

INFORMATION FUSION IN FINGERPRINT  
AUTHENTICATION

By

*Arun Abraham Ross*

A DISSERTATION

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

DOCTOR OF PHILOSOPHY

Department of Computer Science & Engineering

2003

## ABSTRACT

### INFORMATION FUSION IN FINGERPRINT AUTHENTICATION

By

*Arun Abraham Ross*

Although the problem of *automatic* fingerprint matching has been extensively studied, it is nevertheless, not a fully solved problem. In this thesis, an information fusion approach is adopted to address some of the limitations of existing fingerprint matching systems. A hybrid fingerprint system that utilizes both minutiae points and ridge feature maps to represent and match fingerprint images has been developed. The hybrid matcher is shown to perform significantly better than a traditional minutiae-based matcher. The ridge feature maps extracted by this technique have also been used to align and register fingerprint image pairs via a correlation process, thereby obviating the need to rely on minutiae points for image registration. To address the problem of partial prints obtained from small-sized sensors, a fingerprint mosaicking scheme has been developed. The proposed technique constructs a composite fingerprint template from two partial fingerprint impressions by using the iterative control point (ICP) algorithm that determines the transformation parameters relating the two impressions. To mitigate the effect of non-linear distortions

in fingerprint images on the matching process, an average deformation model has been proposed. The model is developed by comparing a fingerprint impression with several other impressions of the same finger and observing the common ridge points that occur in them. An index of deformation has been suggested in this context to aid in the selection of an ‘optimal’ fingerprint impression from a set of impressions. Finally, techniques to combine fingerprint information with the other biometric traits of a subject (viz., face and hand geometry) are presented. To enhance user convenience, a learning methodology has been used to compute user-specific parameters in a multibiometric system. Information fusion systems, as presented in this thesis, are expected to be more reliable and robust than systems that rely on a single source of information.

© Copyright 2003 by Arun Abraham Ross

All Rights Reserved

To My Family - *Love never fails*

## ACKNOWLEDGMENTS

*Trust in the Lord with all your heart, And lean not on your own understanding; In all your ways  
acknowledge Him, And He shall direct your paths.*

### **Proverbs 3:5,6**

In the last three and a half years I have learnt the importance of relying on Jesus completely. Thank you Lord for showing me the way.

I have been privileged to have Dr. Anil Jain as my advisor. His attention to detail, quest for excellence, and love for perfection has inspired me to give my best time and again. He has helped define my research and presentation skills, and I am deeply indebted to him for making the Ph.D. experience a memorable one. I am grateful to Dr. George Stockman for his constant encouragement and support during the course of my program. Special thanks to Dr. James Reisman and Dr. Sarat Dass who have collaborated with me on various occasions; I thoroughly enjoyed conducting research work alongside them. I would like to express my gratitude to Dr. Jian-Zhong Qian for his efforts in ensuring that I spend three productive summers at Siemens Corporate Research (SCR) in Princeton; portions of my research work were made possible by a

research grant from SCR. Many thanks to Dr. Dennis Gilliland for his guidance and willingness to serve in my Ph.D. committee.

I am grateful to Dr. Salil Prabhakar and Dr. Sharath Pankanti for the excellent support rendered over the past several years. The PRIP lab at MSU has proved to be an excellent working atmosphere, and I will fondly remember the late nights and weekends spent in its precincts. I would like to thank Paul Albee, Anoop Namboodiri, Umut Uludag, Silviu Minut, Xiaoguang Lu, Martin Law, Miguel Figueroa-Villanue, Michael Farmer, Shailesh Saini, Dan Gutchess, Friederike Griess, Unsang Park, Hong Chen, Erin McGarrity, Dr. Vincent Hsu, Dr. Aditya Vailaya, Dr. Nicolae Duta, Dr. Scott Connell, Dr. Yilu Zhang, Dr. Vera Bakic, Dr. Shaoyun Chen and Dr. Lin Hong for engaging me in useful discussions many a time.

A special word of appreciation to Linda Moore, Starr Portice, Cathy Davison, Debbie Kruch, Beverly Wallace and Mary Gebbia for their administrative assistance and support. Thanks to Jay Kusler, John Lane, Margaret Wilke and the rest of the team for their role as system administrators.

Finally, I would like to thank my parents, brother and sister for their incredible love, prayers, enthusiasm, and encouragement; fellow believers for their timely counsel and prayer; and my beautiful wife Lisa for saying “I do”. It is to them that I dedicate this thesis to.

## TABLE OF CONTENTS

<b>LIST OF TABLES</b>	<b>xi</b>
<b>LIST OF FIGURES</b>	<b>xii</b>
<b>1 Introduction</b>	<b>1</b>
1.1 Biometrics . . . . .	1
1.2 Fingerprint as a Biometric . . . . .	8
1.2.1 Fingerprint Representation . . . . .	10
1.2.2 Fingerprint Matching . . . . .	10
1.2.3 Difficulties and Challenges in Fingerprint Matching . . . . .	11
1.3 Thesis Contributions . . . . .	13
<b>2 Fingerprint Representation using Ridge Feature Maps</b>	<b>17</b>
2.1 Introduction . . . . .	20
2.2 Fingerprint as Oriented Texture . . . . .	22
2.3 Image Filtering using Gabor Filters . . . . .	25
2.3.1 Fingerprint Enhancement . . . . .	27
2.3.2 Fingerprint Segmentation . . . . .	28
2.3.3 Filtering Enhanced Image . . . . .	29
2.4 Ridge Feature Maps . . . . .	32
2.4.1 Tessellation of Filtered Images . . . . .	32
2.4.2 Ridge Feature Map Definition . . . . .	32
2.5 Minutiae Extraction . . . . .	33
2.6 Alignment Using Minutiae Information . . . . .	37
2.6.1 Aligning Query and Template Images . . . . .	37
2.6.2 Matching Scores . . . . .	40
2.6.3 Combining Matching Scores . . . . .	41
2.6.4 Fingerprint Identification . . . . .	41
2.6.5 Experimental Results . . . . .	43
2.7 Alignment Using Ridge Feature Maps . . . . .	47
2.7.1 Constructing the Ridge Feature Map . . . . .	48
2.7.2 Fingerprint Matching Using Ridge Feature Maps . . . . .	49
2.7.3 Experimental Results . . . . .	52
2.8 Summary . . . . .	54

<b>3 Fingerprint Mosaicking</b>	<b>55</b>
3.1 Introduction . . . . .	57
3.2 Fingerprint Image Registration . . . . .	60
3.3 Fingerprint Mosaicking . . . . .	63
3.3.1 Preprocessing the Fingerprint Image . . . . .	64
3.3.2 Fingerprint Segmentation . . . . .	64
3.3.3 Fingerprint as a Range Image . . . . .	65
3.3.4 Registering Fingerprint Images . . . . .	65
3.4 Experimental Results . . . . .	71
3.5 Summary . . . . .	72
<b>4 A Deformable Model for Fingerprint Matching</b>	<b>74</b>
4.1 Introduction . . . . .	76
4.2 General Warping Methods . . . . .	81
4.3 The Fingerprint Warping Model . . . . .	85
4.3.1 Establishing Ridge Curve Correspondences . . . . .	86
4.3.2 Sampling Ridge Curves . . . . .	88
4.4 Average Deformation Model . . . . .	91
4.4.1 The $\Phi$ Index of Deformation . . . . .	95
4.4.2 Eliminating Erroneous Correspondences . . . . .	96
4.5 Experimental Results . . . . .	100
4.6 Summary . . . . .	104
<b>5 Multibiometrics</b>	<b>106</b>
5.1 Fusion in Biometrics . . . . .	110
5.1.1 Single Biometric Multiple Representation . . . . .	112
5.1.2 Single Biometric Multiple Matchers . . . . .	113
5.1.3 Multiple Biometric Fusion . . . . .	114
5.2 Multibiometric System . . . . .	115
5.2.1 Face Verification . . . . .	116
5.2.2 Hand Geometry . . . . .	117
5.2.3 Combining the three modalities . . . . .	118
5.3 Learning User-specific Parameters . . . . .	126
5.3.1 User-specific matching thresholds . . . . .	127
5.3.2 Weighting individual biometric traits . . . . .	129
5.4 Factors Affecting a User-specific Approach . . . . .	134
5.5 Summary . . . . .	136
<b>6 Conclusions and Future Work</b>	<b>139</b>
6.1 Research Contributions . . . . .	139
6.2 Future Research . . . . .	141
<b>APPENDICES</b>	<b>145</b>

<b>A Hand Geometry-based Biometric System</b>	<b>146</b>
A.1 Why Hand Geometry? . . . . .	146
A.2 Background . . . . .	147
A.3 Image Acquisition . . . . .	148
A.3.1 Enrollment Phase . . . . .	150
A.3.2 Verification Phase . . . . .	150
A.4 Feature Extraction . . . . .	151
A.5 Experimental Results . . . . .	158
<b>BIBLIOGRAPHY</b>	<b>161</b>

## LIST OF TABLES

3.1	Increase in average image size and average number of detected minutiae as a result of mosaicking. . . . .	67
5.1	Error rates associated with fingerprint, face and voice biometric systems. The accuracy estimates of biometric systems depend on a number of test conditions. . . . .	108
5.2	Performance of the linear discriminant classifier on three different trials as indicated by the confusion matrices. In each trial the training and test sets were partitioned differently. . . . .	125
5.3	User-specific thresholds for the biometric traits of 10 users corresponding to a FAR of 1% in each ROC curve. . . . .	131
5.4	Weights of different traits for 10 users. . . . .	133

## LIST OF FIGURES

1.1	Examples of some of the biometric traits used for authenticating an individual. . . . .	2
1.2	The enrollment module and the verification module of a biometric system.	5
1.3	Biometric systems in civilian applications. (a) A border passage system using iris recognition at London’s Heathrow airport ( <i>news.bbc.co.uk</i> ). (b) The INS Passenger Accelerated Service System (INSPASS) at JFK international airport (New York) uses hand geometry to authenticate travellers and significantly reduce their immigration inspection processing time ( <i>www.panynj.gov</i> ). (c) Ben Gurion airport in Tel Aviv (Israel) uses Express Card entry kiosks fitted with hand geometry systems for security and immigration ( <i>www.airportnet.org</i> ). (d) The FacePass system from Viisage is used in point-of-sale verification applications like ATMs, therefore, obviating the need for PINs ( <i>www.viisage.com</i> ). (e) Indivos’ “Pay by Touch” service uses fingerprints to help customers speed up payments in restaurants and cafeterias. When an enrolled customer places her finger on the sensor, the system retrieves her financial account and updates it ( <i>www.kioskbusiness.com</i> ). (f) The Identix TouchClock fingerprint system is used in time and attendance applications ( <i>www.cardsolutions.com</i> ). . . . .	7
1.4	A fingerprint image with the core and four minutiae points marked on it. The ridge pattern along with the core and delta points define the global configuration, while the minutiae points define the local structure.	9
1.5	Fingerprint sensors installed on (a) a keyboard (the Cherry Biometric Keyboard has a smart card reader and a fingerprint sensor attached to it); (b) a mouse (the ID Mouse manufactured by Siemens has a capacitance-based fingerprint sensor placed on a USB mouse). . . . .	9
1.6	A variety of fingerprint sensors with different specifications (e.g., sensing technology, image size, image resolution, image quality, etc.) are now available. . . . .	12
1.7	Effect of noisy images on a fingerprint authentication system. (a) Fingerprint obtained from a user during enrollment. (b) Fingerprint obtained from the same user during verification. The development of scars or cuts can result in erroneous fingerprint matching results. . . . .	14
1.8	Non universality of fingerprints: Four different impressions of a subject’s finger exhibiting poor quality ridges. A fingerprint system might not be able to enroll this subject since minutiae information cannot be reliably extracted. . . . .	15

2.1	Fingerprint images acquired using the solid state Veridicom sensor (a) and the optical Digital Biometrics sensor (b). The detected minutiae points have been marked in both the fingerprint images. 14 minutiae have been detected in the first case (c), and 39 in the second (d). . . . .	18
2.2	Histogram of the number of minutiae points extracted from images acquired using the Veridicom and Digital Biometric sensors. A total of 2,500 fingerprint impressions were used to compute these histograms for each sensor. The histograms suggest that substantially fewer minutiae points are available in images obtained from solid-state sensors. . . . .	19
2.3	Three impressions of a user's fingerprint exhibiting partial overlap: the overlap between (a) and (b), and (a) and (c) is limited. . . . .	19
2.4	Fingerprint as an oriented texture pattern. (a) the constant inter-ridge spacing in a local region of the fingerprint; (b) the dominant direction of the ridges in (a); (c) the power spectrum of (a). . . . .	21
2.5	Tessellating the fingerprint image using a circular and a square grid. The square tessellation, unlike the circular one, is not affected by the location of the core point in the image. (a) Circular tessellation (80 sectors) about a core point. (b) Square tessellation (169 cells) over the entire image. (c) Circular tessellation about a core detected close to the boundary of the image. (d) Square tessellation over image shown in (c). The images were acquired using the Veridicom sensor. . . . .	24
2.6	Gabor filters in spatial domain with eight different orientations used for feature extraction. $f=0.125$ , $\delta_x = \delta_y = \delta = 4$ . . . . .	27
2.7	Fingerprint image (a) before and (b) after enhancement. . . . .	28
2.8	Segmenting a fingerprint image. (a) The original fingerprint image. (b) The binarized fingerprint image. (c) The median filtered image. (d) The edge image outlining the ridges. (e) The segmented fingerprint image. . . . .	30
2.9	Results of the filtering process on the image shown in Figure 2.7(b). The 8 images correspond to the 8 different orientations of the Gabor filter. . . . .	31
2.10	Tessellating the filtered image. (a) A fingerprint image filtered with a Gabor filter oriented at $157.5^\circ$ . (b) A square tessellation of the filtered image. (c) The ridge feature map ( $n_c \times n_c$ ) representation of the fingerprint. . . . .	33
2.11	Feature maps representing the variance in intensity in the filtered images for each cell. For purposes of visualization, the feature values have been scaled to the 0 - 255 range. . . . .	34
2.12	Flowchart of the minutiae extraction algorithm [1]. . . . .	36
2.13	Template feature extraction. A minutiae set and a ridge feature map are extracted from the input fingerprint image. . . . .	38
2.14	The matching process. The minutiae matching module provides the transformation parameters necessary to align the query image with the template. . . . .	39

2.15	Eight $300 \times 300$ fingerprint impressions acquired using the Veridicom sensor. Images (a) - (d) correspond to the right index finger of one subject, and images (e) - (h) correspond to the right middle finger of another subject. The images were resized to $256 \times 256$ to speed-up Fourier operations. . . . .	44
2.16	ROC showing the performances of the three matchers. The hybrid matcher is observed to perform better than the minutiae matcher. . . . .	46
2.17	Two impressions of the same finger that have a high minutiae matching score but a low ridge feature map matching score. The hybrid score results in a true match. . . . .	46
2.18	Two impressions of the same finger that have a low minutiae matching score but a high ridge feature map matching score. The hybrid score results in a true match. . . . .	47
2.19	The standard deviation map, $\{S_\theta\}$ of the filtered images shown in Figure 2.9. Each image is $240 \times 240$ . . . . .	49
2.20	The ridge feature map, $\{R_\theta\}$ , of the filtered images shown in Figure 2.9. Each image is $15 \times 15$ . . . . .	50
2.21	ROC curves depicting matching performance of the correlation-based technique. . . . .	53
3.1	Fingerprint verification: Multiple impressions of the same finger are stored in the database as templates. The query image is matched against the components of the template to verify the claimed identity. . . . .	56
3.2	Two impressions ( $300 \times 300$ ) of the same finger acquired using the Veridicom sensor. The two impressions are observed to have very little overlap. . . . .	57
3.3	Rolled versus dab prints. (a) - (d): A sequence of 4 rolled fingerprint impressions obtained using the Digital Biometric sensor. Successive image frames are known to have spatial proximity. (e) - (h): Four impressions of a finger obtained at different time instances using a Veridicom sensor. The transformation between the impressions is not known. . . . .	59
3.4	Mapping an intensity image to a range image. (a) The original intensity image. (b) The intensity image after median filtering and scaling. (c) The segmented intensity image. (d) The range image corresponding to the boxed region (rotated by $\sim 90^\circ$ ) in (c). . . . .	66
3.5	Composite template construction: (a) First image after segmentation. (b) Second image after segmentation. (c) Initial alignment. (d) Final alignment. (e) Minutiae extracted from mosaicked images. (f) Composite minutiae set obtained by augmenting individual minutiae sets. . . . .	68
3.6	The result of mosaicking six pairs of fingerprint impressions. The spatial extent of the mosaicked image is observed to be larger than that of the component images. . . . .	69
3.7	Examples of poorly mosaicked image pairs. . . . .	70

3.8	The ROC curves indicating improvement in matching performance after mosaicking templates. Utilizing the minutiae points that are extracted from the composite fingerprint image ( $\mathcal{M}_{R_1}$ ) results in the best matching performance. . . . .	72
4.1	Aligning two impressions of the same finger using an affine transformation. Due to non-linear distortions, the alignment is not accurate in some regions. Only fingerprint ridges are shown for clarity [2]. . . . .	75
4.2	Alignment of two impressions of the same finger using affine transformation. (a) and (b) are the gray scale images; (c) and (d) are the thinned (skeletonized) images of (a) and (b), respectively; and (e) shows the alignment based on the thinned images. Ridge lines do not align in (e). . . . .	78
4.3	An example of minutiae correspondences between two impressions of a finger. . . . .	82
4.4	An example of ridge curve correspondences between two impressions of a finger. . . . .	82
4.5	Non-linear deformations (with rotation and translation parameters removed) associated with two pairings involving the same template: (a) Template image; (b) and (c) Query images; (d) and (e) Non-linear deformation of (a) into (b) and (c), respectively. . . . .	84
4.6	Vector representation of ridge bifurcation used to establish correspondences between component ridge curves. $O$ marks the bifurcation points in correspondence, and $X$ marks the points on the ridges at Euclidean distance $d$ from $O$ . . . . .	88
4.7	The average deformation model (shown as deformations on a reference grid) corresponding to 6 templates of a finger sorted in increasing $\Phi$ -values. (a) is chosen to be the optimal template since it has the least $\Phi$ -value. . . . .	93
4.8	The average deformation model (shown as deformations on a reference grid) of 3 different fingers . . . . .	94
4.9	Examples of incorrect minutiae correspondences (a) resulting in erroneous ridge curve correspondences (b). . . . .	97
4.10	Effect of eliminating unreliable minutiae correspondences on the average deformation model; (a) template fingerprint, (b) average deformation model with $p = 100$ , and (c) average deformation model with $p = 60$ . . . . .	98
4.11	Improved alignment of template and query images using ridge curve correspondences (right panel). The alignment using minutiae correspondences are shown in the left panel. Both sets of alignment use the TPS warping model. . . . .	101
4.12	Improvement in matching performance when ridge curve correspondences is used to develop the average deformation model. . . . .	102
4.13	Matching performance when the $\Phi$ index of deformation is used to select optimal templates. Both optimal and suboptimal templates using ridge curve correspondences result in superior matching performance compared to minutiae correspondences. . . . .	102

5.1	Intra-class variation associated with an individual's face image. Due to change in pose, an appearance-based face recognition system will not be able to match these 3 images successfully, although they belong to the same individual [3]. . . . .	109
5.2	A bimodal biometric system showing the three levels of fusion; FU: Fusion Module, MM: Matching Module, DM: Decision Module. . . . .	111
5.3	The problem of face detection is compounded by the effects of complex lighting and cluttered background. . . . .	116
5.4	A Hand Geometry System. The sensor provides both the top and side views of the subject's hand. Features are extracted using the image of the top-view only. . . . .	117
5.5	Scatter Plot showing the genuine and impostor scores in 3-D space. The points correspond to 3,600 genuine scores (+) and 11,025 impostor scores (o). . . . .	119
5.6	ROC curves showing the performance of each of the three individual modalities. . . . .	120
5.7	ROC curves showing an improvement in performance when scores are combined using the sum rule. . . . .	121
5.8	ROC curves showing an improvement in performance when scores are combined using the sum rule. . . . .	122
5.9	Construction and Performance of the C5.0 Decision Tree. The performance is indicated by confusion matrices. . . . .	123
5.10	Linear Discriminant Analysis of the score vectors. The score vectors have been plotted in a 2-dimensional space representing the first and the second discriminant variables. There are 1,800 genuine score vectors(+) and 11,025 impostor score vectors (o). . . . .	124
5.11	Intra-class variability and inter-class similarity of hand geometry features depicted using Chernoff faces. (a) The feature sets associated with a user showing large variability. (b) and (c) The feature sets associated with two different users exhibiting similarity [4] . . . . .	128
5.12	The impostor distributions of the face biometric feature vector of 3 users. (a), (c) and (e) are the histogram of impostor scores associated with the 3 users. (b), (d) and (f) are the corresponding cumulative histograms. For $\gamma = 0.3$ it is observed that the thresholds for each of the users is different. . . . .	130
5.13	ROC curves exhibiting performance improvement when user-specific thresholds are utilized to verify claimed identity. (a) Fingerprint; (b) Face. . . . .	131
5.14	ROC curves when using (a) equal weights for all three traits, and a user-specific matching threshold, (b) user-specific weights for all three traits and a common matching threshold. . . . .	132
5.15	(a) and (b) Fingerprint images of user 4 whose ridge details are not very clear ( $w_1 = 0.2$ ); (c),(d) and (e) Varying face poses of user 3 ( $w_2 = 0.1$ ); (f) and (g) Incorrect placement of hand and the curved finger of user 2 ( $w_3 = 0.2$ .) . . . . .	135

A.1	Hand geometry sensing device. . . . .	149
A.2	The sixteen axes along which feature values are computed. . . . .	152
A.3	The gray scale profile of pixels along a measurement axis. . . . .	153
A.4	Computation of the feature vector. . . . .	156
A.5	Chernoff Faces representing the average feature vectors of 20 different hands.	157
A.6	Incorrect placement of hand. . . . .	159
A.7	The ROC curve depicting the matching performance of the hand geometry system. . . . .	160

# Chapter 1

## Introduction

### 1.1 Biometrics

A wide variety of systems require reliable personal authentication schemes to either confirm or determine the identity of individuals requesting their services. The purpose of such schemes is to ensure that the rendered services are accessed by a legitimate user, and not anyone else. Examples of these systems include secure access to buildings, computer systems, laptops, cellular phones and ATMs. In the absence of robust authentication schemes, these systems are vulnerable to the wiles of an impostor.

Traditionally, passwords (knowledge-based security) and ID cards (token-based security) have been used to restrict access to systems. However, security can be easily breached in these systems when a password is divulged to an unauthorized user or a card is stolen by an impostor; further, simple passwords are easy to guess (by an impostor) and difficult passwords may be hard to recall (by a legitimate user). The emergence of *biometrics* has addressed the problems that plague traditional verifica-

tion methods. Biometrics refers to the automatic identification (or verification) of an individual (or a claimed identity) by using certain physiological or behavioral traits associated with the person. By using biometrics it is possible to establish an identity based on ‘who you are’, rather than by ‘what you possess’ (e.g., an ID card) or ‘what you remember’ (e.g., a password). Biometric systems make use of fingerprints, hand geometry, iris, retina, face, hand vein, facial thermograms, signature, voiceprint, gait, palmprint, etc. (Figure 1.1) to establish a person’s identity [5, 6]. While biometric systems have their limitations [7], they have an edge over traditional security methods in that it is significantly difficult to lose, steal or forge biometric traits; further, they facilitate human recognition at a distance (e.g., face and gait).

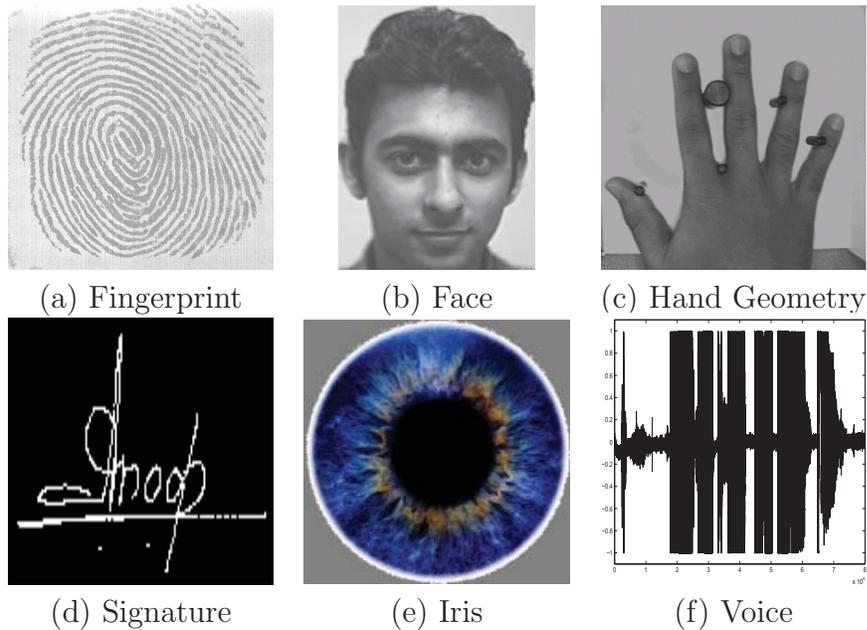


Figure 1.1: Examples of some of the biometric traits used for authenticating an individual.

Biometric systems also introduce an aspect of user convenience that may not be possible using traditional security techniques. For example, users maintaining

different passwords for different applications may find it challenging to recollect the password associated with a specific application. In some instances, the user might even forget the password, requiring the system administrator to intervene and reset the password for that user. A Meta Group study reports that a password-related help desk call may cost as much as \$30 in terms of support staff time and money [8]. Maintaining, recollecting, and remembering passwords can, therefore, be a tedious and expensive task. Biometrics, on the other hand, addresses this problem effectively, thereby enhancing user convenience: a user can use different ‘passwords’ (biometric traits) for different applications, with ‘password’ recollection not being an issue at all.

A typical biometric system operates by acquiring biometric data from an individual, extracting a feature set from the acquired data, and comparing this feature set against the template feature set in the database. In an *identification* scheme the comparison is done against templates corresponding to all the enrolled users in order to recognize the individual (a one-to-many matching); in a *verification* scheme, the comparison is done against only those templates corresponding to the claimed identity in order to verify the claim (a one-to-one matching). Thus, identification (“Whose biometric data is this?”) and verification (“Does this biometric data belong to Bob?”) are two different problems with different inherent complexities [9]. The templates are typically created at the time of enrollment, and depending on the application may or may not require human personnel intervention. Figure 1.2 illustrates the enrollment and verification modules of a typical biometric system.

The verification problem may be formally posed as follows: given an input feature

vector  $X_Q$  and a claimed identity  $I$ , determine if  $(I, X_Q)$  belongs to  $\omega_1$  or  $\omega_2$ , where  $\omega_1$  indicates that the claim is true (a genuine user) and  $\omega_2$  indicates that the claim is false (an impostor). Typically,  $X_Q$  is matched against  $X_I$ , the biometric template corresponding to user  $I$ , to determine its category. Thus,

$$(I, X_Q) \in \begin{cases} \omega_1 & \text{if } \mathcal{S}(X_Q, X_I) \geq \eta, \\ \omega_2 & \text{otherwise,} \end{cases} \quad (1.1)$$

where  $\mathcal{S}$  is the function that measures the similarity between  $X_Q$  and  $X_I$ <sup>1</sup>, and  $\eta$  is a predefined threshold. Therefore, every claimed identity is classified as  $\omega_1$  or  $\omega_2$  based on the variables  $X_Q$ ,  $I$ ,  $X_I$  and  $\eta$ , and the function  $\mathcal{S}$ .

The identification problem, on the other hand, may be stated as follows: given an input feature vector  $X_Q$ , determine the identity  $I_k$ ,  $k \in \{1, 2, \dots, N, N + 1\}$ . Here  $I_1, I_2, \dots, I_N$  are the identities enrolled in the system, and  $I_{N+1}$  indicates the reject case where no suitable identity can be determined. Hence,

$$X_Q \in \begin{cases} I_k & \text{if } \max_k \{\mathcal{S}(X_Q, X_{I_k})\} > \eta, k = 1, 2, \dots, N, \\ I_{N+1} & \text{otherwise,} \end{cases} \quad (1.2)$$

where  $X_{I_k}$  is the biometric template corresponding to identity  $I_k$ , and  $\eta$  is a predefined threshold.

Biometric systems are being increasingly deployed in large scale civilian applications (Figure 1.3). The Schiphol Privium scheme at the Amsterdam airport, for

---

<sup>1</sup>The value  $\mathcal{S}(X_Q, X_I)$  is termed as a similarity score or matching score

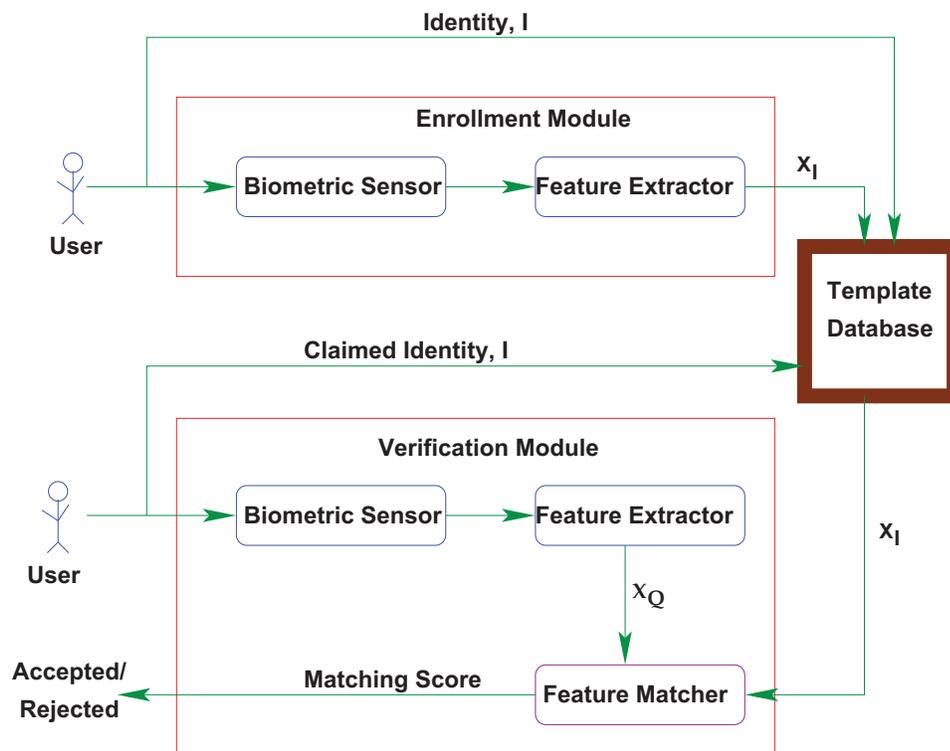


Figure 1.2: The enrollment module and the verification module of a biometric system.

example, employs iris scan cards to speed up the passport and visa control procedures [10]. Passengers enrolled in this scheme insert their card at the gate and look into a camera; the camera acquires the eye image of the traveller and processes it to locate the iris, and compute the Iriscode [11]; the computed Iriscode is compared with the data residing in the card to complete user verification. A similar scheme is also being used to verify the identity of Schiphol airport employees working in high-security areas. Thus, biometric systems can be used to enhance user convenience while improving security.

A simple biometric system has four important modules:

1. *Sensor Module* which captures the biometric data of an individual. An example is a fingerprint sensor that captures fingerprint impressions of a user.
2. *Feature Extraction Module* in which the acquired data is processed to extract feature values. For example, the position and orientation of minutiae points in a fingerprint image would be computed in the feature extraction module of a fingerprint system.
3. *Matching Module* in which the feature values are compared against those in the template by generating a matching score . For example, in this module, the number of matching minutiae between the query and the template can be computed and treated as a matching score.
4. *Decision-making Module* in which the user's claimed identity is either accepted or rejected based on the matching score generated in the matching module



Figure 1.3: Biometric systems in civilian applications. (a) A border passage system using iris recognition at London’s Heathrow airport ([news.bbc.co.uk](http://news.bbc.co.uk)). (b) The INS Passenger Accelerated Service System (INSPASS) at JFK international airport (New York) uses hand geometry to authenticate travellers and significantly reduce their immigration inspection processing time ([www.panynj.gov](http://www.panynj.gov)). (c) Ben Gurion airport in Tel Aviv (Israel) uses Express Card entry kiosks fitted with hand geometry systems for security and immigration ([www.airportnet.org](http://www.airportnet.org)). (d) The FacePass system from Viisage is used in point-of-sale verification applications like ATMs, therefore, obviating the need for PINs ([www.viisage.com](http://www.viisage.com)). (e) Indivos’ “Pay by Touch” service uses fingerprints to help customers speed up payments in restaurants and cafeterias. When an enrolled customer places her finger on the sensor, the system retrieves her financial account and updates it ([www.kioskbusiness.com](http://www.kioskbusiness.com)). (f) The Identix TouchClock fingerprint system is used in time and attendance applications ([www.cardsolutions.com](http://www.cardsolutions.com)).

(verification). Alternately, the system may identify a user based on the matching scores (identification).

## 1.2 Fingerprint as a Biometric

Among all biometric traits, fingerprints have one of the highest levels of reliability [12] and have been extensively used by forensic experts in criminal investigations [13]. A fingerprint refers to the flow of ridge patterns in the tip of the finger. The ridge flow exhibits anomalies in local regions of the fingertip (Figure 1.4), and it is the position and orientation of these anomalies that are used to represent and match fingerprints. Although not scientifically established, fingerprints are believed to be unique across individuals, and across fingers of the same individual [14]. Even identical twins having similar DNA, are believed to have different fingerprints [15]. Traditionally, fingerprint patterns have been extracted by creating an inked impression of the fingertip on paper. The electronic era has ushered in a range of compact sensors that provide digital images of these patterns. These sensors can be easily incorporated into existing computer peripherals like the mouse or the keyboard (Figure 1.5), thereby making this mode of identification a very attractive proposition. This has led to the increased use of *automatic* fingerprint-based authentication systems in both civilian and law-enforcement applications.

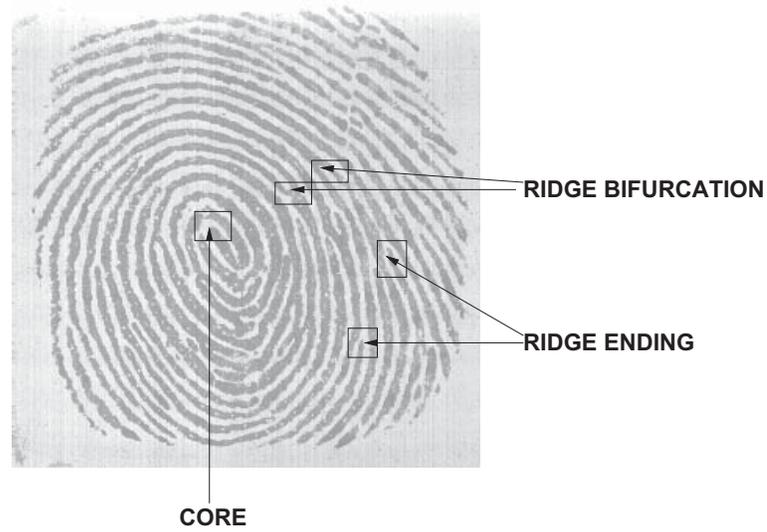


Figure 1.4: A fingerprint image with the core and four minutiae points marked on it. The ridge pattern along with the core and delta points define the global configuration, while the minutiae points define the local structure.

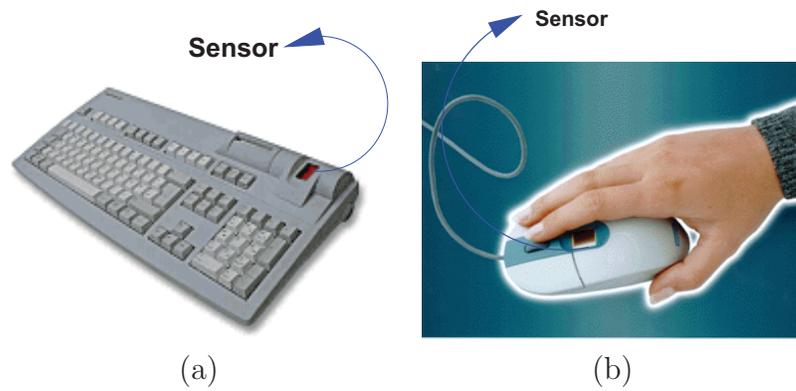


Figure 1.5: Fingerprint sensors installed on (a) a keyboard (the Cherry Biometric Keyboard has a smart card reader and a fingerprint sensor attached to it); (b) a mouse (the ID Mouse manufactured by Siemens has a capacitance-based fingerprint sensor placed on a USB mouse).

### 1.2.1 Fingerprint Representation

The uniqueness of a fingerprint is determined by the topographic relief of its ridge structure and the presence of certain ridge anomalies termed as minutiae points. Typically, the global configuration defined by the ridge structure is used to determine the class [16, 17] of the fingerprint, while the distribution of minutiae points is used to match and establish the similarity between two fingerprints [1, 18]. Automatic fingerprint identification systems, that match a query print against a large database of prints (which can consist of millions of prints), rely on the pattern of ridges in the query image to narrow their search in the database (*fingerprint indexing*), and on the minutiae points to determine an exact match (*fingerprint matching*). The ridge flow pattern itself is seldom used for matching fingerprints.

### 1.2.2 Fingerprint Matching

Fingerprint matching techniques can be broadly classified as being minutiae-based or correlation-based. Minutiae-based techniques attempt to align two sets of minutiae points and determine the total number of matched minutiae [19, 20, 1]. Correlation-based techniques, on the other hand, compare the global pattern of ridges and furrows to see if the ridges in the two fingerprints align [21, 22, 23, 24]. The performance of minutiae-based techniques rely on the accurate detection of minutiae points and the use of sophisticated matching techniques to compare two minutiae sets which undergo non-rigid transformations. The performance of correlation-based techniques is affected by non-linear distortions and noise present in the image. In general, it has

been observed that minutiae-based techniques perform better than correlation-based ones. Correlation-based techniques suffer from the following problems [25]: (a) A fingerprint image may have non-linear warping due to the effect of pressing a convex elastic surface (the finger) on a flat surface (the sensor). Moreover, various sub-regions in the sensed image are distorted differently due to the non-uniform pressure applied by the user. It is difficult to compare two such distorted prints, even if translation and rotation effects are considered. (b) Based on the moisture content of the skin, the acquired images may have either thin or thick ridges. Further, the quality of the images acquired using the sensor may vary with time, thereby complicating the correlation process.

### 1.2.3 Difficulties and Challenges in Fingerprint Matching

Although the problem of *automatic* fingerprint matching has been extensively studied<sup>2</sup>, it is nevertheless, not a fully solved problem. There are a variety of unresolved issues that need to be addressed effectively in this rather popular biometric technique. Some of the challenges are described below:

1. The new generation solid-state sensors are being increasingly used to acquire fingerprint images. These sensors when embedded in compact systems like laptops, mouse, and cellular phones provide a small contact area (e.g., 0.6"  $\times$  0.6" in a Veridicom sensor) for the fingertip and, therefore, sense only a limited portion of the fingerprint. This complicates the problem of matching impressions due

---

<sup>2</sup>The earliest *published* work on automatic fingerprint matching and classification can be traced back to the 1960s [26, 27, 28].



accounted for prior to the matching stage.

4. While it is acknowledged that the fingerprints of a person do not change over time, it is possible for minor cuts and bruises to alter the ridge structure of fingerprints (Figure 1.7). Moreover, the moisture content of the fingertip may change over time affecting the quality of the fingerprint image being acquired from a user. The template fingerprint data obtained during enrollment time may not capture these variations. A protocol to update template data is necessary to maintain system performance.
5. Some users consistently provide poor quality fingerprints due to the dry nature of their skin (Figure 1.8). It is difficult to extract features from such poor quality images. Users providing such noisy fingerprint data might find it difficult to enroll in and interact with a biometric system that uses only fingerprints. To address this issue, a multibiometric system, that uses other biometric traits in addition to fingerprints, has to be considered.

### **1.3 Thesis Contributions**

In this thesis, multiple sources of information are consolidated to enhance the performance of automatic fingerprint authentication systems. A few of the challenges presented in the earlier section are, consequently, addressed. The four major contributions of this thesis are listed below.

1. A hybrid fingerprint matcher that avails of both the minutiae and texture infor-

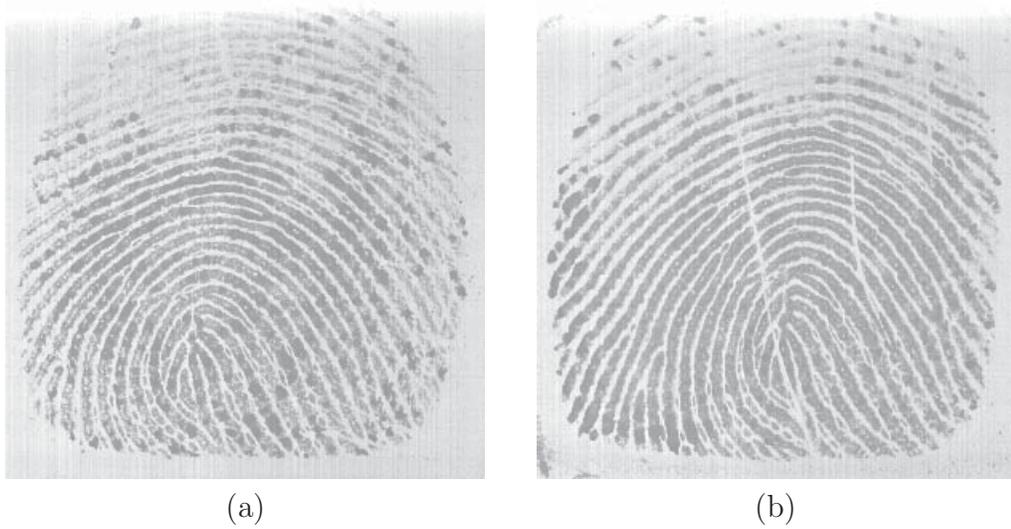


Figure 1.7: Effect of noisy images on a fingerprint authentication system. (a) Fingerprint obtained from a user during enrollment. (b) Fingerprint obtained from the same user during verification. The development of scars or cuts can result in erroneous fingerprint matching results.

mation present in fingerprint images has been developed. The texture information is extracted by applying a set of 8 Gabor filters to an enhanced fingerprint image, and texture features are represented using ridge feature maps.

2. To address the problem of partial prints, a fingerprint mosaicking scheme has been developed. The proposed scheme examines two partial impressions of a finger and constructs a composite template image that includes information from the individual prints.
3. An average deformation model for fingerprint images has been proposed. This model accounts for the non-linear distortions present in fingerprint images. The model is developed by comparing a fingerprint impression with several other impressions of the same finger and observing the common ridge points that

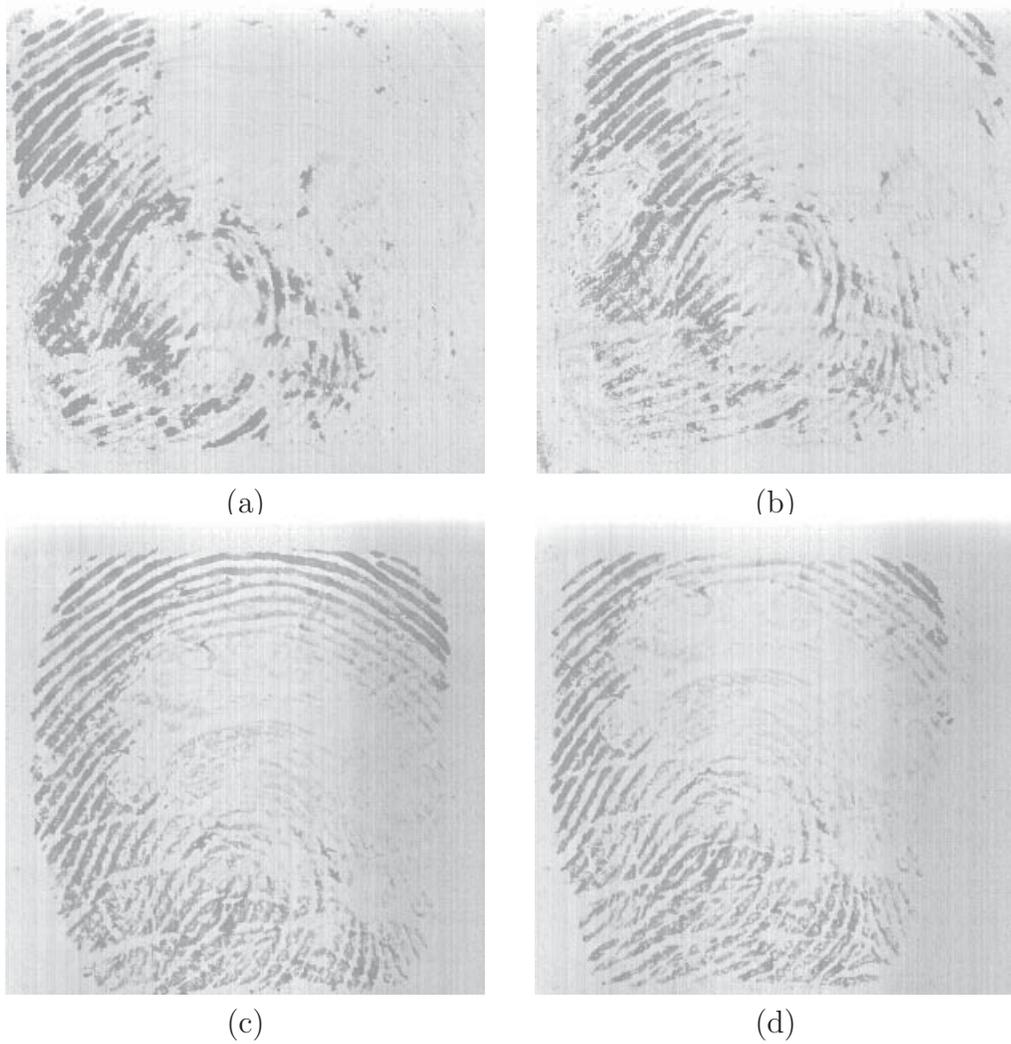


Figure 1.8: Non universality of fingerprints: Four different impressions of a subject's finger exhibiting poor quality ridges. A fingerprint system might not be able to enroll this subject since minutiae information cannot be reliably extracted.

occur in them.

4. To enhance the performance of a fingerprint system, the face and hand geometry traits of a subject are also used. By fusing information gleaned from multiple biometric indicators, the performance of a biometric system can be improved. The proposed multibiometric system also employs a learning technique to compute user-specific parameters in order to improve verification performance.

In the subsequent chapters, a detailed description of each of these contributions is provided.

## Chapter 2

# Fingerprint Representation using Ridge Feature Maps

Most fingerprint matching systems rely on the distribution of minutiae on the fingertip to represent and match fingerprints. The advent of solid-state sensors presents new opportunities as well as fresh challenges to traditional fingerprint matching algorithms. These sensors provide only a small contact area ( $0.6'' \times 0.6''$ ) for the fingertip and, therefore, sample only a limited portion of the fingerprint pattern (e.g.,  $300 \times 300$  pixels at 500 *dpi*). An optical sensor, on the other hand, may have a contact area of approximately  $1'' \times 1''$ , resulting in images of size  $480 \times 508$  pixels at 500 *dpi*. Hence, the number of minutiae points that can be extracted from a fingerprint sample acquired using a solid-state sensor is smaller compared to that acquired using an optical sensor. Figures 2.1 and 2.2 illustrate this difference. Also, multiple impressions of the same finger, acquired at different instances using a solid-state sensor, may overlap over a small region region due to the rotation and translation of subsequent

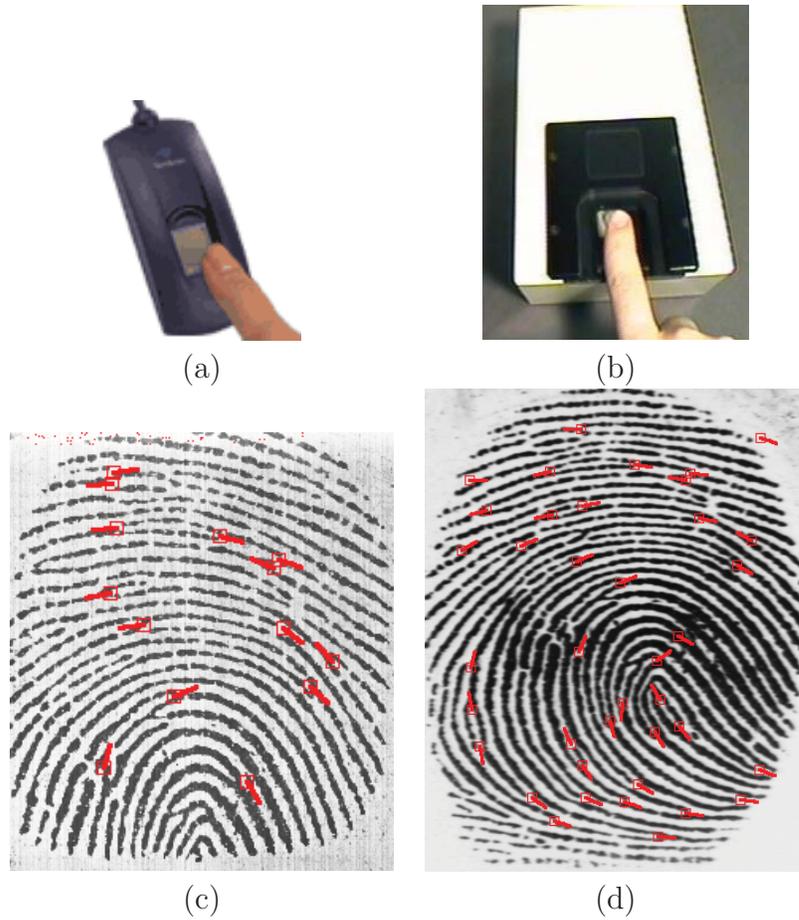


Figure 2.1: Fingerprint images acquired using the solid state Veridicom sensor (a) and the optical Digital Biometrics sensor (b). The detected minutiae points have been marked in both the fingerprint images. 14 minutiae have been detected in the first case (c), and 39 in the second (d).

fingerprints. The minutiae-based matching schemes will not perform well in such situations due to the lack of a sufficient number of common minutiae points between the two impressions. Thus, there is a need to explore complimentary representations of fingerprints.

While the ridge flow pattern is generally used for classifying fingerprints, it is seldom used in matching. We develop a hybrid fingerprint matching scheme that uses *both* minutiae and ridge flow information to represent and match fingerprints.

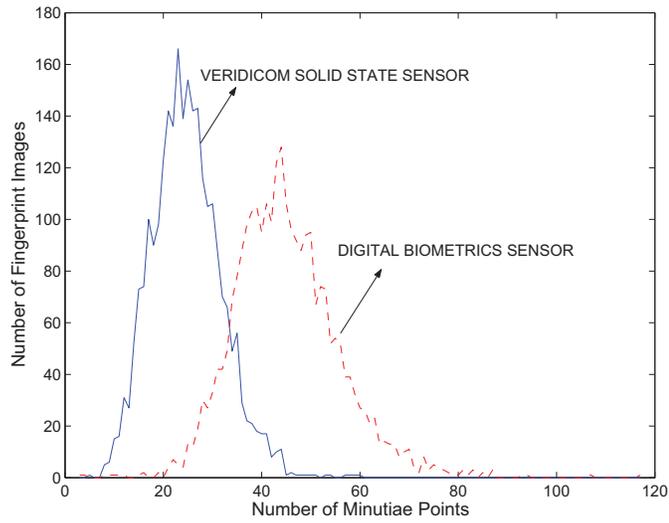


Figure 2.2: Histogram of the number of minutiae points extracted from images acquired using the Veridicom and Digital Biometric sensors. A total of 2,500 fingerprint impressions were used to compute these histograms for each sensor. The histograms suggest that substantially fewer minutiae points are available in images obtained from solid-state sensors.

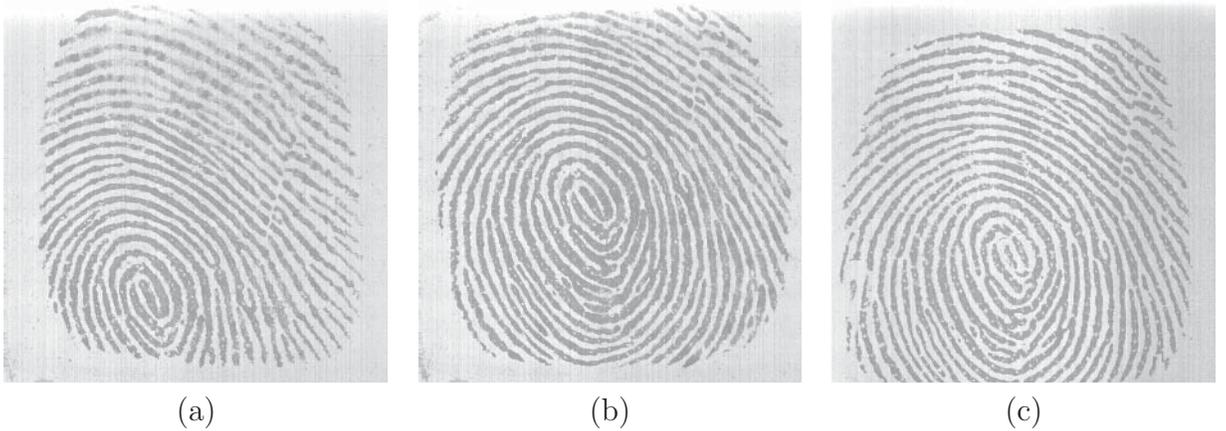


Figure 2.3: Three impressions of a user's fingerprint exhibiting partial overlap: the overlap between (a) and (b), and (a) and (c) is limited.

A set of 8 Gabor filters, whose spatial frequencies correspond to the average inter-ridge spacing in fingerprints, is used to capture the ridge strength at equally spaced orientations. A square tessellation of the filtered images is then used to construct an eight-dimensional feature map, called the ridge feature map. The ridge feature map along with the minutiae set of a fingerprint image is used for matching purposes. The proposed technique has the following features: (i) the entire image is taken into account while constructing the ridge feature map; (ii) minutiae matching is used to determine the translation and rotation parameters relating the query and the template images for ridge feature map extraction; (iii) filtering and ridge feature map extraction are implemented in the frequency domain thereby speeding up the matching process; (iv) filtered query images are cached to greatly increase the one-to-many matching speed. The hybrid matcher performs better than a minutiae-based fingerprint matching system.

## 2.1 Introduction

The ridge pattern in a fingerprint may be viewed as an oriented texture pattern having a fixed dominant spatial frequency and orientation in a local neighborhood. The frequency is due to the inter-ridge spacing present in the fingerprint (figure 2.4(a)), and the orientation is due to the flow pattern exhibited by the ridges (figure 2.4(b)). By capturing the frequency and orientation of ridges in non-overlapping local regions in the fingerprint, a distinct representation of the fingerprint is possible. One such representation has been discussed in [29]. However, to match two fingerprints

using such a representation, a suitable alignment of the underlying ridge structures is essential.

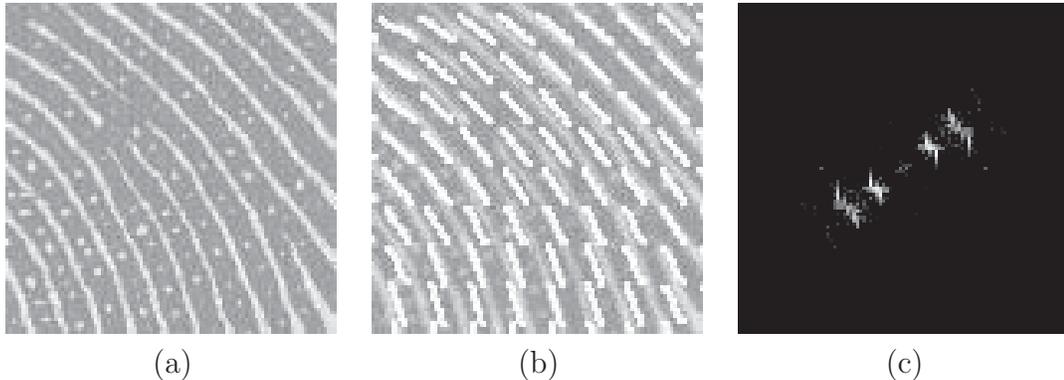


Figure 2.4: Fingerprint as an oriented texture pattern. (a) the constant inter-ridge spacing in a local region of the fingerprint; (b) the dominant direction of the ridges in (a); (c) the power spectrum of (a).

We describe a fingerprint representation scheme, that constructs a feature map by observing the local ridge orientation in a fingerprint image. The local ridge characteristics are extracted via a set of Gabor filters that are pre-tuned to a specific frequency corresponding to the average inter-ridge spacing in a fingerprint image. An input fingerprint image is filtered using this set of Gabor filters; a square tessellation [30] is then applied to each filtered image to examine the local response to the filter; a feature vector which measures the energy in the filtered images for each of the tessellated cells is next obtained. A collection of these feature vectors (over the tessellation) constitutes the ridge feature map used to represent a fingerprint. Fingerprint matching entails determining the similarity between two such ridge feature maps. This representation is used along with the minutiae set of the fingerprint image for matching purposes. The proposed representation and matching scheme are

motivated by the following observations:

1. Global image information, as defined by the ridge pattern of the fingerprint, is not being explicitly used during the matching phase in most of the current matching systems. We believe that the ridge pattern, when observed at various resolutions and orientations, provides discriminatory information that can be used for matching fingerprints.
2. Minutiae information may not be very discriminative in the case of solid-state sensors which typically capture only a small area of the fingertip. For example, the average number of minutiae points extracted from Digital Biometrics optical sensor images ( $500 \times 500$  image at 500 dpi) is 45 compared to 25 minutiae obtained from Veridicom solid-state sensor images ( $300 \times 300$  image at 500 dpi). Alternate representations, to supplement minutiae information, are necessary to maintain sufficient fingerprint identification performance in such cases. Further, in poor quality images, while it is difficult to accurately locate minutiae points, the ridge pattern features may be easier to detect.

## 2.2 Fingerprint as Oriented Texture

Jain *et al.* [29] have proposed a novel representation scheme that captures global and local features of a fingerprint in a compact fixed length feature vector termed as *FingerCode*. This technique makes use of the texture features available in a fingerprint to compute the feature vector. Their scheme for generic representation of oriented texture relies on extracting a core point in the fingerprint. A circular region around

the core point is located and tessellated into sectors as shown in Figure 2.5(a). The pixel intensities in each sector are normalized to a constant mean and variance, and filtered using a bank of 8 Gabor filters to produce a set of 8 filtered images. Gray scale variance within a sector quantifies the underlying ridge structures and is used as a feature. The 640-dimensional feature vector is the collection of all the features, computed from all the 80 sectors, in each of the filtered image. The FingerCode captures the *local* information, and the ordered enumeration of the tessellation captures the invariant *global* relationships among the local patterns. The matching stage simply computes the Euclidean distance between the two corresponding FingerCodes. This technique, however, suffers from the following shortcomings:

1. The frame of reference is based on a global singular point (i.e., the core point). Detection of the core point is non-trivial; furthermore, the core point may not even be present in the captured fingerprint images.
2. The alignment is based on a single reference point and is, therefore, not very robust with respect to errors in the location of the reference point.
3. The tessellation does not cover the entire image. Furthermore, if the core were to be detected close to the boundary of the image, the tessellation will include an extremely small portion of the image (Figure 2.5(c)).

The technique proposed here has the following advantages:

1. Unlike in [29], the filtering is done on the enhanced images rather than the raw input images. The enhanced images have lower noise content than the raw images.

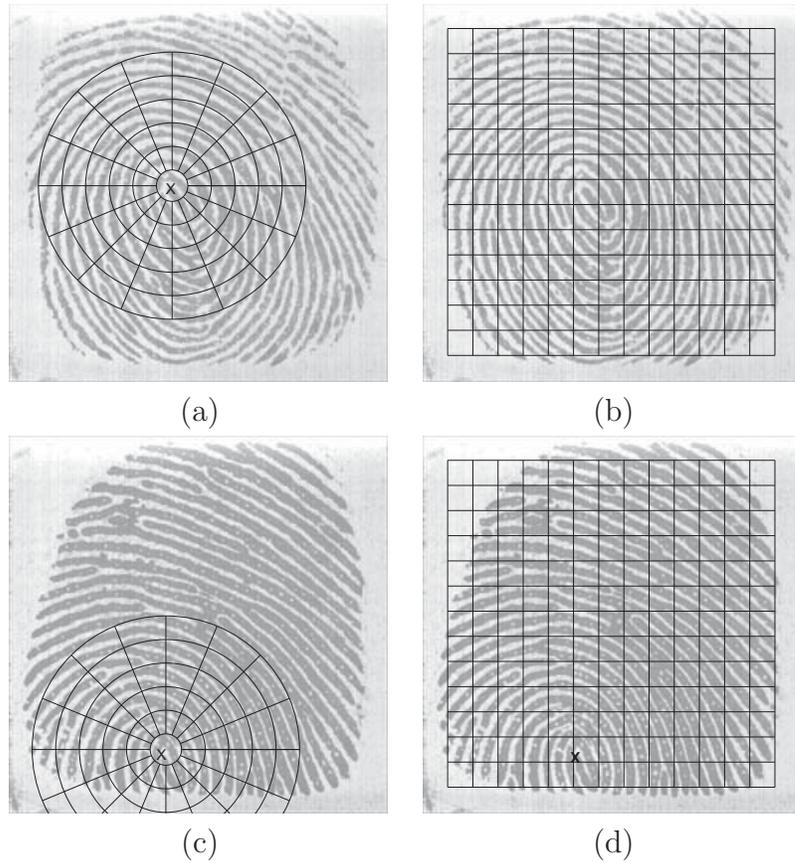


Figure 2.5: Tessellating the fingerprint image using a circular and a square grid. The square tessellation, unlike the circular one, is not affected by the location of the core point in the image. (a) Circular tessellation (80 sectors) about a core point. (b) Square tessellation (169 cells) over the entire image. (c) Circular tessellation about a core detected close to the boundary of the image. (d) Square tessellation over image shown in (c). The images were acquired using the Veridicom sensor.

2. Instead of using circular tessellation, a square tessellation is used (Figure 2.5(b)). The tessellation covers the entire image, and all the tessellated cells are of the same size. Moreover, the tessellation is not based on detecting any landmark points.
3. The fingerprint images are aligned using the overall minutiae information; this is more robust than using only the core point for aligning image pairs.
4. In the absence of reliable minutiae information, we describe a correlation based technique that utilizes ridge feature maps alone.

## 2.3 Image Filtering using Gabor Filters

A  $2D$  Gabor filter can be thought of as a complex plane wave modulated by a  $2D$  Gaussian envelope. These filters optimally capture both local orientation and frequency information<sup>1</sup> and their development was motivated by observing the linear response of the receptive field in simple striate cortex cells. By tuning a Gabor filter to a specific frequency and direction, the local frequency and orientation information can be obtained. Thus, they are suited for extracting texture information from images. Daugman has successfully used these filters to extract salient features from the human iris [11].

An even symmetric Gabor filter has the following general form in the spatial

---

<sup>1</sup>They are optimal in the sense that they try to minimize simultaneously the joint space-spatial frequency uncertainty [31, 32].

domain:

$$\begin{aligned}
 G_{\theta,f}(x,y) &= \exp\left\{\frac{-1}{2}\left[\frac{x'^2}{\delta_x^2} + \frac{y'^2}{\delta_y^2}\right]\right\} \cos(2\pi f x'), & (2.1) \\
 x' &= x \sin\theta + y \cos\theta, \\
 y' &= x \cos\theta - y \sin\theta,
 \end{aligned}$$

where  $f$  is the frequency of the sinusoidal plane wave at an angle  $\theta$  with the  $x$ -axis, and  $\delta_x$  and  $\delta_y$  are the standard deviations of the Gaussian envelope along the  $x$  and  $y$  axes, respectively.

For extracting the response of the ridge at various orientations of the Gabor filter, the parameters  $(f, \delta_x, \delta_y, \theta)$  are set to the following values:

(i) The frequency,  $f$ , corresponds to the inter-ridge distance in fingerprint images.

For the  $300 \times 300$  (500 dpi) images obtained using the Veridicom sensor and resized to  $240 \times 240$  (see section 2.6.5), the average inter-ridge spacing is about 8 pixels.

Hence,  $f = \frac{1}{8} = 0.125$ .

(ii) The selection of the standard deviation values,  $\delta_x$  and  $\delta_y$ , involves a trade-off.

Larger values are more robust to noise, but will not capture ridge information at a fine level. Smaller values, on the other hand, are less robust to noise in the image,

but capture ridge information very well. Based on empirical data [33], both these

values were set to 4, i.e.,  $\delta_x = \delta_y = \delta = 4$ .

(iii) Eight different orientations are examined. These correspond to  $\theta$  values of  $0^\circ$ ,

$22.5^\circ$ ,  $45^\circ$ ,  $67.5^\circ$ ,  $90^\circ$ ,  $112.5^\circ$ ,  $135^\circ$ ,  $157.5^\circ$  (Figure 2.6).

These parameters are fixed during the matching process, allowing for pre-storing

the Gabor filter representations in a lookup table referred to as the Gabor filter bank. This filter bank precalculates the Fourier representation of the Gabor filter for all orientations of interest. This formulation substantially improves the matching time in a one-to-many matching scheme.

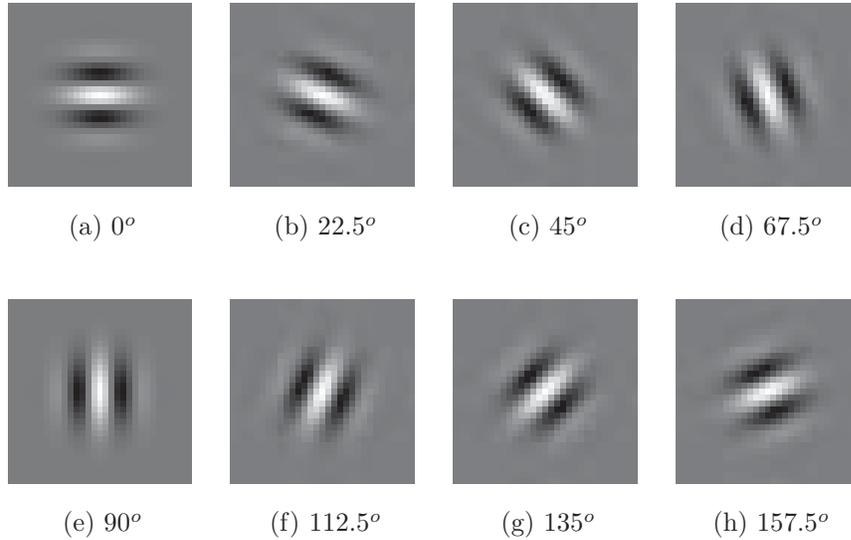


Figure 2.6: Gabor filters in spatial domain with eight different orientations used for feature extraction.  $f = 0.125$ ,  $\delta_x = \delta_y = \delta = 4$ .

### 2.3.1 Fingerprint Enhancement

Enhancement is the process by which the clarity of the ridge and furrow structures in the fingerprint images is improved to facilitate the feature extraction process [33] [34]. Fingerprint enhancement helps in reducing the noise content in the fingerprint image. Enhancement, however, can also introduce false ridges, resulting in spurious or missing minutiae points. Since the ridge feature map representation proposed here relies on the *dominant* ridge directions in each tessellated cell, the introduction of

false ridges is not a serious problem. The minutiae features are also extracted after processing the enhanced fingerprint image. The enhancement algorithm is based on the technique described in [33]. Figure 2.7 shows a fingerprint image before and after enhancement.

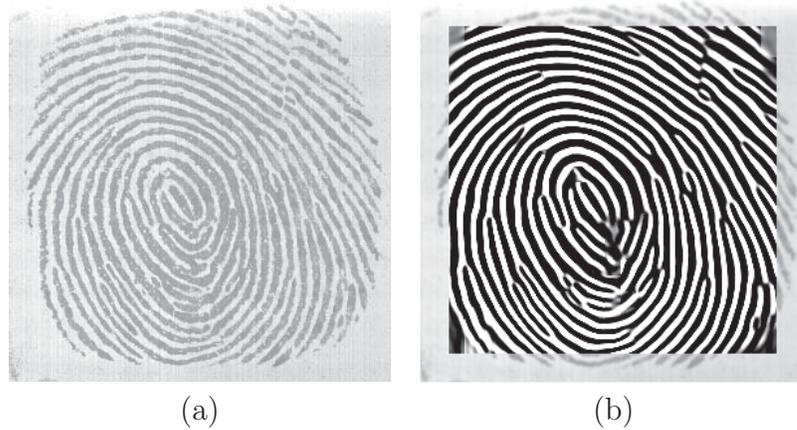


Figure 2.7: Fingerprint image (a) before and (b) after enhancement.

### 2.3.2 Fingerprint Segmentation

The ridge feature map is constructed using the feature values computed at each tessellated cell. Certain cells may predominantly contain background information, and therefore, the feature values computed at these cells will not be an accurate indication of ridge strength. Thus, the purpose of segmentation is to separate the foreground and background regions in a given fingerprint image. The foreground corresponds to those regions in the image that have relevant fingerprint information (i.e., the ridges and valleys of the fingerprint), while the background represents those regions that do not have the relevant information. Cells with predominantly background information

are not used during the matching stage. Segmentation is done by observing the local variation of intensity on the original gray-scale image [1].

The algorithm to segment a fingerprint image is as follows:

1. The intensity of the input fingerprint image is normalized to the range  $[0, 1]$ . Such a normalization ensures that the effect of any rapid changes in intensity over the image is minimized.
2. The image is binarized by selecting a threshold that corresponds to one-half of the mean of the normalized image intensity (Figure 2.8(b)).
3. A  $3 \times 3$  median filter is applied to the binarized image. This operation removes any undesirable “salt-and-pepper” noise that may be present in the binarized image (Figure 2.8(c)).
4. An edge detection algorithm is applied to the binary image to get the outline of ridges (Figure 2.8(d)).
5. Graham’s convex hull algorithm [35] is applied on the edge image to generate a polygon that segments the fingerprint image (Figure 2.8(e)).

### 2.3.3 Filtering Enhanced Image

Filtering requires convolving the enhanced image,  $H$ , with each of the 8 Gabor filters in the spatial domain. However, convolution in the spatial domain would be extremely slow. For example, a  $256 \times 256$  image convolved with a  $16 \times 16$  filter would need  $\sim 10^7$  multiplications (assuming that the convolution operation has not been optimized). In

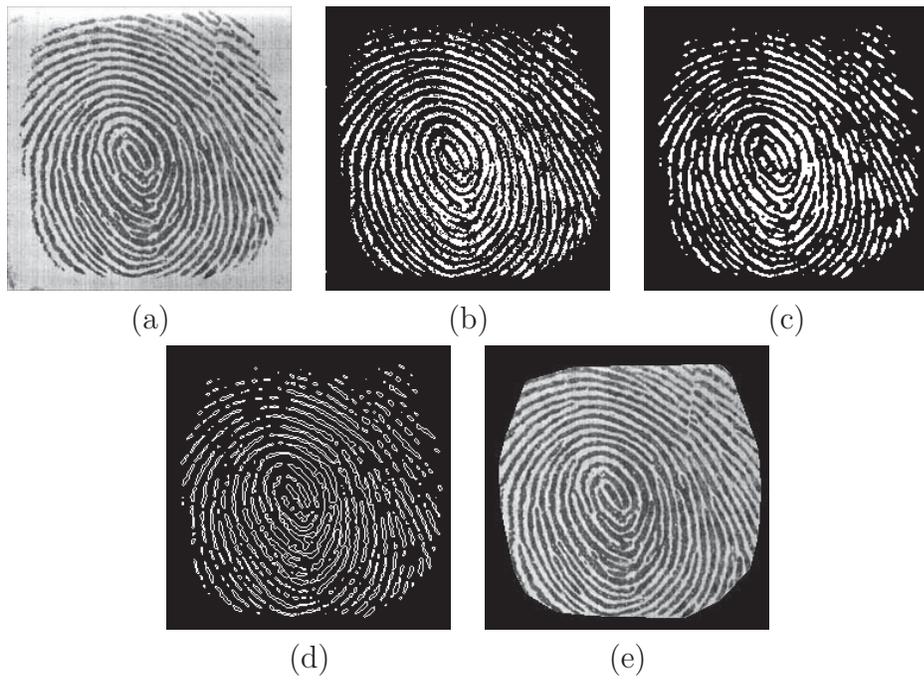


Figure 2.8: Segmenting a fingerprint image. (a) The original fingerprint image. (b) The binarized fingerprint image. (c) The median filtered image. (d) The edge image outlining the ridges. (e) The segmented fingerprint image.

order to speed-up this operation, we perform the convolution in the frequency domain. Let  $\mathcal{F}(H)$  denote the discrete Fourier transform of  $H$ , and let  $\mathcal{F}(G_\theta)$  indicate the discrete Fourier transform of the Gabor filter having the spatial orientation  $\theta$  as described by Equation (2.2). Thus, the Gabor filtered image,  $V_\theta$ , may be obtained as,

$$V_\theta = \mathcal{F}^{-1}[\mathcal{F}(H)\mathcal{F}(G_\theta)], \quad (2.2)$$

where  $\mathcal{F}^{-1}$  is the inverse Fourier transform. Eight filtered images are obtained as a result of this filtering (Figure 2.9).

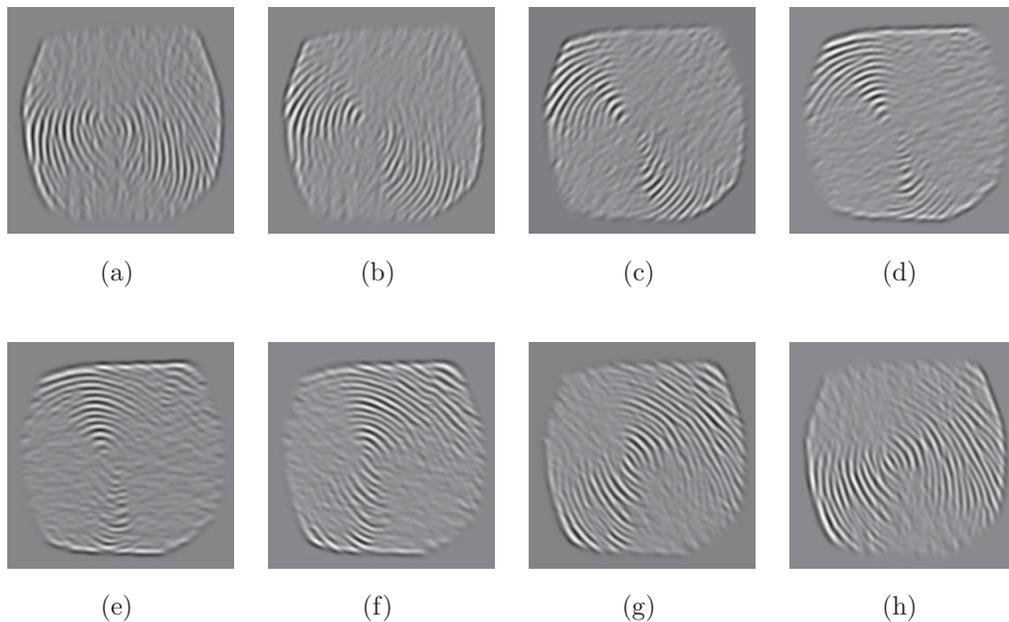


Figure 2.9: Results of the filtering process on the image shown in Figure 2.7(b). The 8 images correspond to the 8 different orientations of the Gabor filter.

## 2.4 Ridge Feature Maps

### 2.4.1 Tessellation of Filtered Images

While a filtered image in its entirety can be used as a representation scheme, the presence of local distortions would adversely affect the matching process. Moreover, it is the local variations in ridge structure (combined with the global ridge configuration) that provide a better representation of the fingerprint. To examine local variations, the image is tessellated into square cells, and features from each of the cells are computed (Figure 2.10). The size of a cell is chosen to correspond to approximately the width of two ridges ( $16 \times 16$ ). A 8 pixel wide border of the image is not included in the tessellation. This results in  $n_c = 15$  cells in each row and column of the square grid. The total number of tessellated cells over the image is, therefore,  $N_c = 225$ . The *variance* of the pixel intensities in each cell across all filtered images is used as a feature vector. The variance corresponds to the energy of the filter response, and is, therefore, a useful measure of ridge orientation in a local neighborhood. Those tessellated cells that contain a certain proportion of background pixels are labeled as background cells and the corresponding feature value is set to 0.

### 2.4.2 Ridge Feature Map Definition

Let  $C_\theta(i, j)$  refer to the  $(i, j)$ th cell in the square grid that is placed on the filtered image  $V_\theta$ . The variance,  $\sigma_\theta^2(i, j)$ , represents the feature value corresponding to the cell. Thus, for each  $V_\theta$ , a feature map of variance values can be obtained. Let  $R_\theta$

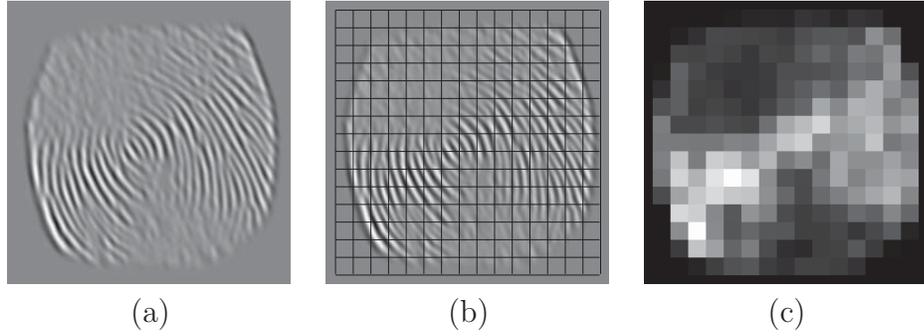


Figure 2.10: Tessellating the filtered image. (a) A fingerprint image filtered with a Gabor filter oriented at  $157.5^\circ$ . (b) A square tessellation of the filtered image. (c) The ridge feature map ( $n_c \times n_c$ ) representation of the fingerprint.

denote the feature map associated with the filtered image  $V_\theta$ . Then,

$$R_\theta = \{\sigma_\theta^2(i, j)\}, \quad (2.3)$$

where  $\theta \in \{0^\circ, 22.5^\circ, 45^\circ, 67.5^\circ, 90^\circ, 112.5^\circ, 135^\circ, 157.5^\circ\}$ ,  $i = 1 \dots n_c$ ,  $j = 1 \dots n_c$ .

An eight-dimensional feature map corresponding to the 8 filtered images is obtained in this way (Figure 2.11). These ridge feature maps are used to represent and match a query image with a template.

## 2.5 Minutiae Extraction

Minutiae extraction refers to the process by which the minutiae points are detected in a fingerprint image. Each minutiae is characterized by its  $(x, y)$  location in the image, and the orientation  $\theta$  of the ridge on which it is detected. The ridge information in a  $64 \times 64$  region around the  $(x, y)$  point is associated with every minutiae which is useful when two minutiae sets are being matched. The minutiae extraction scheme

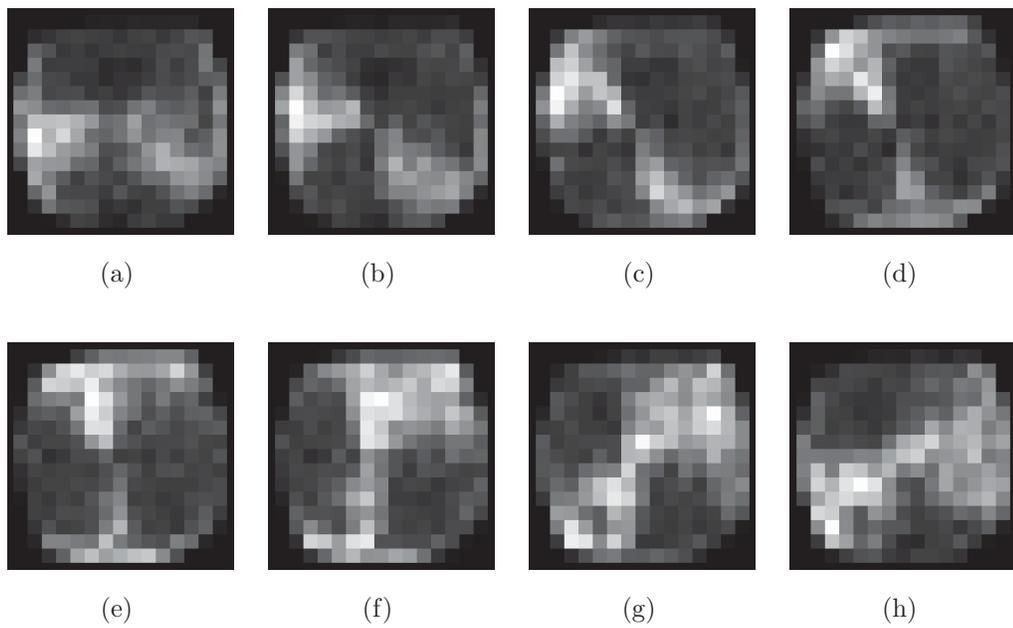


Figure 2.11: Feature maps representing the variance in intensity in the filtered images for each cell. For purposes of visualization, the feature values have been scaled to the 0 - 255 range.

(Figure 2.12) can be broadly classified into the following stages:

(i) Orientation Field Estimation: The orientation of the fingerprint image is computed in non-overlapping blocks by examining the gradients of pixel intensities in the  $x$  and  $y$  directions within the block.

(ii) Ridge Detection: The ridges present in the fingerprint image are identified by applying masks that are capable of accentuating the local maximum gray level values along the normal direction of the local ridge direction.

(iii) Ridge Thinning: The ridge map constructed in the earlier stage is used to obtain a thinned ridge image.

(iv) Minutiae Detection: A set of rules is applied to the thinned ridges to label minutiae points (ridge endings and ridge bifurcations). As a postprocessing step, a refinement algorithm is applied to remove spurious minutiae points.

Minutiae matching involves a point matching operation on the two minutiae sets. Jain et al. apply an elastic string matching technique to compare the two minutiae sets [1]. The output of the matching process is a matching score that indicates the similarity of the two minutiae sets being compared, and a correspondence map that indicates pairing of minutiae points from the two sets. The correspondence map is used to compute the transformation parameters necessary to align the two fingerprint images.

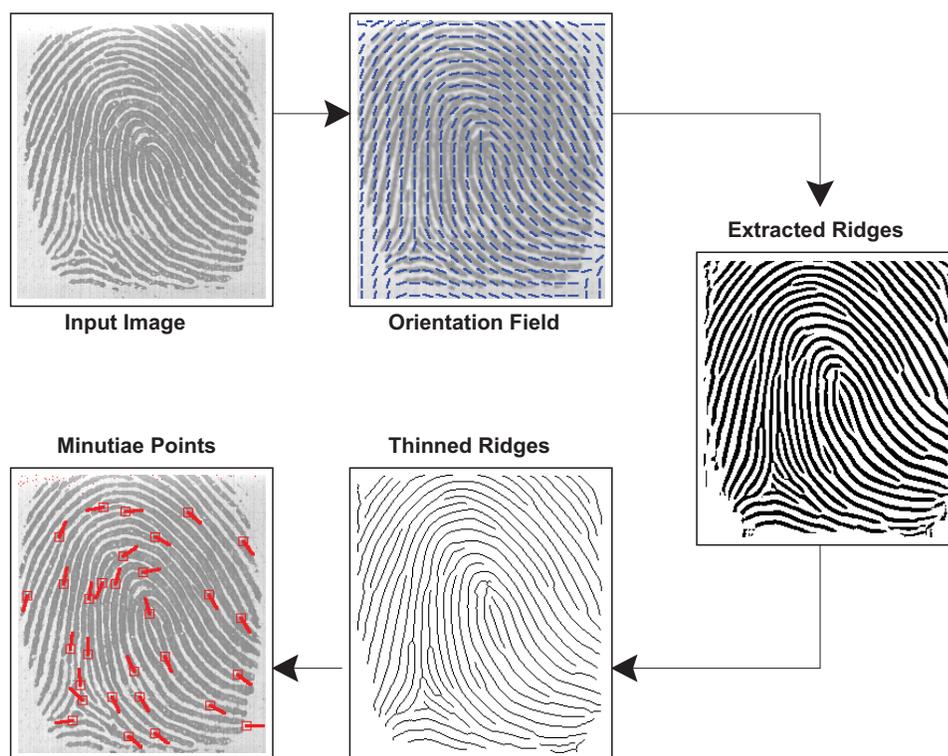


Figure 2.12: Flowchart of the minutiae extraction algorithm [1].

## 2.6 Alignment Using Minutiae Information

The process of fingerprint matching involves comparing a query print with a set of one or more template prints. Prior to the matching process, features are extracted from all the template images (Figure 2.13). The hybrid fingerprint matcher proposed here utilizes two distinct sets of fingerprint information for matching fingerprints: minutiae features, and ridge feature maps. When a query image is presented, the matching proceeds as follows: (i) the query and template minutiae features are matched to generate a minutiae matching score and a transformation parameter (translation and rotation) that relates the query and template fingerprints; (ii) the rotation parameter is used to rotate the 8 Gabor filters and the modified filters are applied to the query image; (iii) the filtered query images are then translated and rotated according to the parameters; (iv) the ridge feature map is extracted from these filtered images; (v) the query and template ridge feature maps are matched; (vi) the minutiae and ridge feature map matching results are combined to generate a single matching score (Figure 2.14).

### 2.6.1 Aligning Query and Template Images

For comparing the ridge feature maps of two images, it is necessary that the images themselves are aligned appropriately to ensure an overlap of common region in the two fingerprint images. This is done by determining the transformation parameters,  $(t_x, t_y, t_\phi)$ , that would align the query image with the template. As indicated in section 2.5, the correspondence map provided by the minutiae matcher is used to compute

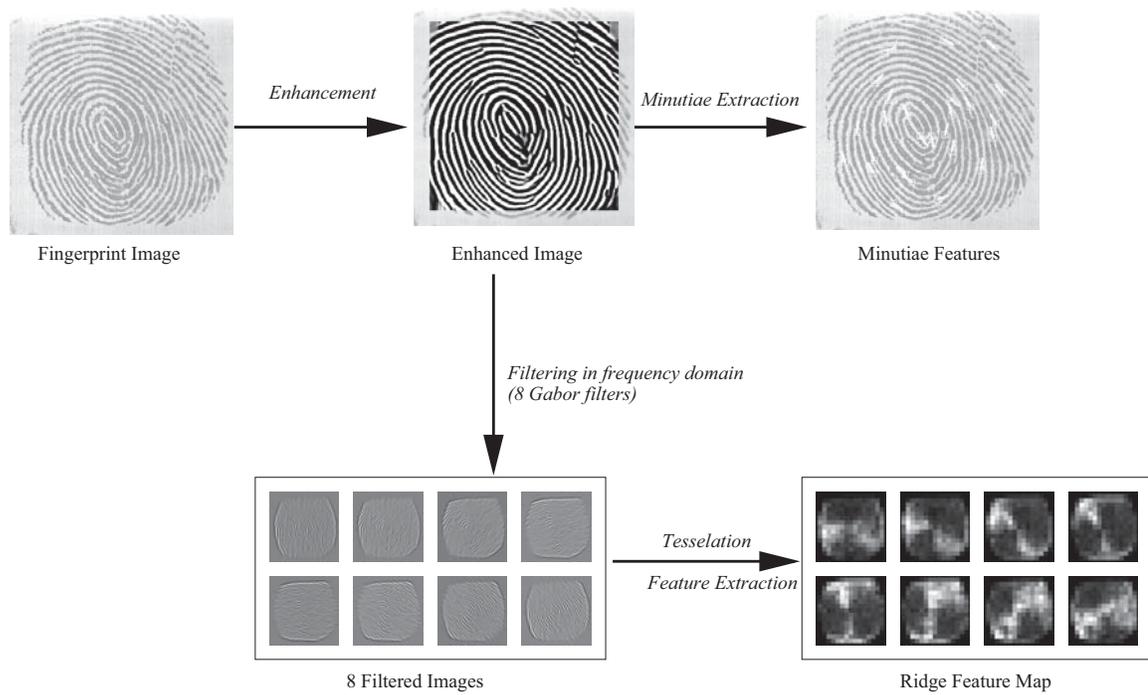


Figure 2.13: Template feature extraction. A minutiae set and a ridge feature map are extracted from the input fingerprint image.

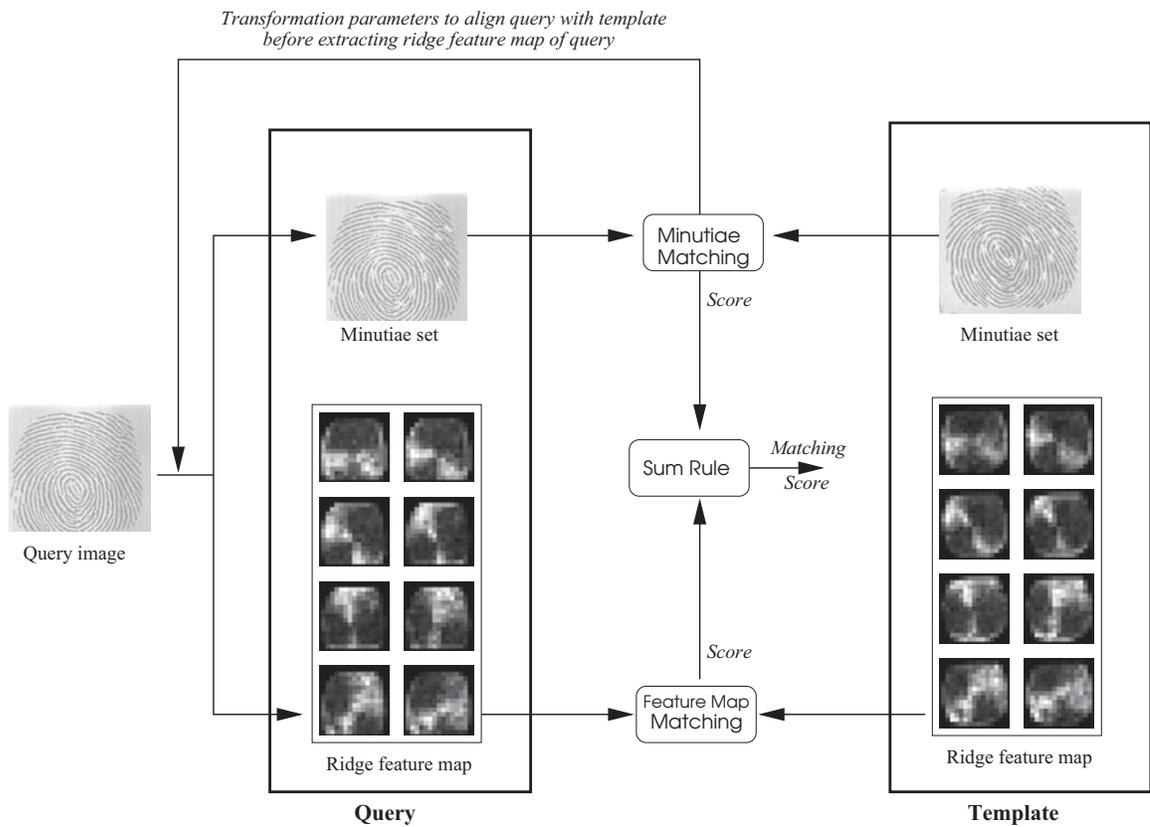


Figure 2.14: The matching process. The minutiae matching module provides the transformation parameters necessary to align the query image with the template.

$(t_x, t_y, t_\phi)$ . The estimated rotation parameter ( $t_\phi$ ) is the average of the difference in individual rotation values of all corresponding minutiae pairs. The translation parameters  $(t_x, t_y)$  are computed using the spatial coordinates of the minutiae pair that resulted in the best alignment.

Once the transformation parameters,  $(t_x, t_y, t_\phi)$ , are obtained, the query image can be aligned with the template. But rotating the query image will result in artifacts that may affect the subsequent filtering operation. In order to avoid this, appropriately rotated Gabor filters are applied to the query image. The resulting *filtered images* are then rotated and feature values extracted. Let  $H$  represent the enhanced query image, and  $(t_x, t_y, t_\phi)$  be the translation and rotation parameters obtained using the minutiae matching information. Then the filtered image,  $V_{\theta, t_\phi}$ , is obtained as,

$$V_{\theta, t_\phi} = Rot_{t_\phi} \mathcal{F}^{-1}[\mathcal{F}(H)\mathcal{F}(G_{\theta-t_\phi})], \quad (2.4)$$

where  $Rot_{t_\phi}$  indicates that the filtered image is rotated by an angle  $t_\phi$ . The notation  $V_{\theta, t_\phi}$  is used to indicate that the filtered image corresponding to filter orientation  $\theta-t_\phi$  was rotated through an angle  $t_\phi$ . The filtered image is then translated by  $(t_x, t_y)$ , in order to ensure that the tessellation of the query image would overlap with that of the template.

## 2.6.2 Matching Scores

The minutiae matching score is a measure of the similarity of the minutiae sets of the query and template images; the higher the matching score the better the match.

The similarity score is normalized in the  $[0, 100]$  range. The ridge feature maps of the query and the template images are compared by computing the sum of the Euclidean distances of the 8-dimensional feature vectors in the corresponding tessellated cells. Cells that are marked as background, are not used in the matching process. This results in a *distance* score measure; a higher distance score indicates a poor match. The distance score is normalized in the  $[0, 100]$  range, and converted to a similarity score by simply subtracting it from 100.

### 2.6.3 Combining Matching Scores

The matching scores generated by comparing the minutiae sets and the ridge feature maps, are combined in order to generate a single matching score. While a variety of strategies [36] may be used to fuse these scores, we adopt the simple sum rule. Let  $S_M$  and  $S_R$  indicate the similarity scores obtained using minutiae matching and ridge feature map matching, respectively. Then, the final matching score,  $S$ , is computed as,

$$S = \alpha S_M + (1 - \alpha) S_R, \tag{2.5}$$

where  $\alpha \in [0, 1]$ . For the experimental results reported in this paper,  $\alpha$  was set to 0.5. It is possible to vary  $\alpha$  to assign different weights to the individual matchers.

### 2.6.4 Fingerprint Identification

Fingerprint identification involves matching a query image against *multiple* templates (corresponding to different users) in order to determine the best matching score and,

therefore, the template that best resembles it. It is obvious that the processing time required to perform identification (one-to-many matching) is substantially more than that required for verification (one-to-one matching). In order to reduce the number of matching operations, most fingerprint identification systems use some indexing mechanism, to narrow the number of templates against which the query image has to be matched. A variety of fingerprint indexing mechanisms have been proposed in the literature [16][37] [38][39]. However, in the identification process described in this paper, we do not use an indexing mechanism to limit the number of matchings. The identification process requires filtering and rotating the query image for every match that is performed (Eq. (2.4)). Computing  $V_{\theta,t_\phi}$  is an expensive operation because of the Fourier operations performed. To decrease the computational complexity involved, a combination of frequency domain filtering, and filtered image-caching, is done. Caching  $V_{\theta,t_\phi}$  avoids recomputing this filtered image. Each time a query image,  $Q$ , is presented, the following sequence of operations is performed:

*Step 1:* Let the image-cache be represented by  $K$ . Set  $K = \Phi$  (the empty set).

*Step 2:* Extract the minutiae set of  $Q$ ,  $M^Q$

For all the templates  $\{T_i\}$  in the database, represented by their minutiae set  $\{M^{T_i}\}$  and ridge feature map  $\{R^{T_i}\}$ , do steps 3 to 7.

*Step 3:* Compute the transformation parameters,  $(t_x, t_y, t_\phi)$ , relating  $Q$  and  $T_i$ , using the minutiae sets  $M^Q$  and  $M^{T_i}$  as described earlier.

*Step 4:* If  $V_{\theta,t_\phi} \in K$ , do step 6.

*Step 5:* Compute  $V_{\theta,t_\phi}$  according to Eq. (2.4).  $K = K \cup V_{\theta,t_\phi}$ .

*Step 6:* Offset  $V_{\theta,t_\phi}$  using  $(t_x, t_y)$  and perform tessellation and ridge feature map ex-

traction. Let  $R^Q$  be the ridge feature map of the query image.

*Step 7:* Use  $M^Q$ ,  $M^{T_i}$ ,  $R^Q$  and  $R^{T_i}$  to generate the matching scores  $S_M$  and  $S_R$ .

Combine scores using Eq. (2.5).

Step 5 is performed only when  $V_{\theta,t_\phi}$  has *not* been computed at an earlier stage, thus improving the speed of the one-to-many matching process.

## 2.6.5 Experimental Results

The fingerprint database used in our experiments consists of fingerprint impressions obtained from 160 non-habituated, cooperative subjects using the Veridicom sensor ( $300 \times 300$  images at 500 dpi). The data was collected over two sessions. The subjects mainly consisted of students, faculty and staff at Michigan State University, and their relatives and friends. Approximately 35% of the subjects were women. In the first session, each subject was asked to provide 2 impressions of each of 4 different fingers - the left index finger, the left middle finger, the right index finger and the right middle finger. A set of 1,280 ( $160 \times 4 \times 2$ ) images were collected in this way. The subjects were requested to provide their fingerprint images again, after a period of 6 weeks. During the second session, the same procedure was adopted, and an additional 1,280 images were obtained. Thus, a total of 2,560 images were acquired over two time sessions (Figure 2.15). We refer to this database as the MSU\_VERIDICOM database. The  $300 \times 300$  images were resized to  $256 \times 256^2$  in order to speed-up the Fourier transformations.

---

<sup>2</sup>The images were first resized to  $240 \times 240$  using a bicubic interpolation; they were then padded with zeros to increase the size to  $256 \times 256$ . The padding was necessary to avoid the wrap-around distortions at the border when the image is convolved with the Gabor filters.



Figure 2.15: Eight  $300 \times 300$  fingerprint impressions acquired using the Veridicom sensor. Images (a) - (d) correspond to the right index finger of one subject, and images (e) - (h) correspond to the right middle finger of another subject. The images were resized to  $256 \times 256$  to speed-up Fourier operations.

The performance of a biometric system can be measured by reporting its False Accept Rate (FAR) and False Reject Rate (FRR) at various thresholds. These two error rates are brought together in a Receiver Operating Characteristic (ROC) curve that plots the FRR against the FAR at different thresholds. (Alternately, the genuine accept rate (GAR), which equals  $1 - \text{FRR}$ , may be plotted against the FAR). The FAR and FRR are computed by generating all possible genuine and impostor matching scores and then setting a threshold for deciding whether to accept or reject a match. A genuine matching score is obtained when two feature vectors corresponding to the *same* individual are compared, and an impostor matching score is obtained when feature vectors from two *different* individuals are compared.

The ROC curves depicting the performances of the minutiae, ridge feature map

and hybrid matchers are shown in Figure 2.16. The hybrid technique outperforms the minutiae-based scheme over a wide range of FAR values. For example, at a FAR of 0.1%, the GAR of the minutiae matcher is  $\sim 67\%$ , while that of the hybrid matcher is  $\sim 84\%$ . The equal error rate of the hybrid technique is observed to be  $\sim 4\%$ . The experiments also show that the minutiae information and ridge flow information complement each other. Consider Figure 2.17 that shows two different impressions of a finger. For this pair, matching the minutiae sets results in a high matching score, but matching the ridge feature map results in a low score (due to the limited amount of foreground overlap between the two impressions). The hybrid score, however, results in a positive match (at a certain matching threshold) between the two impressions. Now consider the fingerprint impressions (of another finger) in Figure 2.18. The minutiae matching score is rather low in this case (due to spurious minutiae being detected in both images), while the ridge feature map matching score is high (enhancing the image provides sharp dominant ridge directions). The hybrid score results in a positive match of the two impressions (at a certain matching threshold), thereby underlining the importance of the proposed technique.

The experiments reported here were conducted on a Pentium III, 800 Mhz processor, running Windows 2000. Minutiae extraction took  $\sim 1$  second, while ridge feature map computation took  $\sim 0.3$  seconds. The time taken to match two minutiae sets and generate the transformation parameters was  $\sim 0.02$  seconds. Matching two ridge feature maps took  $\sim 0.01$  seconds. The total time for fingerprint verification (one-to-one matching) was  $\sim 1.4$  seconds. However, fingerprint identification (one-to-many matching), involving 1,000 templates, took only  $\sim 0.2$  seconds, because of

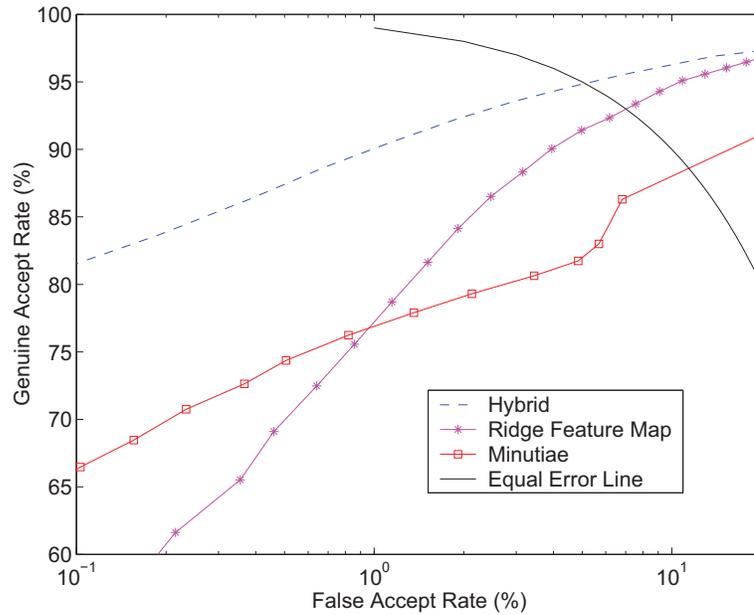


Figure 2.16: ROC showing the performances of the three matchers. The hybrid matcher is observed to perform better than the minutiae matcher.

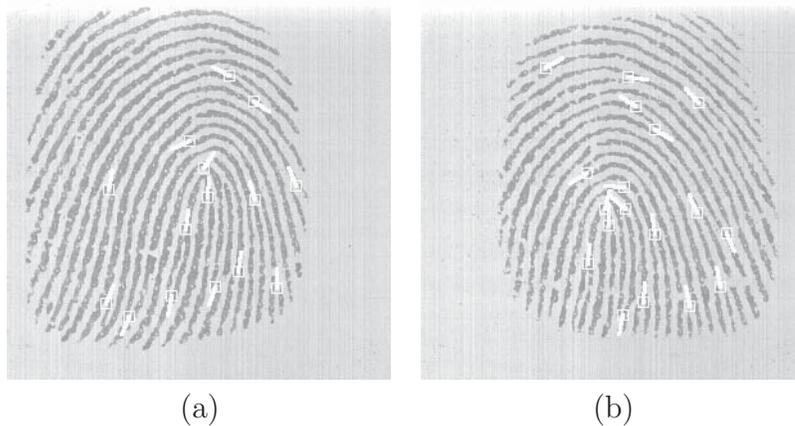


Figure 2.17: Two impressions of the same finger that have a high minutiae matching score but a low ridge feature map matching score. The hybrid score results in a true match.

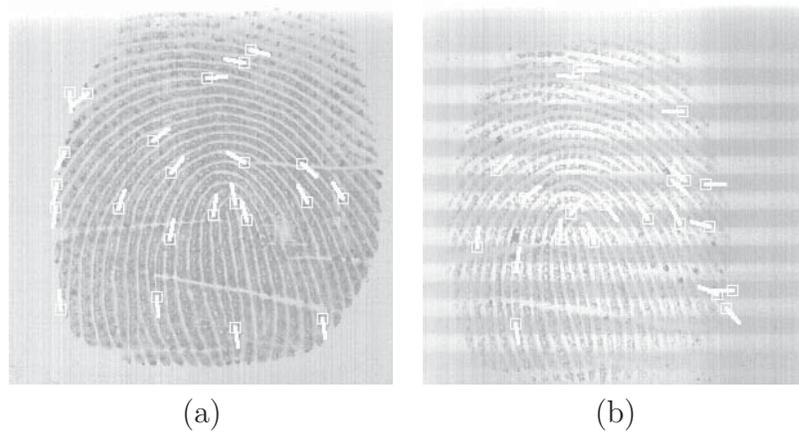


Figure 2.18: Two impressions of the same finger that have a low minutiae matching score but a high ridge feature map matching score. The hybrid score results in a true match.

the filtered image-cache.

## 2.7 Alignment Using Ridge Feature Maps

To circumvent the problem of unreliable landmark points (i.e., core and minutiae points), we propose a technique that uses the extracted feature sets themselves to align and match fingerprint images. The feature set, in this case, is the ridge feature map described in the earlier sections. A template fingerprint image is filtered using a set of Gabor filters; a standard deviation map is next computed using each filtered image; the standard deviation map is then sampled at regular intervals to generate the ridge feature map. Fingerprint verification entails *correlating* the standard deviation map of the query image with the ridge feature map of the template. A two-dimensional correlation is performed thereby taking the spatial relationship between feature values into account. A matching score is generated using the Euclidean distance metric

between corresponding elements in the ridge feature map of the template and the standard deviation map of the query image. Based on the matching score, and a pre-specified threshold, the query image is declared to match successfully (genuine) or unsuccessfully (impostor) with the template. In the following sections we describe the feature extraction and correlation process in more detail.

### 2.7.1 Constructing the Ridge Feature Map

The  $240 \times 240$  input fingerprint image,  $I$ , is convolved with the 8 Gabor filters,  $\{G_\theta\}$ . Since the input image may be noisy, it is first enhanced before applying the filters. A segmentation algorithm is also applied on the input image to identify the foreground and background regions. The foreground corresponds to those regions in the image that have ridges and furrows, while the background represents those regions that do not have this information (Figure 2.7(c)). Segmentation is useful during the matching phase, when the distance between two feature maps is computed.

Let  $H$  indicate the  $240 \times 240$  enhanced image. Let  $\mathcal{F}(H)$  denote the discrete Fourier transform of  $H$ , and let  $\mathcal{F}(G_\theta)$  indicate the discrete Fourier transform of the Gabor filter having the spatial orientation  $\theta$  as described by Equation (2.2). Then the Gabor filtered image,  $V_\theta$ , may be obtained as,

$$V_\theta = \mathcal{F}^{-1}[\mathcal{F}(H)\mathcal{F}(G_\theta)], \quad (2.6)$$

where  $\mathcal{F}^{-1}$  is the inverse Fourier transform. 8 filtered images are obtained in this way (Figure 2.9). Each  $V_\theta$  is used to construct a standard deviation image,  $S_\theta$ ,

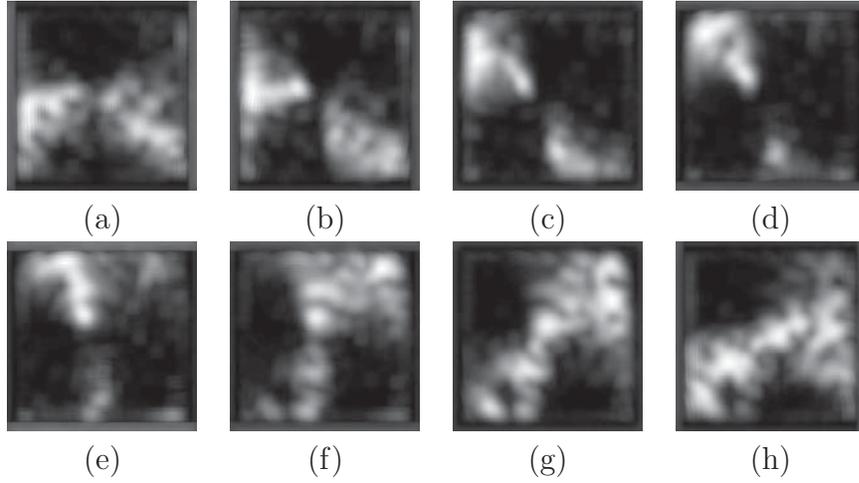


Figure 2.19: The standard deviation map,  $\{S_\theta\}$  of the filtered images shown in Figure 2.9. Each image is  $240 \times 240$ .

where  $S_\theta(x, y)$  represents the standard deviation of the pixel intensities in a  $16 \times 16$  neighborhood of  $(x, y)$  in  $V_\theta$ . The standard deviation map,  $S = \{S_\theta\}$ , comprises of 8 images corresponding to the 8 filtered images. Thus, the standard deviation map,  $S$ , captures the variation in the ridge strength at various orientations (Figure 2.19). Each standard deviation image,  $S_\theta$ , is then sampled at regular intervals (every  $16^{th}$  pixel) in both the horizontal and vertical directions to obtain the ridge feature image,  $R_\theta$  (Figure 2.20). The ridge feature map,  $R = \{R_\theta\}$ , is composed of these 8 images. The size of  $R_\theta$  ( $15 \times 15$ ) is lesser than that of  $S_\theta$  ( $240 \times 240$ ). We, therefore, have a compact fixed-length ( $15 \times 15 \times 8 = 1,800$ -valued) representation for the fingerprint.

## 2.7.2 Fingerprint Matching Using Ridge Feature Maps

The process of fingerprint matching involves comparing a query print with a set of one or more template prints. Prior to the matching process, ridge feature maps are

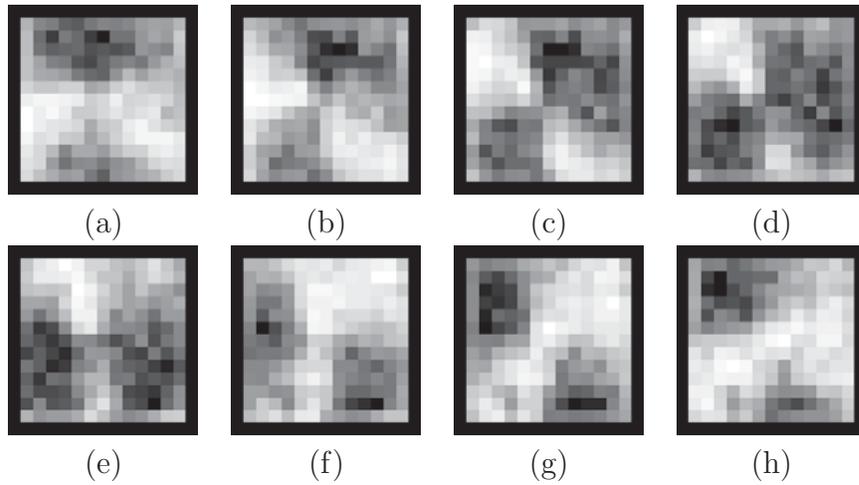


Figure 2.20: The ridge feature map,  $\{R_\theta\}$ , of the filtered images shown in Figure 2.9. Each image is  $15 \times 15$ .

extracted from all template images present in the database. When a query print,  $Q$ , is presented to the system, it is matched against a template ridge map,  $R^T = \{R_\theta^T\}$  as follows:

1. The query image is enhanced and the set of 8 Gabor filters is applied to the enhanced image, resulting in 8 filtered images.
2. The standard deviation map,  $S^Q = \{S_\theta^Q\}$ , for the query image is constructed using these filtered images.
3. Each of the 8 template ridge feature images,  $R_\theta^T$ , is ‘expanded’ to the size of  $S_\theta^Q$  by interpolating with 0’s. Let the ridge feature map consisting of the interpolated images be indicated by  $S^T = \{S_\theta^T\}$ .
4. To determine the alignment between  $S^Q$  and  $S^T$ , a 2D correlation of the two maps is performed. Correlation involves multiplying corresponding entries in

the two maps at all possible translation offsets, and determining the sum. The offset that results in the maximum sum is chosen to be the optimal alignment between the two maps. Correlation is done in the frequency domain, and every offset is appropriately weighted. The weighting is necessary to account for the amount of overlap between the two maps. Let  $U_{TQ}$  represent the unweighted correlation matrix, and  $C_{TQ}$  represent the weighted correlation matrix. Let  $N \times N$  be the size of a standard deviation image ( $N = 240$ ). Then,

$$U_{TQ} = \sum_{\theta} \{ \mathcal{F}^{-1} [ \mathcal{F}(S_{\theta}^Q) \mathcal{F}^*(S_{\theta}^T) ] \} \quad (2.7)$$

$$C_{TQ}(x, y) = \frac{U_{TQ}(x, y) * N * N}{(N - h_x)(N - w_y)}, x = 1 \dots N, y = 1 \dots N \quad (2.8)$$

where,

$$h_x = | [(x + \frac{N}{2}) \bmod N] - \frac{N}{2} | \quad \text{and} \quad w_y = | [(y + \frac{N}{2}) \bmod N] - \frac{N}{2} |$$

The optimal offset  $(t_x, t_y)$  required to align  $S^Q$  with  $S^T$  is then determined as,

$$(t'_x, t'_y) = \arg \max_{x, y} \{ C_{TQ}(x, y) \}, x = 1 \dots N, y = 1 \dots N$$

$$t_x = \begin{cases} t'_x & \text{if } t'_x < \frac{N}{2}, \\ t'_x - N & \text{if } t'_x \geq \frac{N}{2}. \end{cases} \quad (2.9)$$

$$t_y = \begin{cases} t'_y & \text{if } t'_y < \frac{N}{2}, \\ t'_y - N & \text{if } t'_y \geq \frac{N}{2}. \end{cases} \quad (2.10)$$

Equations (2.9) and (2.10) are used to decide if the offsets are negative or positive.

5. At this optimal offset, the Euclidean distance between corresponding non-zero foreground elements in  $\{S_\theta^Q\}$  and  $\{S_\theta^T\}$  is computed. This distance is treated as the matching score between the query print,  $Q$  and the template,  $T$ . Based on the matching score, and the pre-specified threshold, the query image is said to have matched successfully or unsuccessfully with the template.

The above procedure does not account for the rotational offset between the query and the template feature maps. To account for rotational offsets, various rotated versions of the template ridge feature map may be correlated with the query feature map, and the optimal alignment computed. Alternately, FFT-based registration techniques (like the Fourier-Mellin transform) may be employed.

### 2.7.3 Experimental Results

Our experiments on the MSU\_VERIDICOM database described earlier indicate that the proposed technique provides a fairly good alignment of fingerprint image pairs.

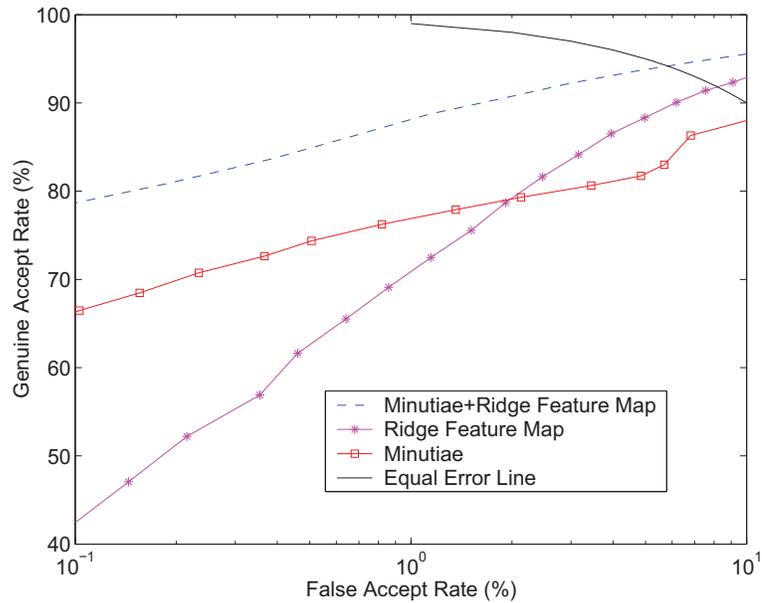


Figure 2.21: ROC curves depicting matching performance of the correlation-based technique.

We compare the proposed technique with a minutiae-based matcher by plotting the Genuine Accept Rate against the False Accept Rate at various thresholds of the matching score. As expected, the minutiae-based matcher demonstrates better performance than the correlation-based matcher. However, fusing the two matchers (by normalizing and adding the matching scores) results in an improved performance of the fingerprint verification system. The ROC curves exhibiting these behaviors is shown in Figure 2.21.

It must be mentioned that the transformation parameters computed using minutiae points is more accurate than those computed using the ridge feature maps. However, in images where minutiae points cannot be reliably derived, the ridge feature maps will prove to be a suitable alternative. This will be especially true in fingerprint images exhibiting cuts and bruises.

## 2.8 Summary

In this chapter a novel fingerprint representation technique that uses ridge feature maps has been presented. A hybrid fingerprint matching technique that combines minutiae information with the ridge feature map has also been described. Experiments indicate that the hybrid technique performs much better than a purely minutiae-based matching scheme. Further, the ridge feature maps have been used to register (align) two fingerprint images via a correlation process. The matching performance of the hybrid system is tightly coupled with the accuracy of aligning the fingerprint images using minutiae points. The following areas of improvement can also be studied:

1. New matching methods for comparing the ridge feature maps of two images.
2. Development of fusion architectures to improve performance of the hybrid matcher.
3. Constructing the ridge feature maps using adaptive methods for optimal selection of the Gabor filters.

## Chapter 3

# Fingerprint Mosaicking

A fingerprint-based verification system has two distinct phases of operation: (i) the enrollment phase, during which multiple impressions of a fingerprint are acquired and stored in the database as templates, and (ii) the authentication phase, where the query image of a user is matched against the stored templates pertaining to that user (Figure 3.1). The solid-state sensors that are being increasingly deployed in commercial applications, however, sense only a limited portion of the fingerprint pattern present in the tip of the finger. The amount of information (e.g., number of minutiae points) that can be extracted from such partial prints is substantially lower compared to that which can be extracted from more elaborate prints sensed using an optical sensor or even inked prints. As observed in Chapter 2, the average number of minutiae points extracted from a Digital Biometrics optical sensor ( $500 \times 500$  image at 500 dpi) is 45 compared to 25 minutiae obtained from a Veridicom sensor image ( $300 \times 300$  image at 500 dpi). This loss of information affects the matching performance of the verification system - the relatively small overlap between the

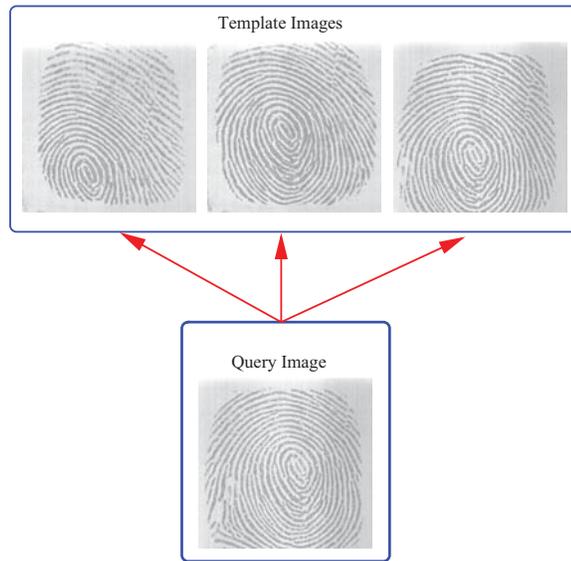


Figure 3.1: Fingerprint verification: Multiple impressions of the same finger are stored in the database as templates. The query image is matched against the components of the template to verify the claimed identity.

template and query impressions results in fewer corresponding points and therefore, results in higher false rejects and/or higher false accepts (Figure 3.2).

To deal with this problem, we have developed a fingerprint mosaicking scheme that constructs a composite fingerprint template using evidence accumulated from multiple impressions. A composite template reduces storage, decreases matching time and alleviates the quandary of selecting the “optimal” fingerprint template from a given set of impressions. In the proposed algorithm, two impressions (templates) of a finger are initially aligned using the corresponding minutiae points. This alignment is used by a modified version of the well-known iterative closest point algorithm (ICP) to compute a transformation matrix that defines the spatial relationship between the two impressions. The resulting transformation matrix is used in two ways: (a) the two template images are stitched together to generate a composite image. Minutiae

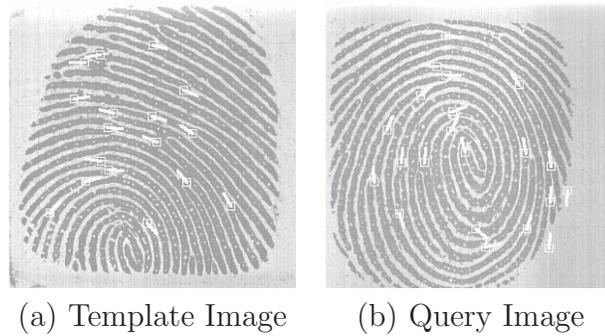


Figure 3.2: Two impressions ( $300 \times 300$ ) of the same finger acquired using the Veridicom sensor. The two impressions are observed to have very little overlap.

points are then detected in this composite image; (b) the minutia sets obtained from each of the individual impressions are integrated to create a composite minutia set. Our experiments show that a composite template improves the performance of the fingerprint matching system by  $\sim 4\%$ .

### 3.1 Introduction

Typically, two types of representations are used to assess the similarity between a pair of fingerprints: (a) the global representation that examines the structure and flow of ridges over the entire print, and (b) the local representation, that exploits the position and orientation of certain singular points, called minutiae, that are present in the print (e.g., ridge endings and ridge bifurcations). By building a composite image, the amount of information available for these two types of representation increases. For example, the number of minutiae points used to represent a fingerprint may increase when minutiae information from two impressions of a finger are integrated. Similarly, the amount of ridge information available for global representation may

increase as well. The challenge lies in accurately registering multiple impressions of the finger in order to extract more information. An accurate registration would aid in efficient mosaicking of the impressions.

Registering fingerprint images, however, is a difficult problem for the following two reasons: (a) The ridges in a fingerprint image may have non-linear distortions due to the effect of pressing a convex elastic surface (the finger) on a flat surface (the sensor). Moreover, these distortions may be present only in certain regions of the sensed image due to the non-uniform pressure applied by the subject. (b) The presence of dirt deposits on the sensor or the finger can result in noisy or occluded images. It is rather difficult to register pairs of fingerprint images that are distorted differently or affected by noise.

Ratha et al. [42] have developed a mosaicking scheme to integrate multiple snapshots of a fingerprint. The multiple snapshots are acquired as the user *rolls* the finger on the surface of the sensor and, therefore, a specific temporal order is imposed on the image frames when constructing the composite image. The authors examine 5 composition schemes that stack the grayscale images together and construct a composite mosaicked image, by associating a confidence value with every pixel. They evaluate the efficacy of these schemes by observing the area and quality (in terms of the number of valid minutiae points detected) of the composite image. Their experiments indicate that the mosaicked image has a substantially larger area and, consequently, more number of minutiae points are detected. It has to be noted that in their technique, successive impressions will have spatial proximity. But, in the case of *dab* fingerprint impressions obtained at different time instances, the parametric

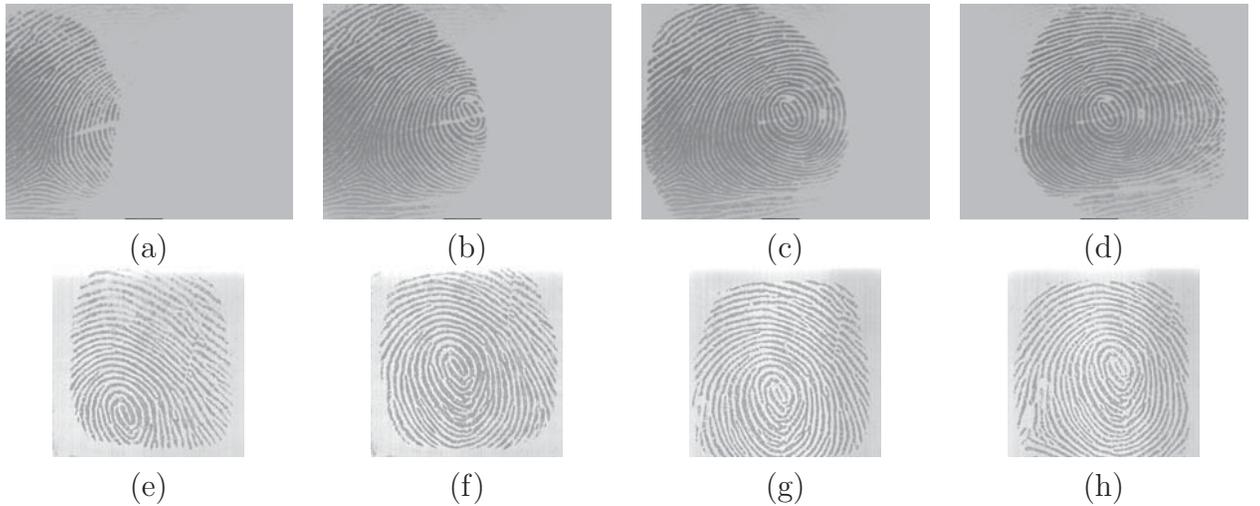


Figure 3.3: Rolled versus dab prints. (a) - (d): A sequence of 4 rolled fingerprint impressions obtained using the Digital Biometric sensor. Successive image frames are known to have spatial proximity. (e) - (h): Four impressions of a finger obtained at different time instances using a Veridicom sensor. The transformation between the impressions is not known.

transformation between impressions is not known. This complicates the problem of mosaicking fingerprint images captured at different time instances.

We approach the problem of fingerprint mosaicking by treating the acquired fingerprint images as 3D surfaces. The rationale behind this is the observation that the imaging process involves pressing the 3D surface of the fingertip on a 2D flat surface. We assume that the resulting 2D intensity image indicates the pressure with which a user holds the finger against the surface of the sensor. Therefore, the intensity images may be treated as 3D range (surface) images.

In order to generate the transformation matrix defining the spatial relationship between two impressions, we employ the *iterative closest point (ICP) algorithm* that registers two 3D surface images when sufficient number of *corresponding points* are available between the two surfaces. The correspondences are used to compute an

initial approximate alignment between the two surfaces; the ICP algorithm then attempts to find an optimal alignment such that the sum of distances between *control points* in one surface and the corresponding tangential planes in the other is minimized. Details of the algorithm is provided in the following section.

## 3.2 Fingerprint Image Registration

The problem of registering multiple 3D surfaces has received much attention in the literature (see [43, 44] and the references therein). A typical application of registration is 3D object model construction where multiple views of an object are integrated [45]. However, a variety of other applications exist for surface registration [46], including medical image analysis [47], terrain matching [48], etc.

A 3D surface registration algorithm seeks to find the best transformation  $\mathcal{T}$  that relates two entities  $P$  and  $Q$  whose range images<sup>1</sup> are given by  $R_P$  and  $R_Q$ , respectively. Thus the goal of a registration algorithm is to find  $\mathcal{T}$  such that the following objective function,  $D(R_P, R_Q)$ , is minimized:

$$D(R_P, R_Q) = \sum_{p \in R_P} \|\mathcal{T}p - f(p)\|, \quad (3.1)$$

where

$$f : P \rightarrow Q \mid \forall p \in R_P, f(p) \in R_Q.$$

The transformation,  $\mathcal{T}$ , that is used to optimally align entities  $P$  and  $Q$ , usually

---

<sup>1</sup>The terms range image and 3D surface are used interchangeably in this chapter. Typically, a 3D surface is *constructed* from a range image

depends upon the distortions present in the range images. Thus,  $\mathcal{T}$  may be rigid, affine, polynomial, or elastic. For our application we assume  $\mathcal{T}$  to be a *rigid* transformation.  $\mathcal{T}$  can, therefore, be expressed as follows in homogeneous coordinates:

$$\mathcal{T} = \begin{bmatrix} \cos \alpha \cos \beta & \cos \alpha \sin \beta \sin \gamma - \sin \alpha \cos \gamma & \cos \alpha \sin \beta \cos \gamma + \sin \alpha \sin \gamma & t_x \\ \sin \alpha \cos \beta & \sin \alpha \sin \beta \sin \gamma + \cos \alpha \cos \gamma & \sin \alpha \sin \beta \cos \gamma - \cos \alpha \sin \gamma & t_y \\ -\sin \beta & \cos \beta \sin \gamma & \cos \beta \cos \gamma & t_z \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (3.2)$$

Here  $\alpha$ ,  $\beta$  and  $\gamma$  are the rotation angles about the  $x$ ,  $y$  and  $z$  axes, respectively, and  $t_x$ ,  $t_y$  and  $t_z$  are the translation components along the three axes. Thus the matrix  $\mathcal{T}$  has 6 independent parameters.

In reality, the function  $f$  is not known and, therefore, the objective function in equation (3.1) has to be replaced by an evaluation function that assumes knowledge of a set of corresponding points in  $R_P$  and  $R_Q$ . Therefore, given a set of  $N$  pairs of corresponding points,  $(p_i, q_i)$ ,  $p_i \in R_P$ ,  $q_i \in R_Q$  and  $i = 1 \dots N$ , one can try to minimize the evaluation function  $e(R_P, R_Q)$ :

$$e(R_P, R_Q) = \sum_{i=1}^N \|\mathcal{T}p_i - q_i\|^2. \quad (3.3)$$

If the correspondences are not known, then it is not possible to register the images. Corresponding points are typically selected by extracting higher level features (e.g., edges, corners, texture, points of locally maximum curvature, etc.) from the two

surfaces, and looking for similarities between the two sets of extracted features.

If the corresponding points are known, then the evaluation function shown in Equation (3.3) can be minimized by simply searching for the global minimum in the 6-dimensional parametric space using an iterative procedure. Such a procedure, however, does not guarantee convergence to a *global* minimum. To circumvent this problem, Chen and Medioni [49] assume that an initial approximate transformation,  $\mathcal{T}_0$ , is known. A good starting approximation assures that the global minimum is reached quickly and surely.

Equation (3.3) imposes a strict correspondence between points  $p_i$  and  $q_i$ . If the pair of points selected are incompatible (i.e., they are located on different surfaces in the two images), then an iterative procedure may converge very slowly. In order to deal with this issue, the ICP algorithm is used. This algorithm tries to minimize the distances between points in one image to geometric entities (as opposed to points) in the other. Chen and Medioni [49] attempt to minimize the distance of a point on one surface, to the tangential plane of the corresponding point in the other surface. Thus, they minimize

$$e^k(R_P, R_Q) = \sum_{i=1}^N d_s^2(\mathcal{T}^k p_i, S_j^k), \quad (3.4)$$

where,  $d_s$  is the distance from the point to the plane, and  $S_j$  is the tangential plane corresponding to  $q_j$  in surface  $R_Q$ . Once an initial alignment is provided, the control points are automatically chosen by examining homogeneous regions in the two images.

An iterative procedure is adopted to converge to the global minimum (and hence the superscript  $k$  in the above equation). Since an approximate initial transformation matrix is known, convergence to the global minimum is assured, and since there is a relaxation in the condition of strict correspondence between points (equation (3.4)), convergence is faster.

### 3.3 Fingerprint Mosaicking

We pose the fingerprint mosaicking problem as a 3D surface registration problem that can be solved using a modified ICP algorithm. The initial alignment of fingerprint images  $I_P$  and  $I_Q$  is obtained by extracting minutiae points from each individual image, and then comparing the two sets of minutiae points using an elastic point matching algorithm [50]. The comparison proceeds by first selecting a reference minutiae pair (one from each image), and then determining the number of corresponding minutiae pairs using the remaining sets of points in both the images. The reference pair that results in the maximum number of corresponding pairs is chosen. Let  $(p_0, q_0)$  be the reference minutiae pair and let  $(p_1, q_1), \dots, (p_N, q_N)$  be the other corresponding minutiae pairs. Here,  $p_i = (p_{x_i}, p_{y_i}, p_{z_i}, p_{\theta_i})$  and  $q_i = (q_{x_i}, q_{y_i}, q_{z_i}, q_{\theta_i})$ , where  $(x, y)$  are the spatial coordinates of the minutiae points,  $z$  is the intensity of the image at  $(x, y)$  and  $\theta$  is the minutiae orientation. The initial transformation,  $\mathcal{T}^0$ , is computed using Horn's method of unit quaternions [51] that operates on the  $(x, y, z)$  values. In this technique, the translation parameters in Equation (3.2) are computed using the centroid of the point sets  $(p_{x_i}, p_{y_i}, p_{z_i})$  and  $(q_{x_i}, q_{y_i}, q_{z_i})$ , and the rotation components

are computed using the cross-covariance matrix between the centroid-adjusted pairs of points.

### 3.3.1 Preprocessing the Fingerprint Image

Since the ICP algorithm uses distances from points to planes, it is very sensitive to rapid and abrupt changes in surface direction. Therefore, the fingerprint images are first median filtered using a  $3 \times 3$  mask. This operation removes any undesirable “salt-and-pepper” noise that may be present in the valleys (furrows) of the fingerprint image (which may contribute to abrupt changes in the range image). The intensity values of the median filtered image are then scaled to a narrow range of values ( $[10, 20]$ ) to ensure a fairly smooth change in surface direction in the corresponding range image of the fingerprints (Figure 3.4(b)).

### 3.3.2 Fingerprint Segmentation

The purpose of segmentation is to separate the foreground and background regions in the given intensity image. The foreground corresponds to those regions in the image that have valid fingerprint information (i.e., the ridges and valleys of the fingerprint), while the background represents those regions that do not have this information. It is useful to mask out the background of the images before registering the images using the ICP algorithm. This prevents the ICP algorithm from choosing control points in the background region (which is possible due to the homogeneity in intensity in these regions) and then attempting to align the images using these points.

The algorithm to segment an image is as follows: (a) The preprocessed grayscale fingerprint image is converted to a binary image by examining the histogram of intensities and choosing a threshold. (b) An edge detection algorithm is applied to the binary image to get the outline of ridges. (c) Graham’s convex hull algorithm [35] is used to generate a polygon that segments the fingerprint image (Figure 3.4(c)).

### 3.3.3 Fingerprint as a Range Image

The intensity values are directly used as range values. Therefore, the intensity value of the image at the planar coordinate  $(x, y)$  is treated as the range value at that location. We now have two range images  $R_P$  and  $R_Q$ , that are obtained from the corresponding intensity images  $I_P$  and  $I_Q$ , respectively. Figure 3.4(d) illustrates this mapping for a portion of the image in 3.4(c).

### 3.3.4 Registering Fingerprint Images

The two range images,  $R_P$  and  $R_Q$ , are now subject to the iterations of the ICP algorithm. At each iteration  $k$ , the transformation  $\mathcal{T}^k$  that minimizes  $E^k$  in Equation (3.4) is chosen. The process is said to have converged when,

$$\frac{|E^k - E^{k-1}|}{N} < \epsilon,$$

where  $\epsilon$  is some threshold,  $\epsilon \approx 0$ .

The final transformation matrix,  $\mathcal{T}^{solution}$ , is used in the following two ways.

1. It is used to integrate the two individual images and create a composite image

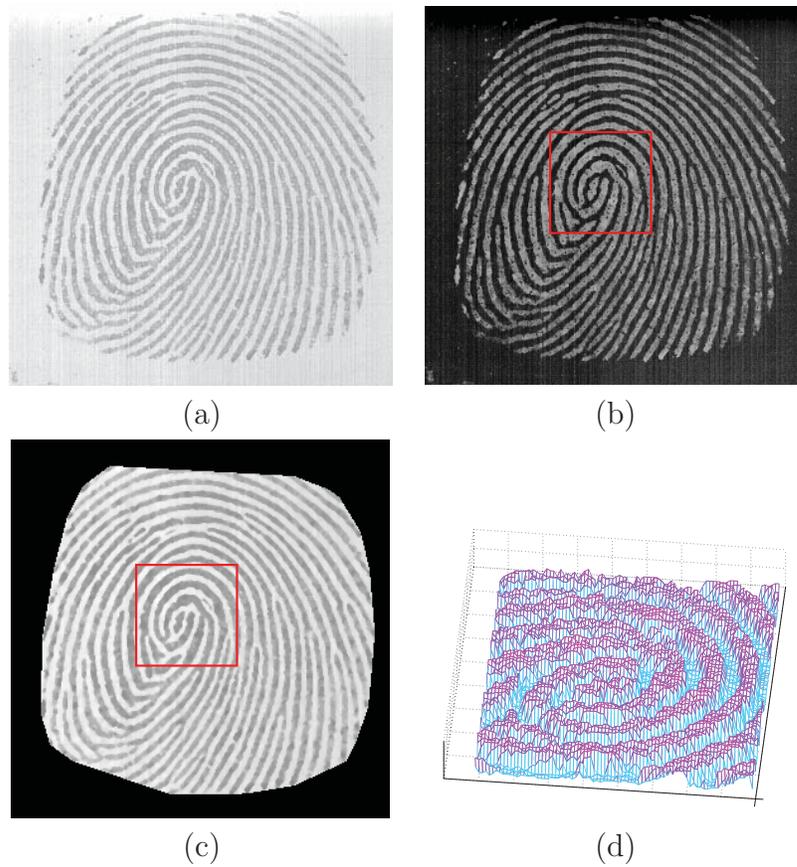


Figure 3.4: Mapping an intensity image to a range image. (a) The original intensity image. (b) The intensity image after median filtering and scaling. (c) The segmented intensity image. (d) The range image corresponding to the boxed region (rotated by  $\sim 90^\circ$  in (c)).

	Average Size	Average Number of Minutiae
Input Image	$300 \times 300$	22
Mosaicked Image	$336 \times 332$	30

Table 3.1: Increase in average image size and average number of detected minutiae as a result of mosaicking.

whose spatial extent is generally larger than the individual images. Minutiae points are then extracted from this larger image.

2. The minutiae sets from the individual images are augmented using  $\mathcal{T}^{solution}$ .

### Mosaicked Images

The given intensity images  $I_P$  and  $I_Q$  are integrated into a new image  $I_R$ . Since  $\mathcal{T}^{solution}$  transforms  $I_P$  into  $I_Q$ , we compute the new spatial coordinates of every pixel  $I_P$  in  $I_R$ . We extract a new minutiae set ( $\mathcal{M}_{R_1}$ ) from this image using the algorithm described in [50] (see figure 3.5(e)). Note that the spatial extent of the composite image is generally larger than the individual images. Figure 3.6 shows the result of mosaicking on six different fingerprint pairs. Table 3.1 lists the increase in image size and the number of detected minutiae in the composite image.

The mosaicking procedure may sometimes result in poorly aligned images. This can happen when: (i) the segmentation of either of the images is erroneous, (ii) the images are noisy, or (iii) there are very few ( $< 5$ ) corresponding points available to provide a valid initial alignment (Figure 3.7).

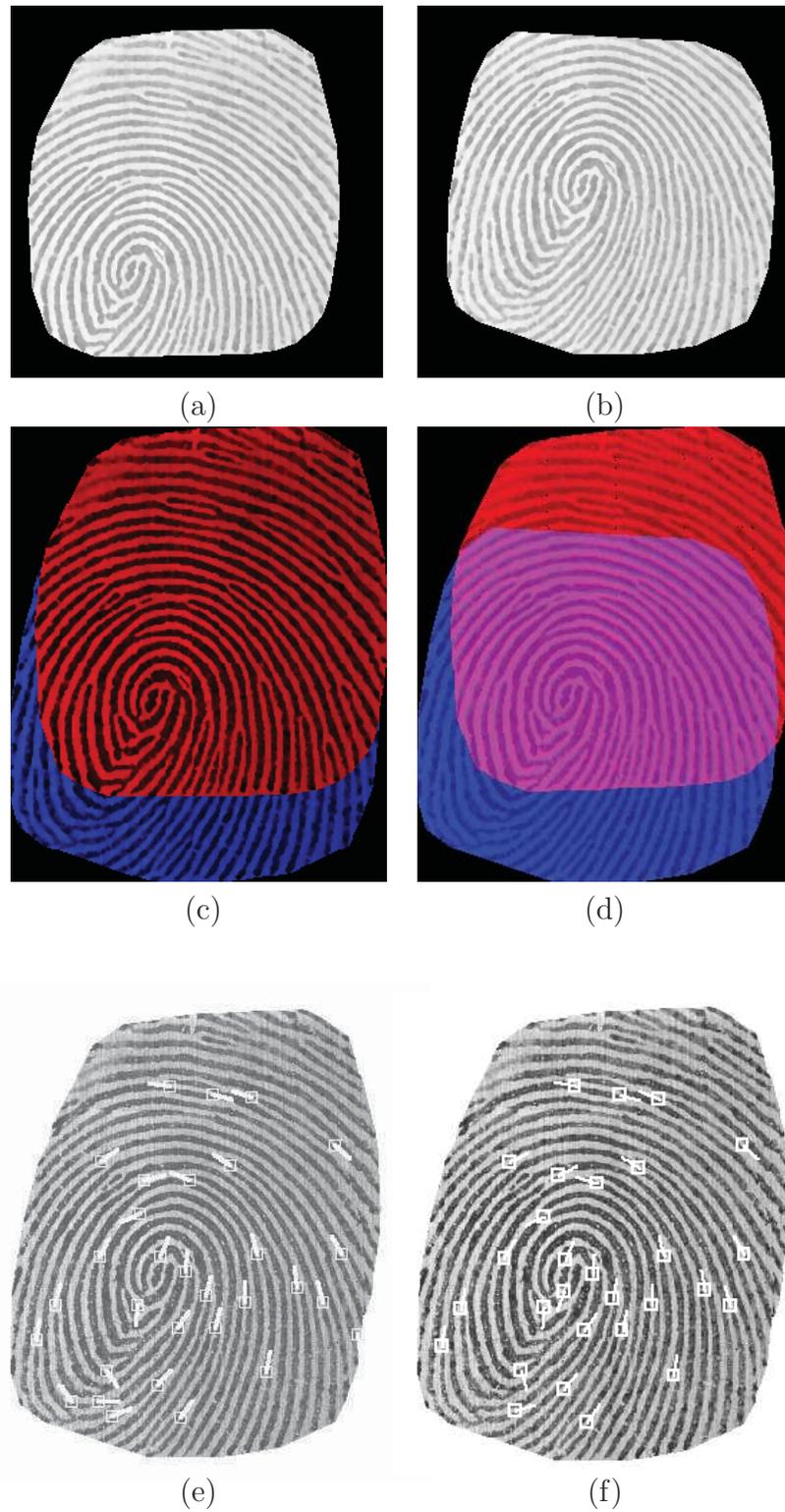


Figure 3.5: Composite template construction: (a) First image after segmentation. (b) Second image after segmentation. (c) Initial alignment. (d) Final alignment. (e) Minutiae extracted from mosaicked images. (f) Composite minutiae set obtained by augmenting individual minutiae sets.

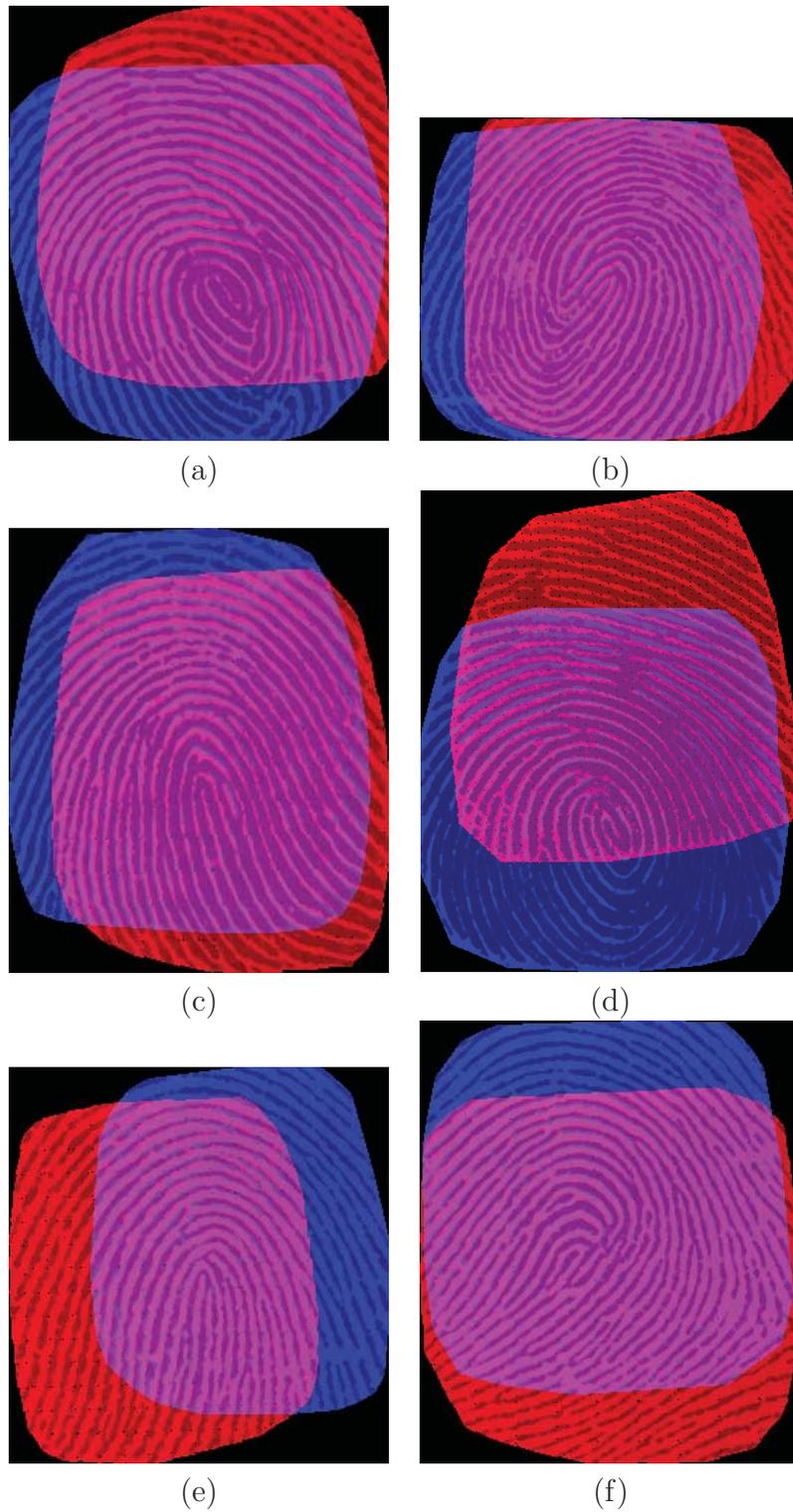


Figure 3.6: The result of mosaicking six pairs of fingerprint impressions. The spatial extent of the mosaicked image is observed to be larger than that of the component images.

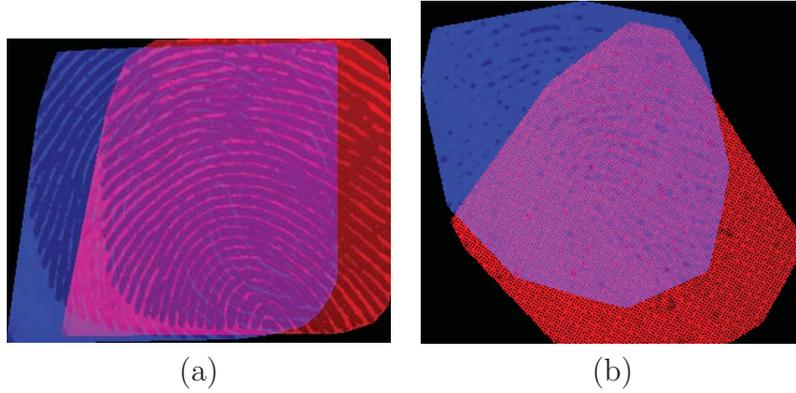


Figure 3.7: Examples of poorly mosaicked image pairs.

### Integrating Images

The intensity images  $I_P$  and  $I_Q$  are integrated into a new image  $I_R$ . Since  $\mathcal{T}_{solution}$  transforms  $I_P$  into  $I_Q$ , we compute the new spatial coordinate of every pixel  $I_P$  in  $I_R$ . We extract a new minutiae set ( $\mathcal{M}_{R_1}$ ) from this composite image using the algorithm described in [50] (see figure 3.5).

### Augmented Minutiae Sets

Let  $\mathcal{M}_P$  refer to the minutiae set extracted from  $I_P$  and  $\mathcal{M}_Q$  refer to the minutiae set extracted from  $I_Q$ . The composed minutiae set  $\mathcal{M}_{R_2}$  is obtained by computing the  $(x, y, \theta)$  parameter of each minutia in the composite image. The new  $(x, y)$  coordinates (i.e., the spatial coordinates) of the minutiae points (of the first image) is determined by simply multiplying the old coordinates with the transformation matrix (Figure 3.5f).<sup>2</sup> The minutiae orientation,  $\theta$ , is not recomputed.

---

<sup>2</sup>Since the first image is transformed to align with the second, this computation has to be done for the first image only.

## 3.4 Experimental Results

We have conducted the following experiments to validate the effectiveness of the transformation and integration described in section 3.3.4. We have two different techniques to obtain a composite minutiae set. The two minutiae sets are indicated by  $\mathcal{M}_{R_1}$  (obtained by extracting minutiae from the composite image), and  $\mathcal{M}_{R_2}$  (obtained by integrating individual minutiae sets). We treat these sets as template minutiae sets against which a query minutiae set can be matched.

Fingerprint images of 640 different fingers, corresponding to 160 different subjects, were acquired using the Veridicom sensor as described in the previous chapter. 4 different impressions of each of these fingers were obtained over two different sessions separated by a period of six weeks (2 impressions in each session). The two impressions acquired at the same session were used to construct the template minutiae set of a finger, while the other two impressions were used as query images during the test phase of the experiment. Thus, 640 pairs of images were used to construct the minutiae templates  $\mathcal{M}_{R_1}$  and  $\mathcal{M}_{R_2}$ , and the rest (1280) were used as query images.

Given a minutiae set  $\mathcal{M}_U$  (of the query image  $I_U$ ), and the template minutiae sets  $\mathcal{M}_P$ ,  $\mathcal{M}_Q$ ,  $\mathcal{M}_{R_1}$  and  $\mathcal{M}_{R_2}$ , we perform the following comparisons: (i)  $\mathcal{M}_U$  with  $\mathcal{M}_P$ , (ii)  $\mathcal{M}_U$  with  $\mathcal{M}_Q$ , (iii)  $\mathcal{M}_U$  with  $\mathcal{M}_{R_1}$ , and (iv)  $\mathcal{M}_U$  with  $\mathcal{M}_{R_2}$ . Thus we get a set of four scores corresponding to these comparisons. The ROC curves depicting the performance of these 4 different matchings are shown in figure 3.8. It is clear from these graphs that a composite template image results in improved matching performance. We further observe that a better matching performance is obtained by using  $\mathcal{M}_{R_1}$

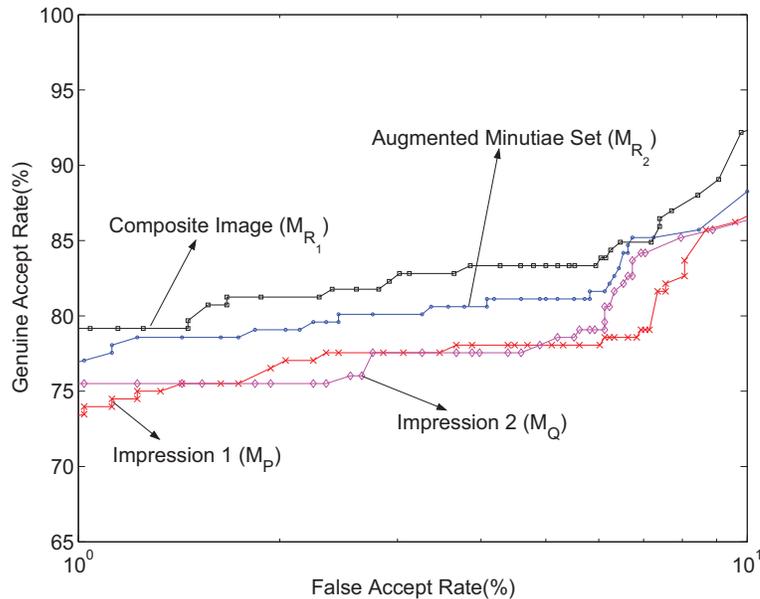


Figure 3.8: The ROC curves indicating improvement in matching performance after mosaicking templates. Utilizing the minutiae points that are extracted from the composite fingerprint image ( $\mathcal{M}_{R_1}$ ) results in the best matching performance.

rather than  $\mathcal{M}_{R_2}$ . This may be due to incorrect minutiae orientation in  $\mathcal{M}_{R_2}$ . Note that when augmenting the two minutiae sets,  $\mathcal{M}_P$  and  $\mathcal{M}_Q$ , no systematic technique is used to adjust the minutiae orientation in the composite minutiae template,  $\mathcal{M}_{R_2}$ .

While the use of  $\mathcal{M}_{R_1}$  results in better matching performance, generating  $\mathcal{M}_{R_1}$  introduces several spurious minutiae that have to be carefully discarded. The spurious minutiae are a consequence of misalignment of the ridges present in the two individual impressions that are being integrated.

### 3.5 Summary

We have described a fingerprint template construction technique that integrates information available in two different impressions of the same finger. The method makes

use of corresponding minutiae points to establish an initial approximate alignment, and a modified ICP algorithm to register the two impressions. The transformation matrix generated by the ICP algorithm is used to construct composite information from the individual impressions. Our experiments indicate that mosaicking the images together and then extracting the (template) minutiae set results in a better matching performance.

The mosaicking scheme suggested here has been used to register two impressions of a finger. By repeated application of this procedure several impressions of a finger ( $> 2$ ) may be integrated. Fingerprint mosaicking elegantly addresses the problem of partial fingerprint images and is, therefore, an essential component of a fingerprint recognition system.

## Chapter 4

# A Deformable Model for Fingerprint Matching

Fingerprint images are typically acquired using a contact-based sensor wherein a user places<sup>1</sup> her finger on the surface of the sensor. The elastic nature of the human skin, coupled with the non-uniform pressure applied by the finger on the sensor, result in fingerprint images whose ridges exhibit non-linear distortions. For reliable matching, these non-linear distortion effects must be accounted for prior to comparing two fingerprint images. Models based on affine transformations have been used to offset the effects of distortion, but they invariably lead to unsatisfactory matching results since the distortions are basically elastic in nature (Figure 4.1).

Given several template impressions of a finger, we estimate the “average” deformation for each template image corresponding to that finger based on the thin plate

---

<sup>1</sup>A live-scan fingerprint is usually acquired using the *dab* method, in which the finger is placed on the surface of the sensor without rolling. (It is possible to capture a *rolled* live-scan fingerprint, although an elaborate scanner arrangement may be necessary in this case).

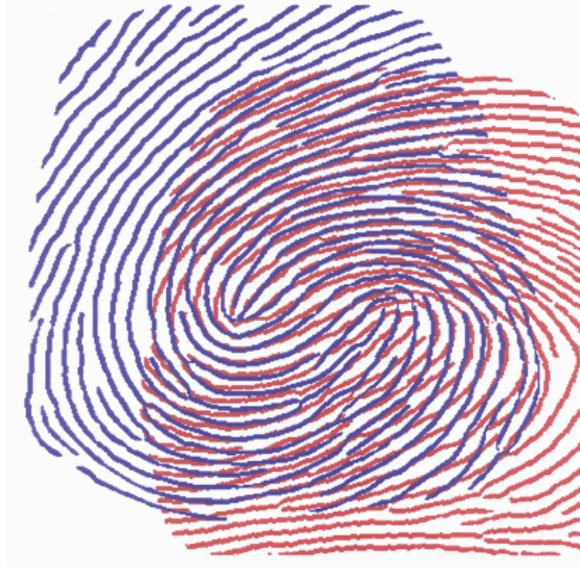


Figure 4.1: Aligning two impressions of the same finger using an affine transformation. Due to non-linear distortions, the alignment is not accurate in some regions. Only fingerprint ridges are shown for clarity [2].

spline (TPS) model. The estimated average deformation is then utilized to align the minutiae points between the template and query images during the matching stage. It is shown that the use of an average deformation model leads to a better alignment between the two sets of points as opposed to a rigid transformation. The average deformation is computed using two types of landmark points: minutiae points and ridge points. Further, an index of deformation is proposed for choosing the best deformation model arising from a set of template impressions corresponding to a finger. Experimental data consists of 1600 fingerprints corresponding to 50 different fingers collected over a period of 2 weeks. It is shown that the average deformation model leads to an improvement in the alignment between impressions originating from the same finger.

## 4.1 Introduction

As indicated earlier, the problem of automatic fingerprint matching involves determining a measure of similarity between two fingerprint impressions by comparing their ridge structure and/or the spatial distribution of the minutiae points [24, 23, 18, 53]. The image acquisition process, however, introduces non-linear distortions in the ridge structure and, consequently, in the spatial location of minutiae points, thereby confounding the matching process. This distortion is a function of several parameters including the orientation of the sensor with respect to the finger, the amount of pressure applied by the subject, the disposition of the subject (sitting or standing), the motion of the finger prior to its placement on the sensor, the moisture content of the skin (dry, oily or wet), the elasticity of the skin, etc. Therefore, the distortions observed in a fingerprint vary from one acquisition to the next. For reliable matching, these non-linear distortions must be accounted for prior to comparing two fingerprint images. Deformation models based on affine transformations invariably lead to unsatisfactory matching results since the distortions are basically elastic in nature (Figure 4.1).

To deal with the problem of non-linear distortion in fingerprint images, four types of approaches have been discussed in the literature. The first approach accounts for distortion in the image *acquisition* stage by capturing the least distorted print from the user. Ratha et al. [54] describe a system which does not accept an input image if the user applies excessive force on the sensor, thereby minimizing the effect of distortions on the acquired image. The system operates by measuring the forces and torques

applied on the sensor. Dorai et al. [55] observe a video sequence of the fingertip as it interacts with the sensor and measure the distortion in successive frames. When excessive distortion is observed, the system requests the user to provide another fingerprint. These systems require specialized hardware and the ability to perform extensive computations in real-time. As a result, they do not offer a practical solution to fingerprint deformation for real-time and embedded fingerprint applications.

In the second approach, the distortion is estimated *during* the matching stage. Thebaud [56] uses a gradient descent technique to compute local warps when comparing two fingerprints. The fingerprint correlation score is used as the objective function. Besides being time consuming, this technique potentially results in a higher False Accept Rate (FAR) since it performs local warping to force a match between the two images. Kovács-Vajna [18] uses minutiae triplets to compare two minutiae sets. By not using the entire minutiae pattern at once, the cumulative effect of distortion is avoided. Bazen and Gerez [57] use a thin-plate spline (TPS) model to account for non-linear distortions while comparing two minutiae sets.

In the third approach, the distortion is removed *before* the matching stage. Senior and Bolle [58] have developed a model which assumes that ridges in a fingerprint are constantly spaced, and that deviations from this model indicate the presence of elastic distortions. They apply local warps in regions exhibiting such deviations to make the local ridge distances nearly equal the average inter-ridge spacing. Their experimental results show a significant improvement in genuine matching scores (i.e., the matching score when comparing two impressions of the same finger), as indicated by the t-statistic. However, their assumption that inter-ridge spacing in a fingerprint

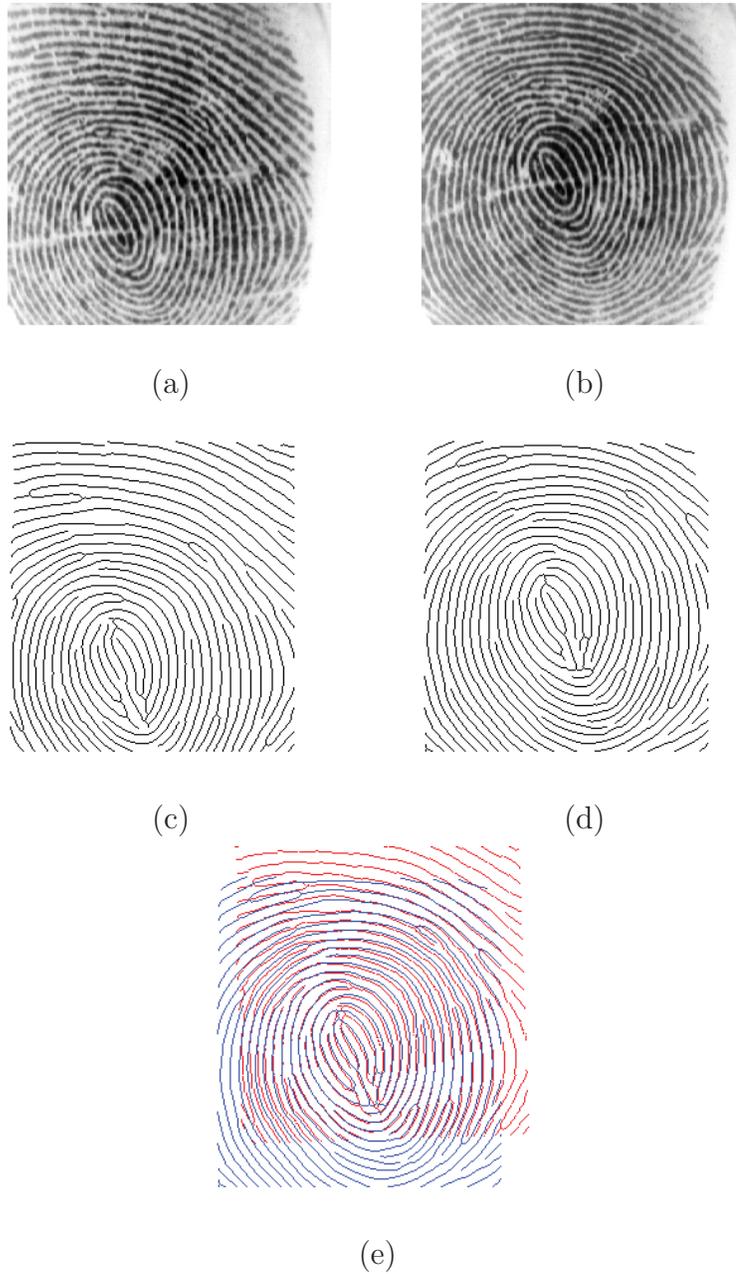


Figure 4.2: Alignment of two impressions of the same finger using affine transformation. (a) and (b) are the gray scale images; (c) and (d) are the thinned (skeletonized) images of (a) and (b), respectively; and (e) shows the alignment based on the thinned images. Ridge lines do not align in (e).

is constantly spaced is not always valid. Watson et al. [59] construct distortion tolerant filters for each (template) fingerprint before performing a correlation type of matching. Their experiments show that applying the filter before matching improves the performance.

The fourth approach is more suited for introducing distortions in synthetic fingerprints. Cappelli et al. [60] have attempted to model the distortions that could occur in a fingerprint image by considering three concentric regions in a fingerprint; the inner and outer regions are assumed to have no distortions although ridges in the outer region can be translated and rotated with respect to the ridges in the inner region; the region in between is assumed to undergo non-linear distortions in order to accommodate the transition of ridges from the inner to the outer region. The authors, however, do not use this model to perform fingerprint matching. Rather, they use it to synthesize multiple impressions of the same finger [61].

Several difficulties arise when we try to ascertain the extent of non-linear distortion in fingerprint impressions. The distortion present in impressions of a finger is not known in advance, thereby making normalization to a true template as in [58] impossible. A reference template fingerprint having no distortion effects is never available given multiple impressions of a finger; in other words, distortion effects should appropriately be measured in relative, rather than in absolute, terms between pairs of impressions. Bazen and Gerez [57] report the estimation of relative distortions between pairs of impressions based on the minutiae point pattern correspondence (see Figure 4.3). In this paper, we present an approach of estimating relative distortions based on ridge curve correspondence (see Figure 4.4). Modelling distortion effects

based on ridge curve correspondence offers several advantages over minutiae point pattern matching, consequently leading to improved authentication performance. Unlike minutiae points, which can be sparsely distributed in regions of a fingerprint image, ridge curves are spread all over the image domain resulting in a more reliable estimate of the distortion. The spatial continuity of ridge curves enables sampling of a large number of points on the ridges for establishing correspondences, including points in the vicinity of undetected minutiae points. Obtaining correspondences for undetected minutiae points is not possible when correspondences are based solely on (detected) minutiae point patterns. Also, in some poor quality images, minutiae information cannot be reliably extracted and thus, should not be used to construct a fingerprint distortion model. For the above reasons, ridge curve-based warping techniques result in a more robust and reliable estimate of the distortion in fingerprint impressions, and consequently, incorporating this distortion model in the authentication stage yields superior matching performance.

Most of the earlier techniques deal with the problem of non-linear distortion on a case by case basis, i.e., for *every* pair of fingerprint impressions (or for *every* fingerprint impression), the distortion removal technique, via a deformation model, is applied and a matching score generated. No attempt has been made to develop a *finger-specific deformation model* that can be computed offline and later used for matching. The main advantage of such a scheme is that once a finger-specific model has been computed and stored along with the template, re-computation of the model is not necessary during the matching stage. When multiple impressions of a user's fingerprint are available, a technique is proposed for computing the *average* defor-

mation model in the training stage using the thin plate spline (TPS) warping model based on ridge curve correspondence. The average deformation model corresponding to each fingerprint impression is an overall description of the relative distortions of the remaining impressions with that impression (called the baseline impression). In other words, we describe distortion effects by (i) a baseline impression and (ii) the average deformation model with respect to the baseline impression. An optimal baseline impression with the most consistent distortions (that is, distortions that deviate the least from the average) is selected by means of an index of deformation. The optimal baseline impression is not an impression with no distortion effects; it is the impression relative to which other impressions of the same finger have the most consistent (least variable) distortions. We show that by removing distortion effects using the optimal baseline impressions (and their average deformation models), superior matching performance is obtained in the authentication stage.

## 4.2 General Warping Methods

Warping methods can be used to obtain global deformation models for image registration. Applications of warping techniques abound in the statistical, medical imaging and computer vision literature. There have been a variety of image registration techniques motivated from different principles; examples include warping by elastic deformations [62, 63], optical or fluid flow [64, 65, 66], diffusion processes [67], Bayesian prior distributions [68, 69], and thin-plate splines (TPS) [70, 71, 72]. Only recently have warping techniques based on deformation models been used to model distortions

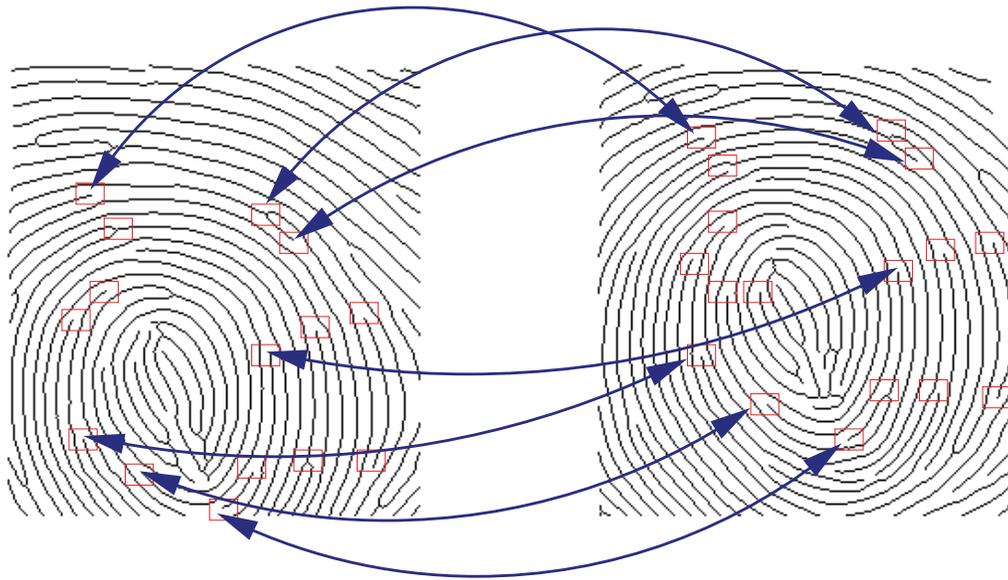


Figure 4.3: An example of minutiae correspondences between two impressions of a finger.

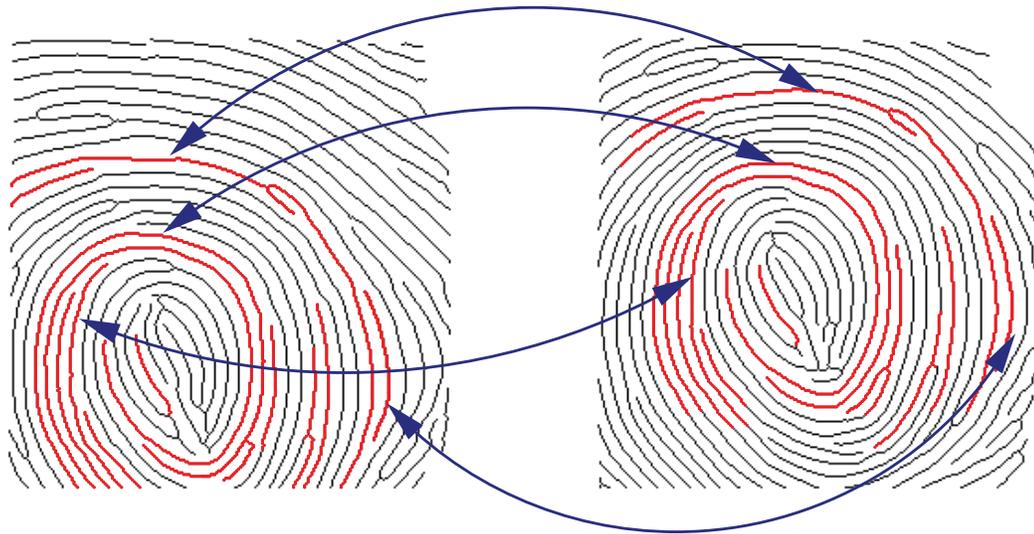


Figure 4.4: An example of ridge curve correspondences between two impressions of a finger.

in fingerprint images for the purpose of matching [73, 57]. Warping enables the distortions to be estimated and subsequently removed prior to matching. It is shown in [57] that this procedure results in superior matching performance compared to algorithms which either do not model distortions or model them using rigid transformations. Distortion models based on TPS were used in [73] and [57].

Earlier work in modelling the non-linear distortion in fingerprint images used only the spatial distribution of the minutiae points [73, 57]. One disadvantage of using the minutiae point pattern for estimating the non-linear deformation is that minutiae points may be sparse in some areas of the fingerprint. As a result, the estimated deformation model exhibits high variability in such regions leading to sub-optimal deformation models. A more fundamental and rich feature of fingerprint images is their ridge structure. A skeletonized version of a fingerprint image, known as the thinned image, can be used to extract ridge curve information (see Figure 4.2). Obtaining the deformation model based on aligning the ridge curves offers several advantages. Firstly, ridge lines are distributed over the entire fingerprint image and thus, a more reliable deformation model can be obtained. Secondly, the likelihood of incorrectly corresponding two ridge curves is much less than corresponding two minutiae points, due to the richer intrinsic information available in curves compared to points. Consequently, the deformation model based on ridge curves yields better matching performance compared to minutiae points as is shown in this paper.

When multiple impressions of a finger are available, it is observed that the non-linear distortion present in them vary significantly. Further, these distortions are different for different pairings of the impressions (Figure 4.5). Thus, we address the



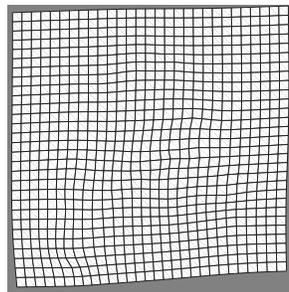
(a)



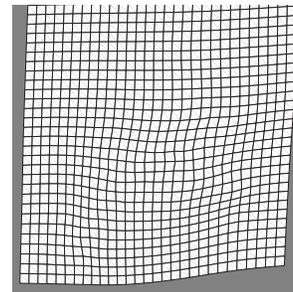
(b)



(c)



(d)



(e)

Figure 4.5: Non-linear deformations (with rotation and translation parameters removed) associated with two pairings involving the same template: (a) Template image; (b) and (c) Query images; (d) and (e) Non-linear deformation of (a) into (b) and (c), respectively.

following two problems: (i) obtain a deformation model based on ridge curve correspondence that can be incorporated in the matching stage, and (ii) given multiple deformation models for a finger (each model corresponds to one impression of the finger), select the optimal model that gives the most consistent distortion effects measured from a baseline impression. In this paper, we develop an *average* deformation model given multiple impressions of a single finger together with an estimate of its variability based on ridge curve correspondences. An index of deformation is suggested as a means of selecting a baseline impression with an associated average deformation model with the least variability. The average deformation model is incorporated in the matching stage when a query fingerprint is compared against a template fingerprint. Experimental results indicate that better matching performance is achieved by incorporating deformation models (and average deformation models) based on ridge curves as opposed to using only minutiae points.

### 4.3 The Fingerprint Warping Model

Let  $I_0(x, y)$  and  $I_1(x, y)$  denote two fingerprint impressions, where  $(x, y) \in S$  for a domain  $S \subset R^2$ . Our convention is to refer to  $I_0$  and  $I_1$  as the template and query images, respectively. A warping of  $I_0$  to  $I_1$  is defined as the function  $F : S \rightarrow S$  such that

$$I_0(F(x, y)) = I_1(x, y) \tag{4.1}$$

for  $(x, y) \in S$ . The function  $F$  is called the warping function which takes  $I_0$  to  $I_1$ . In our application we register the two impressions  $I_0$  and  $I_1$  by matching corresponding

ridge curves. Thus, in equation (4.1), the warping function,  $F : S \rightarrow S$ , registers two sets of ridge curves derived from  $I_0$  and  $I_1$ . Let  $u_k \equiv u_k(t) = (u_{k1}(t), u_{k2}(t))^T$ ,  $t \in C_k$ , denote a parameterized ridge curve in  $I_0$  for  $k = 1, 2, \dots, n$ , and let  $v_k \equiv v_k(t) = (v_{k1}(t), v_{k2}(t))^T$ ,  $t \in D_k$  and  $k = 1, 2, \dots, n$ , denote the corresponding parameterized ridge curves in  $I_1$ ; here,  $n$  is the total number of corresponding curves. The two sets of ridge curves, one set in  $I_0$  and the other in  $I_1$ , with known correspondences is denoted by the pair  $(U, V)$  where  $U = (u_1, u_2, \dots, u_n)^T$  and  $V = (v_1, v_2, \dots, v_n)^T$ . We assume that each correspondence pair is aligned as close as possible using rigid transformation prior to non-linear warping. This can be achieved using the Procrustes analysis (see [74]) after pairs of corresponding points are obtained using the methodology outlined below. For  $n$  pairs of ridge curve correspondences, a warping function,  $F$ , that warps  $U$  to  $V$ , subject to perfect alignment, is given by the conditions

$$F(u_k) = v_k \tag{4.2}$$

for  $k = 1, 2, \dots, n$ .

### 4.3.1 Establishing Ridge Curve Correspondences

Given a pair of grayscale fingerprint images,  $I_0$  and  $I_1$ , we obtain their thinned versions,  $R_0$  and  $R_1$ , using the algorithm described in [1]. A thinned image is a binary image (see Figures 4.2 (c) and (d)) with grayscale values of 0 (indicating ridges) and 255 (indicating valleys). Each thinned image can be thought of as a collection of ridge curves. In order to develop ridge curve correspondences, we proceed as follows:

1. Minutiae points are extracted from  $I_0$  and  $I_1$  using the algorithm described in [1].  
 Let  $M_0 = (m_{0,1}, m_{0,2}, \dots, m_{0,K_0})$  and  $M_1 = (m_{1,1}, m_{1,2}, \dots, m_{1,K_1})$  denote the two minutiae sets of cardinalities  $K_0$  and  $K_1$ , respectively. Here, each minutiae point  $m_{i,j}$  is characterized by its location in the image, the orientation of the associated ridge, and the grayscale intensity of pixels in its vicinity.
  
2. Minutiae correspondences between  $M_0$  and  $M_1$  is obtained using the elastic string matching technique described in [1]. The output of the matcher is a similarity score in the range  $[0,1000]$  and a set of correspondences of the form  $C = \{(m_{0,a_j}, m_{1,b_j}) : j = 1, 2, \dots, K\}$  where  $K \leq \min\{K_0, K_1\}$ , and the  $a_j$ s ( $b_j$ s) are all distinct. Figure 4.3 shows an example of the minutiae point pattern correspondence for two impressions of a finger.
  
3. Once the correspondence between  $M_0$  and  $M_1$  is established, the ridge curves associated with these minutiae points are extracted from  $R_0$  and  $R_1$  using a simple ridge tracing technique. A minutiae point that is a ridge ending has one ridge curve associated with it while a ridge bifurcation has three associated ridges. In the case of a ridge ending, the ridge curve correspondence between the two images can be easily established since each minutiae point has only one associated ridge curve. However, in the case of a ridge bifurcation, the problem of establishing ridge curve correspondences is non-trivial due to the presence of multiple ridge curves for each minutiae point; each of the three component ridge curves of one minutiae point can potentially match with any component of the other impression.

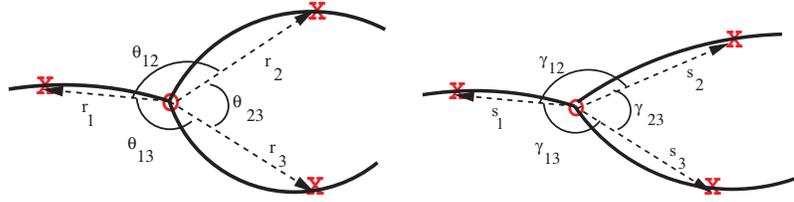


Figure 4.6: Vector representation of ridge bifurcation used to establish correspondences between component ridge curves.  $O$  marks the bifurcation points in correspondence, and  $X$  marks the points on the ridges at Euclidean distance  $d$  from  $O$ .

To resolve this ambiguity, each ridge curve corresponding to the minutiae point in  $I_0$  ( $I_1$ ) is represented as a directional vector  $r_j(s_j)$ ,  $j = 1, 2, 3$ , based on two points on the ridge curve: the minutiae point and the  $d$ -th point ( $d = 20$ ) on the ridge from the minutiae (see Figure 4.6). We define  $\theta_{j,k}$  ( $\gamma_{j,k}$ ) to be the angle that  $r_j(s_j)$  makes with  $r_k(s_k)$ , for  $k \neq j$ . We find the vector  $r_j(s_j)$  for which the angles  $\{\theta_{j,k}, k \neq j\}$  ( $\{\gamma_{j,k}, k \neq j\}$ ) are both obtuse. This establishes the first ridge curve correspondence, say,  $r_1 \sim s_1$ , without loss of generality. We then compute the cross products  $c_r = r_2 \times r_3$  and  $c_s = s_2 \times s_3$ . We assign the correspondence  $r_2 \sim s_2$  and  $r_3 \sim s_3$  if  $c_r$  and  $c_s$  are of the same sign, and  $r_2 \sim s_3$  and  $r_3 \sim s_2$ , otherwise. Figure 4.4 shows an example of ridge curve correspondence for a pair of impressions of a finger.

### 4.3.2 Sampling Ridge Curves

Having determined the corresponding ridge curves, we next establish a correspondence between *points* on these curves by sampling every  $q$ -th point ( $q = 20$ ) on each of the ridge curves. For the correspondence pair  $(U, V)$ , we have  $u_k \equiv u_k(t)$  and  $v_k \equiv v_k(t)$  for  $k = 1, 2, \dots, n$ . The sampling of the  $k$ -th corresponding ridge curves, say at

points  $t_1, t_2, \dots, t_{g_k}$ , yields  $g_k$  pairings of the form  $(u_k(t_j), v_k(t_j))$  for  $j = 1, 2, \dots, g_k$ . Thus, we have a total of  $N = \sum_{k=1}^n g_k$  points in establishing the correspondence. We denote this set of corresponding points by  $\mathcal{U} = (\Pi_\infty^*, \Pi_\epsilon^*, \dots, \Pi_{\mathcal{N}}^*)^T$  and  $\mathcal{V} = (\Xi_\infty^*, \Xi_\epsilon^*, \dots, \Xi_{\mathcal{N}}^*)^T$ . We use TPS to estimate the non-linear deformation  $F$  based on these points. TPS represents a natural parametric generalization from rigid to mild non-rigid deformations. The deformation model for TPS is given in terms of the warping function  $F(u)$ , with

$$F(u) = c + A \cdot u + W^T s(u), \quad (4.3)$$

where  $u \in S$ ,  $c$  is a  $2 \times 1$  translation vector,  $A$  is a  $2 \times 2$  affine matrix,  $W^T$  is a  $N \times 2$  coefficient matrix,  $s(u) = (\sigma(u - u_1^*), \sigma(u - u_2^*), \dots, \sigma(u - u_N^*))^T$  where

$$\sigma(u) = \begin{cases} \|u\|^2 \log(\|u\|), & \|u\| > 0 \\ 0, & \|u\| = 0. \end{cases} \quad (4.4)$$

In equation (4.3), there are 6 and  $2N$  parameters corresponding to the rigid and non-rigid parts of the deformation model, respectively, resulting in a total of  $2N + 6$  parameters to be estimated. The restrictions

$$F(u_j^*) = v_j^*, \quad (4.5)$$

$j = 1, 2, \dots, N$  provide  $2N$  constraints. For the parameters to be uniquely estimated, we further assume that  $W$  satisfies the two conditions (i)  $1_N^T W = 0$  and (ii)  $U_s^T W = 0$ ,

where  $1_N$  is the vector of ones of length  $N$ . Thus, the parameters of the TPS model can be obtained from the matrix equation

$$\begin{bmatrix} H & 1_N & \mathcal{U} \\ 1_N^T & 0 & 0 \\ \mathcal{U}^T & 0 & 0 \end{bmatrix} \begin{bmatrix} W \\ c^T \\ A^T \end{bmatrix} = \begin{bmatrix} \mathcal{V} \\ 0 \\ 0 \end{bmatrix}, \quad (4.6)$$

where  $H$  is the  $N \times N$  matrix with entries  $h_{ij} = \sigma(u_i^* - u_j^*)$ .

The matrix equation in (4.6) gives rise to a TPS model that minimizes the bending energy subject to the perfect alignment constraints in (4.5). A more robust TPS model can be obtained by relaxing the constraints in equation (4.5), and instead determining the function  $F$  which minimizes the expression

$$\sum_{j=1}^N (v_j^* - F(u_j^*))^T (v_j^* - F(u_j^*)) + \lambda J(F), \quad (4.7)$$

where

$$\begin{aligned} J(F) = & \sum_{j=1}^2 \int_S \left\{ \left( \frac{\partial^2 F_j(x, y)}{\partial x^2} \right)^2 + 2 \left( \frac{\partial^2 F_j(x, y)}{\partial x \partial y} \right)^2 \right. \\ & \left. + \left( \frac{\partial^2 F_j(x, y)}{\partial y^2} \right)^2 \right\} dx dy \end{aligned} \quad (4.8)$$

represents the bending energy associated with  $F = (F_1, F_2)^T$ ,  $F_j$  is the  $j^{\text{th}}$  component of  $F$ , and  $\lambda > 0$ . The case  $\lambda = 0$  gives rise to the TPS model described by equation (4.6). For general  $\lambda > 0$ , the parameters of the resulting TPS model can be obtained using equation (4.6) with  $H$  replaced by  $H + \lambda I_N$ , where  $I_N$  is the  $N \times N$  Identity

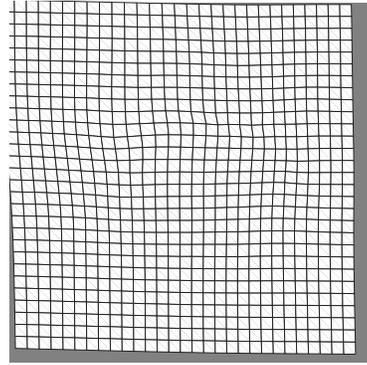
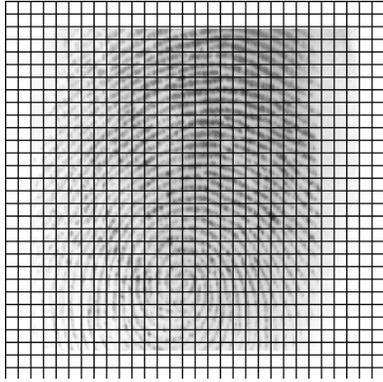
matrix.

## 4.4 Average Deformation Model

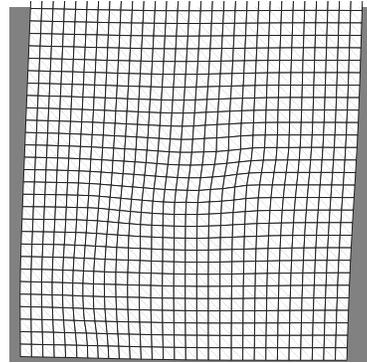
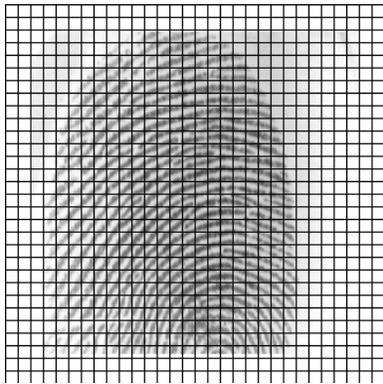
Suppose we have  $L$  impressions of a finger,  $T_1, T_2, \dots, T_L$ . Each impression,  $T_i$ , can be paired with the remaining impressions,  $T_j, j \neq i$ , to create  $L - 1$  pairs of the form  $(T_i, T_j)$ . For the pair  $(T_i, T_j)$ , we obtain a non-linear transformation  $F_{ij}$  by employing the technique described in section 4.3. Note that  $F_{ij}$  transforms *every* pixel in the template fingerprint,  $T_i$ , to a new location. Thus, we can compute the *average* deformation of each pixel  $u$  in  $T_i$  as

$$\bar{F}_i(u) = \frac{1}{L-1} \sum_{j \neq i} F_{ij}(u). \quad (4.9)$$

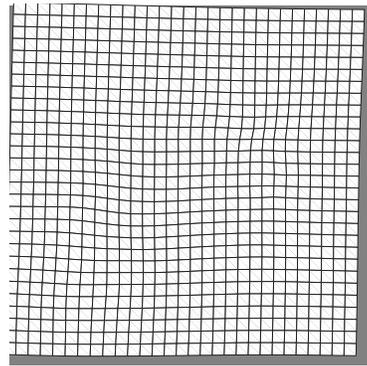
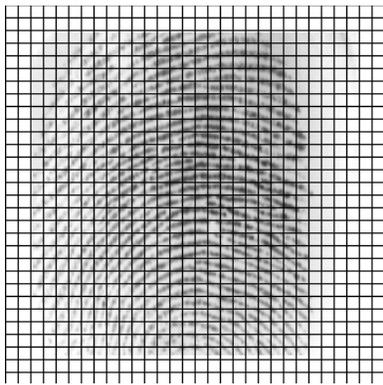
There will be  $L$  average deformation models corresponding to the  $L$  impressions of the finger. The average deformation is the typical deformation that arises when we compare one fingerprint impression of a finger (the baseline impression) with other impressions of the same finger. Figure 4.7 shows that changing the baseline impression for the finger will result in a different average deformation model for that finger (the  $\Phi$  values are as discussed in section 4.4.1). Figure 4.8 shows the average deformation for 3 different fingers; it can be clearly seen that the average warping functions are different for the 3 fingers indicating that the fingerprint deformation is finger-specific.



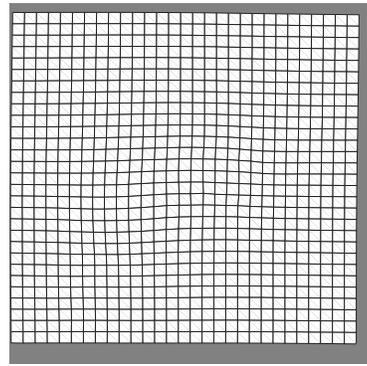
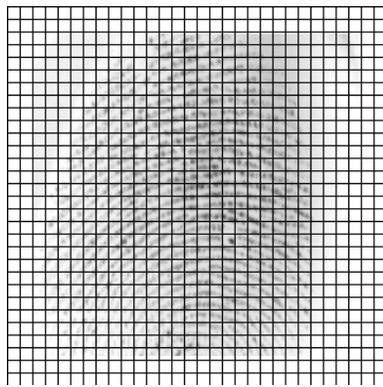
(a)  $\Phi = 15.54$



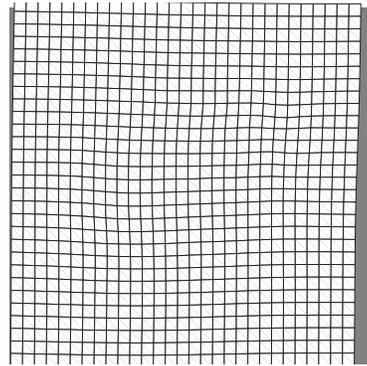
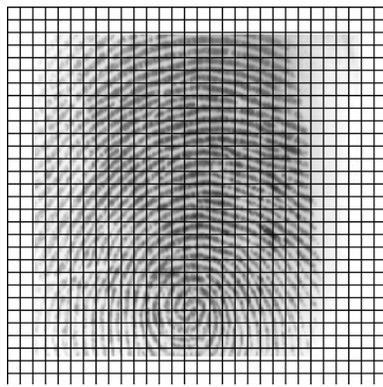
(b)  $\Phi = 17.97$



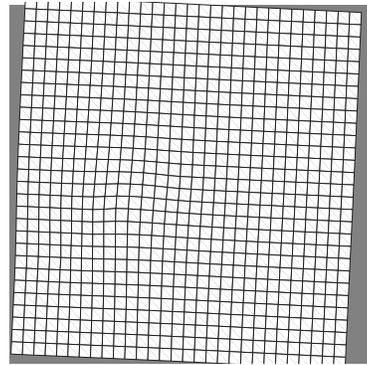
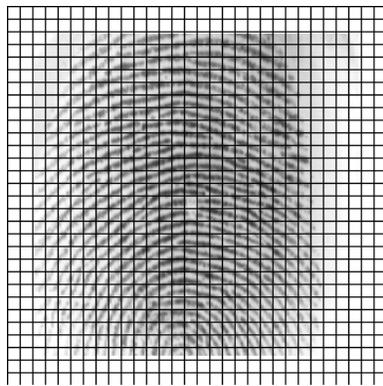
(c)  $\Phi = 48.79$



(d)  $\Phi = 83.12$

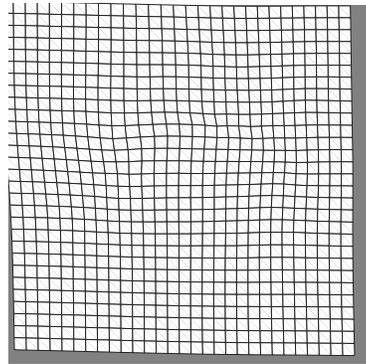
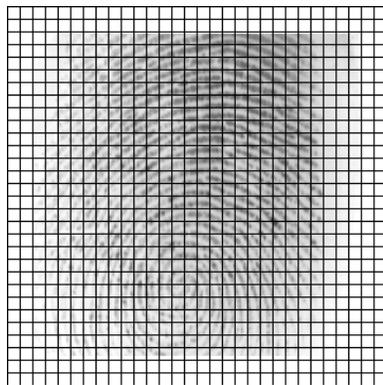


(e)  $\Phi = 94.34$

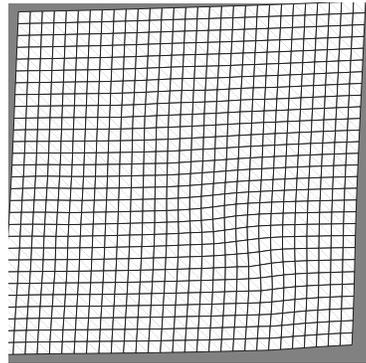
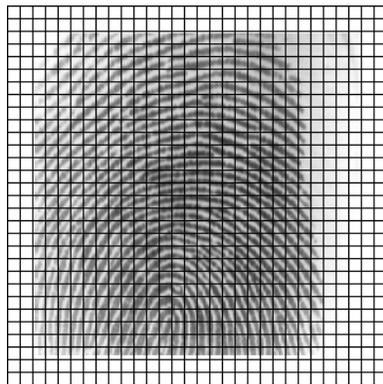


(f)  $\Phi = 232.53$

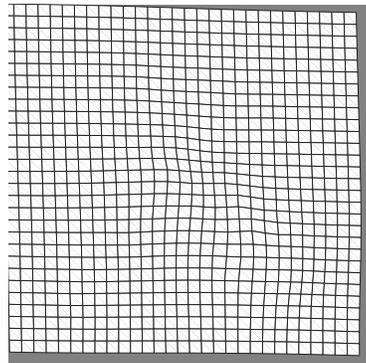
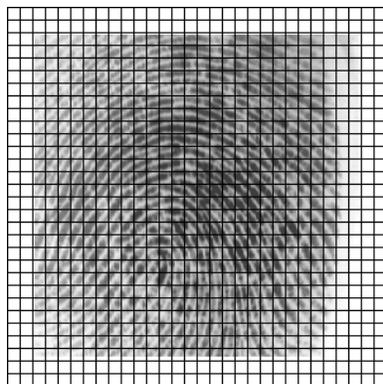
Figure 4.7: The average deformation model (shown as deformations on a reference grid) corresponding to 6 templates of a finger sorted in increasing  $\Phi$ -values. (a) is chosen to be the optimal template since it has the least  $\Phi$ -value.



(a)  $\Phi = 46.68$



(b)  $\Phi = 37.59$



(c)  $\Phi = 85.18$

Figure 4.8: The average deformation model (shown as deformations on a reference grid) of 3 different fingers .

### 4.4.1 The $\Phi$ Index of Deformation

We consider the following two questions in this section:

1. Which of the  $L$  average deformation models can be considered to be the optimal model for this finger?
2. Will the optimal model, when incorporated in the matching stage, result in improved performance compared to the suboptimal models?

In order to address these questions, we first define the pixel-wise covariance matrix associated with the  $i$ -th average deformation,  $\bar{F}_i$ , as follows:

$$D_{\bar{F}_i}(u) = \frac{1}{L-1} \sum_{j \neq i} (F_{ij}(u) - \bar{F}_i(u)) \cdot (F_{ij}(u) - \bar{F}_i(u))^T, \quad (4.10)$$

where  $F_{ij}$  is the deformation function that warps  $T_i$  to  $T_j$ . The covariance matrix defined at each pixel  $u$ , is a measure of the variability associated with the estimated deformation functions. Two choices of pixel-wise measures of variability are given by (i) the determinant,  $\phi(D_{\bar{F}_i}(u)) = |D_{\bar{F}_i}(u)|$ , and (ii) the trace,  $\phi(D_{\bar{F}_i}(u)) = \text{tr}(D_{\bar{F}_i}(u))$ . Pixels with large (small) values of  $\phi$  indicate high (low) variability in the deformations  $F_{ij}$ . We propose to use the values of  $\phi$  to determine the optimal model for a given finger. We define the  $i^{\text{th}}$  index of deformation,  $\Phi_i$ , as

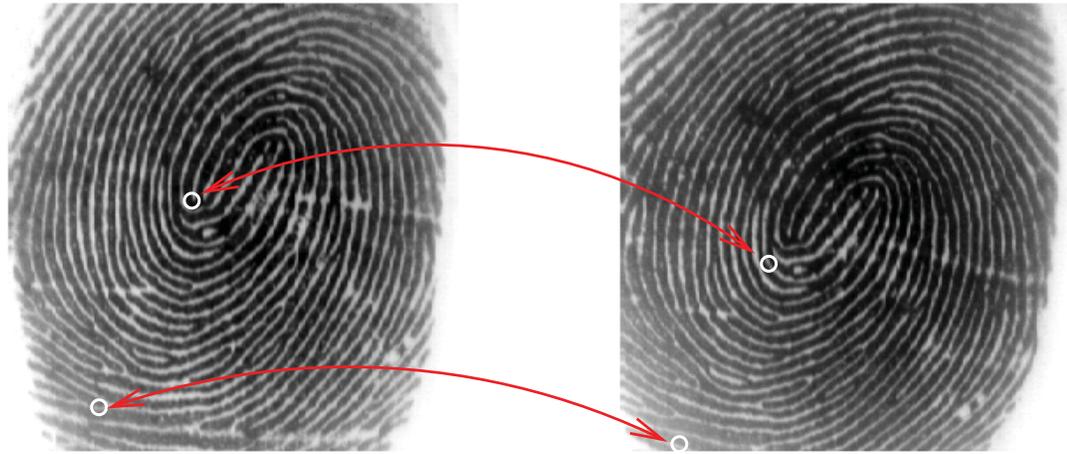
$$\Phi_i = \frac{1}{|S|} \sum_{u=1}^{|S|} \phi(D_{\bar{F}_i}(u)), \quad (4.11)$$

where,  $\phi(D) = \text{tr}(D)$ , and  $|S|$  is the number of pixels in the domain  $S$ . Subsequently, we choose  $T_{i^*}$  as the template with the smallest variability in deformation if  $i^* = \arg \min_i \Phi_i$ . In effect, we choose that template  $T_i$  that minimizes the average variation across pixels measured in terms of  $\Phi_i$ . Low (high) values of the index of deformation indicate that the warping functions are similar (dissimilar) to each other.

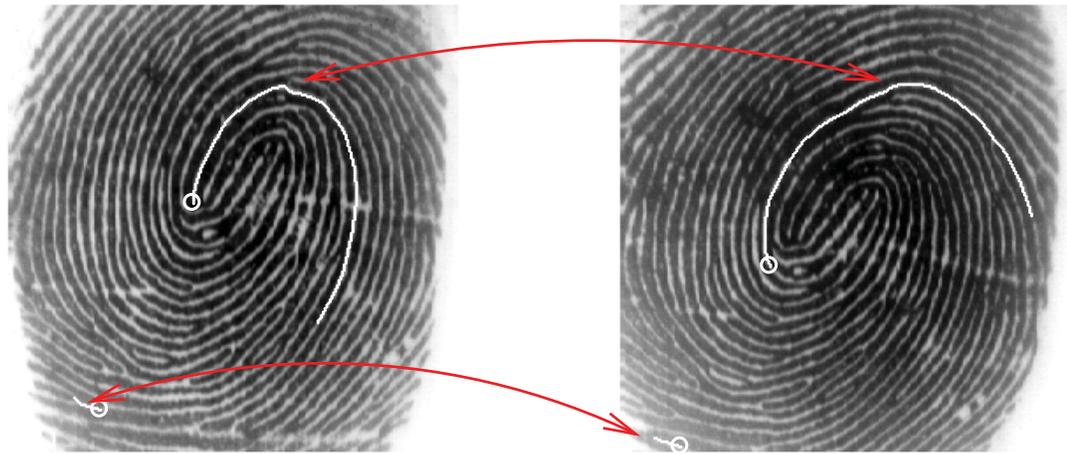
#### 4.4.2 Eliminating Erroneous Correspondences

For each baseline fingerprint impression, it is important to determine the set of minutiae points that are correctly paired to form a correspondence. The reason for this is that the average deformation model is sensitive to the accuracy of the ridge curve correspondence, which in turn depend on the minutiae correspondence. It is, therefore, necessary to check the correctness of the minutiae correspondences prior to obtaining the ridge curve correspondences. Figure 4.9(a) gives an example of two incorrect minutiae correspondences which result in incorrect ridge curve correspondences (Figure 4.9(b)). These erroneous correspondences have to be eliminated prior to computing the average deformation model; failure to exclude such minutiae points results in a warping model that exhibits spurious distortions. This is done using the technique described below.

For the given baseline fingerprint impression, minutiae points that have a correspondence with at least  $\ell$  ( $\ell = 5$ ) of the remaining  $L - 1$  impressions are extracted. We denote the set of all extracted minutiae points by  $M = \{m_i, i = 1, 2, \dots, K\}$ , where  $K$  is the total number of such minutiae points. Each  $m_i$  has a correspond-

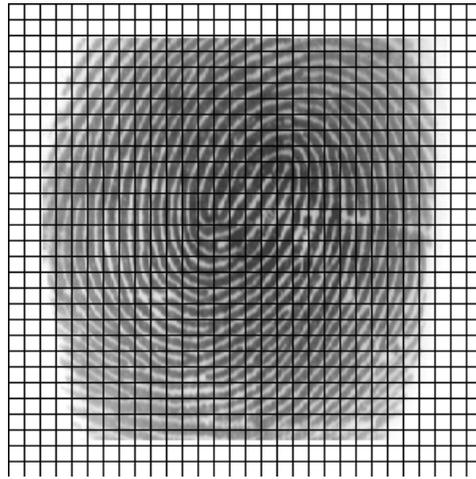


(a)

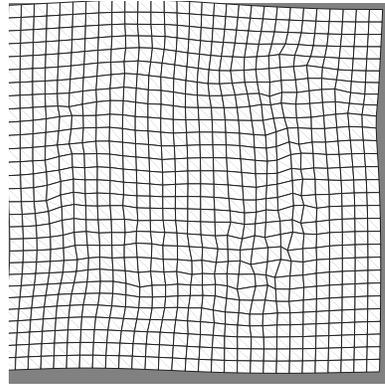


(b)

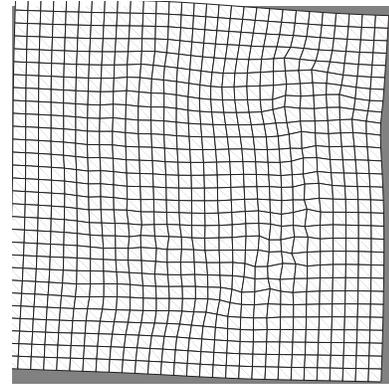
Figure 4.9: Examples of incorrect minutiae correspondences (a) resulting in erroneous ridge curve correspondences (b).



(a)



(b)  $\Phi = 102.16$



(c)  $\Phi = 67.78$

Figure 4.10: Effect of eliminating unreliable minutiae correspondences on the average deformation model; (a) template fingerprint, (b) average deformation model with  $p = 100$ , and (c) average deformation model with  $p = 60$ .

ing minutiae point in at least  $\ell$  of the  $L - 1$  impressions. We denote these pairings by  $(m_i, p_1), (m_i, p_2), \dots, (m_i, p_{\ell_i})$ , where  $\ell_i$  is the total number of pairings. We now develop a measure of reliability for minutiae point  $m_i$  as follows:

1. Sampled ridge point correspondences are obtained for each  $(m_i, p_j)$ ,  $j = 1, 2, \dots, \ell_i$  based on which a TPS deformation model,  $F_{(m_i, p_j)}$  is computed. The average deformation model for the minutiae point  $m_i$  is given by

$$\bar{F}_{m_i}(u) = \frac{1}{\ell_i} \sum_{j=1}^{\ell_i} F_{(m_i, p_j)}(u).$$

Here, the average deformation model is obtained in a  $10 \times 10$  square region, say  $S_{m_i}$ , centered at  $m_i$ .

2. Let

$$D_{\bar{F}_{m_i}}(u) = \frac{1}{\ell_i} \sum_{j=1}^{\ell_i} (F_{(m_i, p_j)}(u) - \bar{F}_{m_i}(u)) \cdot (F_{(m_i, p_j)}(u) - \bar{F}_{m_i}(u))^T \quad (4.12)$$

denote the site-wise variability measure of the deformations  $F_{(m_i, p_j)}$  around  $\bar{F}_{m_i}$ .

The average variability is measured by

$$R_{m_i} = \frac{1}{|S_{m_i}|} \sum_{u=1}^{|S_{m_i}|} \text{trace}(D_{\bar{F}_{m_i}}(u))$$

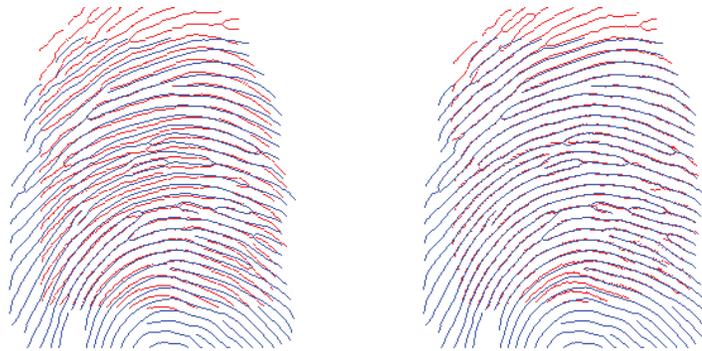
with small values of  $R_{m_i}$  indicating better reliability. Correspondences pertaining to those minutiae points with  $R_{m_i}$  values lower than the  $p$ -th percentile (e.g.,  $p = 60$ ) are used to develop the average deformation model for the template

fingerprint.

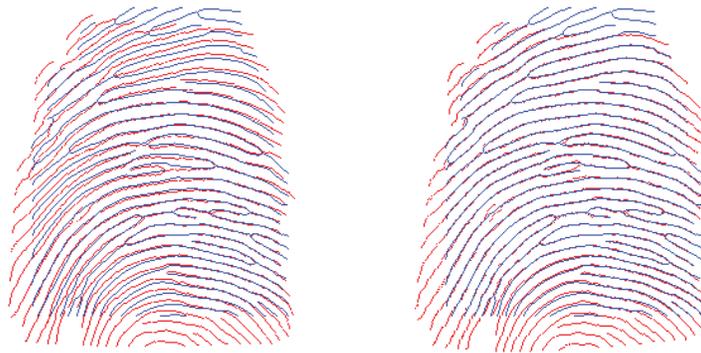
For the incorrect minutiae correspondences in Figure 4.9, the value of  $R$  for the top minutiae point was 93.2 (the 60-th percentile value of  $R$  was 55.5 for this template) while the lower minutiae point occurred in less than 5 corresponding pairs and hence was eliminated. Figure 4.10(a) shows the average deformation model that results for this template when all correspondences are used (i.e.,  $p = 100$ ); Figure 4.10(b) gives the deformation model for  $p = 60$ .

## 4.5 Experimental Results

In order to apply the TPS model to reliably estimate fingerprint deformation, we need to have several impressions of the same finger. Large number of impressions of a finger are not available in standard fingerprint databases (e.g., FVC 2002 [75]). Therefore, fingerprint images of 50 fingers were acquired using the Identix sensor ( $256 \times 255$ , 380 dpi) over a period of two weeks in our lab. There were 32 impressions corresponding to every finger, resulting in a total of 1600 impressions. One half of the impressions ( $L = 16$  for each finger, resulting in 800 impressions) were used as templates to compute the average deformation model for each finger, while the remaining 800 impressions were used as query images for testing. For each template image,  $T$ , the minutiae set,  $M_T$ , and the thinned image,  $R_T$ , were extracted. The average deformation model of  $T$ ,  $\bar{F}_T$ , was obtained based on pairings with the remaining 15 impressions of the same finger (equation (4.7) with  $\lambda = 0.1$ ). The minutiae set  $M_T$  was transformed to the deformed set,  $MD_T \equiv \bar{F}_T(M_T)$  using  $\bar{F}_T$ . A total of



(a)



(b)



(c)

Figure 4.11: Improved alignment of template and query images using ridge curve correspondences (right panel). The alignment using minutiae correspondences are shown in the left panel. Both sets of alignment use the TPS warping model.

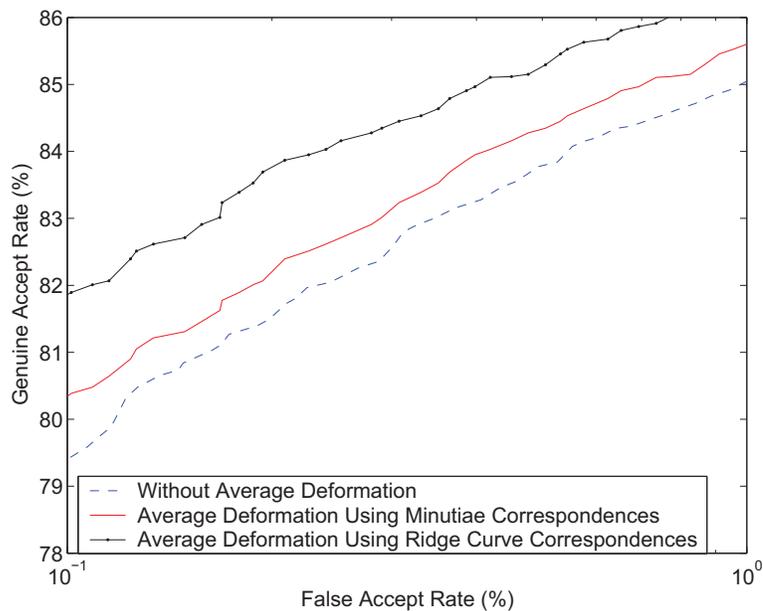


Figure 4.12: Improvement in matching performance when ridge curve correspondences is used to develop the average deformation model.

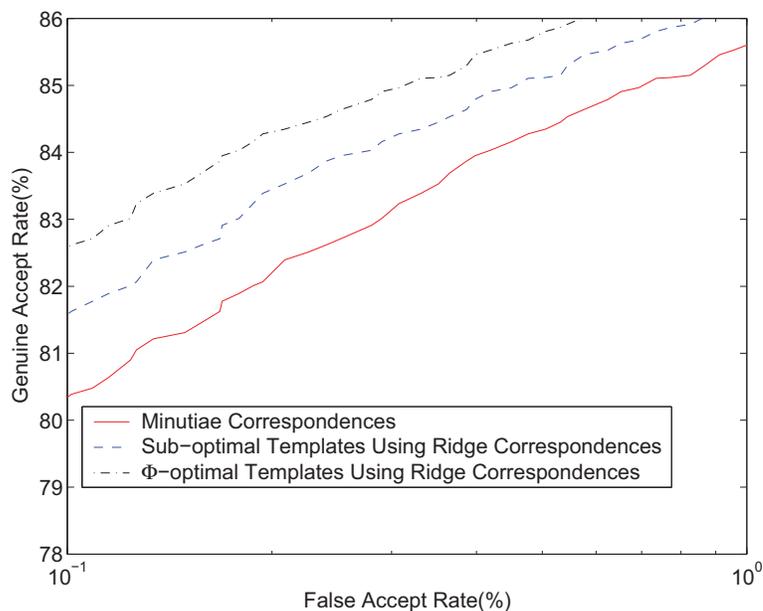


Figure 4.13: Matching performance when the  $\Phi$  index of deformation is used to select optimal templates. Both optimal and suboptimal templates using ridge curve correspondences result in superior matching performance compared to minutiae correspondences.

800 sets ( $50 \times 16$ ) of deformed minutiae points were thus obtained. In order to test the matching performance of the deformed minutiae sets, and the utility of the index of deformation,  $\Phi$ , the following two experiments were conducted. In both these experiments, the minutiae matcher described in [1] was used to generate the matching (similarity) score.

In the first experiment, the matching performance using the average deformation model was evaluated. Every template image,  $T$ , was compared with every query image,  $Q$ , and two types of matching scores were generated for each comparison: the matching score obtained by matching (i)  $M_T$  with  $M_Q$ , and (ii)  $MD_T$  with  $M_Q$ . The Receiver Operating Characteristic (ROC) curve plotting the genuine accept rate (GAR) against the false accept rate (FAR) at various matching thresholds is presented in Figure 4.12. An overall improvement of 2% is observed when the average deformation model is used to distort  $M_T$  prior to matching.

In the second experiment, the advantage of using the index of deformation is demonstrated. The  $\Phi$ -index of deformation (with  $\phi(D) = \text{tr}(D)$ ) of every template image is used to rank the templates according to their variability in the distortion. The template images can now be split into two sets: (i) impressions with the least  $\Phi$  values for every finger (the  $\Phi$ -optimal templates) and (ii) the remaining impressions for every finger (the  $\Phi$ -suboptimal templates). We repeated the matching procedure outlined above using these two template sets. The resulting ROC curve is shown in Figure 4.13. From the figure, it is clear that using  $\Phi$ -optimal templates results in better performance compared to using  $\Phi$ -suboptimal templates. Further, the  $\Phi$ -suboptimal templates still yield better performance compared to the non-distorted

templates thus demonstrating the importance of the average deformable model.

## 4.6 Summary

In this chapter, a deformation model for estimating the distortion effects in fingerprint impressions has been proposed. The distortion is computed based on ridge curve correspondence. It has been shown that the deformation model based on ridge curve correspondence gives superior authentication performance compared to minutiae point pattern matching. The warping model samples the ridge curve and uses thin-plate splines for estimating the non-linear deformation. An index of deformation has also been proposed for selecting the “optimal” template from a given set of fingerprint impressions.

The work presented here can be expanded in several ways. An interesting exercise would be to design an incremental approach to updating the average deformation model, i.e., updating the current average deformation model of a finger by using information presented by newly acquired fingerprint impressions. The technique proposed here uses a simple pixel-wise averaging measure to compute the average deformation model. This measure is sensitive to extreme deformations borne out by outliers; thus, more robust measures of describing the finger specific average deformation model are needed. The effect of the number of training samples (used to develop the average deformation model) on the matching performance has to be systematically studied.

Non-linear deformation in fingerprints is a consequence of utilizing contact-based sensors for imaging the fingertip. Using an acoustic (ultrasound) camera to procure

fingerprint images may help avoid this problem [76]. However, an ultrasound camera introduces other challenges that may affect the quality of the image (e.g., a high quality reconstruction procedure is required to “assemble” an image). Further, the cost of the camera and its size may limit its deployment in real-world applications.

# Chapter 5

## Multibiometrics

In the previous chapters information integration involved looking for complementary information present in a single biometric trait, namely, fingerprint. However, biometric systems using a single biometric trait for authentication purposes have the following limitations:

1. Noise in sensed data: The sensed data might be noisy or distorted. A fingerprint with a scar, or a voice altered by cold are examples of noisy data. Noisy data could also be the result of defective or improperly maintained sensors (e.g., accumulation of dirt on a fingerprint sensor) or unfavorable ambient conditions (e.g., poor illumination of a user's face in a face recognition system). Noisy biometric data may be incorrectly matched with templates in the database (Figure 1.7), resulting in a user being incorrectly rejected.
2. Intra-class variations: The biometric data acquired from an individual during authentication may be very different from the data that was used to generate

the template during enrollment, thereby affecting the matching process. This variation is typically caused by a user who is incorrectly interacting with the sensor (Figure 5.1), or when the sensing operation is modified during the verification phase (e.g., by changing sensors - the sensor interoperability problem). As another example, the varying psychological makeup of an individual, might result in vastly different behavioral traits at various time instances.

3. Restricted degrees of freedom: While a biometric trait is expected to vary significantly across individuals, there may be similarities in the feature sets used to represent these traits. This limitation restricts the degrees of freedom provided by the biometric trait. For example, Gorman [7] states that the discrimination capability of fingerprints (i.e., keyspace) is about  $10^4 - 10^5$ , whereas an 8-character password of randomly chosen characters has a keyspace of  $6.6 \times 10^{15}$ . Every biometric trait has some upper bound in terms of its discrimination capability.
4. Non-universality: While every user is expected to possess the biometric trait being acquired, in reality it is possible for a subset of the users to not possess a particular biometric. A fingerprint biometric system, for example, may be unable to extract features from the fingerprints of certain individuals, due to the poor quality of the ridges (Figure 1.8). Thus, there is a failure to enroll (FTE) rate associated with using a single biometric trait. It has been empirically estimated that around 4% of fingerprint images have poor quality ridges. den Os *et al.* [77] report the FTE problem in a speaker recognition system.

5. Spoof attacks: An impostor may attempt to spoof the biometric trait of a legitimate enrolled user in order to circumvent the system. This type of attack is especially relevant when behavioral traits such as signature [78, 79] and voice [80] are used. However, physical traits are also susceptible to spoof attacks. For example, it has been convincingly demonstrated that it is possible to construct artificial fingers/fingerprints in a reasonable amount of time to circumvent a fingerprint authentication system [81].

	Test	Test Parameter	False Rate	Reject	False Rate	Accept
<b>Fingerprint</b>	FVC 2002 [75]	Users mostly in age group 20-39	0.2%		0.2%	
<b>Face</b>	FRVT 2002 [82]	Enrolment and test images were collected in indoor environment and could be on different days	10%		1%	
<b>Voice</b>	NIST 2000 [83]	Text dependent	10-20%		2-5 %	

Table 5.1: Error rates associated with fingerprint, face and voice biometric systems. The accuracy estimates of biometric systems depend on a number of test conditions.

Some of the limitations imposed by single biometric systems can be overcome by installing multiple sensors that capture different biometric traits. Such systems, known as *multibiometric systems* [84], are expected to be more reliable due to the presence of multiple, independent pieces of evidence [85]. These systems are also able to meet the stringent performance requirements imposed by various applications

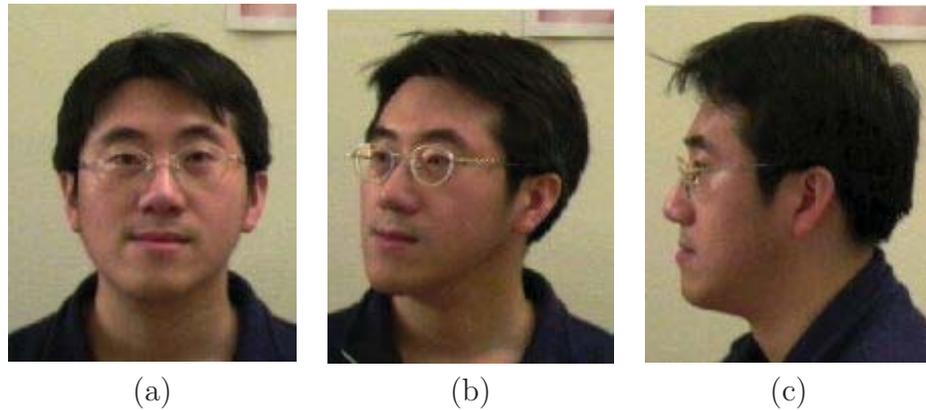


Figure 5.1: Intra-class variation associated with an individual’s face image. Due to change in pose, an appearance-based face recognition system will not be able to match these 3 images successfully, although they belong to the same individual [3].

[86]. Multibiometric systems address the problem of non-universality, since multiple traits ensure sufficient population coverage. Further, multibiometric systems provide anti-spoofing measures by making it difficult for an intruder to spoof the multiple biometric traits of a genuine user simultaneously. By asking the user to present a random subset of biometric traits, the system ensures that a ‘live’ user is indeed present at the point of data acquisition.

Multibiometric biometric systems have received much attention in recent literature. Brunelli et al. [87] describe a multibiometric system that uses the face and voice traits of an individual for identification. Their system combines the scores of five different classifiers operating on the voice and face feature sets, to generate a single matching score that is used for identification. Bigun et al. develop a statistical framework based on Bayesian statistics to integrate information presented by the speech (text-dependent) and face data of a user [88]. Hong et al. use face and fingerprints for person identification [86]. Their system consolidates multiple cues

by associating different confidence measures with the individual biometric matchers. A commercial product called BioID [89] uses voice, lip motion and face features of a user to verify identity. General strategies for combining multiple classifiers have been suggested in [90] and [36]. All the approaches presented in [90] (the highest rank method, the Borda count method and logistic regression) attempt to reduce or rerank a given set of classes. These techniques are thus relevant to the identification problem in which a large number of classes (identities) are present. Prabhakar and Jain show in [91] that selecting classifiers based on some “goodness” statistic may be necessary to avoid performance degradation when using classifier combination techniques. There is also a large amount of literature dealing with combination strategies for matching scores (see for example [92], [93], [94]).

## 5.1 Fusion in Biometrics

Multibiometric systems integrate information presented by single or multiple biometric indicators. The information can be consolidated at various levels. Figure 5.2 illustrates the various levels of fusion possible when combining two (or more) biometric systems. The three possible levels of fusion are: (a) fusion at the feature extraction level, (b) fusion at the matching score level, (c) fusion at the decision level.

1. Fusion at the feature extraction level: The data obtained from each sensor is used to compute a feature vector. As the features extracted from one biometric indicator are independent of those extracted from the other, it is reasonable to concatenate the two vectors into a single new vector. The new feature vector

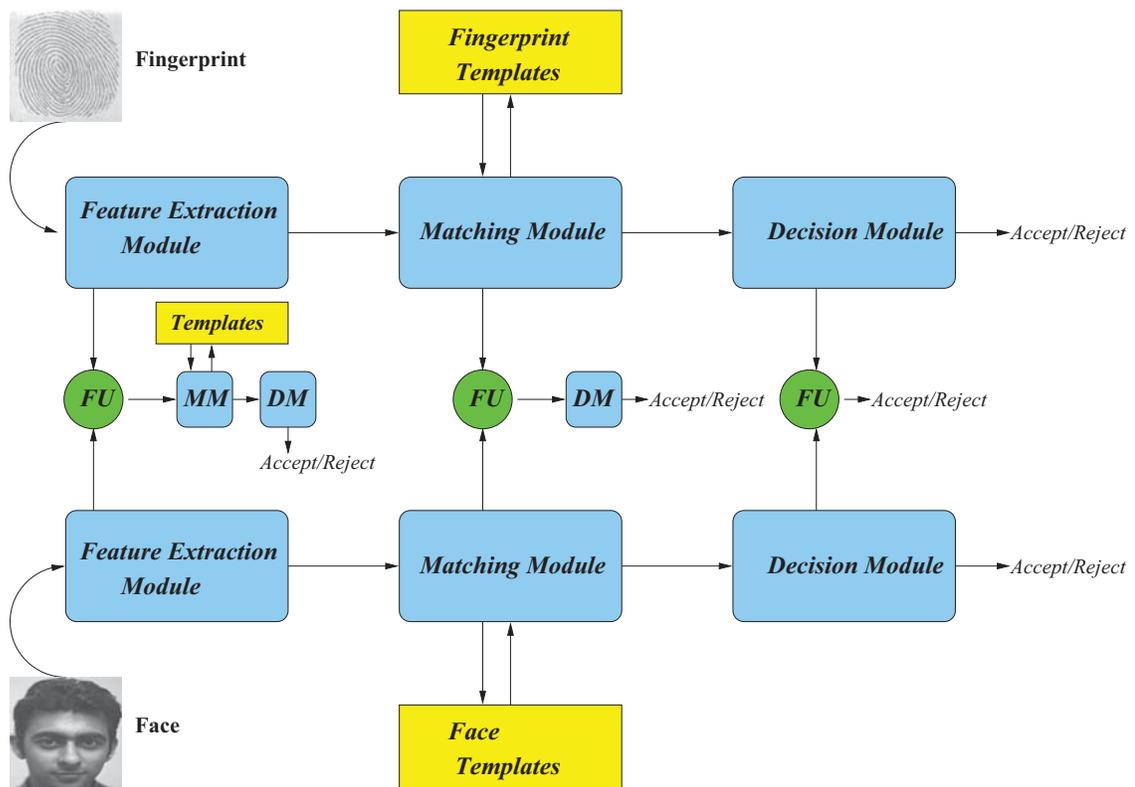


Figure 5.2: A bimodal biometric system showing the three levels of fusion; FU: Fusion Module, MM: Matching Module, DM: Decision Module.

now has a higher dimensionality and represents a person's identity in a different (and, hopefully, more discriminating) hyperspace. Feature reduction techniques may be employed to extract useful features from the larger set of features.

2. Fusion at the matching score level: Each system provides a similarity score indicating the proximity of the input feature vector with the template vector. These scores can be combined to assert the veracity of the claimed identity. Techniques such as logistic regression may be used to combine the scores reported by the two sensors. These techniques attempt to minimize the FRR for a given FAR [95].
3. Fusion at the decision level: Each sensor captures a biometric attribute and the resulting feature vectors are individually classified into the two classes - accept or reject. A majority vote scheme, such as that employed in [96] can be used to make the final decision.

Fusion in the context of biometrics can take the following forms:

1. Single biometric multiple representation.
2. Single biometric multiple matchers.
3. Multiple biometric fusion.

### **5.1.1 Single Biometric Multiple Representation**

This type of fusion involves using multiple representations of a single biometric indicator. Typically, each representation has its own associated matcher/classifier. The

similarity scores reported by these classifiers are then consolidated. Cappelli et al. [97] describe a fingerprint *classification* system <sup>1</sup> that combines a structural classifier with a KL transform-based classifier by integrating the scores generated by the two classifiers. This is done by first mapping the scores (which are distance measures) into a common domain via a double sigmoid function and then taking a weighted average in the new domain. Jain et al. [98] also use multiple classifiers for fingerprint indexing. Their technique uses a  $K$  nearest neighbor classifier and a set of 10 neural network classifiers to classify fingerprints. General strategies for combining multiple classifiers have been suggested in [90]. All the approaches presented there (the highest rank method, the Borda count method and logistic regression) attempt to reduce or rerank a given set of classes. These techniques are thus relevant to the identification problem in which a large number of classes (identities) are present.

The fusion in this approach takes place at the matching stage, after the classifiers report a similarity score for each class. Prabhakar and Jain [91] observe that the classifiers used for fusion have to be carefully selected in order to avoid performance degradation.

### 5.1.2 Single Biometric Multiple Matchers

It is also possible to incorporate multiple matching strategies in the matching module of a biometric system and combine the scores generated by these strategies. Jain et al. [95] use the logistic function to map the matching scores obtained from two

---

<sup>1</sup>The system classifies a fingerprint into one of five classes - Arch, Left loop, Right loop, Whorl and Tented arch. Thus it is not a biometric system by itself, as it does not perform person identification or verification.

different fingerprint matching algorithms into a single score. The authors demonstrate that such an integration strategy improves the overall performance of a fingerprint verification system.

This type of fusion also takes place at the matching stage of a biometric system. Although there are multiple matchers in this case, all matchers operate on the same representation of the biometric data.

### 5.1.3 Multiple Biometric Fusion

Multimodal fusion refers to the fusion of multiple biometric indicators. Such systems seek to improve the speed and reliability (accuracy) of a biometric system [86] by integrating matching scores obtained from multiple biometric sources. A variety of fusion schemes have been described in the literature to combine these various scores. These include majority voting, sum and product rules,  $k$ -NN classifiers, SVMs, decision trees, Bayesian methods, etc. (see for example [92, 36, 88, 99, 93, 94]).

An important aspect that has to be addressed in fusion at the matching score level is the normalization of the scores obtained from the different domain experts [87]. Normalization typically involves mapping the scores obtained from multiple domains into a common domain before combining them. This could be viewed as a two-step process in which the distributions of scores for each domain is first estimated using robust statistical techniques [100] and these distributions are then scaled or translated into a common domain.

Besides the techniques described above, other types of fusion are also possible in

biometrics. (i) A fingerprint biometric system may store multiple templates of a user's fingerprint (same finger) in its database. When a fingerprint impression is presented to the system for verification, it is compared against each of the templates, and the matching score generated by these multiple matchings are integrated. (ii) A system may store a single template of a user's finger, but acquire multiple impressions of the finger during verification. (iii) Another possibility would be to acquire and use impressions of multiple fingers for every user. These possibilities have been discussed in [101].

## 5.2 Multibiometric System

The choice and number of biometric traits used in designing a multibiometric system are variables that have to be decided before deploying the system. Fingerprint systems exhibit very good performance and are very popular since the sensors used to acquire them can be easily embedded in a system. Most of the working population have hands and, therefore, hand geometry would be a good choice also. Face images can be easily acquired and even non-experts can compare face images in the case of repudiation claims where an individual uses a facility and then denies having used it. We use face, hand geometry and fingerprints to design and evaluate our multibiometric system. A brief description of the face and hand geometry indicators used in our experiments is given below.

### 5.2.1 Face Verification

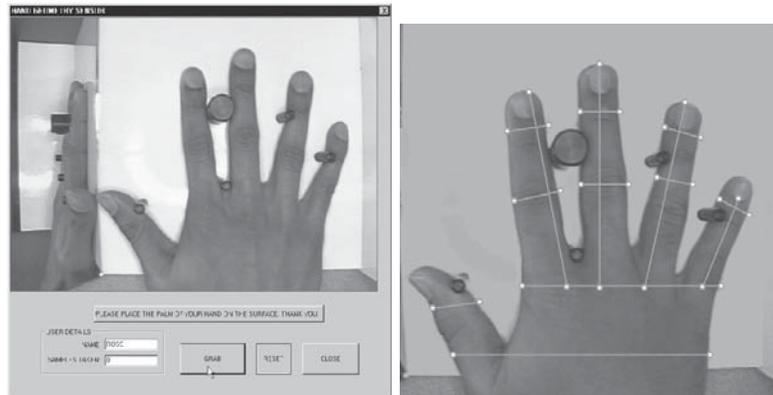
Face verification involves extracting a feature set from a two-dimensional image of the user's face and matching it with the template stored in the database. The feature extraction process is often preceded by a face detection process during which the location and spatial extent of the face is determined within the given image. This is a difficult process given the high degree of variability associated with human faces (color, texture, expression, pose, etc.). The problem is further compounded by the presence of complex backgrounds and variable lighting conditions (Figure 5.3). A variety of techniques have been described in the literature to locate the spatial coordinates of a face within an image [102, 103, 104]. Once the boundary of the face is established, we use the eigenface approach to extract features from the face [105, 106]. In this approach a set of orthonormal vectors (or images) that span a lower dimensional subspace is first computed using the principal component analysis (PCA) technique. The feature vector of a face image is the projection of the (original face) image on the (reduced) eigenspace. Matching involves computing the Euclidean distance between the eigenface coefficients of the template and the detected face.



Figure 5.3: The problem of face detection is compounded by the effects of complex lighting and cluttered background.

## 5.2.2 Hand Geometry

Hand geometry, as the name suggests, refers to the geometric structure of the hand. This structure includes width of the fingers at various locations, width of the palm, thickness of the palm, length of the fingers, etc. Although these metrics do not vary significantly across the population, they can nevertheless be used to verify the identity of an individual. Hand geometry measurement is non-intrusive and the verification involves a simple processing of the resulting features. Figure 5.4 shows the hand geometry system that was used in our experiments. The system computes 14 feature values comprising of the lengths of the fingers, widths of the fingers and widths of the palm at various locations (Appendix A).



(a) GUI for capturing hand geometry. The five pegs aid in proper placement of the hand on the platen.

(b) 14 hand features corresponding to various length and width measurements.

Figure 5.4: A Hand Geometry System. The sensor provides both the top and side views of the subject's hand. Features are extracted using the image of the top-view only.

### 5.2.3 Combining the three modalities

The database for our experiments consisted of matching scores obtained from three different modalities - face, fingerprint and hand geometry. However, data pertaining to all three modalities were not available for a single set of users. The mutual non-dependence of the biometric indicators allows us to assign the biometric data of one user to another.

The database itself was constructed as follows: The fingerprint and face data were obtained from user set I consisting of 50 users. Each user was asked to provide nine face images and nine fingerprint impressions (of the same finger). This data was used to generate 3,600 ( $50 \times 9 \times 8$ ) genuine scores and 22,050 ( $50 \times 49 \times 9$ ) impostor scores for each modality. The hand geometry data was collected separately from user set II also consisting of 50 users <sup>2</sup>. This resulted in 3,600 genuine scores and 22,050 impostor scores for this modality also. Each user in set I was randomly paired with a user in set II. Thus corresponding genuine and impostor scores for all three modalities were available for testing. All scores were mapped to the range  $[0, 100]$ . Since the face and hand scores were not similarity scores (they were distance scores), they were converted to similarity scores by simply subtracting them from 100. A similarity score  $x$  represents the proximity of two feature sets as computed by a classifier. A *score vector* represents the scores of multiple classifiers. Thus, the vector  $(x_1, x_2, x_3)$  is a score vector, where  $x_1$ ,  $x_2$  and  $x_3$  correspond to the (similarity) scores obtained from the classifiers corresponding to the face, fingerprint and hand geometry systems,

---

<sup>2</sup>Some users from set I were present in set II.

respectively.

The scatter plot of the genuine and impostor scores (3,600 genuine scores, and 11,025 impostor scores) is shown in Figure 5.5. The plot indicates that the two classes are reasonably separated in 3-dimensional space; therefore, a relatively simple classifier should perform well on this dataset. The ROC curves depicting the performance of the individual modalities are shown in Figure 5.6.

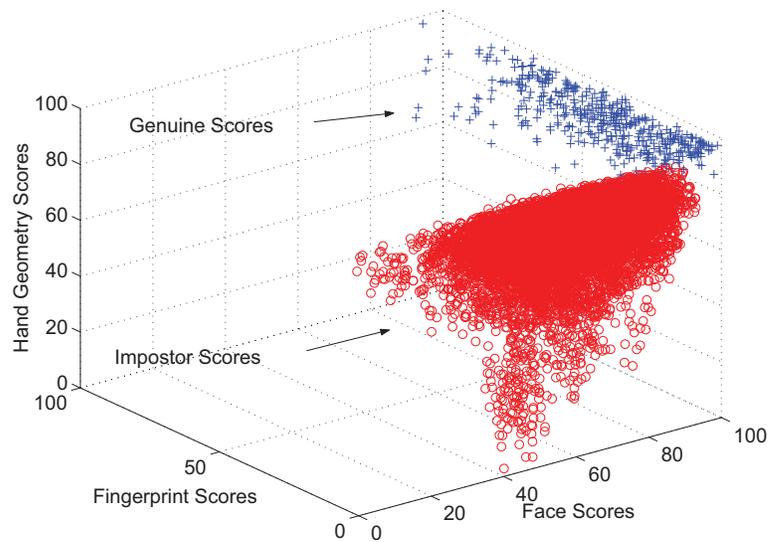


Figure 5.5: Scatter Plot showing the genuine and impostor scores in 3-D space. The points correspond to 3,600 genuine scores (+) and 11,025 impostor scores (o).

## Sum Rule

The simplest form of combination would be to take the weighted average of the scores from the multiple modalities. This strategy was applied to all possible combinations of the three modalities. Equal weights were assigned to each modality as the bias of each classifier was not computed. Figure 5.7 and Figure 5.8 show the performance of

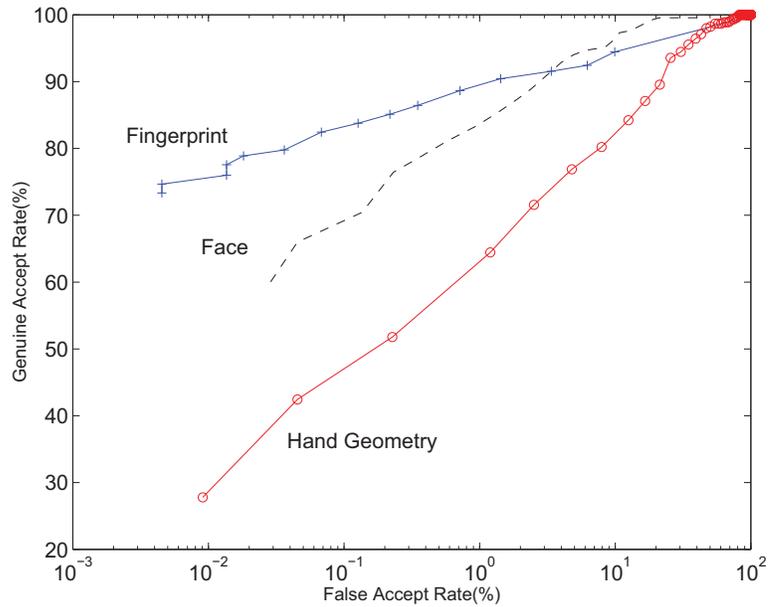


Figure 5.6: ROC curves showing the performance of each of the three individual modalities.

the sum rule on the three modalities.

## Decision Trees

A decision tree derives a sequence of *if-then-else* rules using the training set in order to assign a class label to the input data. It does this by finding out an attribute (feature) that maximizes information gain at a particular node. The *C5.0* program [107] was used to generate a tree from the training set of genuine and impostor score vectors. The training set consisted of 11,025 impostor score vectors and 1,800 genuine score vectors. The test set consisted of the same number of impostor and genuine score vectors. Figure 5.9 shows the construction and performance of the *C5.0* decision tree on the training and test sets.

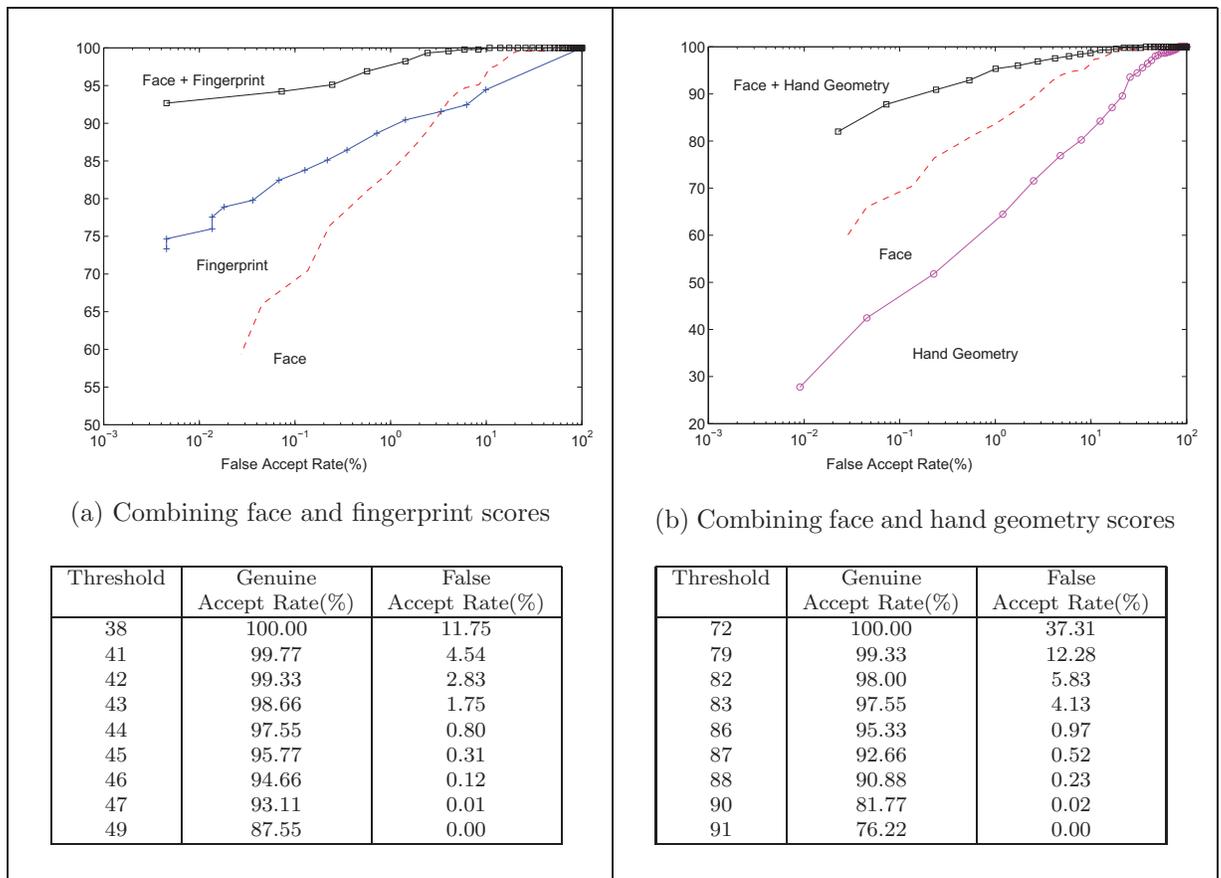


Figure 5.7: ROC curves showing an improvement in performance when scores are combined using the sum rule.

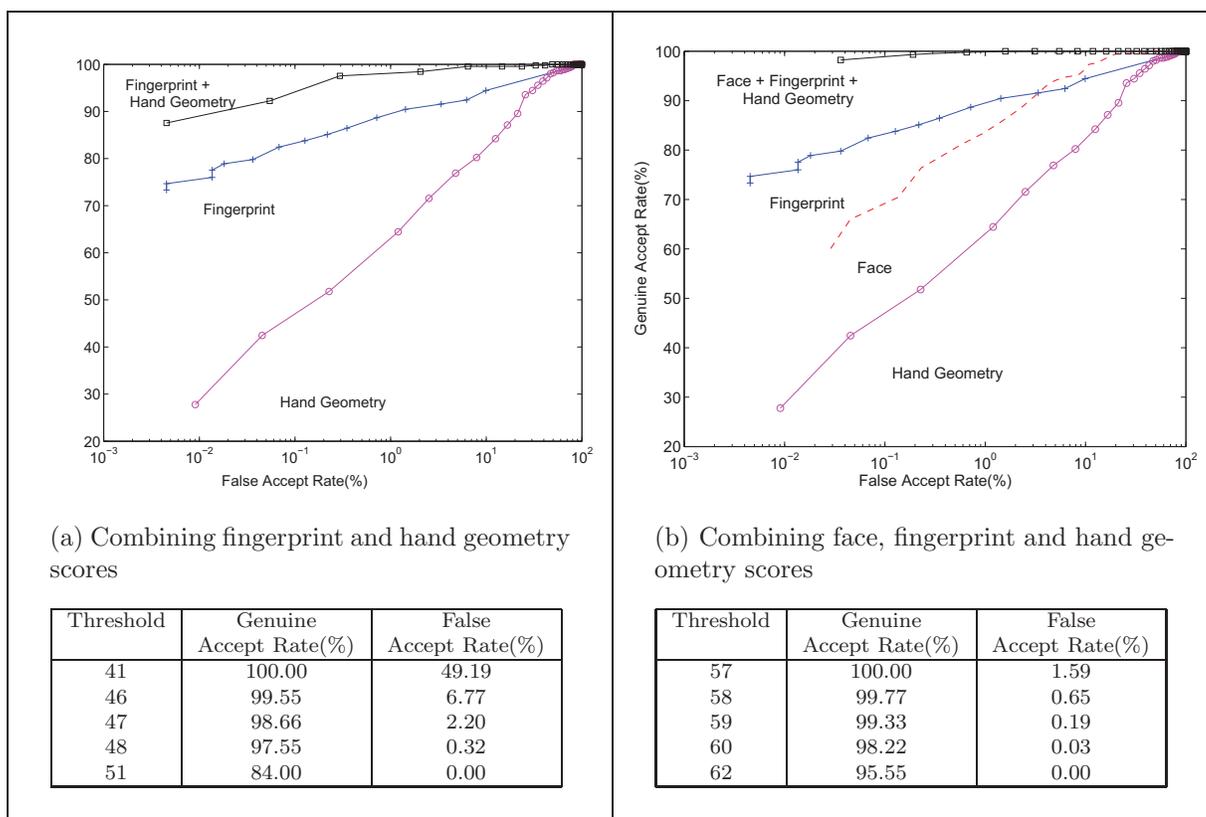
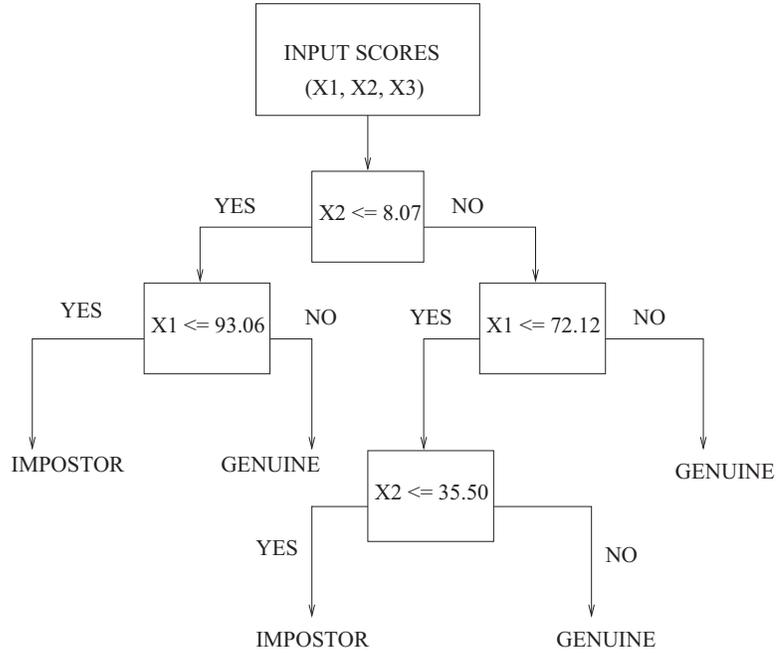


Figure 5.8: ROC curves showing an improvement in performance when scores are combined using the sum rule.



Evaluation on training data (11, 250 score vectors):

	Genuine Class	Impostor Class
Genuine Class	1, 720	80
Impostor Class	2	11, 023

Evaluation on test data (11, 250 score vectors):

	Genuine Class	Impostor Class
Genuine Class	1, 624	176
Impostor Class	4	11, 021

Figure 5.9: Construction and Performance of the C5.0 Decision Tree. The performance is indicated by confusion matrices.

## Linear Discriminant Function

Linear discriminant analysis of the training set helps in transforming the 3-dimensional score vectors into a new subspace that maximizes the between-class separation. Figure 5.10 shows the plot of the score vectors using the first and the second discriminant variables.

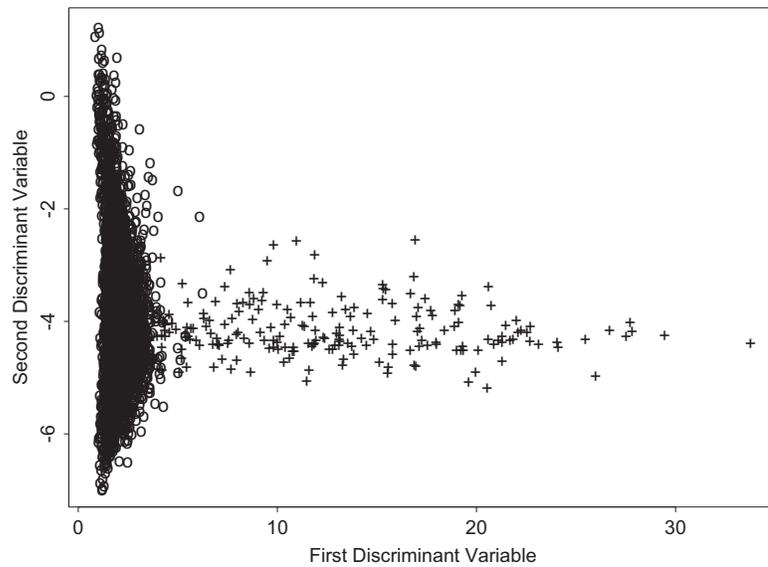


Figure 5.10: Linear Discriminant Analysis of the score vectors. The score vectors have been plotted in a 2-dimensional space representing the first and the second discriminant variables. There are 1,800 genuine score vectors(+) and 11,025 impostor score vectors (o).

The test set vectors are classified by using the minimum Mahalanobis distance rule (after first calculating the centroids of the two classes in the new feature space, and then measuring the Mahalanobis distance). We assume that the two classes have unequal covariance matrices. Table 5.2 shows the confusion matrices resulting from using this rule on different partitioning of the data into training and test sets.

		Genuine Class	Impostor Class
Trial 1:	Genuine Class	1,800	0
	Impostor Class	54	10,971
		Genuine Class	Impostor Class
Trial 2:	Genuine Class	1,800	0
	Impostor Class	50	10,975
		Genuine Class	Impostor Class
Trial 3:	Genuine Class	1,800	0
	Impostor Class	72	10,953

Table 5.2: Performance of the linear discriminant classifier on three different trials as indicated by the confusion matrices. In each trial the training and test sets were partitioned differently.

## Discussion

The experiments described above suggest that the sum rule performs better than the decision tree and linear discriminant classifiers. The FAR of the tree classifier is 0.036% ( $\pm 0.03\%$ ) and the FRR is 9.63% ( $\pm 0.03\%$ ). The FAR of the linear discriminant classifier is 0.47% ( $\pm 0.3\%$ ) and its FRR is 0.00%. The FRR value in this case is a consequence of overfitting the genuine class as it has fewer samples in both the test and training sets. The sum rule that combines all three scores has a corresponding FAR of 0.03% and a FRR of 1.78% suggesting better performance than the other two classifiers. It has to be noted that it is not possible to fix the FRR (and then compute the FAR) in the case of the decision tree and linear discriminant classifiers.

### 5.3 Learning User-specific Parameters

We develop techniques to further improve system performance in a multibiometric system by learning user-specific matching thresholds and weighting, for individual traits. Matching thresholds are used to decide if a matching score corresponds to a genuine user or an impostor. Scores greater than the matching threshold indicate a genuine user; scores lower than the threshold indicate an impostor. Biometric systems typically use a common threshold across users. We show that by setting user-specific thresholds, it is possible to improve system performance. Weighting is used to vary the importance of matching scores of each biometric trait. By learning user-specific weights, the performance of the system is shown to improve.

The automatic learning and update of system parameters help reduce the error rates associated with an individual, thereby improving the performance accuracy of the system. In a multibiometric system, it is essential that different biometric traits be given various degrees of importance for different users. This is important especially when the biometric trait of some users cannot be reliably acquired. For example, users with persistently dry fingers may not be able to provide good quality fingerprints. Such users might experience higher false rejects when interacting with a fingerprint system. By reducing the weight of the fingerprint trait of such users, and increasing the weights associated with the other traits the false reject error rate of these users can be reduced. Further, the intra-class and inter-class variability associated with a single biometric trait varies between users. Consequently, different users are prone to different types of errors. The *false reject rate* (FRR) of users with large inter-class

variations may be high. Similarly, the *false accept rate* (FAR) associated with users having small inter-class variations may be high. Thus, the system invokes a specific set of parameters based on the claimed identity,  $I$ . The biometric system learns user-specific parameters by observing system performance over a period of time. This will appeal to that segment of the population averse to interacting with a system that constantly requests a user to provide multiple readings of the same biometric. The emphasis is on tuning the system variables automatically, yet appropriately, to attain performance gain. Therefore, adaptation in a multibiometric system entails the following: (a) developing user-specific matching thresholds, and (b) assigning weights to individual biometric traits.

### 5.3.1 User-specific matching thresholds

The matching thresholds for each user is computed using the cumulative histogram of impostor scores corresponding to that user. Since genuine scores (matching scores generated when comparing feature sets from the same user) would not be available when the user begins to use the system, the impostor scores are used for this purpose. The impostor scores are generated by comparing the feature sets of the user with feature sets of other users or with feature sets available in a predetermined impostor database. The cumulative histogram at a value  $x$ ,  $x = 1, 2, \dots, 100$ , is the sum of all those impostor scores lesser than or equal to  $x$ . The user-specific matching thresholds are computed as follows:

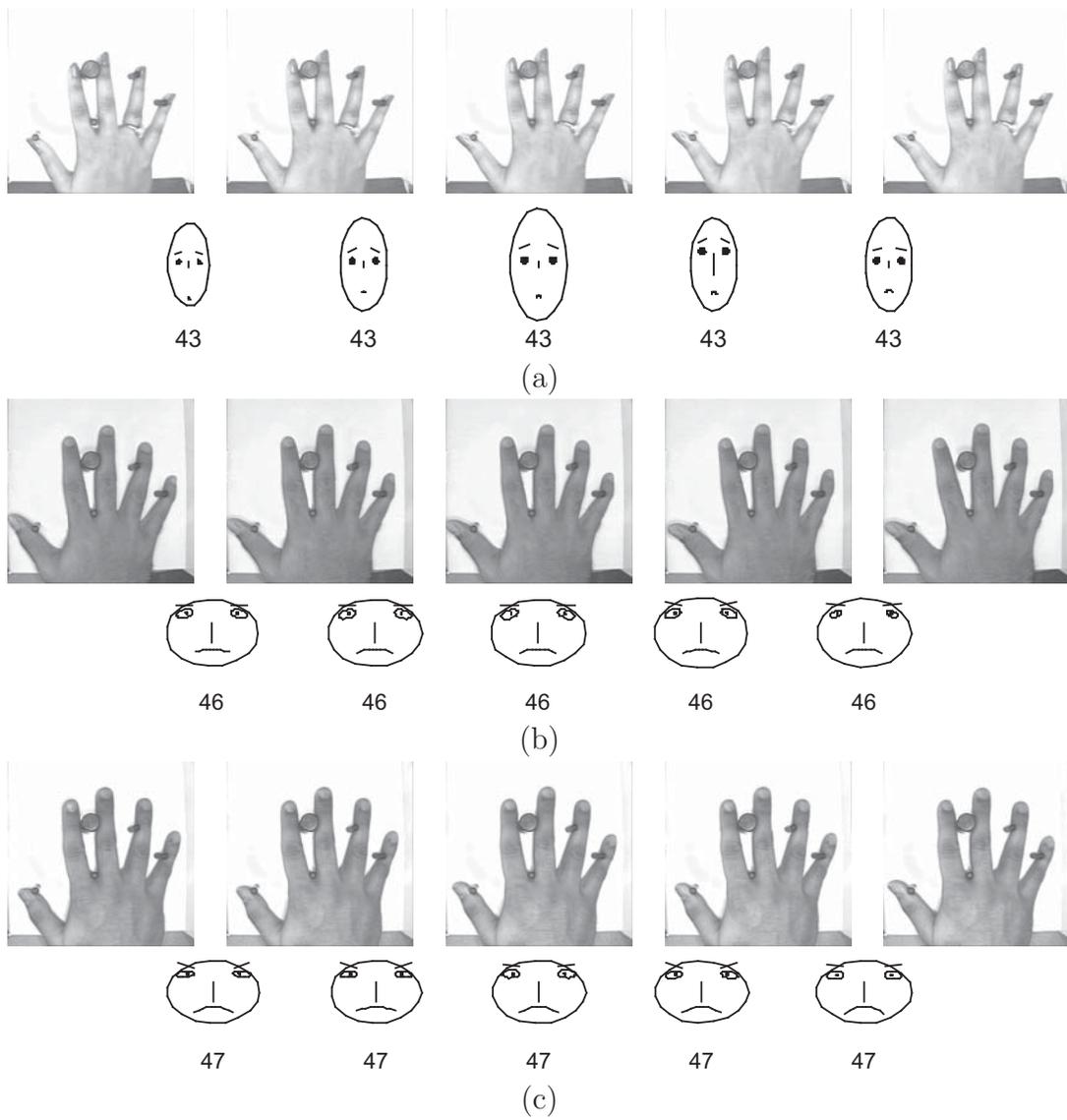


Figure 5.11: Intra-class variability and inter-class similarity of hand geometry features depicted using Chernoff faces. (a) The feature sets associated with a user showing large variability. (b) and (c) The feature sets associated with two different users exhibiting similarity [4]

1. For the  $i^{th}$  user in the database, let  $t_i(\gamma)$  correspond to the threshold in the cumulative histogram that retains  $\gamma$  fraction of scores,  $0 \leq \gamma \leq 1$ .
2. Using  $\{t_i(\gamma)\}$  as the matching threshold, compute  $\{FAR_i(\gamma), GAR_i(\gamma)\}$ , where GAR is the genuine accept rate.
3. Compute the total FAR and GAR as,  $FAR(\gamma) = \sum_i FAR_i(\gamma)$ ,  $GAR(\gamma) = \sum_i GAR_i(\gamma)$ .
4. Use  $\{FAR(\gamma), GAR(\gamma)\}$  to generate the ROC curve.

We observe that the choice of threshold relies on the distribution of impostor scores for each user (Figure 5.12). This is in contrast to traditional methods where the threshold is established by pooling together the impostor scores associated with all users. When the multibiometric system is deployed, the  $\gamma$  corresponding to a specified FAR is used to invoke the set of user-specific thresholds,  $\{t_i(\gamma)\}$ . Table 5.3 shows the user-specific thresholds (corresponding to a FAR of 1%) associated with the 10 users whose data was collected over a period of two months. The ROC curves indicating the improved performance is shown in Figure 5.13.

### 5.3.2 Weighting individual biometric traits

Each biometric trait provides a matching score based on the input feature set provided and the template against which the input is compared with. These scores are weighted according to the biometric trait used ( $w_1$  for finger,  $w_2$  for face and  $w_3$  for hand geometry), in order to reduce the importance of less reliable biometric traits (and

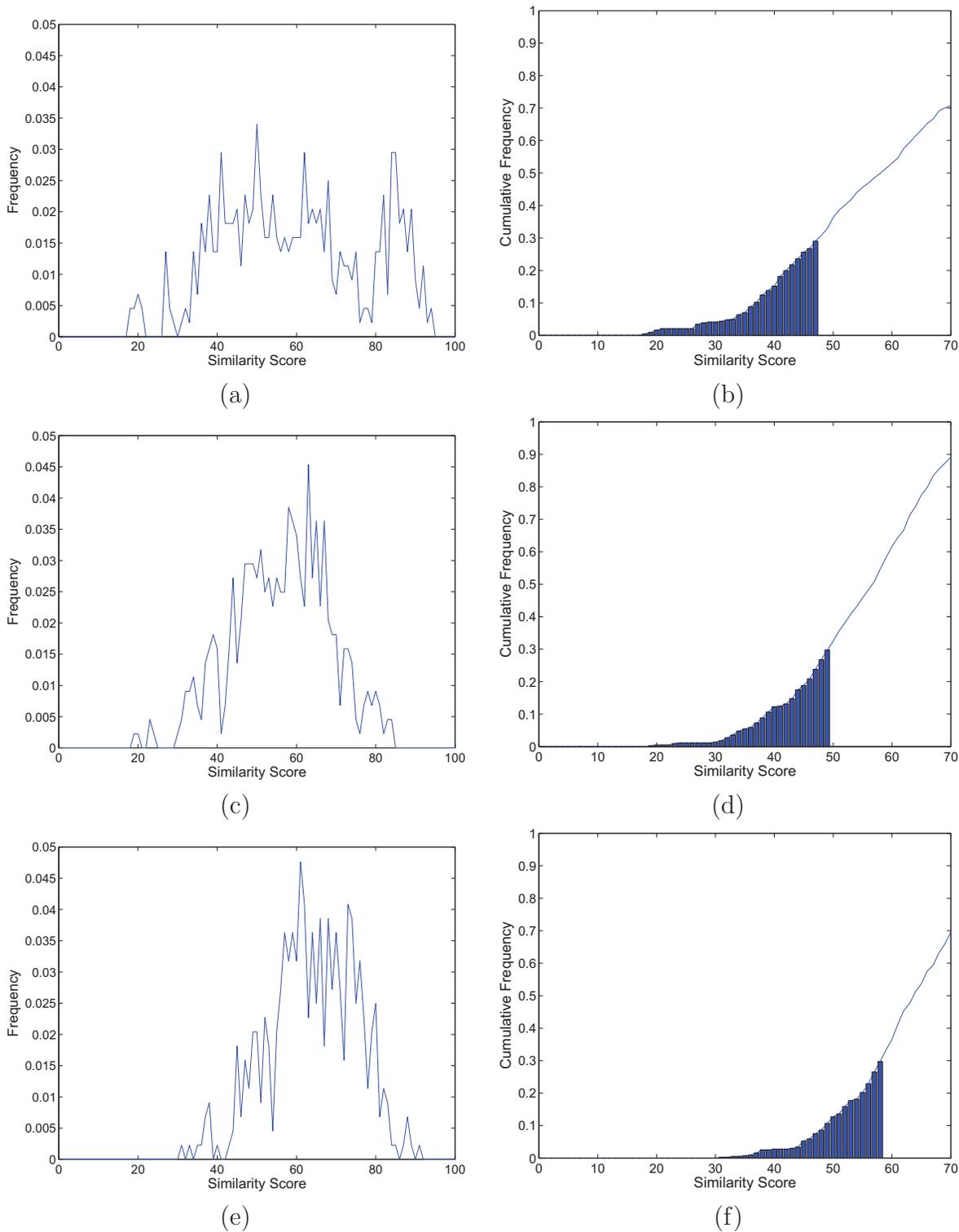


Figure 5.12: The impostor distributions of the face biometric feature vector of 3 users. (a), (c) and (e) are the histogram of impostor scores associated with the 3 users. (b), (d) and (f) are the corresponding cumulative histograms. For  $\gamma = 0.3$  it is observed that the thresholds for each of the users is different.

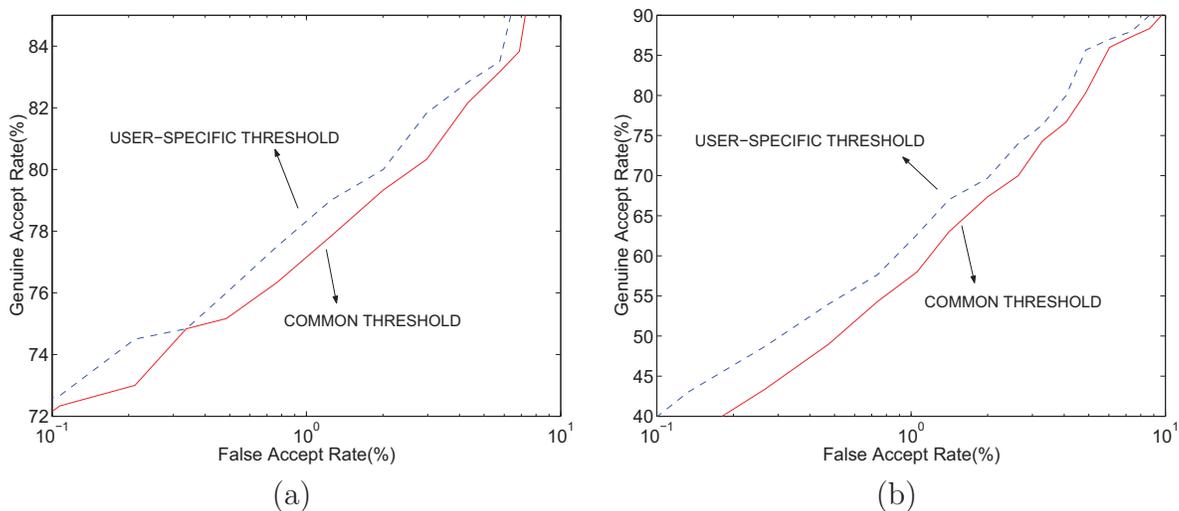


Figure 5.13: ROC curves exhibiting performance improvement when user-specific thresholds are utilized to verify claimed identity. (a) Fingerprint; (b) Face.

User #	Fingerprint	Face	Hand Geometry
1	14	91	94
2	17	91	95
3	15	92	95
4	12	94	95
5	11	91	90
6	11	90	92
7	16	95	94
8	19	92	97
9	11	90	96
10	19	94	93

Table 5.3: User-specific thresholds for the biometric traits of 10 users corresponding to a FAR of 1% in each ROC curve.

increase the influence of more reliable traits). Weighting the matching scores can be done in the following ways.

### Weighting all traits equally and using a user-specific threshold

Equal weights are assigned to the face, hand and fingerprint matching scores, and a new score is obtained as,

$$S_{fus} = \sum_{k=1}^3 \frac{1}{3} S_k \quad (5.1)$$

The user specific threshold is computed using the impostor distribution of  $S_{fus}$  (for each user) using the procedure outlined in section 5.3.1. The performance improvement can be seen in Figure 5.14(a).

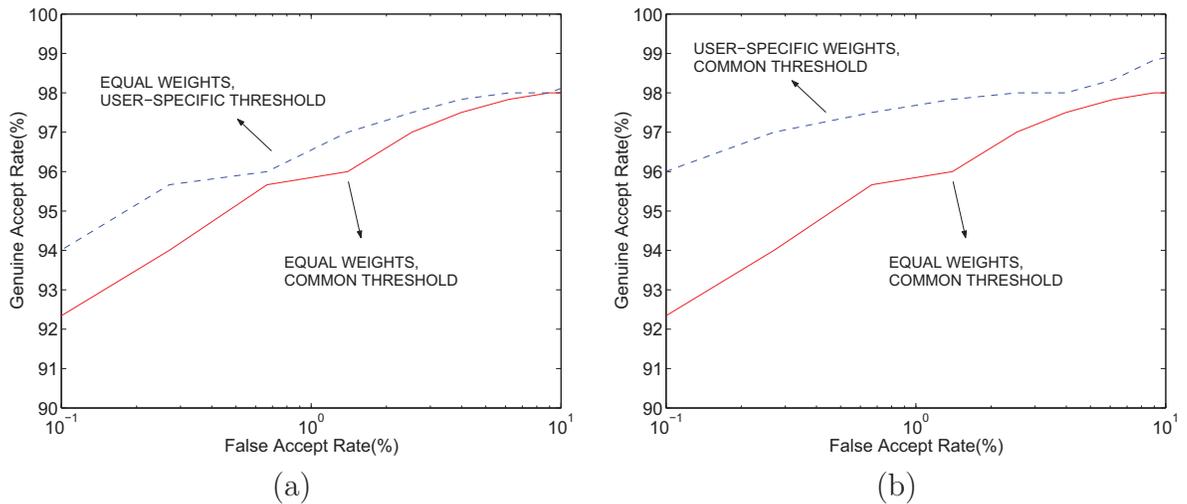


Figure 5.14: ROC curves when using (a) equal weights for all three traits, and a user-specific matching threshold, (b) user-specific weights for all three traits and a common matching threshold.

User #	Fingerprint ( $w_1$ )	Face ( $w_2$ )	Hand Geometry ( $w_3$ )
1	0.5	0.3	0.2
2	0.6	0.2	0.2
3	0.4	0.1	0.5
4	0.2	0.4	0.4
5	0.5	0.2	0.3
6	0.6	0.1	0.3
7	0.6	0.1	0.3
8	0.4	0.2	0.4
9	0.5	0.1	0.4
10	0.6	0.2	0.2

Table 5.4: Weights of different traits for 10 users.

### Estimating user-specific weights by ‘exhaustive’ search and using a common matching threshold

Here, the optimal weight for each trait is found by searching the space of weights  $((w_1, w_2, w_3))$  for each user with the constraint,  $w_1 + w_2 + w_3 = 1$ , such that the total error rate is minimized. The total error rate is the sum of the false accept and false genuine rates.

1. For the  $i^{th}$  user in the database, vary weights  $w_{1,i}$ ,  $w_{2,i}$  and  $w_{3,i}$  over the range  $[0, 1]$ , with the constraint  $w_{1,i} + w_{2,i} + w_{3,i} = 1$ . Compute  $S_{fus} = w_{1,i}S_1 + w_{2,i}S_2 + w_{3,i}S_3$ .
2. Choose that set of weights that minimizes the total error rate associated with the scores.

The total error rate is the sum of the false accept and false reject rates.

Table 5.4 lists the weights computed for the 10 users listed in table 5.3. Since the weight estimation procedure utilizes the histograms of both the genuine and impostor

scores, computing user-specific thresholds using impostor scores does not further improve performance. We, therefore, use a common matching threshold. The resulting performance is indicated by the ROC curve in Figure 5.14(b). From table 5.4, we observe that for user 4,  $w_1 = 0.2$ , a very low value. To understand this, we examined the fingerprints corresponding to this user (Figures 5.15(a) and 5.15(b)). We note that the ridge details are not very clear, and therefore the minutiae matching algorithm did not provide correct matching scores. This demonstrates the importance of assigning user-specific weights to the individual biometric trait. Similarly, user 3 has a very small weight attached to the face biometric, possibly due to varying face poses and lighting during data acquisition (Figure 5.15(c), 5.15(d) and 5.15(e)). User 2 has a small weight attached to hand geometry due to incorrect placement of the hand and a curved little finger (Figures 5.15(f) and 5.15(g)).

## 5.4 Factors Affecting a User-specific Approach

The following issues will have to be considered while deploying biometric systems with user-specific thresholds and weights:

1. A ‘smart’ user may deliberately provide poor quality biometric data constantly (e.g., by touching the fingerprint sensor lightly), thereby forcing the system to reduce the weights associated with a specific biometric. The user may then claim that the biometric data belongs to someone else. Thus, the user accesses a privilege, and later denies using it. To address such repudiation claims, one could use the face images acquired during the verification phase to invalidate

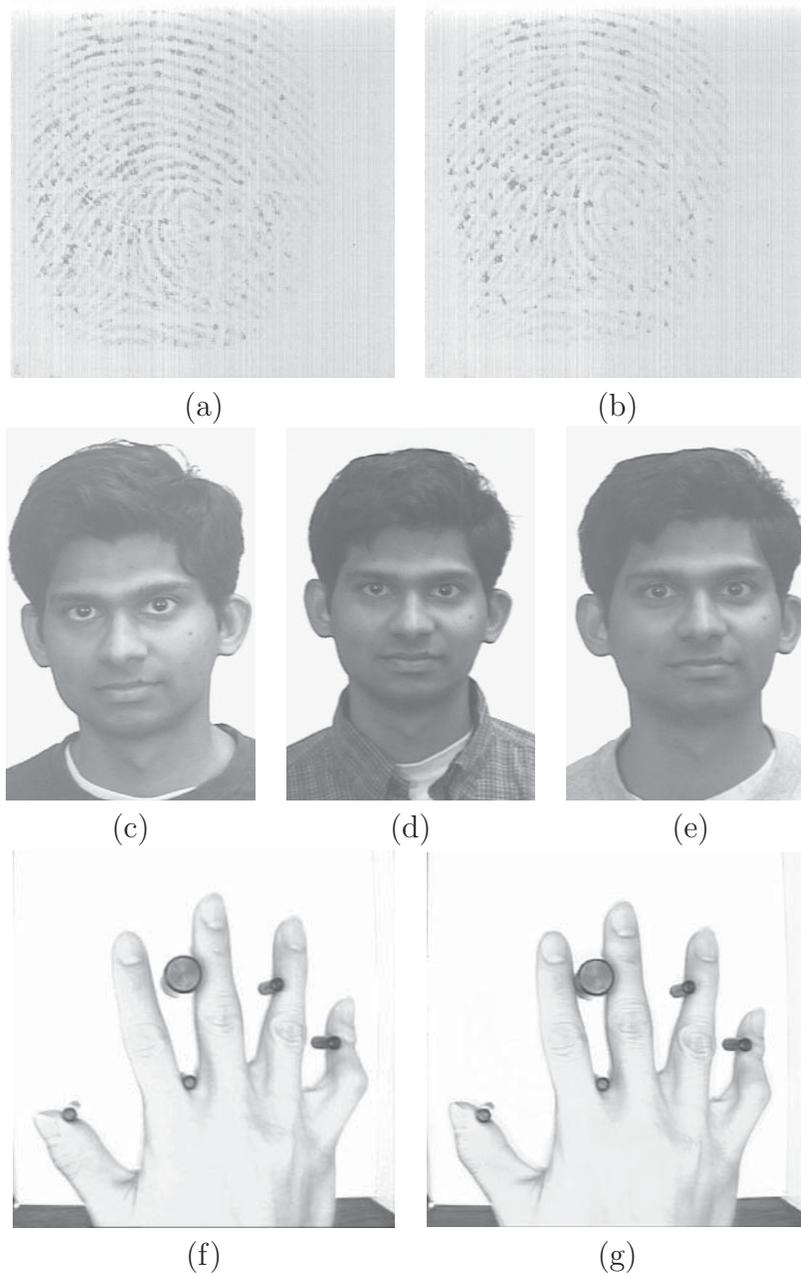


Figure 5.15: (a) and (b) Fingerprint images of user 4 whose ridge details are not very clear ( $w_1 = 0.2$ ); (c),(d) and (e) Varying face poses of user 3 ( $w_2 = 0.1$ ); (f) and (g) Incorrect placement of hand and the curved finger of user 2 ( $w_3 = 0.2$ .)

the user's claim. However, this is a post-incident solution in that it is invoked after the weights have been lowered. A more effective solution would be to flag unnatural alterations observed in the weight vector for every user.

2. An intruder attempting to violate a biometric system might target users with known problems with their biometric data (e.g., users with calloused fingers or arthritis of the hand). Such users will have low weights attached with certain biometric traits and, therefore, the intruder has to spoof only those traits having higher weights.
3. While acquiring biometric data from a user, certain extraneous (non-biometric) information could also be used to verify the claimed identity. For example, if a certain user typically enters a high-security room only during the day, then the system could raise a flag when the user attempts to access the facility at night. At this time, additional more-expensive biometric measurements may be obtained to facilitate access.
4. The techniques mentioned in the chapter could also be used for a uni-biometric system. For example, a minutiae-based fingerprint matcher could learn to weight various regions of a user's fingerprint differently by merely observing matching minutiae pairs.

## 5.5 Summary

In this chapter we have demonstrated the following:

1. The performance of a biometric system can be enhanced by computing user-specific thresholds. These thresholds may be computed for individual biometric traits, or may be computed after consolidating the scores provided by individual traits. This improves user convenience by reducing the false reject rate.
2. The performance can be further improved by assigning user-specific weights to the various traits when combining the matching scores. This weighting can be computed from the training data of every user.

Multibiometrics eliminates the failure to enroll problem. By assigning smaller weights to those traits for which a user's sample are consistently noisy, one can accommodate more people in the system. These parameters can be estimated from the training data provided by the user. Future work could involve looking at ways to perform template selection and template update. The choice of templates to represent feature sets, and the methodology initiated to update them would help improve system performance (and user convenience) over a period of time. However, automatic template selection (and update) has to be done in a reliable manner to prevent unauthorized users from progressively replacing the biometric template of a legitimate user with their traits.

The evidence presented by multiple traits can be used in different ways depending on whether the authentication scheme requires *identifying* the individual or *verifying* the claimed identity. As observed earlier, identification potentially requires matching the extracted feature set against *all* templates (identities) in the database in order to find out the best match. This could be prohibitive in terms of the number of compar-

isons to be performed (especially when feature sets of multiple traits are involved), resulting in a slow response time which is ill-suited in a real-time identification system. To circumvent this, an indexing scheme, wherein the extracted feature set is compared against a small subset of the database, is used to reduce the number of comparisons to be performed and improve processing time. Hong and Jain [86] use a face matching system to retrieve the top 5 matches for a user's identity; a fingerprint matcher is then applied to these 5 matches, and a final decision made by combining the scores provided by these two matchers. Thus, the face and fingerprint systems operate in the cascade mode, where one system is used to narrow down the number of possible identities, before the other is used. In a verification scheme, on the other hand, indexing is not required since the extracted feature set has to be compared with only those templates corresponding to the claimed identity. In such systems the feature sets extracted from multiple traits may be simultaneously used to produce matching scores which are then consolidated to arrive at a decision. The individual biometric systems, in this case, operate in the parallel mode. The techniques presented in this chapter are more suited for a verification system.

# Chapter 6

## Conclusions and Future Work

### 6.1 Research Contributions

Researchers have invested a substantial amount of effort in studying fingerprints in the context of forensics, computer vision, image processing, pattern recognition, and biometrics. This seemingly simple pattern of ridges and valleys on the tip of a finger has, therefore, received considerable attention. In spite of this attention, the problem of automatic fingerprint matching continues to harbor plenty of challenges.

In this thesis we have approached the challenges in fingerprint matching from an information fusion perspective. We first developed a hybrid matcher that combined the minutiae and texture information present in a fingerprint. In the proposed technique, the texture information was represented using ridge feature maps which could also be utilized to align and register pairs of fingerprint images. The hybrid system presented here was shown to enhance the matching performance of a fingerprint system. One of the advantages of using a hybrid approach is the complementary nature

of the information being combined. This aids in decreasing the false accept rate (i.e., the false match rate) as follows: by examining the underlying ridge information via ridge feature maps after aligning the images using minutiae points, the integrity of the matcher is improved.

We then described a mosaicking scheme that integrated the information present in two impressions of a finger. In this technique, fingerprint images were viewed as range images, and the iterative control point (ICP) algorithm was used to register them. The technique presented here utilized minutiae points to determine an initial alignment between image pairs. The mosaicking process resulted in a composite template which was more elaborate than the individual images. Mosaicking is an essential component of a fingerprint system since it elegantly consolidates the information available in several partial prints.

To account for non-linear distortions in fingerprints, an “average” deformation model was proposed. In this approach, a fingerprint impression (baseline) was compared with several other impressions of the same finger in order to determine the “relative” non-linear deformation present in it. The average deformation model was developed using thin-plate splines (TPS) and ridge curves were used to establish correspondences between image pairs. The estimated average deformation was utilized to pre-distort the minutiae points in the template image before matching it with the minutiae points in the query image. The use of an average deformation model resulted in a better alignment between the template and query minutiae points. An index of deformation was also defined for choosing the deformation model with the least variability from a set of template impressions corresponding to a finger.

Finally, the fingerprint evidence of a user was combined with the face and hand geometry traits to design a multibiometric system. A multibiometric system not only improves matching performance as was demonstrated in this thesis, but it also addresses the problem of non-universality and spoofing that are prevalent in unimodal systems. Evidence consolidation was done at the matching score level in which the matching scores generated by the various matchers were combined. The matching performance was further improved by employing user-specific weights and user-specific thresholds in the matching stage.

## 6.2 Future Research

Significant efforts are currently being undertaken to integrate biometrics into the fabric of society (e.g., National ID card, US-VISIT program, etc.). It is, therefore, imperative that researchers and practitioners systematically study the engineering aspects of biometric systems that would ensure their successful installation in real-world applications. The social and legal implications of biometric systems will also have to be separately studied and understood, before deploying these systems on a large scale.

We conclude this thesis by suggesting possible ways in which the research presented here may be expanded in order to build robust fingerprint (biometric) systems.

1. The hybrid matcher utilizes square tessellations to compare the texture (i.e., ridge flow) information across images. In view of the non-linear deformations present in fingerprint images, it would be instructive to ‘distort’ the square

cells in a non-linear fashion prior to the matching process. In fact, the average deformation model suggested in this thesis may be used to pre-distort the square tessellations in the template image.

2. The performance of the hybrid matcher is closely related to the accuracy of the image registration (alignment) process. In the procedure presented in this thesis, it is the minutiae points that are used to accomplish registration. However, for the sake of robustness, multiple alignment hypothesis may be used to determine the transformation parameters that best relate two fingerprint images. For example, the minutiae points, ridge curves, orientation field, ridge feature maps and singular points could all be used to generate an alignment hypothesis. These various hypothesis may then be consolidated to recover the true transformation parameters. The availability of multiple alignment hypothesis is especially useful when poor quality fingerprint images are involved.
3. The fingerprint mosaicking technique utilizes partial impressions of a fingerprint to construct a more elaborate one. Given several partial impressions of a finger, there is no systematic technique for determining which of these should be used for generating the composite template. Thus, a method to automatically select candidate impressions for mosaicking would improve the integrity of the composite template. The candidate selection process could also be used to perform automatic template selection and update in a biometric system.
4. The average deformation model for a fingerprint may be used to detect the presence of a fake finger on the sensor. The minutiae points on a fake fingertip

may not be distorted in the same way as those present on a live fingertip. (This is true, for example, when the material used to synthesize the fake fingerprint does not exhibit “skin-like” properties). Therefore, the system may request a user to present multiple impressions of a finger during verification and derive the relative non-linear distortion between the acquired impressions. These distortions can then be compared with the average deformation model for the finger in order to observe if they are similar. A lack of similarity might indicate the presence of a fake finger.

5. The effect of score normalization on the matching performance of a multibiometric system has to be studied. In this thesis, a simple min-max normalization technique is utilized to transform the matching scores of multiple modalities into a common domain (i.e., all matching scores are mapped in a linear fashion to the  $[0,100]$  range). However, other robust normalization techniques have to be examined in order to offset the effect of outliers on the matching performance.
6. Fusion at the matching score level is the most popular approach to multibiometrics due to the ease in accessing and consolidating the scores generated by multiple matchers. However, fusion at the feature extraction (representation) level is expected to be more effective due to the richer source of information available at this level. Therefore, it is important to study the possibility of fusing information at this level. Concatenating compatible feature sets would be one way to integrate information at this level. Integration at this level is not without its challenges. For example, it would be difficult to concatenate

two incompatible feature sets like the eigen-coefficients (for face) and minutiae points (for fingerprint).

7. While multibiometric systems typically integrate “typical” physiological (e.g., fingerprint, iris, hand) and behavioral (e.g., gait, signature, voice) traits for recognition , one could consider the use of atypical traits like the color of hair, color of eyes, height of individual, and gender of individual in conjunction with typical biometric traits to enhance recognition performance.

## APPENDICES

# Appendix A

## Hand Geometry-based Biometric System

In this appendix a biometric system that uses the geometry of a person's hand for recognition is described. A technique for computing the various features (invariant to the lighting conditions of the device, presence of noise and the color of the skin) is summarized. A prototype image acquisition system was developed to capture the profile of the hand. Experiments on a database containing 50 users is presented.

### A.1 Why Hand Geometry?

The suitability of a particular biometric to a specific application depends upon several factors [50]; among these factors, the user acceptability seems to be the most significant. For many access control applications, like immigration, border control and dormitory meal plan access, very distinctive biometrics, e.g., fingerprint and iris,

may not be acceptable for the sake of protecting an individual's privacy. In such situations, it is desirable that the given biometric indicator be only distinctive enough for verification but not for identification. As hand geometry information is not very distinctive, it is one of the biometrics of choice in applications like those mentioned above.

Hand geometry-based authentication is also very effective for various other reasons. Almost all of the working population have hands and exception processing for people with disabilities could be easily engineered [111]. Hand geometry measurements are easily collectible due to both the dexterity of the hand and due to a relatively simple method of sensing which does not impose undue requirements on the imaging optics. Note that good frictional skin is required by fingerprint imaging systems, and a special illumination setup is needed by iris or retina-based identification systems. Further, hand geometry is ideally suited for integration with other biometrics, in particular, fingerprints. For instance, an identification/verification system may use fingerprints for (infrequent) identification and use hand geometry for (frequent) verification. It is easy to conceptualize a sensing system which can simultaneously capture both fingerprints and hand geometry.

## A.2 Background

Hand Geometry, as the name suggests, refers to the geometric structure of the hand. This structure includes width of the fingers at various locations, width of the palm, thickness of the palm, length of the fingers, etc. Although these metrics do not vary

significantly across the population, they can however be used to verify the identity of an individual. Hand geometry measurement is non-intrusive and the verification involves a simple processing of the resulting features. Unlike palmprint verification methods [112, 113], this method does not involve extraction of detailed features of the hand (for example, wrinkles on the skin).

Hand geometry-based verification systems are not new and have been available since the early 1970s. However, there is not much open literature addressing the research issues underlying hand geometry-based identity authentication; much of the literature is in the form of patents [114, 115, 116] or application-oriented description. Sidlauskas [117] discusses a 3D hand profile identification apparatus that has been used for hand geometry recognition.

Authentication of identity of an individual based on a set of hand features is an important research problem. It is well known that the individual hand features themselves are not very descriptive; devising methods to combine these non-salient individual features to attain robust positive identification is a challenging pattern recognition problem in its own right.

### **A.3 Image Acquisition**

The image acquisition system which we have used comprises of a light source, a camera, a single mirror and a flat surface (with five pegs on it). The user places his right hand - palm facing downwards - on the flat surface of the device. The five pegs serve as control points for appropriate placement of the right hand of the user. The

device also has knobs to change the intensity of the light source and the focal length of the camera. The lone mirror projects the side-view of the user's hand onto the camera. The device is hooked to a PC with a GUI application which provides a live visual feedback of the top-view and the side-view of the hand (Figure A.1) and has the following functionality: (i) assists the user in correct positioning of the hand on the surface of the device; (ii) acquires images of the user's hand; (iii) displays images that were captured previously; (iv) extracts features from a given image; (v) registers the user in a database along with the extracted feature vector; (vi) checks whether a given image of the hand matches any of the entries in the database; (vii) updates a particular user's entry in the database by recomputing the feature vector. In the current prototype implementation, a  $640 \times 480$  8-bit grayscale image of the hand is captured.

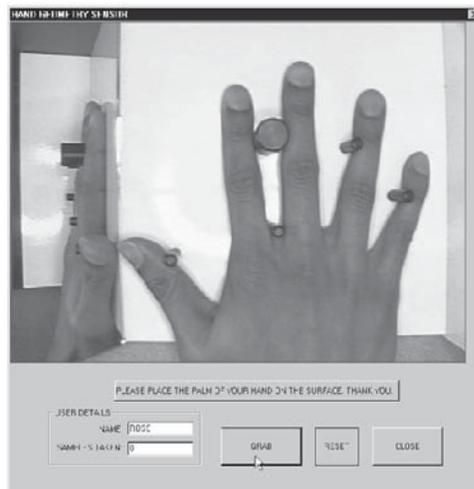


Figure A.1: Hand geometry sensing device.

### A.3.1 Enrollment Phase

This process involves one of the following two tasks: (i) add a new user to the database; (ii) update a current user's feature vector. During the enrollment phase, five images of the same hand are taken in succession; the user removes his hand completely from the device before every acquisition. These five images are then used to compute the feature vector of the given hand. Recomputing a feature vector simply involves averaging the individual feature values.

### A.3.2 Verification Phase

This process involves matching a given hand to a person previously enrolled in the system. Two snapshots of the hand are taken and the "average" feature vector is computed. The given feature vector is then compared with the feature vector stored in the database associated with the claimed identity. Let  $F = (f_1, f_2, \dots, f_d)$  represent the  $d$ -dimensional feature vector in the database associated with the claimed identity and  $Y = (y_1, y_2, \dots, y_d)$  be the feature vector of the hand whose identity has to be verified. The verification is positive if the distance between  $F$  and  $Y$  is less than a threshold value. Four distance metrics, absolute, weighted absolute, Euclidean, and

weighted Euclidean, corresponding to the following four equations were explored:

$$\sum_{j=1}^d |y_j - f_j| < \epsilon_a, \quad (\text{A.1})$$

$$\sum_{j=1}^d \frac{|y_j - f_j|}{\sigma_j} < \epsilon_{wa}, \quad (\text{A.2})$$

$$\sqrt{\sum_{j=1}^d (y_j - f_j)^2} < \epsilon_e, \text{ and} \quad (\text{A.3})$$

$$\sqrt{\sum_{j=1}^d \frac{(y_j - f_j)^2}{\sigma_j^2}} < \epsilon_{we}, \quad (\text{A.4})$$

where  $\sigma_j^2$  is the feature variance of the  $j$ th feature and  $\epsilon_a$ ,  $\epsilon_{wa}$ ,  $\epsilon_e$ , and  $\epsilon_{we}$  are threshold values for each respective distance metric.

## A.4 Feature Extraction

The hand geometry-based authentication system relies on geometric invariants of a human hand. Typical features include length and width of the fingers, aspect ratio of the palm or fingers, thickness of the hand, etc. [118]. To our knowledge, the existing commercial systems do not take advantage of any non-geometric attributes of the hand, e.g., color of the skin.

Figure A.2 shows the 16 axes along which the various features mentioned above have been measured. The five pegs on the image serve as control points and assist in choosing these axes. The hand is represented as a vector of the measurements selected above. Since the positions of the five pegs are fixed in the image, no attempt is made to remove these pegs in the acquired images.

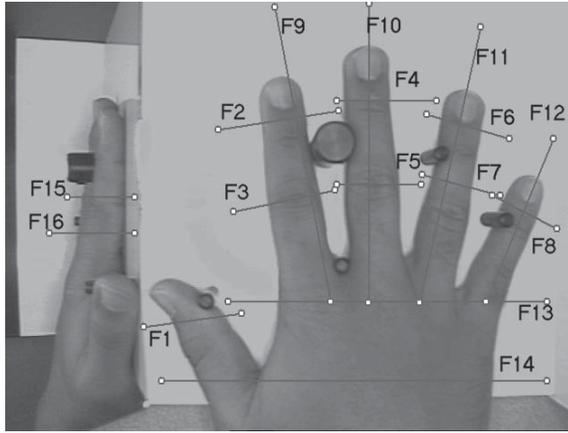
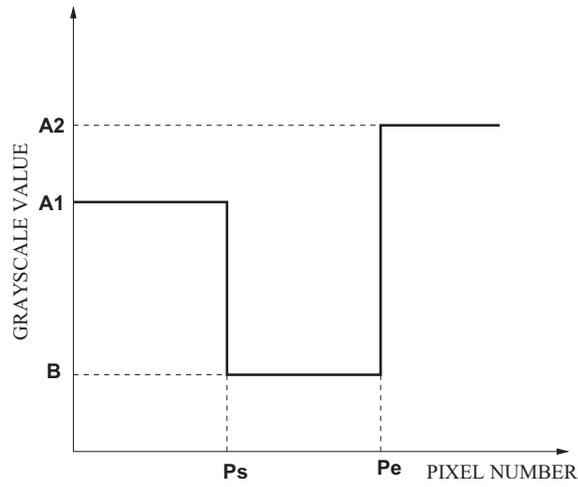


Figure A.2: The sixteen axes along which feature values are computed.

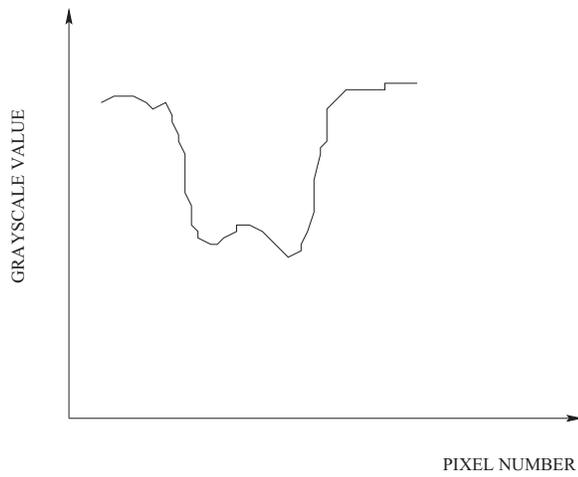
In order to offset the effects of background lighting, color of the skin, and noise, the following approach was devised to compute the various feature values. A sequence of pixels along a measurement axis will have an ideal gray scale profile as shown in Figure A.3(a). Here  $Len$  refers to the total number of pixels considered,  $P_s$  and  $P_e$  refer to the end points within which the object (viz., finger) to be measured is located, and  $A_1$ ,  $A_2$  and  $B$  are the gray scale values.

The actual gray scale profile tends to be spiky as shown in Figure A.3(b). Our first step is to model the above profile. Let the pixels along a measurement axis be numbered from 1 to  $Len$ . Let  $X = (x_1, x_2, \dots, x_{Len})$  be the gray values of the pixels along that axis. We make the following assumptions about the profile:

1. The observed profile (Figure A.3(b)) is obtained from the ideal profile (Figure A.3(a)) by the addition of Gaussian noise to each of the pixels in the latter. Thus, for example, the gray level of a pixel lying between  $P_s$  and  $P_e$  is assumed



(a) The ideal profile



(b) An observed profile

Figure A.3: The gray scale profile of pixels along a measurement axis.

to be drawn from the distribution

$$G(x/B, \sigma_B^2) = \frac{1}{\sqrt{2\pi\sigma_B^2}} \exp \left\{ -\frac{1}{2\sigma_B^2} (x - B)^2 \right\} \quad (\text{A.5})$$

where  $\sigma_B^2$  is the variance of  $x$  in the interval  $R$ ,  $P_s < R \leq P_e$ .

2. The gray level of an arbitrary pixel along a particular axis is *independent* of the gray level of other pixels in the line. This assumption holds good because of the absence of pronounced shadows in the acquired image.

Operating under these assumptions, we can write the joint distribution of all the pixel values along a particular axis as

$$P(X/\Theta) = \left[ \prod_{j=1}^{P_s} \frac{1}{\sqrt{2\pi\sigma_{A1}^2}} \exp \left\{ -\frac{1}{2\sigma_{A1}^2} (x_j - A1)^2 \right\} \right] \left[ \prod_{j=P_s+1}^{P_e} \frac{1}{\sqrt{2\pi\sigma_B^2}} \exp \left\{ -\frac{1}{2\sigma_B^2} (x_j - B)^2 \right\} \right] \left[ \prod_{j=P_e+1}^{Len} \frac{1}{\sqrt{2\pi\sigma_{A2}^2}} \exp \left\{ -\frac{1}{2\sigma_{A2}^2} (x_j - A2)^2 \right\} \right], \quad (\text{A.6})$$

where  $\Theta = (P_s, P_e, A1, A2, B, \sigma_{A1}^2, \sigma_{A2}^2, \sigma_B^2)$  and  $\sigma_{A1}^2$ ,  $\sigma_{A2}^2$  and  $\sigma_B^2$  are the variances of  $x$  in the three intervals  $[1, P_s]$ ,  $[P_s + 1, P_e]$  and  $[P_e + 1, Len]$ , respectively.

The goal now is to estimate  $P_s$  and  $P_e$  using the observed pixel values along the chosen axis. We use the Maximum Likelihood Estimate (MLE) method to estimate  $\Theta$ . By taking logarithm on both sides of Eq. (A.6) and simplifying, we obtain the

likelihood function:

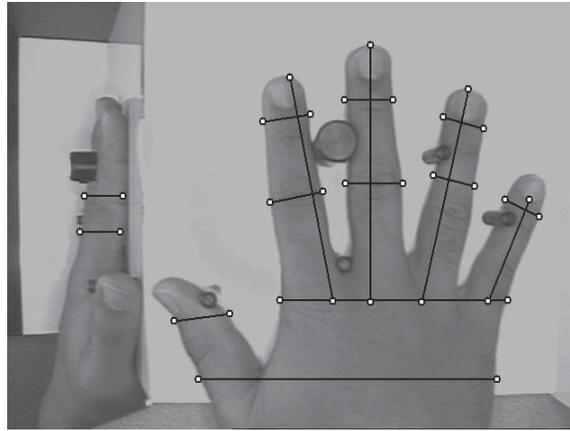
$$\begin{aligned}
L(\Theta) &= \frac{1}{\sigma_{A1}^2} \sum_1^{P_s} (x_j - A1)^2 + \frac{1}{\sigma_B^2} \sum_{P_s+1}^{P_e} (x_j - B)^2 \\
&+ \frac{1}{\sigma_{A2}^2} \sum_{P_e+1}^{Len} (x_j - A2)^2 + P_s \log \sigma_{A1}^2 \\
&+ (P_e - P_s) \log \sigma_B^2 + (Len - P_e) \log \sigma_{A2}^2
\end{aligned} \tag{A.7}$$

The parameters can now be estimated iteratively; the parameter estimates at the  $(k+1)^{st}$  stage, given the observation  $X = (x_1, x_2, \dots, x_{Len})$ , are given below.

$$\begin{aligned}
\widehat{P}_s^{(k+1)} &= \arg \min_{P_s} L \left( \begin{array}{c} P_s, \widehat{P}_e^{(k)}, \widehat{A1}^{(k)}, \widehat{A2}^{(k)}, \\ \widehat{B}^{(k)}, \widehat{\sigma}_{A1}^2^{(k)}, \widehat{\sigma}_{A2}^2^{(k)}, \widehat{\sigma}_B^2^{(k)} \end{array} \right) \\
\widehat{P}_e^{(k+1)} &= \arg \min_{P_e} L \left( \begin{array}{c} \widehat{P}_s^{(k+1)}, P_e, \widehat{A1}^{(k)}, \widehat{A2}^{(k)}, \\ \widehat{B}^{(k)}, \widehat{\sigma}_{A1}^2^{(k)}, \widehat{\sigma}_{A2}^2^{(k)}, \widehat{\sigma}_B^2^{(k)} \end{array} \right) \\
\widehat{B}^{(k+1)} &= \frac{\sum_{\widehat{P}_s^{(k+1)}+1}^{\widehat{P}_e^{(k+1)}} x_j}{\widehat{P}_e^{(k+1)} - \widehat{P}_s^{(k+1)}} \\
\widehat{\sigma}_B^2^{(k+1)} &= \frac{\sum_{\widehat{P}_s^{(k+1)}+1}^{\widehat{P}_e^{(k+1)}} x_j^2}{\widehat{P}_e^{(k+1)} - \widehat{P}_s^{(k+1)}} - \left\{ \widehat{B}^{(k+1)} \right\}^2 \\
\widehat{A1}^{(k+1)} &= \frac{\sum_1^{\widehat{P}_e^{(k+1)}} x_j}{\widehat{P}_s^{(k+1)}} \\
\widehat{\sigma}_{A1}^2^{(k+1)} &= \frac{\sum_1^{\widehat{P}_e^{(k+1)}} x_j^2}{\widehat{P}_s^{(k+1)}} - \left\{ \widehat{A1}^{(k+1)} \right\}^2 \\
\widehat{A2}^{(k+1)} &= \frac{\sum_{\widehat{P}_e^{(k+1)}+1}^{Len} x_j}{Len - \widehat{P}_e^{(k+1)}} \\
\widehat{\sigma}_{A2}^2^{(k+1)} &= \frac{\sum_{\widehat{P}_e^{(k+1)}+1}^{Len} x_j^2}{Len - \widehat{P}_e^{(k+1)}} - \left\{ \widehat{A2}^{(k+1)} \right\}^2
\end{aligned} \tag{A.8}$$

The initial estimates of  $A1$ ,  $\sigma_{A1}^2$ ,  $A2$ ,  $\sigma_{A2}^2$ ,  $B$  and  $\sigma_B^2$  are obtained as follows: (i)  $A1$  and  $\sigma_{A1}^2$  are estimated using the gray values of the first  $N_{A1}$  pixels along the axis; (ii)  $A2$  and  $\sigma_{A2}^2$  are estimated using the gray values of the pixels from  $(Len - N_{A2})$  to  $Len$ ; (iii)  $B$  and  $\sigma_B^2$  are estimated using the gray values of the pixels between  $(Len/2 - N_B)$  and  $(Len/2 + N_B)$ . The values of  $N_{A1}$ ,  $N_{A2}$  and  $N_B$  are fixed for the system;  $N_{A1} = 5$ ,  $N_{A2} = 4$  and  $N_B = 5$ . The initial values of  $P_s$  and  $P_e$  are set to  $Len/2 - 10$  and  $Len/2 + 10$ , respectively.

Figure A.4 shows a hand image along with the positions of detected points ( $P_s$  and  $P_e$ ) along each of the 16 axes and the corresponding feature vector.



(a) Estimates of  $P_s$  and  $P_e$  along the 16 axes

(akasapuv 65 53 59 52 62 47 47 45 255 333 253 287 243 149 34 35)

(b) The corresponding database entry

Figure A.4: Computation of the feature vector.

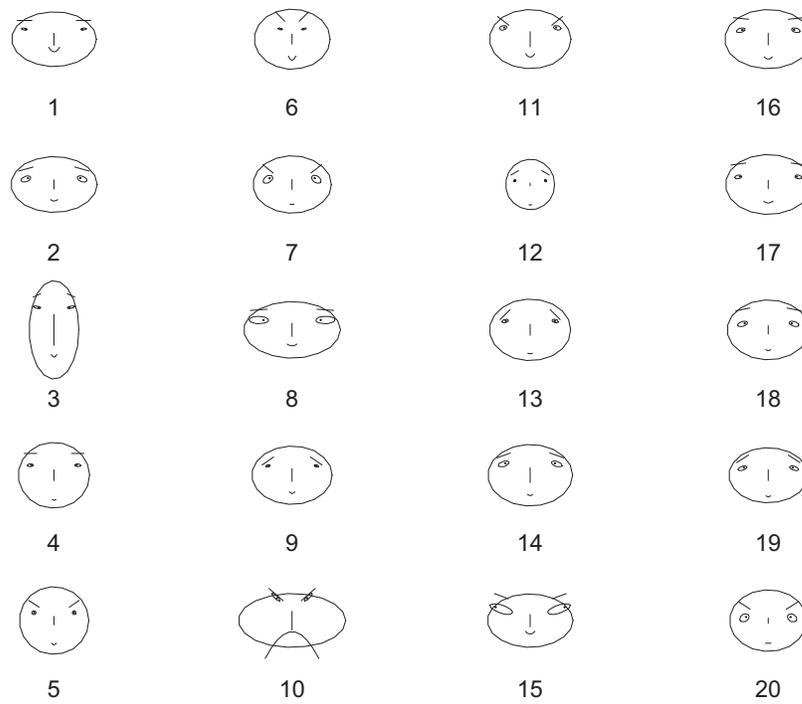


Figure A.5: Chernoff Faces representing the average feature vectors of 20 different hands.

## A.5 Experimental Results

The hand geometry authentication system was trained and tested using a database of 50 users. Ten images of each user's hand were captured over two sessions; in each session the background lighting of the acquisition device was changed. Thus a total of 500 images were made available. Out of 500 images, only 360 were used for testing our hand geometry system. The remaining 140 images were discarded due to incorrect placement of the hand by the user (see for example, Figure A.6). Thus, user adaptation of this biometric is necessary. Two images of each user's hand were randomly selected to compute the feature vector which is stored in the database along with the user's name. Figure A.5 shows the chernoff faces [4] representing the average feature vector of 20 of the users. 15 hand features have been mapped to the attributes of the cartoon face as follows: 1-area of face; 2-shape of face; 3-length of nose; 4-location of mouth; 5-curve of smile; 6-width of mouth; 7, 8, 9, 10 and 11-location, separation, angle, shape and width of eyes; 12-location and width of pupil; 13, 14 and 15 -location, angle and width of eyebrow. The difference between any two hand geometries as reflected in these cartoon faces appears to be significant.

Eqs. (A.1)-(4) are used for verifying whether the feature vector of a hand matches with the feature vector stored in the database. In order to study the effectiveness of various distance metrics, the genuine and impostor distributions are plotted for matching scores obtained using each distance metric and an ROC curve generated from each pair of distributions. A genuine matching score is obtained by comparing two feature vectors from the same hand while an impostor matching score is obtained

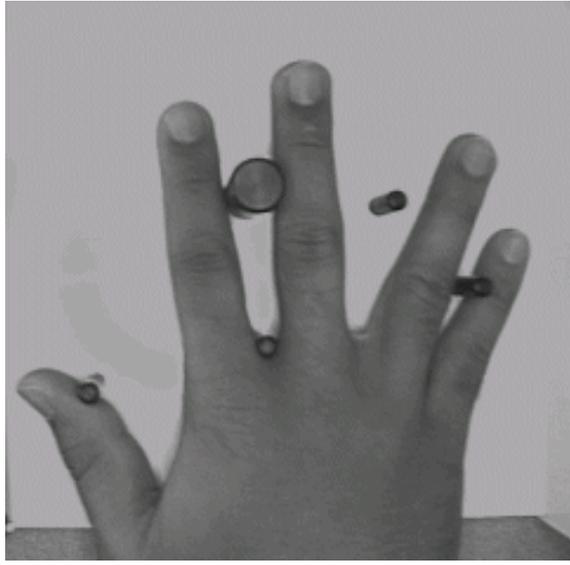


Figure A.6: Incorrect placement of hand.

by comparing the feature vectors of two different hands. Let us define the hit rate to be the percentage of time the system matches a hand to the right entry in the database, and the false acceptance rate to be the percentage of time the system matches a hand to an incorrect entry in the database for a given threshold. The ROC curve that plots the hit rate against the false acceptance rate is then computed using the leave-one-out method. A feature vector in the database is matched against those feature vectors representing a different user. The minimum of these distances is taken as an impostor matching score. If the matching score falls below the chosen threshold, it is considered to be a false acceptance by the system. This process is repeated for all the users in the database. A genuine matching score is obtained by matching a feature vector against those feature vectors that belong to the same user and then taking the minimum of all such distances. If the matching score falls below the chosen threshold then it is considered to be a hit. The ROC curve shown in

Figure A.7 depicts the performance of the system for the weighted Euclidean distance (Eq. A.4) which gave the best result. The system performance could be significantly improved by (i) having habituated users; (ii) better registration of hand geometry measurements; and (iii) using higher level features (like color of the skin, wrinkles and folds on the skin etc.). Among these factors, registration appears to be the most critical. Even though the pegs are used for registration in our system, the registration accomplished by the pegs is not sufficiently accurate.

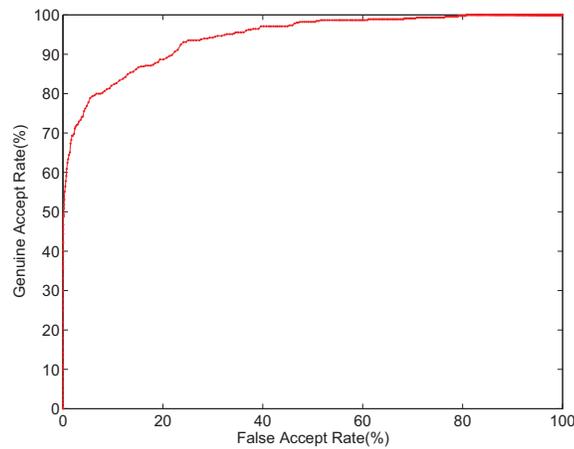


Figure A.7: The ROC curve depicting the matching performance of the hand geometry system.

## BIBLIOGRAPHY

# Bibliography

- [1] A. K. Jain, L. Hong, and R. Bolle, “On-line fingerprint verification,” *IEEE Transactions on PAMI*, vol. 19, pp. 302–314, April 1997.
- [2] L. Hong, “Automatic personal identification using fingerprints,” PhD Thesis, Michigan State University, 1998.
- [3] R.-L. Hsu, “Face detection and modeling for recognition,” PhD Thesis, Michigan State University, 2002.
- [4] H. Chernoff, “The use of faces to represent points in k-dimensional space graphically,” *Journal of the American Statistical Association*, vol. 68, pp. 361–368, 1973.
- [5] A. K. Jain, R. Bolle, and S. Pankanti, eds., *Biometrics: Personal Identification in Networked Society*. Kluwer Academic Publishers, 1999.
- [6] J. L. Wayman, “Fundamentals of biometric authentication technologies,” *International Journal of Image and Graphics*, vol. 1, no. 1, pp. 93–113, 2001.
- [7] L. O’Gorman, “Seven issues with human authentication technologies,” in *Proc. of Workshop on Automatic Identification Advanced Technologies (AutoID)*, (Tarrytown, New York), pp. 185–186, Mar 2002.
- [8] R. Yasin, “Password pain relief,” *Information Security Magazine*, April 2002. Available at <http://www.infosecuritymag.com/2002/apr/cover.shtml>.
- [9] J. Daugman, “Statistical demands of identification versus verification.” Available at <http://www.cl.cam.ac.uk/users/jgd1000/veri/veri.html>.
- [10] “Schiphol backs eye scan security.” CNN World News, March 27 2002. Available at <http://www.cnn.com/2002/WORLD/europe/03/27/schiphol.security/>.
- [11] J. Daugman, “Recognizing persons by their iris patterns,” in *Biometrics: Personal Identification in a Networked Society* (A. K. Jain, R. Bolle, and S. Pankanti, eds.), pp. 103–121, Kluwer Academic Publishers, 1999.
- [12] J. Berry and D. A. Stoney, “The history and development of fingerprinting,” in *Advances in Fingerprint Technology* (H. C. Lee and R. Gaensslen, eds.), pp. 1–40, Florida: CRC Press, 2nd ed., 2001.

- [13] Federal Bureau of Investigation, *The Science of Fingerprints: Classification and Uses*. Washington, D.C., 1984. U.S. Government Printing Office.
- [14] S. Pankanti, S. Prabhakar, and A. K. Jain, “On the individuality of fingerprints,” *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 24, no. 8, pp. 1010–1025, 2002.
- [15] A. K. Jain, S. Prabhakar, and S. Pankanti, “On the similarity of identical twin fingerprints,” *Pattern Recognition*, vol. 35, no. 8, pp. 2653–2663, 2002.
- [16] K. Karu and A. K. Jain, “Fingerprint classification,” *Pattern Recognition*, vol. 29, no. 3, pp. 389–404, 1996.
- [17] A. Senior, “A combination fingerprint classifier,” *IEEE Transactions on PAMI*, vol. 23, pp. 1165–1174, Oct 2001.
- [18] Z. M. Kovács-Vajna, “A fingerprint verification system based on triangular matching and dynamic time warping,” *IEEE Transactions on PAMI*, vol. 22, pp. 1266–1276, Nov 2000.
- [19] F. Pernus, S. Kovacic, and L. Gyergyek, “Minutiae-based fingerprint recognition,” in *Proceedings of the Fifth International Conference on Pattern Recognition*, pp. 1380–1382, 1980.
- [20] B. Mehtre and M. Murthy, “A minutiae based fingerprint identification system,” in *Proceedings of the Second International Conference on Advances in Pattern Recognition and Digital Techniques*, (Calcutta, India), January 1986.
- [21] A. Sibbald, “Method and apparatus for fingerprint characterization and recognition using auto-correlation pattern,” US Patent 5633947, 1994.
- [22] A. Stoianaov, C. Soutar, and A. Graham, “High-speed fingerprint verification using an optical correlator,” in *Proceedings SPIE*, vol. 3386, pp. 242–252, 1998.
- [23] D. Roberge, C. Soutar, and B. Vijaya Kumar, “High-speed fingerprint verification using an optical correlator,” in *Proceedings SPIE*, vol. 3386, pp. 123–133, 1998.
- [24] A. M. Bazen, G. T. B. Verwaaijen, S. H. Gerez, L. P. J. Veelenturf, and B. J. van der Zwaag, “A correlation-based fingerprint verification system,” in *Proceedings of the ProRISC2000 Workshop on Circuits, Systems and Signal Processing*, (Veldhoven, Netherlands), Nov 2000.
- [25] L. O’Gorman, “Fingerprint verification,” in *Biometrics: Personal Identification in a Networked Society* (A. K. Jain, R. Bolle, and S. Pankanti, eds.), pp. 43–64, Kluwer Academic Publishers, 1999.
- [26] M. Trauring, “Automatic comparison of finger-ridge patterns,” *Nature*, pp. 938–940, 1963.

- [27] C. Shelman, "Machine classification of fingerprints," in *Proceedings of the First National Symposium on Law Enforcement Science and Technology*, pp. 467–477, 1967.
- [28] A. Grasselli, "On the automatic classification of fingerprints - some considerations on the linguistic interpretation of pictures," in *Methodologies of Pattern Recognition* (S. Watanabe, ed.), pp. 253–273, Academic Press, 1969.
- [29] A. K. Jain, S. Prabhakar, L. Hong, and S. Pankanti, "Filterbank-based fingerprint matching," *IEEE Transactions on Image Processing*, vol. 9, pp. 846–859, May 2000.
- [30] A. K. Jain, A. Ross, and S. Prabhakar, "Fingerprint matching using minutiae and texture features," in *Proc. International Conference on Image Processing (ICIP)*, (Thessaloniki, Greece), pp. 282–285, Oct 2001.
- [31] J. Daugman, "Uncertainty relation for resolution in space, spatial frequency, and orientation optimized by two-dimensional visual cortical filters," *Journal of the Optical Society of America*, vol. 2, pp. 1160–1169, 1985.
- [32] S. Klein and B. Beutner, "Minimizing and maximizing the joint space-spatial frequency uncertainty of gabor-like functions: Comment," *Journal of the Optical Society of America*, vol. 9, no. 2, pp. 337–340, 1992.
- [33] L. Hong, Y. Wan, and A. K. Jain, "Fingerprint image enhancement: Algorithms and performance evaluation," *IEEE Transactions on PAMI*, vol. 20, pp. 777–789, Aug 1998.
- [34] D. Sherlock, D. M. Monro, and K. Millard, "Fingerprint enhancement by directional fourier filtering," *IEE Proceedings on Vision, Image and Signal Processing*, vol. 141, no. 2, pp. 87–94, 1994.
- [35] R. L. Graham, "An efficient algorithm for determining the convex hull of a finite planar set," *Information Processing Letters*, vol. 1, no. 4, pp. 132–133, 1972.
- [36] J. Kittler, M. Hatef, R. P. Duin, and J. G. Matas, "On combining classifiers," *IEEE Transactions on PAMI*, vol. 20, pp. 226–239, Mar 1998.
- [37] R. Germain, A. Califano, and S. Colville, "Fingerprint matching using transformation parameters," *IEEE Computational Science and Engineering*, vol. 4, no. 4, pp. 42–49, 1997.
- [38] B. Bhanu and X. Tan, "A triplet based approach for indexing of fingerprint database for identification," in *Proc. Third International Conference on Audio and Video-based Biometric Person Authentication (AVBPA)*, (Halmstad, Sweden), pp. 205–210, Jun 2001.

- [39] R. Cappelli, D. Maio, and D. Maltoni, "Fingerprint classification by directional image partitioning," *IEEE Transactions on PAMI*, vol. 21, no. 5, pp. 402–421, 1999.
- [40] A. Ross, A. K. Jain, and J. Reisman, "A hybrid fingerprint matcher," in *Proc. of the International Conference on Pattern Recognition (ICPR)*, (Quebec City, Canada), Aug 2002.
- [41] A. Ross, A. K. Jain, and J. Reisman, "A hybrid fingerprint matcher," *Pattern Recognition*, vol. 36, pp. 1661–1673, Jul 2003.
- [42] N. K. Ratha, J. H. Connell, and R. M. Bolle, "Image mosaicing for rolled fingerprint construction," in *Proc. of 14th International Conference on Pattern Recognition*, vol. 2, pp. 1651–1653, 1998.
- [43] P. J. Besl and N. D. McKay, "A method for registration of 3-D shapes," *IEEE Transactions on PAMI*, vol. 14, pp. 239–256, February 1992.
- [44] L. G. Brown, "A survey of image registration techniques," *ACM Computing Surveys*, vol. 24, no. 4, pp. 325–376, 1992.
- [45] B. Vemuri and J. Aggarwal, "3-D model construction from multiple views using range and intensity data," in *Proc. of IEEE Conference on Computer Vision and Pattern Recognition*, pp. 435–437, 1986.
- [46] P. J. Besl, "The free-form surface matching problem," in *Machine Vision for Three-Dimensional Scenes* (H. Freeman, ed.), pp. 25–71, New York: Academic Press, 1990.
- [47] J. B. A. Maintz and M. A. Viergever, "A survey of medical image registration," *Medical Image Analysis*, vol. 4, no. 1, pp. 1–36, 1998.
- [48] T. S. Huang, D. B. Goldgof, and H. Lee, "Feature extraction and terrain matching," in *Proc. of the IEEE conference on Computer Vision and Pattern Recognition*, pp. 899–904, June 1988.
- [49] Y. Chen and G. Medioni, "Object modelling by registration of multiple range images," *Image and Vision Computing*, vol. 10, pp. 145–155, April 1992.
- [50] A. K. Jain, L. Hong, S. Pankanti, and R. Bolle, "An identity authentication system using fingerprints," *Proceedings of the IEEE*, vol. 85, no. 9, pp. 1365–1388, 1997.
- [51] B. K. P. Horn, "Closed-form solution of absolute orientation using unit quaternions," *Journal of the Optical Society of America*, vol. 4, pp. 629–642, April 1987.
- [52] A. K. Jain and A. Ross, "Fingerprint mosaicking," in *Proc. of IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP)*, (Orlando, Florida), May 2002.

- [53] D. Maio and D. Maltoni, “Direct gray-scale minutiae detection in fingerprints,” *IEEE Transactions on PAMI*, vol. 19, pp. 27–40, Jan 1997.
- [54] N. K. Ratha and R. M. Bolle, “Effect of controlled acquisition on fingerprint matching,” in *Proceedings of the International Conference on Pattern Recognition*, vol. 2, (Brisbane, Australia), pp. 1659–1661, 1998.
- [55] C. Dorai, N. Ratha, and R. Bolle, “Detecting dynamic behavior in compressed fingerprint videos: Distortion,” in *Proceedings of Computer Vision and Pattern Recognition*, pp. 320–326, Jun 2000.
- [56] L. R. Thebaud, “Systems and methods with identity verification by comparison and interpretation of skin patterns such as fingerprints,” US Patent 5,909,501, 1999.
- [57] A. M. Bazen and S. Gerez, “Fingerprint matching by thin-plate spline modelling of elastic deformations,” *Pattern Recognition*, vol. 36, pp. 1859–1867, August 2003.
- [58] A. Senior and R. Bolle, “Improved fingerprint matching by distortion removal,” *IEICE Transactions on Information and Systems*, vol. E84-D, pp. 825–831, Jul 2001.
- [59] C. Watson, P. Grother, and D. Cassasent, “Distortion-tolerant filter for elastic-distorted fingerprint matching,” in *Proceedings of SPIE Optical Pattern Recognition*, pp. 166–174, 2000.
- [60] R. Cappelli, D. Maio, and D. Maltoni, “Modelling plastic distortion in fingerprint images,” in *Proceedings of the Second International Conference on Advances in Pattern Recognition (ICAPR)*, (Rio de Janeiro), Mar 2001.
- [61] R. Cappelli, R. Erol, D. Maio, and D. Maltoni, “Synthetic fingerprint-image generation,” in *Proceedings of the 15th International Conference on Pattern Recognition (ICPR)*, vol. 3, (Barcelona, Spain), Sep 2000.
- [62] D. J. Burr, “A dynamic model for image registration,” *Computer Graphics and Image Processing*, vol. 15, pp. 102–112, 1981.
- [63] L. Younes, “Optimal matching between shapes via elastic deformations,” *Image and Vision Computing*, vol. 17, pp. 381–389, 1999.
- [64] J. L. Barron, D. J. Fleet, and S. S. Beauchemin, “Performance of optical flow techniques,” *International Journal Computer Vision*, vol. 12, pp. 43–77, 1994.
- [65] G. E. Christensen, R. D. Rabbitt, and M. I. Miller, “Deformable templates using large deformation kinetics,” *IEEE Transactions on Image Processing*, vol. 5, pp. 1435–1447, 1996.

- [66] S. C. Joshi and M. I. Miller, “Landmark matching via large deformation diffeomorphisms,” *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 9, pp. 1357–1370, 2000.
- [67] Y. Amit, U. Grenander, and M. Piccioni, “Structural image restoration through deformable templates,” *Journal of American Statistical Association*, vol. 86, pp. 376–387, 1991.
- [68] J. M. Cartensen, “An active lattice model in a Bayesian framework,” *Computer Vision and Image Understanding*, vol. 63, no. 2, pp. 380–387, 1996.
- [69] C. A. Glasbey and K. V. Mardia, “A penalized likelihood approach to image warping,” *Journal of the Royal Statistical Society. Series B. Statistical Methodology*, vol. 63, no. 3, pp. 465–514, 2001.
- [70] F. L. Bookstein, “Principal warps: thin-plate splines and the decomposition of deformations,” *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 11, pp. 567–585, 1989.
- [71] K. V. Mardia and T. J. Hainsworth, “Image warping and Bayesian reconstruction with gray-level templates,” in *Statistics and Images: Volume 1* (K. V. Mardia and G. K. Kanji, eds.), pp. 257–280, Carfax, Oxford, 1993.
- [72] K. V. Mardia, T. J. Hainsworth, and J. F. Haddon, “Deformable templates in image sequences,” *Proceedings of the International Conference on Pattern Recognition (ICPR)*, pp. 132–135, 1992.
- [73] A. Almansa and L. Cohen, “Fingerprint image matching by minimization of a thin-plate energy using a two-step algorithm with auxiliary variables,” *IEEE Workshop on Appl. of Comp. Vision (WACV '00)*, pp. 35–40, December 2000.
- [74] I. L. Dryden and K. V. Mardia, *Statistical Shape Analysis*. John Wiley and Sons, 1998.
- [75] D. Maio, D. Maltoni, R. Cappelli, J. L. Wayman, and A. K. Jain, “FVC2002: Fingerprint verification competition,” in *Proceedings of the International Conference on Pattern Recognition (ICPR)*, (Quebec City, Canada), pp. 744–747, August 2002.
- [76] W. Bicz, Z. Gumieny, D. Kosz, and M. Pluta, “Ultrasonic setup for fingerprint patterns detection and evaluation,” *Acoustical Imaging*, vol. 22, 1996.
- [77] E. den Os, H. Jongbloed, A. Stijssiger, and L. Boves, “Speaker verification as a user-friendly access for the visually impaired,” in *Proceedings of the European Conference on Speech Technology*, (Budapest), pp. 1263–1266, 1999.
- [78] D. A. Black, “Forgery above a genuine signature,” *Journal of Criminal Law, Criminology, and Police Science*, vol. 50, pp. 585–590, 1962.

- [79] W. Harrison, *Suspect Documents, Their Scientific Examination*. Nelson-Hall Publishers, 1981.
- [80] A. Eriksson and P. Wretling, “How flexible is the human voice? A case study of mimicry,” in *Proceedings of the European Conference on Speech Technology*, (Rhodes), pp. 1043–1046, 1997.
- [81] T. Matsumoto, H. Matsumoto, K. Yamada, and S. Hoshino, “Impact of artificial gummy fingers on fingerprint systems,” in *Proceedings SPIE*, vol. 4677, (San Jose, USA), pp. 275–289, Feb 2002.
- [82] P. J. Philips, P. Grother, R. J. Micheals, D. M. Blackburn, E. Tabassi, and M. Bone, “Frvt 2002: Overview and summary,” Technical Report available at <http://www.frvt.org/frvt2002/documents.htm>, 2003.
- [83] A. Martin and M. Przybocki, “The NIST 1999 speaker recognition evaluation - an overview,” *Digital Signal Processing*, vol. 10, pp. 1–18, 2000.
- [84] L. Hong, A. K. Jain, and S. Pankanti, “Can multibiometrics improve performance?,” in *Proceedings AutoID’99*, (Summit(NJ), USA), pp. 59–64, Oct 1999.
- [85] L. I. Kuncheva, C. J. Whitaker, C. A. Shipp, and R. P. W. Duin, “Is independence good for combining classifiers?,” in *Proc. of the International Conference on Pattern Recognition (ICPR)*, vol. 2, (Barcelona, Spain), pp. 168–171, 2001.
- [86] L. Hong and A. K. Jain, “Integrating faces and fingerprints for personal identification,” *IEEE Transactions on PAMI*, vol. 20, pp. 1295–1307, Dec 1998.
- [87] R. Brunelli and D. Falavigna, “Person identification using multiple cues,” *IEEE Transactions on PAMI*, vol. 12, pp. 955–966, Oct 1995.
- [88] E. Bigun, J. Bigun, B. Duc, and S. Fischer, “Expert conciliation for multimodal person authentication systems using Bayesian Statistics,” in *First International Conference on AVBPA*, (Crans-Montana, Switzerland), pp. 291–300, March 1997.
- [89] R. W. Frischholz and U. Dieckmann, “Bioid: A multimodal biometric identification system,” *IEEE Computer*, vol. 33, no. 2, pp. 64–68, 2000.
- [90] T. K. Ho, J. J. Hull, and S. N. Srihari, “Decision combination in multiple classifier systems,” *IEEE Transactions on PAMI*, vol. 16, pp. 66–75, January 1994.
- [91] S. Prabhakar and A. K. Jain, “Decision-level fusion in fingerprint verification,” *Pattern Recognition*, vol. 35, no. 4, pp. 861–874, 2002.
- [92] U. Dieckmann, P. Plankensteiner, and T. Wagner, “Sesam: A biometric person identification system using sensor fusion,” *Pattern Recognition Letters*, vol. 18, no. 9, pp. 827–833, 1997.

- [93] P. Verlinde and G. Cholet, “Comparing decision fusion paradigms using k-NN based classifiers, decision trees and logistic regression in a multi-modal identity verification application,” in *Second International Conference on AVBPA*, (Washington D.C., USA), pp. 188–193, March 1999.
- [94] S. Ben-Yacoub, Y. Abdeljaoued, and E. Mayoraz, “Fusion of face and speech data for person identity verification,” Research Paper IDIAP-RR 99-03, IDIAP, CP 592, 1920 Martigny, Switzerland, Jan 1999.
- [95] A. K. Jain, S. Prabhakar, and S. Chen, “Combining multiple matchers for a high security fingerprint verification system,” *Pattern Recognition Letters*, vol. 20, pp. 1371–1379, 1999.
- [96] Y. Zuev and S. Ivanon, “The voting as a way to increase the decision reliability,” in *Foundations of Information/Decision Fusion with Applications to Engineering Problems*, (Washington D.C., USA), pp. 206–210, August 1996.
- [97] R. Cappelli, D. Maio, and D. Maltoni, “Combining fingerprint classifiers,” in *First International Workshop on Multiple Classifier Systems*, pp. 351–361, June 2000.
- [98] A. K. Jain, S. Prabhakar, and L. Hong, “A multichannel approach to fingerprint classification,” *IEEE Transactions on PAMI*, vol. 21, no. 4, pp. 348–359, 1999.
- [99] A. K. Jain, L. Hong, and Y. Kulkarni, “A multimodal biometric system using fingerprint, face and speech,” in *Second International Conference on AVBPA*, (Washington D.C., USA), pp. 182–187, March 1999.
- [100] F. Hampel, P. Rousseeuw, E. Ronchetti, and W. Stahel, *Robust Statistics: The Approach Based on Influence Functions*. John Wiley & Sons, 1986.
- [101] A. K. Jain, S. Prabhakar, and A. Ross, “Fingerprint matching: Data acquisition and performance evaluation,” Technical Report MSU-TR:99-14, Michigan State University, 1999.
- [102] H. Rowley, S. Baluja, and T. Kanade, “Neural network-based face detection,” *IEEE Transactions on PAMI*, vol. 20, pp. 23–38, Jan 1998.
- [103] G. Burel and C. Carel, “Detection and localization of faces on digital images,” *Pattern Recognition Letters*, vol. 15, pp. 963–967, Oct 1994.
- [104] G. Yang and T. Huang, “Human face detection in a complex background,” *Pattern Recognition*, vol. 27, no. 1, pp. 53–63, 1994.
- [105] M. Kirby and L. Sirovich, “Application of the Karhunen-Loeve procedure for the characterization of human faces,” *IEEE Transactions on PAMI*, vol. 12, no. 1, pp. 103–108, 1990.

- [106] M. Turk and A. Pentland, "Eigenfaces for recognition," *Journal of Cognitive Neuroscience*, vol. 3, no. 1, pp. 71–86, 1991.
- [107] R. J. Quinlan, *C4.5: Programs for Machine Learning*. The Morgan Kaufmann Series in Machine Learning, Morgan Kaufmann Publishers, Oct 1992.
- [108] A. Ross, A. K. Jain, and J. Qian, "Information fusion in biometrics," in *Proceedings of the Third International Conference on Audio- and Video-based Person Authentication (AVBPA)*, (Halmstad, Sweden), pp. 354–359, Jun 2001.
- [109] A. K. Jain and A. Ross, "Learning user-specific parameters in a multibiometric system," in *Proc. of the International Conference on Image Processing (ICIP)*, (Rochester, USA), pp. 57–60, Sep 2002.
- [110] A. Ross and A. K. Jain, "Information fusion in biometrics," *Pattern Recognition Letters*, vol. 24, pp. 2115–2125, Sep 2003.
- [111] R. Zunkel, "Hand geometry based authentication," in *Biometrics: Personal Identification in Networked Society* (A. K. Jain, R. Bolle, and S. Pankanti, eds.), ch. 4, pp. 87–102, Kluwer Academic Publishers, 1998.
- [112] J. Young and H. Hammon, "Automatic palmprint verification study," Final Technical Report RADC-TR-81-161, Rome Air Development Center, 1981.
- [113] H. Yoshikawa and S. Ikebata, "A microcomputer-based personal identification system," in *International Conference on Industrial Electronics, Control, and Instrumentation*, 1984.
- [114] R. Miller, "Finger dimension comparison identification system," US Patent 3576538, 1971.
- [115] R. Ernst, "Hand id system," US Patent 3576537, 1971.
- [116] I. Jacoby, A. Giordano, and W. Fioretti, "Personnel identification apparatus," US Patent 3648240, 1972.
- [117] D. Sidlauskas, "3d hand profile identification apparatus," US Patent 4736203, 1988.
- [118] "Hasis - a hand shape identification system."  
<http://bias.csr.unibo.it/research/biolab/hand.html>.
- [119] A. K. Jain, A. Ross, and S. Pankanti, "A prototype hand geometry-based verification system," in *Second International Conference on Audio and Video-based Biometric Person Authentication (AVBPA)*, (Washington, D.C., USA), pp. 166–171, March 1999.