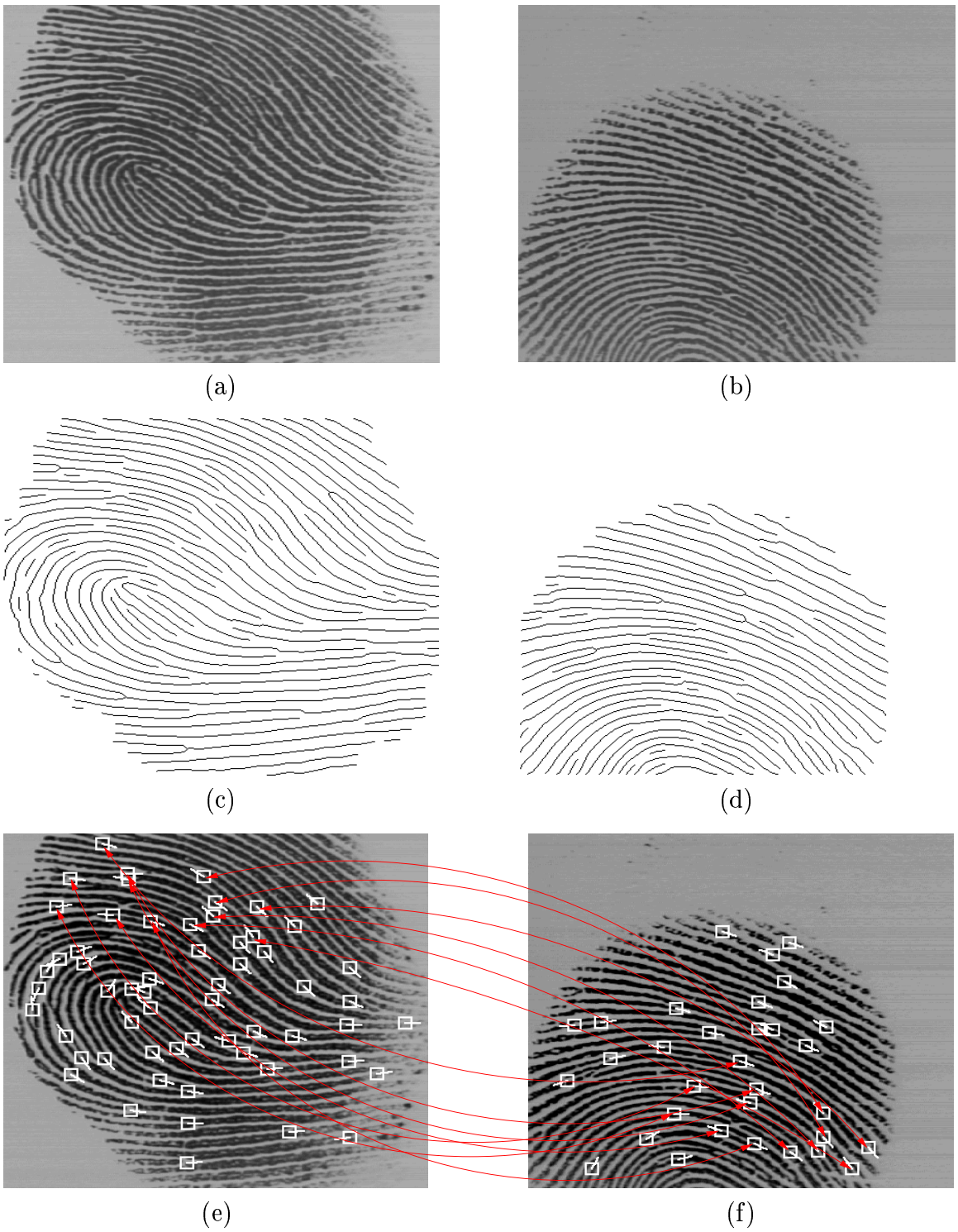


Figure 9.10: A mated pair in the MSU database that has a relatively low matching score: (a) and (b) fingerprint images from the same finger; (c) and (d) thinned ridge maps; (e) and (f) extracted minutiae superimposed on the input images and the corresponding minutiae pairs established using our matching algorithm.

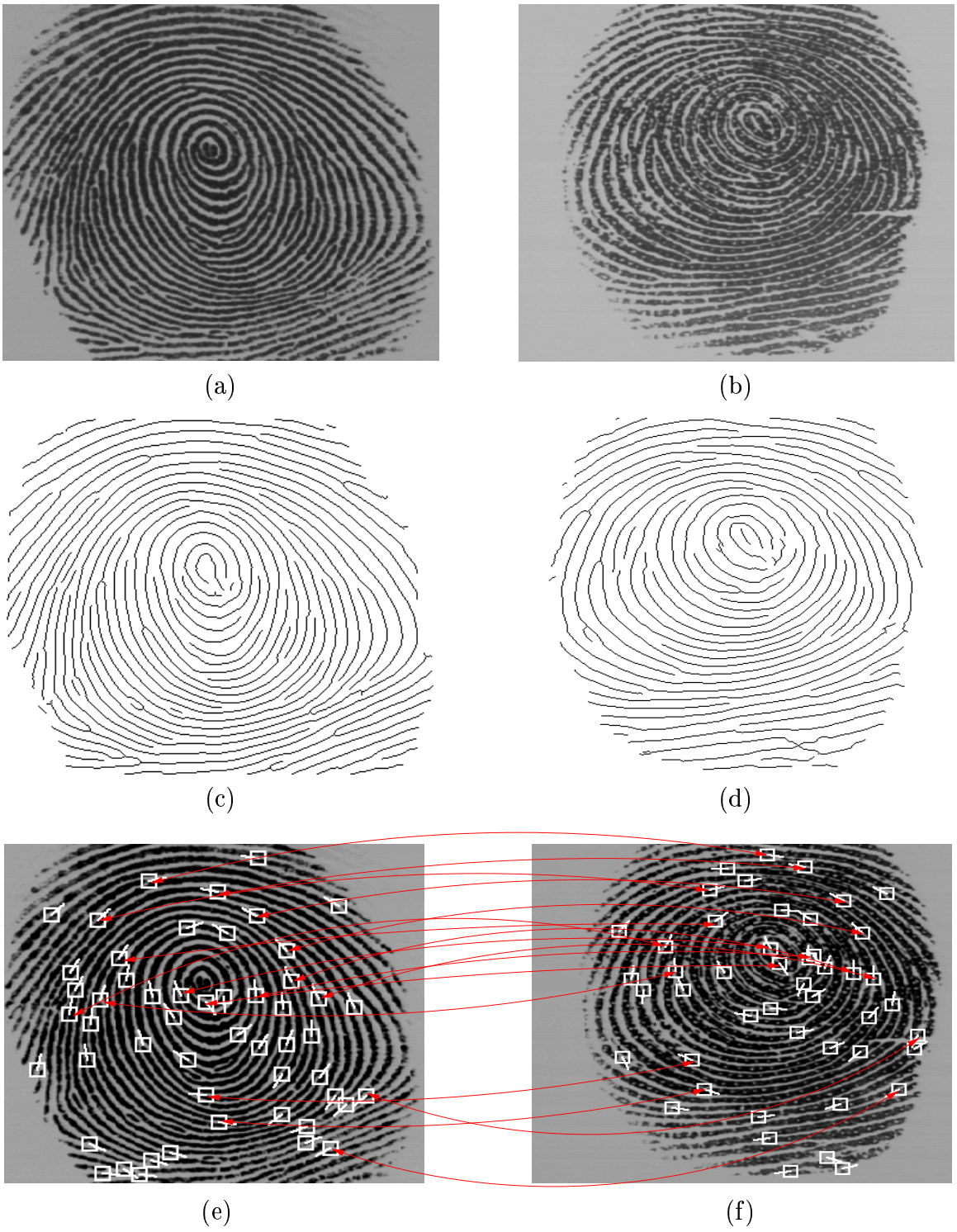
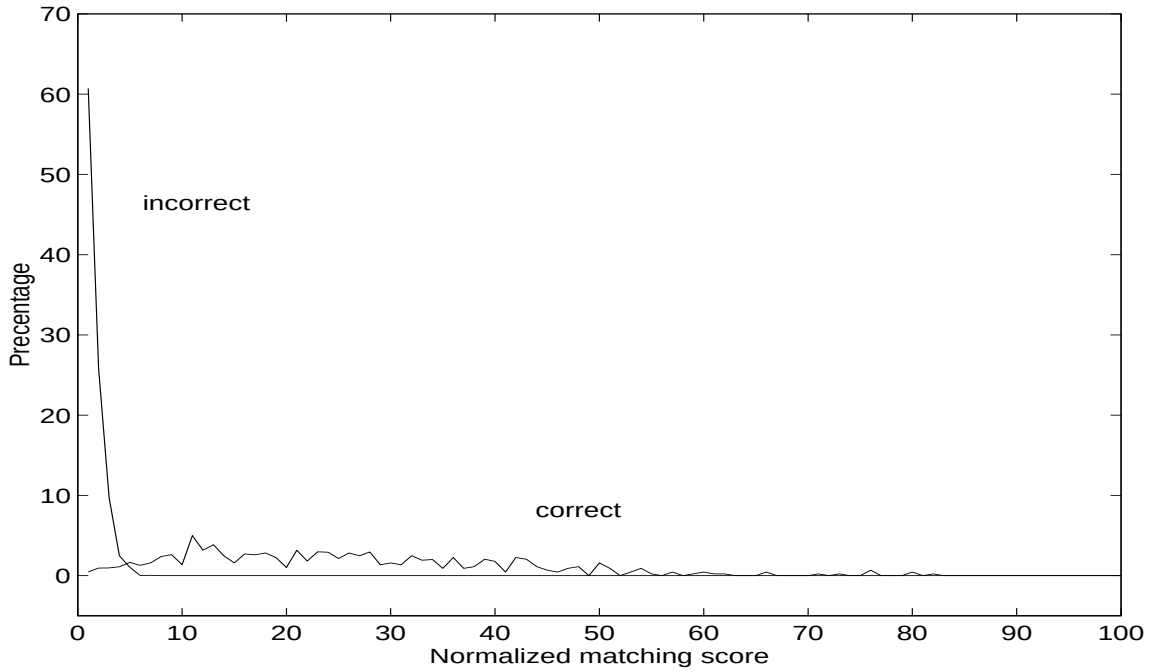
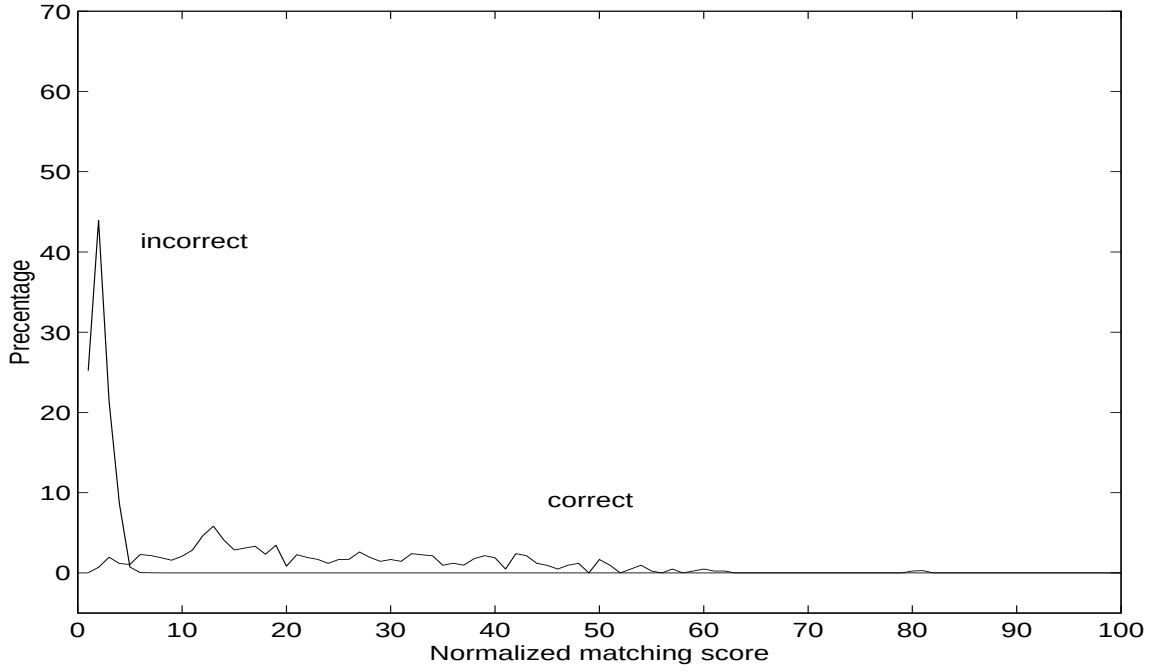


Figure 9.11: A pair of fingerprints from different fingers in the MSU database that have a relatively high matching score: (a) and (b) fingerprint images from different fingers; (c) and (d) thinned ridge maps; (e) and (f) extracted minutiae superimposed on the input images and the corresponding minutiae pairs established using our matching algorithm.



(a)



(b)

Figure 9.12: Distributions of correct and incorrect matching scores; vertical axis represents distribution of matching scores in percentage; (a) MSU database; (b) NIST 9 (CD No. 1).

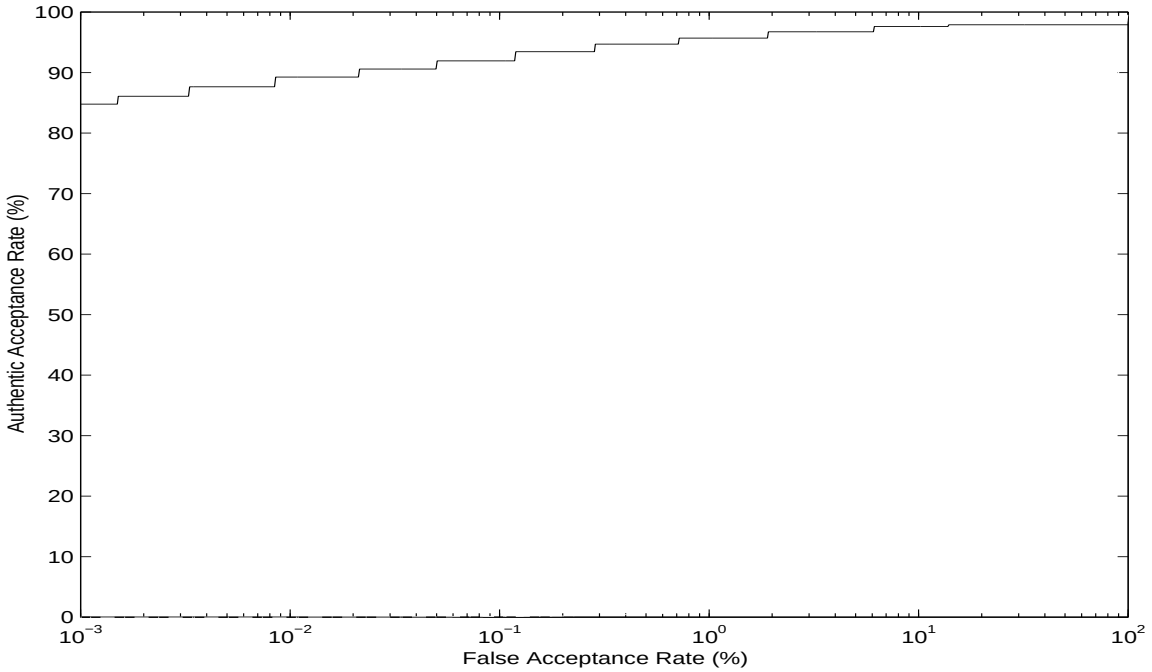
is that a significant number of acquired fingerprint images from the same finger in the MSU database do not have a sufficient amount of common region of interest due to the unrestricted acquisition process. Even if two images are of good quality, if they share a small common region of interest, it is unlikely that the matching algorithm can establish a sufficient number of corresponding minutiae pairs to reach a correct decision. Using more than one template is a simple solution although it may result in a higher FAR. The remaining 1,050 (150×7) fingerprint images were used as input fingerprints to test the performance of the system. An identity is established if at least one of the 3 matching scores is above a certain threshold value. Otherwise, the input fingerprint is rejected as an impostor. In test 2, we used 798 out of the 900 fingerprints of card type 1 in NIST 9 database (CD. No. 1) as templates, which pass the quality checking. The 900 fingerprints of card type 2 were used as input fingerprints. An identity is established if the matching score is above a certain threshold value. The false acceptance rates and false reject rates with different threshold values on the matching score are shown in Table 9.2, which are obtained based on 157,500 ($150 \times 1{,}050$) matches for test 1 and 718,200 (798×900) matches on test 2. Since the matching scores are discretized with a large sampling interval, only an approximate EER can be obtained by averaging the most similar FAR and FRR. The EER was approximated to be 3.07% in test 1 and 2.69% in test 2. The ROCs of the two tests are shown in Figure 9.13. In each ROC, authentic acceptance rate (the percentage of a genuine individual being accepted) is plotted against the FAR. Each point on the curve corresponds to a decision criterion. In the ideal case, if the genuine distribution and the impostor distribution are disjoint, *i.e.* each genuine individual is accepted

Threshold Value	False Acceptance Rate (MSU)	False Reject Rate (MSU)	False Acceptance Rate (NIST 9)	False Reject Rate (NIST 9)
7	0.07%	7.1%	0.073%	12.4%
8	0.02%	9.4%	0.023%	14.6%
9	0.01%	12.5%	0.012%	16.9%
10	0	14.3%	0.003%	19.5%

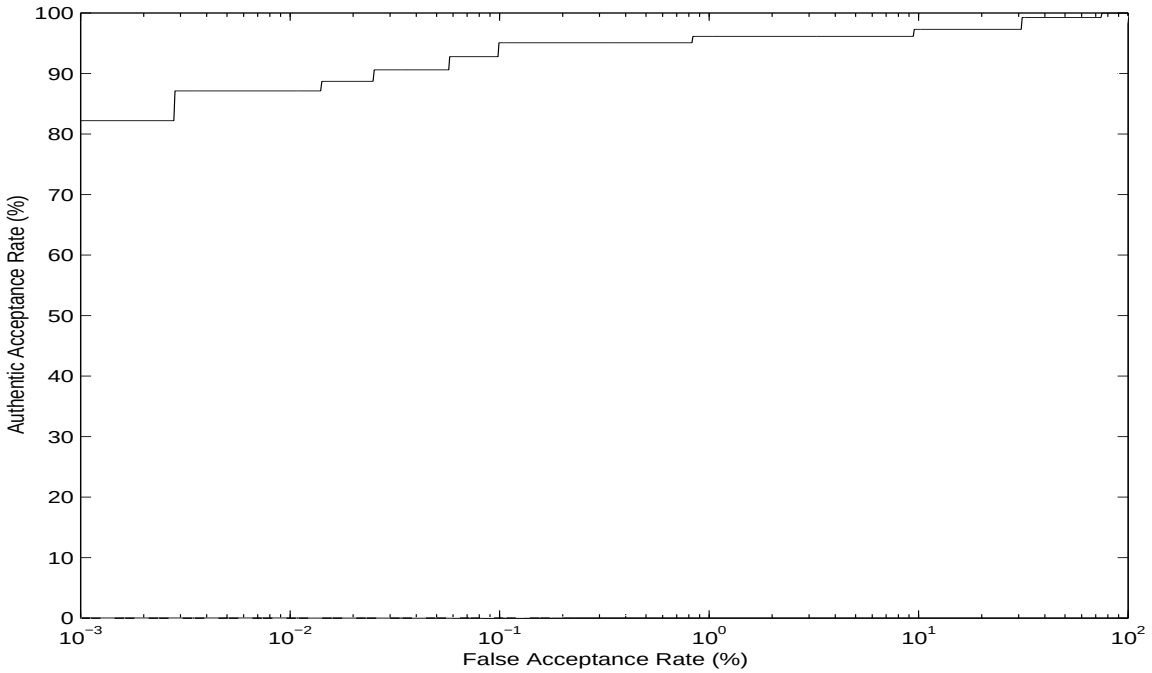
Table 9.2: False acceptance and false reject rates on test sets with different threshold values.

and each impostor is rejected correctly, then the ROC is a horizontal line segment hovering at the authentic acceptance rate of 100%. On the other hand, if the genuine distribution and the impostor distribution are exactly the same, then the ROC is a 45° line segment with one end point at the origin. In this case, decisions can only be made by a random choice. In practice, a ROC is a curve between these two extremes. The closer the ROC is to the upper boundary, the better the system performance. The numbers shown in Table 9.2 are the performance measures of our verification algorithm. They should not be treated as the ultimate performance numbers of the system. In practice, a number of techniques can be employed to ensure a sufficient amount of common region of interest in fingerprint images and good image quality and to restrict the distortion of input images, which can substantially decrease both the FAR and FRR.

In order for an automatic identity authentication system to be acceptable in practice, the response time of the system needs to be within a few seconds. Table 9.3 shows that our implemented system does meet the practical response time requirement.



(a)



(b)

Figure 9.13: Receiver Operating Curves; (a) MSU database; (b) NIST 9 (CD No. 1).

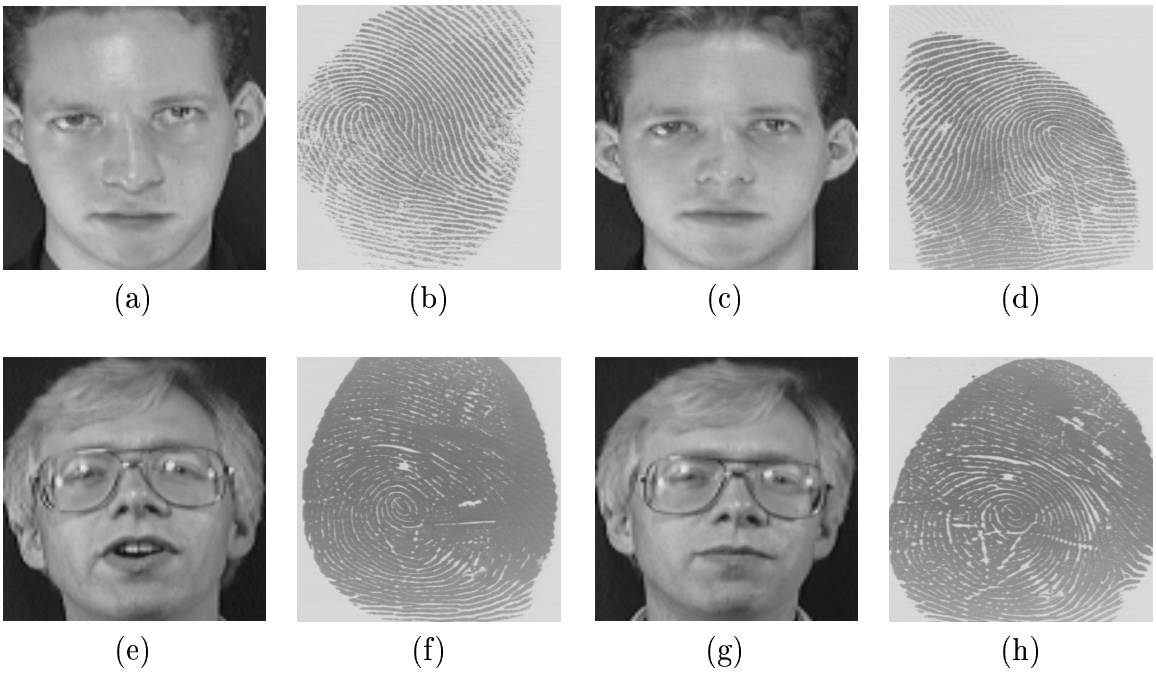
Minutiae Extraction (seconds)	Minutiae Matching (seconds)	Total (seconds)
1.1	0.3	1.4

Table 9.3: Average CPU time for minutiae extraction and matching on a Sun ULTRA 1 workstation.

9.4.3 Identification Test

We randomly assigned each of the remaining 86 individuals in the MSU fingerprint database to an individual in the face database (see Figure 9.14 for some examples). Since the DFFS between two different individuals is statistically independent of the fingerprint matching scores between the two individuals, such a random assignment of a face to a fingerprint is admissible. One fingerprint for each individual is randomly selected as the template for the individual. To simulate the practical identification scenario, each of the remaining 590 faces was paired with a fingerprint to produce a test pair. In the test, with a pre-specified confidence value (FAR), for each of the 590 fingerprint and face pairs, the top 5 matches are retrieved using face recognition. Then fingerprint verification is applied to each of the top 5 matches and a final decision is made by decision fusion.

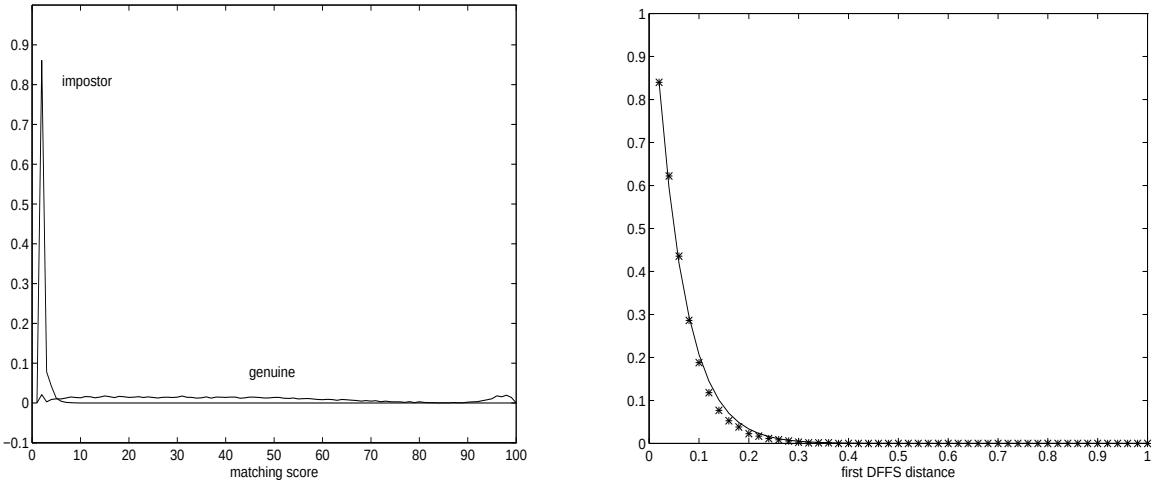
We randomly selected 640 fingerprints of 64 individuals as the training set and the remaining as the test set. The mean and variance of the impostor distribution (Figure 9.15 (a)) were estimated to be 0.70 and 0.64 from the 403,200 (640×630) impostor



(a) (b) (c) (d)

(e) (f) (g) (h)

Figure 9.14: Face and fingerprint pairs; the face images (92×112) are from the Olivetti Research Lab.; the fingerprint images (640×480) are captured with a scanner manufactured by Digital Biometrics.



(a) (b)

Figure 9.15: Impostor distributions; (a) impostor distribution for fingerprint verification; (b) the impostor distribution for face recognition at rank no. 1, where the stars (*) represent empirical data and the solid curve represents the fitting result.

FAR	False Reject Rate (FRR)		
	Face	Fingerprint	Integration
1%	15.8%	3.9%	1.8%
0.1%	42.2%	6.9%	4.4%
0.01%	61.2%	10.6%	6.6%
0.001%	64.1%	14.9%	9.8%

Table 9.4: False reject rates (FRR) on the test set with different values of FAR.

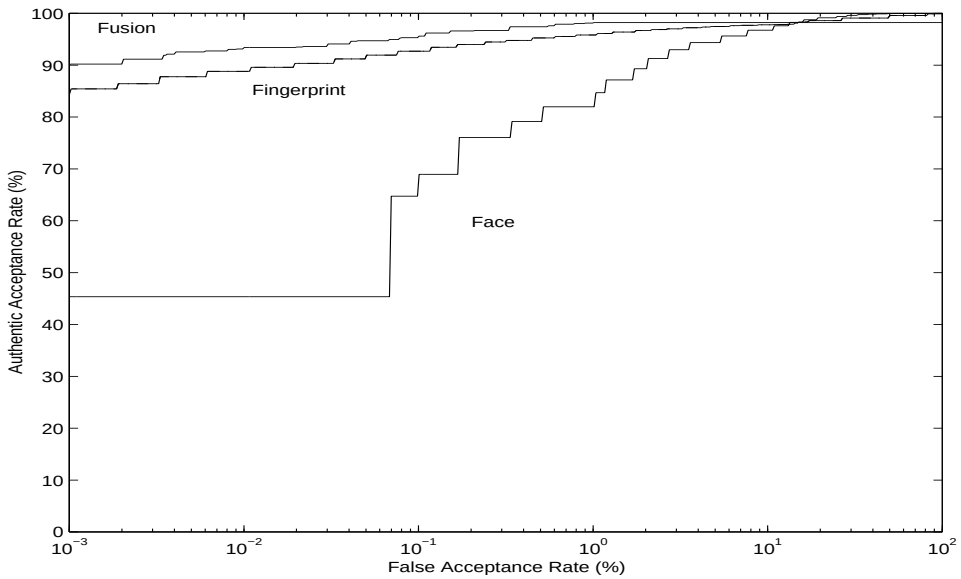


Figure 9.16: Receiver Operating Curves; the vertical axis is (1-FRR).

matching scores of "all against all" verification test by fitting the probability model described in Section 4.1. A total of 542 face images were used as training samples. Since variations in position, orientation, scale, and illumination exist in the face database, the 542 training samples were selected such that the training set contained several representative views. Eigenfaces were estimated from the 542 training samples and the first 64 eigenfaces were used. The top 5 impostor distributions were approximated. Figure 9.15 (b) shows the impostor distribution at rank no. 1.

The pre-specified FAR for a biometric system is usually very small (< 0.0001). In order to demonstrate that the biometric system does meet such a specification, a large

Face Recognition (seconds)	Fingerprint Verification (seconds)	Total (seconds)
0.9	2.1	3.0

Table 9.5: Average CPU time for one test on a Sun UltraSPARC 1 workstation.

set of representative samples is needed. Unfortunately, obtaining such a large number of test samples is both expensive and time consuming. In our test, we re-use faces by different assignment practices. In order to diminish the possible gain in performance due to such a re-use schema, we multiplied the estimated impostor distribution for face recognition by a constant of 1.25, which is sufficiently conservative. On the other hand, fingerprint verification operates in the one-to-one verification mode, so different assignments may be deemed as different impostor forgeries. Therefore, the test results using such a random assignment schema are able to reasonably estimate the underlying performance numbers. In our test, 1000 different assignments were tried. A total of 590,000 (590×1000) face and fingerprint test pairs were generated and tested. The FRRs of our system with respect to different pre-specified FARs, as well as the FRRs using only fingerprints or faces are listed in Table 9.4. Note that the FRRs in integration column include the error rate (1.8%) of genuine individuals not present in the top 5 matches. The receiver operating curves are plotted in Figure 9.16, in which the authentic acceptance rate (the percentage of genuine individuals being accepted, *i.e.*, $1 - FRR$) is plotted against FAR. We can conclude from these test results that integration of fingerprints and faces does result in a significantly better recognition performance. Table 9.5 shows that our implemented system does meet the response time requirement.

9.5 Classification Performance

We have tested our fingerprint classification algorithms on (*i*) NIST 4 fingerprint database, (*ii*) NIST 9 fingerprint database, and (*iii*) a set of live-scan fingerprint images from IBM.

9.5.1 Classification Scheme I

We first present the performance of our classification algorithm on the NIST 4 database. The five-class error rate in classifying these 4,000 fingerprints is 12.5%. The confusion matrix is given in Table 9.6; numbers shown in bold font are correct classifications. Since a number of fingerprints in NIST 4 database are labeled as belonging to two different classes, each row of the confusion matrix in Table 9.6 does not sum up to 800. For the five-class problem, most of the classification errors are due to misclassifying a tented arch as an arch. By combining these two arch categories into a single class, the error rate drops to 7.7%. Besides the tented arch-arch errors, the other errors mainly come from misclassifications between arch/tented arch and loops and due to poor image quality. Four examples of misclassified fingerprints are shown in Figure 9.17. A lower error rate can be achieved by adding the reject option, which is based on the quality index of the input image. The error rates corresponding to different reject rates are listed in Table 9.8.

When testing on the NIST 9 database, if a fingerprint is not an arch or tented arch or loop then classify it as a whorl, since our classification scheme only classifies fingerprints into five classes. The five-class error rate in classifying the 5,400 finger-

True Class	Assigned Class				
	Arch	Tented Arch	Left Loop	Right Loop	Whorl
Arch	**885**	13	10	11	0
Tented Arch	179	**384**	54	14	5
Left Loop	31	27	**755**	3	20
Right Loop	30	47	3	**717**	16
Whorl	6	1	15	15	**759**

Table 9.6: Five-class classification results on NIST 4 database.

True Class	Assigned Class			
	Arch	Left Loop	Right Loop	Whorl
Arch	**1461**	64	25	5
Left Loop	58	**755**	3	20
Right Loop	77	3	**717**	16
Whorl	7	15	15	**759**

Table 9.7: Four-class classification results on NIST 4 database.

Reject rate	0%	5%	10%	20%
5-class Error	12.5%	11.6%	10.1%	7.5%
4-class Error	7.7%	6.6%	5.1%	2.4%

Table 9.8: Error-reject tradeoff.

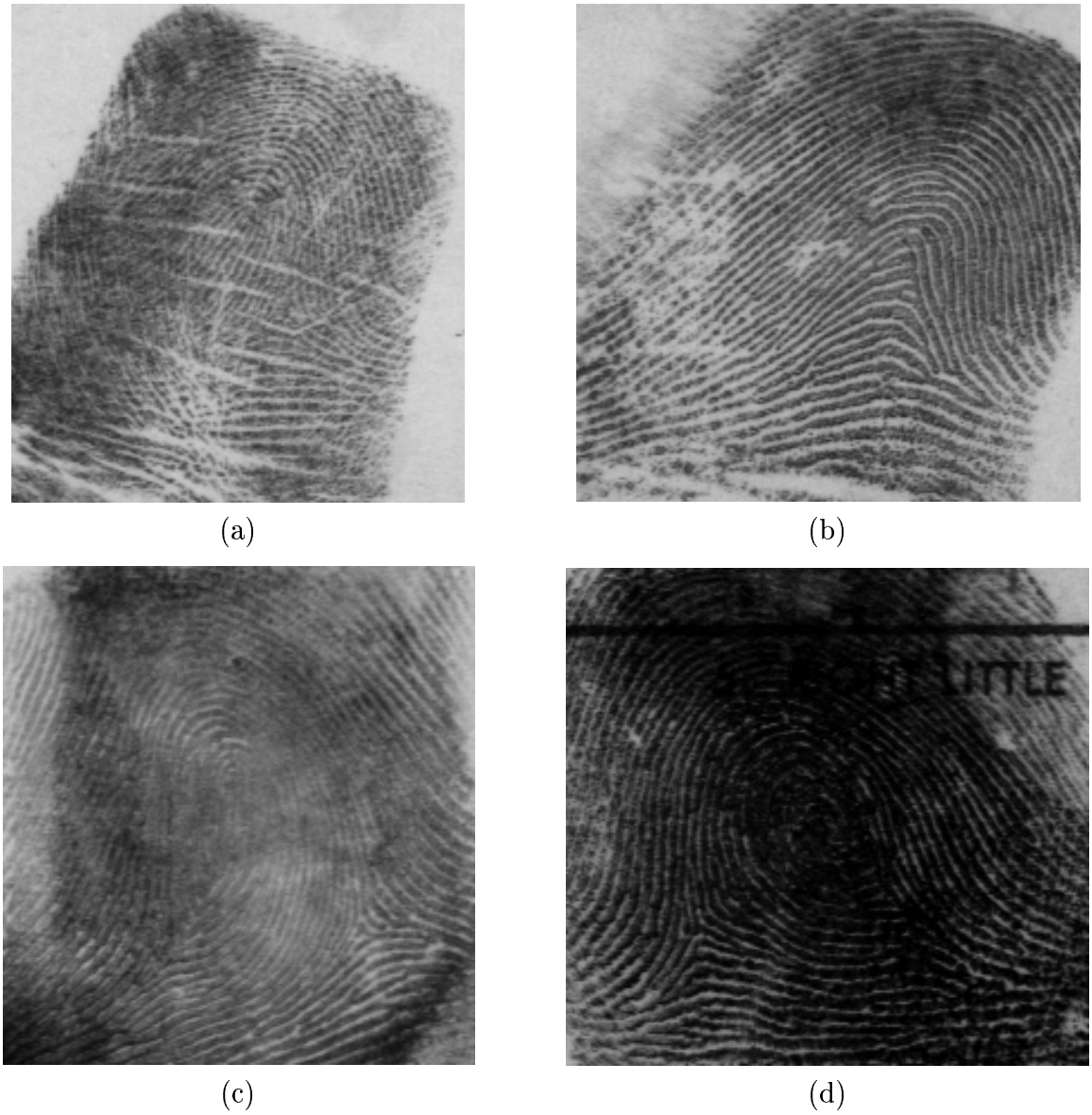


Figure 9.17: Misclassified fingerprints in NIST 4 fingerprint database; (a) a left loop is misclassified as an arch; (b) a tented arch is misclassified as an arch; (c) a left loop is misclassified as a whorl; (d) a whorl is misclassified as a right loop.

True Class	Assigned Class				
	Arch	Tented Arch	Left Loop	Right Loop	Whorl
Arch	**357**	10	6	3	3
Tented Arch	38	**121**	8	5	2
Left Loop	32	32	**1506**	15	65
Right Loop	21	82	23	**1481**	72
Whorl	5	8	93	58	**1293**

Table 9.9: Five-class classification results on NIST 9 database (5,400 images).

True Class	Assigned Class			
	Arch	Left Loop	Right Loop	Whorl
Arch	**526**	14	8	5
Left Loop	64	**1506**	15	65
Right Loop	103	23	**1481**	72
Whorl	13	93	58	**1293**

Table 9.10: Four-class classification results on NIST 9 database.

prints in NIST 9 database is 10.9%. The confusion matrix is shown in Table 9.9. The classification errors are due to misclassifications between arch/tented arch and loops and poor image quality. Examples of misclassified fingerprints are shown in Figure 9.18. A lower error rate can be achieved by adding the reject option, which is based on the quality index of the input image. Table 9.11 shows the error rates corresponding to different reject rates.

In classifying the fingerprints in the IBM database, we assume that twins and composites are whorls. The classification error rate on the IBM database is 10.1% with a 2.6% reject rate. The confusion matrix is listed in Table 9.12. Examples of misclassified fingerprints are shown in Figure 9.19. The error rates corresponding to

Reject rate	0	5%	10%	15%	20%
5-class Error rate	10.9%	9.6%	8.2%	7.4%	6.4%
4-class Error rate	10.0%	8.6%	7.7%	6.9%	6.2%

Table 9.11: Error rates corresponding to different reject rates on NIST 9 database.

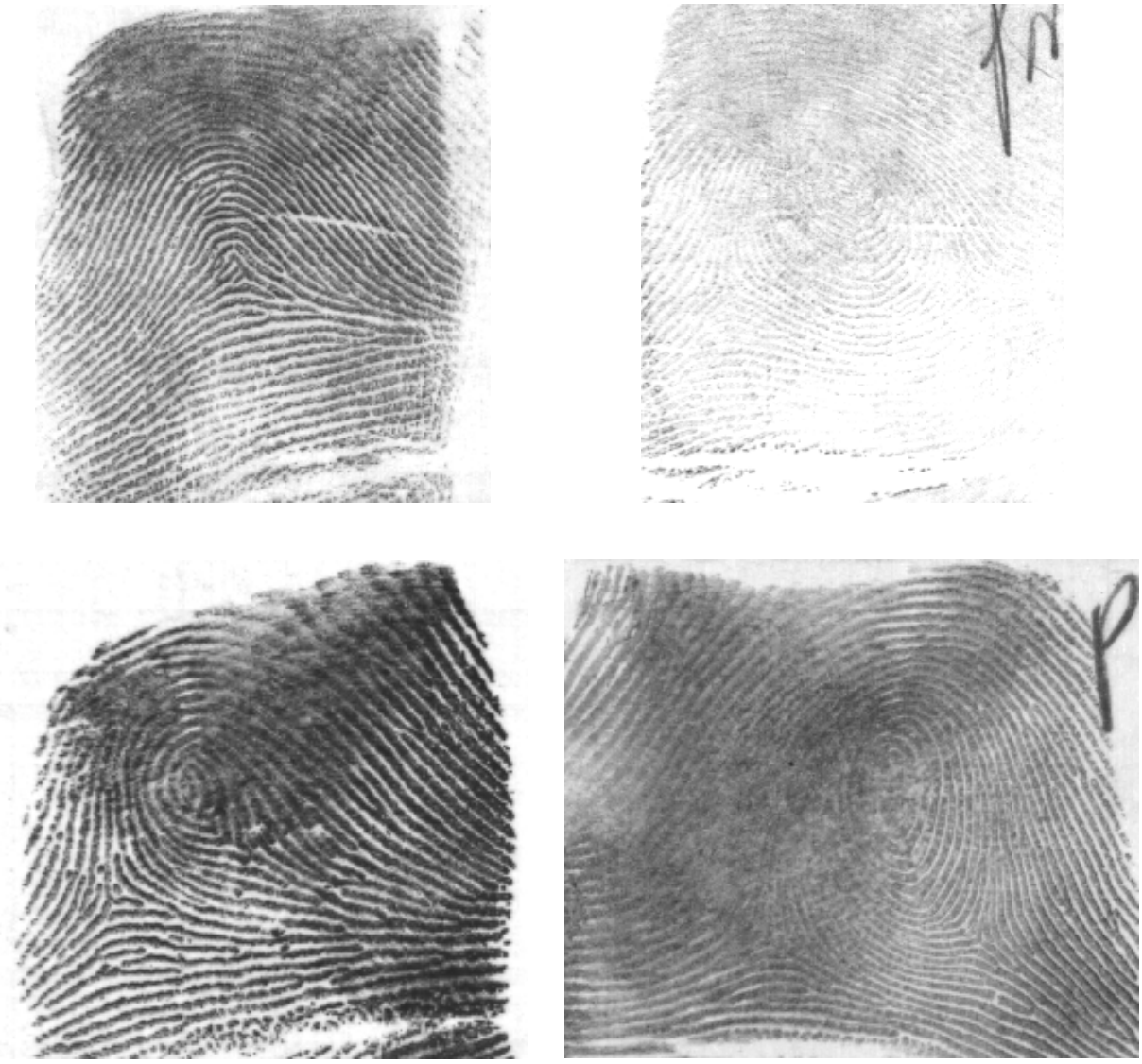


Figure 9.18: Misclassified fingerprints in NIST 9 database; (a) a tented arch is misclassified as an arch; (b) a whorl is misclassified as an arch; (c) a whorl is misclassified as a left loop; (d) a whorl is misclassified as a right loop.

True Class	Assigned Class				
	Arch	Tented Arch	Left Loop	Right Loop	Whorl
Arch	**62**	0	0	1	1
Tented Arch	25	**65**	5	2	3
Left Loop	10	4	**266**	3	6
Right Loop	9	7	3	**345**	4
Whorl	3	0	7	10	**173**

Table 9.12: Five-class classification results on the IBM database.

True Class	Assigned Class			
	Arch	Left Loop	Right Loop	Whorl
Arch	**152**	5	3	4
Left Loop	14	**266**	3	6
Right Loop	16	3	**345**	4
Whorl	3	7	10	**173**

Table 9.13: Four-class classification results on the IBM database.

different reject rates are listed in Table 9.14.

9.5.2 Classification Scheme II

It is difficulty to evaluate the performance of classification scheme II, because the ground truth is not available. However, since the essence of fingerprint classification is to classify fingerprints from the same finger to the same category, we can use individuality property to verify the validity of the classification accuracy implicitly. If two fingerprints from the same finger are classified into the same category, then we claim that the classification is correct, otherwise incorrect. Let N be the number of mated pairs in a database (a total of $2N$ fingerprints), and N_c be the number of

Reject rate	2.6%	5%	10%	15%	20%
5-class Error rate	10.1%	9.3%	7.5%	6.9%	5.8%
4-class Error rate	7.7%	7.0%	5.2%	4.9%	4.1%

Table 9.14: Error rates corresponding to different reject rates on the IBM database.

Figure 9.19: Misclassified fingerprints in the IBM database; (a) a left loop is misclassified as an arch; (b) a right loop is misclassified as a tented arch; (c) a whorl is misclassified as an arch; (d) a tented arch is misclassified as an arch.

fingerprint pairs that are consistently classified. Then, the inconsistency rate, e_i, is defined as

$$e_i = \frac{100N_c}{N}. \tag{9.2}$$

Let e be the misclassification rate, and define $c = 1 - e$. For a given mated pair, there are four outcomes: (i) both images are correctly classified, (ii) the first image is correctly classified and the second image is incorrectly classified, (iii) the first image is incorrectly classified and the second image is correctly classified, and (iv) both images are incorrectly classified. Outcome (i) corresponds to a consistent classification. Outcomes (ii) and (iii) correspond to an inconsistent classification. For outcome (iv), the mated images may be misclassified into (a) the same category and (b) different categories. Outcome (a) corresponds to a consistent classification and outcome (b) corresponds to an inconsistent classification. Therefore,

$$\begin{aligned} e_i &= ce + ce + be^2, & (9.3)\\ &= e + e + (1-b)e^2, & (9.4)\\ &> e, & (9.5) \end{aligned}$$

where $b < 1$ denote the probability of outcome (b).

We have tested classification scheme II on both NIST 4 database and NIST 9 database. The inconsistency rate in classifying all the 2,000 mated pairs is 7.0% with a 1.4% reject rate. The confusion matrix is given in Table 9.15; numbers shown in

First Class	Second Class			
	Class-1	Class-2	Class-3	Class-4
Class-1	**774**	28	15	3
Class-2	19	**355**	6	9
Class-3	30	7	**360**	8
Class-4	3	9	5	**339**

Table 9.15: Consistency test results on the NIST 4 database.

Reject rate	1.4%	5.0%	10.0%	15.0%	20.0%
Inconsistency rate	7.0%	4.3%	4.3%	3.1%	2.7%

Table 9.16: Inconsistency rates corresponding to different reject rates on the NIST 4 database.

bold font are the number of times when the mated pairs are classified in the same category. A lower inconsistency rate can be achieved by incorporating the reject option, which is based on the quality index of the input image. The inconsistency rates corresponding to different reject rates are listed in Table 9.16.

The inconsistency rate in classifying all the 5,400 NIST 9 fingerprints is 7.6% with a reject rate of 4.2%. The confusion matrix is given in Table 9.17; numbers shown in bold font are the number of consistent classifications. A lower inconsistency rate can be achieved by adding the reject option, which is based on the quality index of the input image. The error rates corresponding to different reject rates are listed in Table 9.18.

First Class	Second Class			
	Class-1	Class-2	Class-3	Class-4
Class-1	**349**	18	17	5
Class-2	21	**714**	14	22
Class-3	27	8	**718**	21
Class-4	2	21	30	**599**

Table 9.17: Consistency results on the NIST 9 database.

Reject rate	4.2%	5.0%	1.0%	15.0%	20.0%
Inconsistency rate	7.6%	7.3%	5.4%	4.8%	3.3%

Table 9.18: Inconsistency rates corresponding to different reject rates on the NIST 9 database.

9.6 Summary

In summary, the number of tests conducted on an automatic fingerprint identification/verification system is never enough. Performance measures are as much a function of the algorithm as they are a function of the database used for testing. The biometrics community is slow at establishing benchmarks and the ultimate performance numbers of a fingerprint identification/verification system are those which you find in a deployed system. Therefore, one can carry out only a limited amount of testing in a laboratory environment to show the anticipated system performance. In field testing, in addition to the real performance of the system, the system designer has to pay attention to the perceived performance of the system, especially in the context of the authentication applications which are sensitive to false negatives.

Chapter 10

Summary and Future Research

In this chapter, we will summarize the work we have done, discuss the limitations of our current approaches and some possible solutions.

10.1 Summary

The goal of our research is to design a fingerprint-based biometric system which is capable of achieving a *fully automatic positive personal identification* with a high level of confidence. We have developed a prototype verification system and a prototype identification system. The verification depends solely on fingerprints to *authenticate* the identity claimed by an individual. The identification system which is designed for a limited environment, uses multiple biometric characteristics (fingerprint and face) to make a personal identification. In the design of these two prototype systems, we have identified and investigated the following problems:

- **Minutiae extraction**

  We have developed a new minutiae extraction algorithm which is faster and more reliable than the earlier algorithms reported in the literature, *e.g.*, [120]. The new orientation field estimation algorithm results in a smoother orientation field which greatly improves the performance of ridge extraction. The adaptive ridge finder is capable of tolerating, to a certain extent, low ridge contrast and various sources of noise in fingerprint images such as short breaks and small amount of smudges.

- **Fingerprint enhancement**

  We have developed a new fingerprint image enhancement algorithm. Unlike other enhancement algorithms, we expend a large amount of effort on the estimation of orientation field, which plays a critical role in fingerprint enhancement. We can obtain a relatively good estimate of orientation field even if the quality of input fingerprint image is poor. The algorithm also identifies the unrecoverable corrupted regions in the fingerprint and removes them. Experimental results reveal that the proposed fingerprint enhancement algorithm can significantly improve the quality of input images, resulting in better matching performance.

- **Minutiae matching**

  An alignment-based elastic matching algorithm has been developed to implement minutiae matching. This algorithm is capable of finding the correspondences between minutiae without resorting to an exhaustive search. It can

achieve a good performance in minutiae matching because of its capability to adaptively compensate for the nonlinear deformations and inexact transformations between mated fingerprints.

- **Decision fusion**

  We have developed an integration scheme which can be used to fuse multiple biometrics that complement each other in terms of speed and accuracy. We have tested our scheme by integrating fingerprint verification and face recognition. Experimental results demonstrate that our scheme is able to improve both the accuracy and speed of identification. The multimodal biometric system overcomes some of the limitations of face recognition and fingerprint verification.

- **Fingerprint classification**

  We have designed two fingerprint classification algorithms. The first algorithm classifies input fingerprints into five categories according to the number of singular points detected, their relative positions, and presence of type-1 and type-2 recurring ridges. The second algorithm classifies fingerprints into *four* categories according to the number of singular points detected and their relative positions. Our algorithm invests a significant amount of effort in feature extraction to make the system robust to intraclass variations as well as poor quality of input images. Experiment results demonstrate that our algorithm has better classification performance than previously reported in the literature on the NIST 9 and NIST 4 databases.

- **Performance evaluation**

  The performance of the entire verification system and the identification system as well as the performance of various system components (*e.g.*, minutiae extraction and the fingerprint enhancement) were evaluated extensively on a number of real fingerprint databases. The experimental results reveal that both the verification system and the identification system can achieve a good performance.

10.2 Future Research

Despite the fact that both the verification system and the identification system can achieve a good performance in making a personal identification, we believe that a number of problems still need to be solved to make these systems more effective in practice. The following is a list of limitations of our current approaches and the directions which are significant for the performance improvement of our implementation.

- In order to determine whether a pair of fingerprints are from the same finger, two conditions must be assessed: (*i*) the two fingerprints must be of the same pattern configuration, *e.g.*, whorl, arch, *etc.* and (*ii*) they must share a substantial number of identical minute details. Currently, our minutiae matching algorithm depends only on the assessment of the second condition to make a decision. Obviously, this is not sufficient. A fingerprint classification scheme assigns a fingerprint into one of the prespecified categories based on its global pattern configuration. If two fingerprints are from the same finger, they must belong to the same category. Although fingerprint classification is still a challenging

Figure 10.1: Minutiae with different degrees of importance; the minutiae labeled by the circle is more important than the minutiae labeled by the square.

problem and it is very difficult to achieve a very high classification rate, it is nonetheless beneficial to incorporate the category information into a minutiae matching algorithm to improve its performance.

- All the minutiae extracted by our algorithm are weighted equally in minutiae matching. In fact, different minutiae have varying degrees of importance in matching (see Figure 10.1). Therefore, it will be beneficial to systematically assign a weight to each minutiae to indicate its importance and to incorporate this information in minutiae matching.

- Currently, the minutiae extraction algorithm does not assume any model of the

input fingerprints. In fact, in a verification system, as soon as the individual indicates his/her identity, the template of the individual is available. Such a template provides a model of the input fingerprint. The minutiae extraction algorithm should use this model to facilitate the minutiae extraction.

- Since the minutiae matching does not make any assumptions about the relative translation and rotation between the input minutiae pattern and the template, it needs to evaluate the edit distance corresponding to each pair of aligning ridges to find the best match. Computing the edit distance is computationally expensive. The speed of the minutiae matching can be greatly accelerated if the unreasonable alignments achieved by the ridge matching can be quickly rejected.

- A number of factors are detrimental to the correct localization of minutiae. Among them, poor image quality is the most serious one. By integrating an enhancement mechanism into the minutiae extraction module, this problem can, to a limited extent, be solved. However, currently, our fingerprint enhancement algorithm is computationally expensive, which is not suitable for incorporation into the minutiae extraction algorithm. A fast enhancement algorithm is needed.

- In order to show that a deployed biometric system is capable of achieving certain benchmarks, a systematic and objective performance assessment of the system is necessary, which, unfortunately, is far from established. The expected error rate of a deployed biometric system is usually a very small number ($\ll 1\%$). A

maximum likelihood estimate of the error rate is usually not reliable. In order to estimate such a small number reliably and accurately, large representative data sets are needed and the corresponding confidence intervals should be provided to characterize the reliability of the estimate. Two major issues need to be addressed: (*i*) how should the samples be selected from the population? and (*ii*) how many samples are needed to estimate the error rate with the expected confidence?

Bibliography

[1] *Access Control Applications using Optical Computing.* http://www.mytec.com/, 1997.

[2] *Edge Lit Hologram for Live-scan Fingerprinting.* http://eastview.org/ImEdge/, 1997.

[3] *Scanner Specifications.* ftp://ard.fbi.gov/pub/IQS/spec/, 1997.

[4] N. Ansari, M. H. Chen, and E. S. H. Hou. A genetic algorithm for point pattern matching. In B. Souček and the IRIS Group, editors, *Dynamic, Genetic, and Chaotic Programming.* John Wiley & Sons, 1992.

[5] K. Asai, Y. Hoshino, Y. Kato, and K. Kiji. Automatic reading and matching for single-fingerprint identification. In *Proc. 65th Int. Ass. for Identification Conf.*, pages 1–7, Ottawa, Canada, 1991.

[6] J. Atick, P. Griffin, and A. Redlich. Statistical approach to shape from shading: Reconstruction of 3D face surfaces from single 2D images. *Neural Computation*, 1998. to appear.

[7] R. Bahuguna. Fingerprint verification using hologram matched filterings. In *Proc. Biometric Consortium Eighth Meeting*, San Jose, California, June 1996.

[8] H. Baird. *Model Based Image Matching Using Location*. MIT Press, Cambridge, MA, 1984.

[9] K. Balck and K. Rao. A hybrid optical computer processing technique for fingerprint identification. *IEEE Trans. Computer*, 24:358–369, 1975.

[10] P. Baldi and Y. Chauvin. Neural networks for fingerprint recognition. *Neural Computation*, 5(3):402–418, 1993.

[11] D. H. Ballard. Generalized hough transform to detect arbitrary patterns. *IEEE Trans. Pattern Anal. and Machine Intell.*, 3(2):111–122, 1981.

[12] C. Banner and R. Stock. The FBI's approach to automatic fingerprint identification (part I). *FBI Law Enforcement Bulletin, U.S.A. Government Publication*, 44(1), 1975.

[13] C. Banner and R. Stock. The FBI's approach to automatic fingerprint identification (part II). *FBI Law Enforcement Bulletin, U.S.A. Government Publication*, 44(2), 1975.

[14] L. Berdan and R. Chiralo. Adaptive digital enhancement of latent fingerprints. In *Proc. Int. Carnahan Conf. on Electronic Crime Countermeasures*, pages 131–135, University of Kentucky, Lexington, Kentucky, 1978.

[15] Z. Bian and Q. Xiao. An approach to fingerprint identification by using the attributes of feature lines of fingerprint. In *Proc. 8th ICPR*, pages 27–31, Paris, France, 1986.

[16] E. S. Bigun, J. Bigun, B. Duc, and S. Fischer. Expert conciliation for multi modal person authentication systems by Bayesian statistics. In *Proc. 1st Int. Conf. on Audio Video-Based Personal Authentication*, pages 327–334, Crans-Montana, Switzerland, March 1997.

[17] Biometrics Consortium Homepage. *INS Passenger Accelerated Service System (INSPASS).* http://www.accentdesign.com/bc/publications.html, 1998.

[18] E. Botha and L. Coetzee. Fingerprint recognition with a neural-net classifier. In *Proc. First South African Workshop on Pattern Recognition*, volume 1, pages 33–40, 1990.

[19] F. Bouchier, J. S. Ahrens, and G. Wells. *Laboratory Evaluation of the IriScan Prototype Biometric Identifier.* http://infoserve.library.sandia.gov/sand_doc/1996/961033.pdf, SAND96-1033, 1996.

[20] R. Brunelli and D. Falavigna. Personal identification using multiple cues. *IEEE Trans. Pattern Analysis and Machine Intelligence*, 17(10):955–966, 1995.

[21] R. Brunelli and T. Poggio. Face recognition: Features versus templates. *IEEE Trans. Pattern Analysis and Machine Intelligence*, 15(10):1042–1052, 1993.

[22] G. Candela and R. Chellappa. *Comparative Performance of Classification Methods for Fingerprints.* Tech. Report: NIST TR 5163, 1993.

[23] G. Candela, P. Grother, C. Watson, R. Wikinson, and C. Wilson. *Evaluation of Pattern Classifiers for Fingerprint and OCR Applications.* Tech. Report: NIST TR 3162, 1993.

[24] G. T. Candela, P. J. Grother, C. I. Watson, R. A. Wilkinson, and C. L. Wilson. *PCASYS: A Pattern-Level Classification Automation System for Fingerprints.* NIST Tech. Report NISTIR 5647, August 1995.

[25] S. H. Chang, F. H. Cheng, W. H. Hsu, and G. Z. Wu. Fast algorithm for point pattern matching: Invariant to translations, rotations, and scale changes. *Pattern Recognition*, 30(2):321–339, 1997.

[26] T. Chang. Texture analysis of digitized fingerprints for singularity detection. In *Proc. 5th ICPR*, pages 478–480, 1980.

[27] C. Chapel. *Fingerprinting - A Manual of Identification.* Coward McCann, New York, 1971.

[28] B. Chatterjee, S. Kapoor, B. Mehtre, and N. Murthy. Segmentation of fingerprint images using the directional image. *Pattern Recognition*, 20:429–435, 1987.

[29] B. Chatterjee, A. Majumdar, and M. Verma. Edge detection in fingerprints. *Pattern Recognition*, 20:513–523, 1987.

[30] B. Chatterjee and B. Mehtre. Automatic fingerprint identification. *Journal of the Institution of Electronics and Telecom.*, 37(5/6):493, 1991.

[31] R. Chellappa, C. Wilson, and A. Sirohey. Human and machine recognition of faces: A survey. *Proceedings IEEE*, 83(5):705–740, 1995.

[32] M. Chong, R. Gay, J. Liu, and H. Tan. Automatic representation of fingerprints for data compression by B-spline functions. *Pattern Recognition*, 25(10):1199–1210, 1992.

[33] M. Chong, T. Ngee, L. Jun, and R. Gay. Geometric framework for fingerprint image classification. *Pattern Recognition*, 30(9):1475–1488, 1997.

[34] J. Clark and A. Yuille. *Data Fusion for Sensory Information Processing Systems.* Kluwer Academic Publishers, Boston, 1990.

[35] R. Clarke. Human identification in information systems: Management challenges and public policy issues. *Information Technology & People*, 7(4):6–37, 1994.

[36] Louis Coetzee and Elizabeth Botha. Fingerprint recognition in low quality images. *Pattern Recognition*, 26(10):1441–1460, 1993.

[37] T. H. Cormen, C. E. Leiserson, and R. L. Rivest. *Introduction to Algorithms.* McGraw-Hill, New York, 1990.

[38] J. Cowger. *Friction Ridge Skin: Comparison and Identification of Fingerprints.* Elsevier, New York, 1983.

[39] Thomson CSF. *Thomson CSF Homepage.* http://www.tcs.thomson-csf.com/Us/standard/finger.htm, 1998.

[40] H. Cummins and C. Mildo. *Finger Prints, Palms and Soles.* Dover Publication Inc., New York, 1961.

[41] P. E. Danielsson and Q. Z. Ye. Rotation-invariant operators applied to enhancement of fingerprints. In *Proc. 9th ICPR*, pages 329–333, Rome, 1988.

[42] J. G. Daugman. Uncertainty relation for resolution in space, spatial-frequency, and orientation optimized by two-dimensional visual cortical filters. *J. Opt. Soc. Am.*, 2:1160–1169, 1985.

[43] J. G. Daugman. High confidence visual recognition of persons by a test of statistical independence. *IEEE Trans. Pattern Anal. and Machine Intell.*, 15(11):1148–1161, 1993.

[44] J. G. Daugman and G. O. Williams. A proposed standard for biometric decidability. In *Proc. CardTech/SecureTech Conference*, pages 223–234, Atlanta, GA, 1996.

[45] S. G. Davies. Touching big brother: How biometric technology will fuse flesh and machine. *Information Technology & People*, 7(4):60–69, 1994.

[46] U. Dieckmann, P. Plankensteiner, and T. Wagner. Sesam: A biometric person identification system using sensor fusion. *Pattern Recognition Letters*, 18(9):827–833, 1997.

[47] M. Eleccion. Automatic fingerprint identification. *IEEE Spectrum*, 10(9):36–45, 1973.

[48] W. Engeler, P. Frank, and M. Leung. Fingerprint image processing using neural network. In *Proc. IEEE Region 10 Conf. on Computer and Comm. Systems*, Hong Kong, 1990.

[49] P. Engler, Y. Shi, and F. You. Fingerprint pattern recognition for medical uses- a frequency domain approach. In *Proc. Annual Northeast Bioengineering Conference*, volume 19, page 176, 1993.

[50] J. Ferrante, J. Maier, J. Nelson, and J. Woodard. Automated entry control: Radc technology development results and future plans. In *Proc. Int. Carnahan Conf. on Electronic Crime Countermeasures*, pages 77–82, University of Kentucky, Lexington, Kentucky, 1981.

[51] K. Fielding, J. Homer, and C. Makekau. Optical fingerprint identification by binary joint transform correlation. *Optical Engineering*, 30:1958, 1991.

[52] L. Frye, F. Gamble, and D. Grieser. Real-time fingerprint verification system. *Applied Optics*, 31(5):652, 1992.

[53] Y. Fumio, I. Seigo, and E. Shin. Real-time fingerprint sensor using a hologram. *Applied Optics*, 31(11):1794, 1992.

[54] F. Galton. *Finger Prints.* Da Capo Press, New York, 1961.

[55] S. Gold and A. Rangarajan. A graduated assignment algorithm for graph matching. *IEEE Trans. Pattern Anal. and Machine Intell.*, 18(4):377–388, 1996.

[56] K. Goto, T. Minami, and O. Nakamura. Fingerprint classification by directional distribution patterns. *System Computer Controls*, 13:81–89, 1982.

[57] Q. Guisheng, C. Minde, Q. Shi, and N. Xue. A new automated fingerprint identification system. *Computer Science Technology*, 4(4):289–294, 1989.

[58] E. J. Gumbel. *Statistics of Extremes.* Columbia University Press, New York, 1958.

[59] L. Gyergyek, S. Kovacic, and F. Pernus. Minutiae based fingerprint recognition. In *Proc. 5th ICPR*, pages 1380–1382, 1980.

[60] M. Hartman. Compact fingerprint scanner techniques. In *Proc. Biometric Consortium Eighth Meeting*, San Jose, California, June 1996.

[61] M. Hase and A. Shimisu. Entry method of fingerprint image using a prism. *Trans. Inst. Electron. Commum. Eng. Japan*, J67-D:627–628, 1984.

[62] I. Hideki, K. Ryuj, and H. Yu. A fast automatic fingerprint identification method based on a weighted-mean of binary image. *IEICE Transactions on Fundamentals of Electronic*, 76:1469, 1993.

[63] L. Hong and A. Jain. Integrating faces and fingerprints for personal identification. In *Proc. 3rd Asian Conference on Computer Vision*, pages 16–23, Hong Kong, China, 1998.

[64] L. Hong, Y. Wan, and A. Jain. Fingerprint image enhancement: Algorithms and performance evaluation. *to appear in IEEE Trans. on PAMI*, 1998.

[65] Z. Hong. Algebraic feature extraction of image for recognition. *Pattern Recognition*, 24(2):211–219, 1991.

[66] A. Hrechak and J. McHugh. Automated fingerprint recognition using structural matching. *Pattern Recognition*, 23(8):893–904, 1990.

[67] D. C. Huang. Enhancement and feature purification of fingerprint images. *Pattern Recognition*, 26(11):1661–1671, 1993.

[68] D. P. Huttenlocher and S. Ullman. Object recognition using alignment. In *Proc. First Intern. Conf. Comput. Vision*, pages 102–111, London, 1987.

[69] D. Isenor and S. Zaky. Fingerprint identification using graph matching. *Pattern Recognition*, 19:113–122, 1986.

[70] Jr. J. Campbell. Speaker recognition: A tutorial. *Proceedings of IEEE*, 85(9):1437–1462, 1997.

[71] A. Jain, L. Hong, and R. Bolle. On-line fingerprint verification. *IEEE Trans. Pattern Anal. and Machine Intell.*, 19(4):302–314, 1997.

[72] A. K. Jain and F. Farrokhnia. Unsupervised texture segmentation using Gabor filters. *Pattern Recognition*, 24(12):1167–1186, 1991.

[73] G. Johnson, D. McMahon, S. Teeter, and G. Whitney. A hybrid optical computer processing technique for fingerprint identification. *IEEE Trans. Computers*, 24:358–369, 1975.

[74] P. Jones, B. Santer, and T. Wigley. Correlation methods in fingerprint detection studies. *Climate Dynamics*, 8(6):265, 1993.

[75] J. Campbell Jr., L. Alyea, and J. Dunn. Biometric security: Government applications and operations. *http:/www.vitro.bloomington.in.us:8080/~BC/*, 1996.

[76] T. Kamei and M. Mizoguchi. Image filter design for fingerprint enhancement. In *Proc. ISCV' 95*, pages 109–114, Coral Gables, FL, 1995.

[77] F. Karen. Encryption, smart cards, and fingerprint readers. *IEEE Spectrum*, 26(8):22, 1989.

[78] K. Karu and A. K. Jain. Fingerprint classification. *Pattern Recognition*, 29(3):389–404, 1996.

[79] M. Kass and A. Witkin. Analyzing oriented patterns. *Comput. Vision Graphics Image Process.*, 37(4):362–385, 1987.

[80] M. Kawagoe and A. Tojo. Fingerprint pattern classification. *Pattern Recognition*, 17(3):295–303, 1984.

[81] E. Kaymaz and S. Mitra. A novel approach to Fourier spectral enhancement of laser-luminescent fingerprint images. *Journal of Forensic Sciences*, 38(3):530, 1993.

[82] J. Kittler, Y. Li, J. Matas, and M. U. Sanchez. Combining evidence in multimodal personal identity recognition systems. In *Proc. 1st Int. Conf. on Audio Video-Based Personal Authentication*, pages 327–334, Crans-Montana, Switzerland, March 1997.

[83] J. Klett. Thermal imaging fingerprint technology. In *Proc. Biometric Consortium Ninth Meeting*, Crystal City, Virginia, April 1997.

[84] W. Lau, S. Leung, W. Leung, and A. Luk. Fingerprint recognition using neural network. In *Proc. IEEE Workshop Neural Network for Signal Processing*, 1991.

[85] H. C. Lee and R. E. Gaensslen. *Advances in Fingerprint Technology.* Elsevier, New York, 1991.

[86] Lexington Technology, Inc. *Lexington Technology, Inc. Homepage.* http://www.lexingtontech.com/index.html, 1998.

[87] J. Li, Y. Shan, and P. Shi. Fingerprint preclassification using key-points. In *Proc. IEEE International Symposium on Speech, Image Proc. and Neural Network*, Hong Kong, 1994.

[88] F. R. Livingstone, L. King, J. Beraldin, and M. Rioux. Development of a real-time laser scanning system for object recognition, inspection, and robot control. In *Proc. SPIE on Telemanipulator Technology and Space Telerobotics*, volume 2057, pages 454–461, Boston, Massachusetts, September 1993.

[89] S. Maes and H. Beigi. Open sesame! speech, password or key to secure your door? In *Proc. 3rd Asian Conference on Computer Vision*, pages 531–541, Hong Kong, China, 1998.

[90] D. Maio and D. Maltoni. A structural approach to fingerprint classification. In *Proc. 13th ICPR*, pages 578–585, Vienna, 1996.

[91] D. Maio and D. Maltoni. Direct gray-scale minutiae detection in fingerprints. *IEEE Trans. Pattern Anal. and Machine Intell.*, 19(1):27–40, 1997.

[92] D. Marr. *Vision*. W. H. Freeman, San Francisco, 1982.

[93] K. McCalley, D. Setlak, S. Wilson, and J. Schmitt. A direct fingerprint reader. In *Proc. CardTech/SecurTech, Volume I: Technology*, pages 271–279, Atlanta, Georgia, May 1996.

[94] B. Mehtre. Fingerprint image analysis for automatic identification. *Machine Vision and Applications*, 6(2-3):124–139, 1993.

[95] B. M. Mehtre, N. N. Murthy, and Kapoor. Segmentation of fingerprint images using the directional image. *Pattern Recognition*, 20(4):429–435, 1987.

[96] K. Millard, D. Monro, and B. Sherlock. Algorithm for enhancing fingerprint images. *Electronics Letters*, 28(18):1720, 1992.

[97] B. Miller. Vital signs of identity. *IEEE Spectrum*, 31(2):22–30, 1994.

[98] D. Mintie. Welfare id at the point of transaction using fingerprint and 2D bar codes. In *Proc. CardTech/SecurTech, Volume II: Applications*, pages 469–476, Atlanta, Georgia, May 1996.

[99] Miros. *Miros Homepage*. http://www.miros.com, 1998.

[100] B. Moayer and K. Fu. A syntactic approach to fingerprint pattern recognition. *Pattern Recognition*, 6, 1974.

[101] B. Moayer and K. Fu. An application of stochastic languages to fingerprint pattern recognition. *Pattern Recognition*, 8:173–179, 1976.

[102] B. Moayer and K. Fu. A tree system approach for fingerprint pattern recognition. *IEEE Trans. Pattern Anal. and Machine Intell.*, 8(3):376–388, 1986.

[103] A. Moenssens. *Fingerprint Techniques*. Chilton Book Company, London, 1971.

[104] D. Monro and B. Sherlock. A model for interpreting fingerprint topology. *Pattern Recognition*, 26(7):1047–1055, 1993.

[105] V. Nalwa. Automatic on-line signature verification. *Proceedings of IEEE*, 85(2):213–239, 1997.

[106] E. Newham. *The Biometric Report*. SJB Services, New York, 1995.

[107] J. Nickerson and L. O'Gorman. Matched filter design for fingerprint image enhancement. In *Proc. IEEE Int. Conf. on Acoustic, Speech and Signal Processing*, pages 916–919, New York, 1988.

[108] Federal Bureau of Investigation. *The Science of Fingerprints: Classification and Uses.* U.S. Government Printing Office, Washington, D. C., 1984.

[109] National Institute of Standards and Technology. *Guideline for The Use of Advanced Authentication Technology Alternatives.* Federal Information Processing Standards Publication 190, 1994.

[110] L. O'Gorman and J. V. Nickerson. An approach to fingerprint filter design. *Pattern Recognition*, 22(1):29–38, 1989.

[111] H. Okada and B. Sheu. An analog VLSI edge detector chip and digital multiprocessor chip for neural-based vision processing. In *Proc. IEEE Int. Conf. System Engineering*, 1992.

[112] J. Osterberg, T. Parthasarathy, T. Raghavan, and S. Sclove. Development of a mathematical formula for the calculation of fingerprint probabilities based on individual characteristic. *Journal American Statistic Association*, 72:772–778, 1977.

[113] A. Papoulis. *Probability, Random Variables, and Stochastic Processes.* McGraw-Hill, New York, 1965.

[114] P. J. Phillips, P. J. Rauss, and S. Z. Der. *FERET (Face Recognition Technology) Recognition Algorithm Development and Test Results.* U.S. Government Publication, ALR-TR-995, Army Research Laboratory, Adelphi, MD, 1996.

[115] H. Raafat and Q. Xiao. Fingerprint image postprocessing: A combined statistical and structural approach. *Pattern Recognition*, 24(10):985–992, 1991.

[116] J. Rafferty and J. Wegstein. *The LX39 latent Fingerprint Matcher.* U.S.A. Government Publication. National Bureau of Standards, Institute for Computer Sciences and Technology, 1978.

[117] A. Ranade and A. Rosenfeld. Point pattern matching by relaxation. *Pattern Recognition*, 12(2):269–275, 1983.

[118] A. Rao. *A Taxonomy for Texture Description and Identification.* Springer-Verlag, New York, 1990.

[119] K. Rao and K. Balck. Type classification of fingerprints: A syntactic approach. *IEEE Trans. Pattern Anal. and Machine Intell.*, 2(3):223–231, 1980.

[120] N. Ratha, S. Chen, and A. K. Jain. Adaptive flow orientation based feature extraction in fingerprint images. *Pattern Recognition*, 28(11):1657–1672, 1995.

[121] N. Ratha, K. Karu, S. Chen, and A. K. Jain. A real-time matching system for large fingerprint database. *IEEE Trans. on PAMI*, 18(8):799–813, 1996.

[122] J. Riganati. An overview of algorithms employed in automated fingerprint processing. In *Proc. Int. Carnahan Conf. on Electronic Crime Countermeasures*, pages 125–131, University of Kentucky, Lexington, Kentucky, 1977.

[123] A. Roddy and J. Stosz. Fingerprint features – statistical analysis and system performance estimates. *Proceedings of IEEE*, 85(9):1390–1421, 1997.

[124] K. Saviers. Friction skin characteristics: A study and comparison of proposed standards. In *Garden Grove*. California Police Department, 1987.

[125] J. Schneider. Improved image quality of live scan fingerprint scanners using acoustic backscatter measurements. In *Proc. Biometric Consortium Eighth Meeting*, San Jose, California, June 1996.

[126] J. Schneider and D. Wobschall. Live scan fingerprint imagery using high resolution c-scan ultrasonography. In *Proc. 25th Int. Carnahan Conf. on Security Technology*, pages 88–95, 1991.

[127] S. Sclaroff and A. P. Pentland. Modal matching for correspondence and recognition. *IEEE Trans. Pattern Anal. and Machine Intell.*, 17(6):545–561, 1995.

[128] G. Scott and C. Longuet-Higgins. An algorithm for associating the features of two images. *Proc. Royal Society of London*, 244:21–26, 1991.

[129] I. Seigo, E. Shin, and S. Takashi. Holographic fingerprint sensor. *Fujitsu Scientific & Technical Journal*, 25(4):287, 1989.

[130] Harris Semiconductor. *Harris Semiconductor Homepage.* http://www.semi.harris.com/fngrloc/index.htm, 1998.

[131] D. Sherlock, D. M. Monro, and K. Millard. Fingerprint enhancement by directional Fourier filtering. *IEE Proc. Vis. Image Signal Processing*, 141(2):87–94, 1994.

[132] A. Sherstinsky and R. W. Picard. Restoration and enhancement of fingerprint images using m-lattice: A novel non-linear dynamical system. In *Proc. 12th ICPR-B*, pages 195–200, 1994.

[133] A. Solberg, T. Taxt, and A. Jain. A Markov random field model for classification of multisource satellite imagery. *IEEE Trans. Geoscience and Remote Sensing*, 34(1):100–113, 1996.

[134] M. Sparrow and P. Sparrow. *A Topological Approach to The Matching of Single Fingerprints: Development of Algorithms for Use on Latent Fingermarks.* U.S.A. Government Publication. Gaithersburg, MD: U.S. Dept. of Commerce, National Bureau of Standards, Washington, D.C., 1985.

[135] C. Stanley. Are fingerprints a genetic marker for handedness? *Behavior Genetics*, 24(2):141, 1994.

[136] J. P. P. Starink and E. Backer. Finding point correspondence using simulated annealing. *Pattern Recognition*, 28(2):231–240, 1995.

[137] R. Stock. Automatic fingerprint reading. In *Proc. Int. Carnahan Conf. on Electronic Crime Countermeasures*, pages 16–28, University of Kentucky, Lexington, Kentucky, 1977.

[138] G. Stockman, S. Kopstein, and S. Benett. Matching images to models for registration and object detection via clustering. *IEEE Trans. Pattern Anal. and Machine Intell.*, 4(3):229–241, 1982.

[139] D. L. Swets and J. Weng. Using discriminant eigenfeatures for image retrieval. *IEEE Trans. PAMI*, 18(8):831–836, 1996.

[140] E. Szekly and V Szekly. Image recognition problems of fingerprint identification. *Microprocessors and Microsystems*, 17(4):215–218, 1993.

[141] J. Ton and A. K. Jain. Registering Landsat images by point matching. *IEEE Transactions on Geoscience and Remote Sensing*, 27(5):642–651, 1989.

[142] M. Trauring. Automatic comparison of fingerprint-ridge patterns. *Nature*, 197(4871):938–940, 1963.

[143] TRS. *Technology Recognition Systems Homepage*. http://www.betac.com/trs/, 1998.

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

[145] D. Valentin, H. Abdi, A. J. O'Toole, and G. Cottrell. Connectionist models of face processing: A survey. *Pattern Recognition*, 27(9):1209–1230, 1994.

[146] Veridicom. *Veridicom Homepage*. http://www.veridicom.com/, 1998.

[147] V. V. Vinod and S. Ghose. Point matching using asymmetric neural networks. *Pattern Recognition*, 26(8):1207–1214, 1993.

[148] Visionics. *Visionics Homepage*. http://www.visionics.com, 1998.

[149] C. I. Watson. *NIST Special Database 9, Mated Fingerprint Card Pairs*. National Institute of Standards and Technology, 1993.

[150] C. I. Watson. *NIST Special Database 4, Mated Fingerprint Card Pairs*. National Institute of Standards and Technology, 1994.

[151] J. Wegstein. *The M40 Fingerprint Matcher.* U.S.A. Government Publication. Washington D.C.: National Bureau of Standards, Technical Note 878, U.S Government Printing Office, 1972.

[152] J. H. Wegstein. *An Automated Fingerprint Identification System.* U.S.A. Government Publication, Washington, 1982.

[153] R. Wildes. Iris recognition: An emerging biometric technology. *Proceedings of IEEE*, 85(9):1348–1363, 1997.

[154] J. Woodward. Biometrics: Privacy's foe or privacy's friend? *Proceedings of IEEE*, 85(9):1480–1492, 1997.

[155] J. Zhang, Y. Yan, and M. Lades. Face recognition: Eigenface, elastic matching, and neural nets. *Proceedings of IEEE*, 85(9):1423–1436, 1997.

[156] Y. A. Zuev and S. K. Ivanov. The voting as a way to increase the decision reliability. In *Proc. Foundations of Information/Decision Fusion with Applications to Engineering Problems*, pages 206–210, Washington, D.C., August 1996.