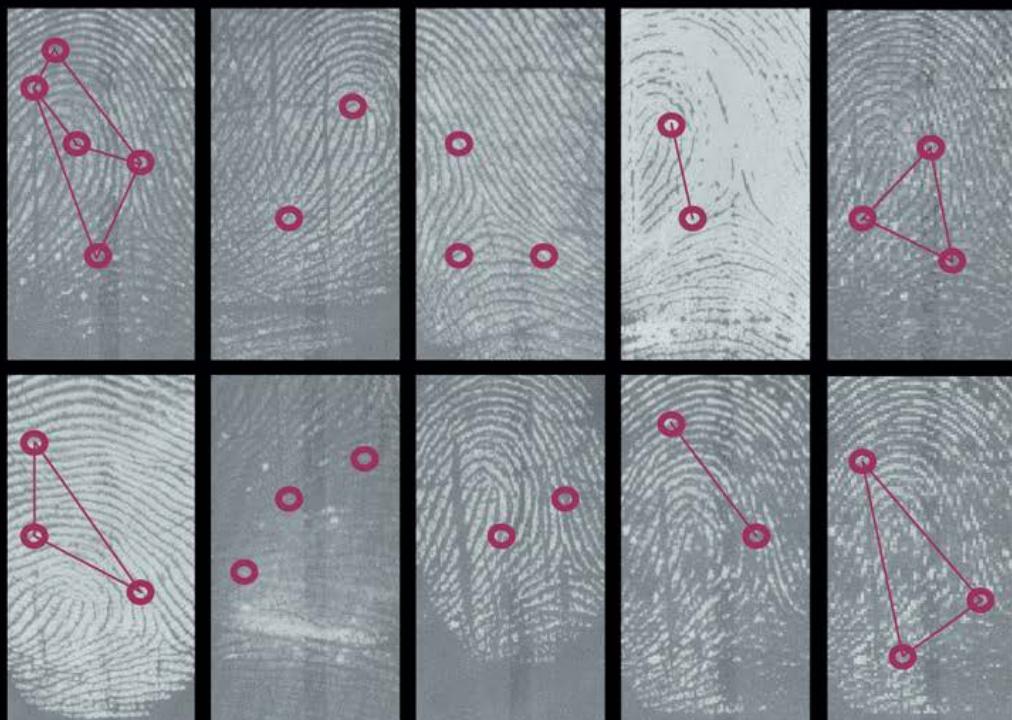


Computational Algorithms for Fingerprint Recognition

Bir Bhanu Xuejun Tan



- Feature Extraction
- Matching
- Performance Prediction

- Indexing
- Classification
- Identification

Kluwer Academic Publishers

COMPUTATIONAL ALGORITHMS FOR FINGERPRINT RECOGNITION

Kluwer International Series on Biometrics

Professor David D. Zhang
Consulting editor

*Department of Computer Science
Hong Kong Polytechnic University
Hung Hom, Kowloon, Hong Kong*

email: csdzhang@comp.polyu.edu.hk

Additional information about this series can be obtained from our website:
<http://www.wkap.nl>

COMPUTATIONAL ALGORITHMS FOR FINGERPRINT RECOGNITION

by

**Bir Bhanu
Xuejun Tan**
*University of California at Riverside
U.S.A.*



SPRINGER SCIENCE+BUSINESS MEDIA, LLC

Library of Congress Cataloging-in-Publication

COMPUTATIONAL ALGORITHMS FOR FINGERPRINT RECOGNITION

by Bir Bhanu, Xuejun Tan

ISBN 978-1-4613-5103-0 ISBN 978-1-4615-0491-7 (eBook)

DOI 10.1007/978-1-4615-0491-7

Copyright © 2004 Springer Science+Business Media New York

Originally published by Kluwer Academic Publishers in 2004

Softcover reprint of the hardcover 1st edition 2004

All rights reserved. No part of this publication may be reproduced, stored in a retrieval system or transmitted in any form or by any means, electronic, mechanical, photo-copying, microfilming, recording, or otherwise, without the prior written permission of the publisher, with the exception of any material supplied specifically for the purpose of being entered and executed on a computer system, for exclusive use by the purchaser of the work.

Permissions for books published in the USA: permissions@wkap.com

Permissions for books published in Europe: permissions@wkap.nl

Printed on acid-free paper.

Contents

LIST OF FIGURES	IX
LIST OF TABLES	XV
PREFACE	XVII
1. INTRODUCTION	1
1.1 RESEARCH HISTORY	1
1.2 FINGERPRINT COLLECTION	6
1.3 FINGERPRINT FORMATION	6
1.4 FUNDAMENTAL CONCLUSIONS	8
1.5 FUNDAMENTAL RECOGNITION SYSTEMS	8
1.6 OUTLINE OF THE BOOK	11
2. LEARNED TEMPLATES FOR MINUTIAE EXTRACTION	13
2.1 INTRODUCTION	13
2.2 RELATED RESEARCH AND CONTRIBUTIONS	14
2.2.1 Related Research	14
2.2.2 Contributions	18
2.3 TECHNICAL APPROACH	18
2.3.1 Offline Learning of Templates	19
2.3.2 Run Time Feature Extraction	24

2.4	EXPERIMENTS	26
2.4.1	Database	26
2.4.2	Learned Templates	27
2.4.3	Results	29
2.5	CONCLUSIONS	32
3.	FINGERPRINT INDEXING	33
3.1	INTRODUCTION	33
3.2	RELATED RESEARCH AND CONTRIBUTIONS	35
3.2.1	Related Research	35
3.2.2	Contributions	37
3.3	TECHNICAL APPROACH	38
3.3.1	Indexing Components	39
3.3.2	Geometric Constraints	43
3.3.3	Indexing Score	44
3.3.4	Algorithms	44
3.3.5	Probability of False Indexing	46
3.4	EXPERIMENTS	47
3.4.1	Database	47
3.4.2	Performance Evaluation Measures for Indexing	48
3.4.3	Indexing Results	49
3.4.4	Effect of Geometric Constraints	52
3.4.5	Extrapolation of Indexing Performance	52
3.4.6	Comparison of Approaches	55
3.5	CONCLUSIONS	56
4.	FINGERPRINT MATCHING BY GENETIC ALGORITHMS	59
4.1	INTRODUCTION	59
4.2	RELATED RESEARCH AND CONTRIBUTIONS	60
4.2.1	Related Research	60
4.2.2	Contributions	61
4.3	TECHNICAL APPROACH	62
4.3.1	Fingerprint Matching Problem	62
4.3.2	Selection of an Optimization Technique	62
4.3.3	Optimization Based on GA	63
4.3.4	GA Based Fingerprint Representation	66
4.4	EXPERIMENTS	71
4.4.1	Database	71
4.4.2	Estimation of Parameters for GA	72
4.4.3	Results	78

4.4.4	Effectiveness of Selection and Crossover Operators	80
4.4.5	Computation Time	81
4.5	CONCLUSIONS	82
5. GENETIC PROGRAMMING FOR FINGERPRINT CLASSIFICATION		83
5.1	INTRODUCTION	83
5.2	RELATED RESEARCH AND CONTRIBUTIONS	84
5.2.1	Related Research	84
5.2.2	Contributions	89
5.3	TECHNICAL APPROACH	90
5.3.1	Design Considerations	92
5.3.2	Reproduction, Crossover and Mutation	97
5.3.3	Steady-State and Generational Genetic Programming	100
5.4	EXPERIMENTS	103
5.4.1	Database	103
5.4.2	Results	104
5.5	CONCLUSIONS	116
6. CLASSIFICATION AND INDEXING APPROACHES FOR IDENTIFICATION		117
6.1	INTRODUCTION	117
6.2	RELATED RESEARCH AND CONTRIBUTIONS	118
6.2.1	Related Research	118
6.2.2	Contributions	123
6.3	TECHNICAL APPROACH	123
6.3.1	Indexing	123
6.3.2	Verification	125
6.4	EXPERIMENTS	127
6.4.1	Classification results	128
6.4.2	Indexing Results	130
6.4.3	Identification Results	130
6.5	CONCLUSIONS	133
7. FUNDAMENTAL PERFORMANCE ANALYSIS – PREDICTION AND VALIDATION		135
7.1	INTRODUCTION	135
7.2	RELATED RESEARCH AND CONTRIBUTIONS	138
7.2.1	Related Research	138
7.2.2	Contributions	142

7.3	TECHNICAL APPROACH	142
7.3.1	Two-Point Model	147
7.3.2	Three-Point Model	148
7.4	EXPERIMENTS	155
7.4.1	Database	155
7.4.2	Parameters	156
7.4.3	Estimation of $P_{3,M}\{M_3 = s \mid M = i\}$	158
7.4.4	Results	160
7.4.5	Real Data	166
7.4.6	Error Rates between Fingerprint and Iris	167
7.5	CONCLUSIONS	168
8.	SUMMARY AND FUTURE WORK	169
8.1	SUMMARY	169
8.2	FUTURE WORK	172
REFERENCES		175
INDEX		189

List of Figures

Chapter 1

<i>Figure 1.1.</i> A piece of ancient Chinese document which used fingerprint as identity (courtesy of [131]).	2
<i>Figure 1.2.</i> Fingerprint science pioneers: (a) Sir Francis Galton and (b) Sir Edward Henry (courtesy of [131]).	3
<i>Figure 1.3.</i> Examples of fingerprints from each class of <i>Henry System</i> for fingerprint classification: (a) Right Loop (R); (b) Left Loop (L); (c) Whorl (W); (d) Arch (A) and (e) Tented Arch (T).	5
<i>Figure 1.4.</i> AFS8500 fingerprint sensor manufactured by AuthenTec Inc. (courtesy of [126]).	7
<i>Figure 1.5.</i> Examples of minutiae: endpoint and bifurcation. Singular points, core and delta, are also shown in this figure.	7
<i>Figure 1.6.</i> (a) Fingerprint authentication system (AFAS) and (b) Fingerprint identification system (AFIS).	9
<i>Figure 1.7.</i> General block diagram of a fingerprint recognition system.	11

Chapter 2

<i>Figure 2.1.</i> Block diagram for minutiae-based feature extraction.	14
<i>Figure 2.2.</i> Block diagram of our approach for minutiae extraction.	19
<i>Figure 2.3.</i> Illustration of an ideal endpoint template T.	20
<i>Figure 2.4.</i> Sample fingerprints in NIST-4.	27

<i>Figure 2.5.</i> Examples of training data: endpoint (first row) and bifurcation (second row).	28
<i>Figure 2.6.</i> Binarized examples of training data in Figure 2.5.	28
<i>Figure 2.7.</i> Learned templates: endpoint (left) and bifurcation (right).	28
<i>Figure 2.8.</i> Examples of minutiae extraction (\square : endpoint, \times : bifurcation).	29
<i>Figure 2.9.</i> Goodness Value of fifteen fingerprints.	30
<i>Figure 2.10.</i> Comparison of experimental results.	31

Chapter 3

<i>Figure 3.1.</i> Block diagram of different approaches to solve identification problem.	34
<i>Figure 3.2.</i> Illustration of variables.	39
<i>Figure 3.3.</i> Definition of feature points' labels.	41
<i>Figure 3.4.</i> $\Delta_{ABC'}$ and Δ_{ABC} have the same angles but different handedness.	42
<i>Figure 3.5.</i> Examples of minutiae with different v values.	42
<i>Figure 3.6.</i> Algorithms used in our experiments.	45
<i>Figure 3.7.</i> $P\{v_k > T\}$ with respect to T .	47
<i>Figure 3.8.</i> Sample fingerprints in NIST-4 database.	48
<i>Figure 3.9.</i> Number of corresponding triangles of the 2000 query fingerprints.	49
<i>Figure 3.10.</i> Distribution of corresponding triangles among 2000 query fingerprints.	50
<i>Figure 3.11.</i> CIP performance varies with number of corresponding triangles.	51
<i>Figure 3.12.</i> Effect of constraints.	52
<i>Figure 3.13.</i> Performance on NIST-4.	53
<i>Figure 3.14.</i> Performance with large M .	54
<i>Figure 3.15.</i> Comparison of Germain et al.'s and our approaches on NIST-4 dataset.	54
<i>Figure 3.16.</i> Comparison of Germain et al.'s approach and our approach on NIST-4.	55

Chapter 4

<i>Figure 4.1.</i> Block diagram of GA based approach.	65
<i>Figure 4.2.</i> Illustration of $\hat{F}_{\hat{e}}(\bullet)$, where $\hat{e} = (\hat{s}, \hat{\theta}, \hat{t}_1, \hat{t}_2)$.	67

<i>List of Figures</i>	xi
------------------------	----

<i>Figure 4.3.</i> Examples of ridge count and minutiae density.	68
<i>Figure 4.4.</i> An example of uniform crossover operator in GA.	70
<i>Figure 4.5.</i> Sample fingerprints in NIST-4 database. This pair of fingerprints are quite challenging to match.	72
<i>Figure 4.6.</i> Histograms of transformation parameters (first 100 pairs of fingerprints in NIST-4).	73
<i>Figure 4.7.</i> Fitness value changes for matching between the fingerprints in Figure 4.5.	74
<i>Figure 4.8.</i> Examples of maximum fitness value between two fingerprints which are from the <i>same</i> fingers (genuine matching).	75
<i>Figure 4.9.</i> Examples of maximum fitness value between two fingerprints which are from the <i>different</i> fingers (impostor matching).	76
<i>Figure 4.10.</i> PDF of fitness value for genuine matchings.	77
<i>Figure 4.11.</i> PDF of fitness value for imposter matchings.	77
<i>Figure 4.12.</i> Comparison of ROC curves of two approaches using entire NIST-4 database (2000 pairs of fingerprints).	79
<i>Figure 4.13.</i> Sample fingerprints with low fitness value in genuine matching.	80
<i>Figure 4.14.</i> Comparison of the average performance of pure GA and its two other variations.	81

Chapter 5

<i>Figure 5.1.</i> Examples of fingerprints from each class of <i>Henry System</i> for fingerprint classification: (a) Right Loop; (b) Left Loop; (c) Whorl; (d) Arch and (d) Tented Arch.	85
<i>Figure 5.2.</i> Examples of <i>Core</i> and <i>Delta</i> .	86
<i>Figure 5.3.</i> Block diagram of our approach.	91
<i>Figure 5.4.</i> An example of fingerprint image from NIST-4 fingerprint database and the primitive feature images derived from the original image: (a) original image, f0760_06; and (b) primitive feature images. Note that 16 primitive feature images are sorted from left to right and top to bottom.	94
<i>Figure 5.5.</i> An example of a composite operator, which includes three computation operators and three feature generation operators. PFs are primitive features.	96
<i>Figure 5.6.</i> An example of crossover between two composite operators.	98
<i>Figure 5.7.</i> An example of mutation.	99

<i>Figure 5.8.</i> Steady-state Genetic Programming.	101
<i>Figure 5.9.</i> Generational Genetic Programming.	102
<i>Figure 5.10.</i> Sample fingerprints from NIST-4.	103
<i>Figure 5.11.</i> Average fitness values based on the number of generations: (a) 5-class and (b) 4-class.	105
<i>Figure 5.12.</i> Learned composite operators: (a) 5-class, size is 61; and (b) 4-class, size is 149.	106
<i>Figure 5.13.</i> Tree structure for the composite operator in Figure 5.12(a). All these operators are defined in Table 5.2, and the size of the tree is 61. Feature generation operators are shown in black font and start with SPE, and all the others are computation operators. PFs are primitive features.	107
<i>Figure 5.14.</i> Output images of each node for the composite operator in Figure 5.12(a). The input fingerprint is s1000_07 shown in Figure 5.10. Those images are sorted from left to right and top to bottom according to the preorder traversal of the composite operator. The size of the composite operator is 61.	108
<i>Figure 5.15.</i> Output images of each node for the composite operator in Figure 5.12(b). The input fingerprint is s1000_07 shown in Figure 5.10. Those images are sorted from left to right and top to bottom according to the preorder traversal of the composite operator. The size of the composite operator is 149.	109
<i>Figure 5.16.</i> Feature vectors generated by the composite operators shown in Figure 5.12 on fingerprint s1000_07: (a) 5-class, length is 87 and (b) 4-class, length is 102. Note that, vertical scales are different in (a) and (b).	110
<i>Figure 5.17.</i> Fingerprints with two ground-truth labels: T and L.	111
<i>Figure 5.18.</i> Errors in classification: (a) ground-truth: R, classification: W; (b) ground-truth: T, classification: A; (c) ground-truth: R, classification: T; (d) ground-truth: T/R, classification: A.	113
<i>Figure 5.19.</i> Fingerprints that are misclassified in our approach for 5- class classification: Fingerprint_ID (class, ground-truth). 0, 1, 2, 3 and 4 represent classes T, A, W, R and L, respectively.	114
<i>Figure 5.20.</i> Fingerprints that are misclassified in our approach for 4- class classification: Fingerprint_ID (class, ground-truth). 0, 1, 2 and 3 represent classes T/A, W, R and L, respectively.	115

Chapter 6

<i>Figure 6.1.</i> Block diagram of different approaches to solve identification problem.	119
<i>Figure 6.2.</i> Two step approach for fingerprint identification.	122
<i>Figure 6.3.</i> Number of potential corresponding triangles between each pair of genuine indexing.	129
<i>Figure 6.4.</i> Indexing performance.	130
<i>Figure 6.5.</i> Identification results using classification based approach.	131
<i>Figure 6.6.</i> Identification results using indexing based approach.	132

Chapter 7

<i>Figure 7.1.</i> Illustration of the variables. The matching takes place from query to template.	143
<i>Figure 7.2.</i> After transform T, minutia B is in minutia A's uncertainty area. If minutia B's ridge direction is in minutia A's directional uncertainty area, then we think their directions to be the same and minutiae A and B are the corresponding minutiae.	146
<i>Figure 7.3.</i> Examples of minutiae and ridge count.	146
<i>Figure 7.4.</i> Basic procedure to estimate the lower bound of $P_{3,M}(M_3=s \mid M=i)$.	150
<i>Figure 7.5.</i> To construct 2 triangles, the minimum number of sides and points that are needed are 5 and 4, respectively. There are 6 variations of 4 points and 5 associated sides to form 2 triangles.	151
<i>Figure 7.6.</i> Relations between two sets of feature points F_1 and F_2 .	152
<i>Figure 7.7.</i> Distribution of ridge counts.	157
<i>Figure 7.8.</i> Distribution of ridge counts' difference.	157
<i>Figure 7.9.</i> Basic simulation procedure.	158
<i>Figure 7.10.</i> Estimated $P_{3,M}(M_3=s \mid M=i)$.	159
<i>Figure 7.11.</i> Simulation results of $P_{3,M}(M_3=s \mid M=i)$ and its 95% confidence intervals for various M : (a) $M=5$; (b) $M=8$; (c) $M=12$; (d) $M=18$.	160
<i>Figure 7.12.</i> PDF of 2-point model's error rate.	161
<i>Figure 7.13.</i> PDF of 3-point model's error rate.	162
<i>Figure 7.14.</i> Lower bound of the performance.	163
<i>Figure 7.15.</i> Comparison of different models' estimation.	163

<i>Figure 7.16.</i> PDF of the number of matching minutiae in Pankanti et al.'s work [81].	165
<i>Figure 7.17.</i> Ratio of estimation between our approach and Pankanti et al. [81].	165
<i>Figure 7.18.</i> Comparison of theoretical results and real-data results.	167

List of Tables

Chapter 2

<i>Table 2.1.</i> Properties of <i>Crossing Number</i> .	15
<i>Table 2.2.</i> Other approaches for minutiae extraction.	16

Chapter 3

<i>Table 3.1.</i> Comparisons between Germain et al.'s [33] and our approach.	36
<i>Table 3.2</i> NIST-4 database used in published fingerprint recognition research.	38
<i>Table 3.3.</i> Expectation of the percentage of angle changes less than various thresholds.	40

Chapter 4

<i>Table 4.1.</i> Parameters in experiments.	74
--	----

Chapter 5

<i>Table 5.1.</i> Representative fingerprint classification approaches.	88
<i>Table 5.2.</i> Primitive operators used in our approach. The first nine rows are computation operators and the last five rows are feature generation operators.	95

<i>Table 5.3.</i> Confusion matrix of the testing results for 5 and 4-class classifications.	111
<i>Table 5.4.</i> Classification results on NIST-4.	116

Chapter 6

<i>Table 6.1.</i> Confusion matrix for 5-class classifications.	128
<i>Table 6.2.</i> Classification results on NIST-4.	129

Chapter 7

<i>Table 7.1.</i> Simulation results for different s triangles.	151
<i>Table 7.2.</i> Parameters used in estimation.	156
<i>Table 7.3.</i> Estimated error rate for different m , n and s .	162

Preface

Recently the world has seen an increasing activity in biometrics for both the defense and security related applications and the commercial applications such as e-commerce. Biometrics such as fingerprint, face, gait, iris, voice, signature, etc. recognize a person's identity using his/her physiological or behavioral characteristics. These biometrics are needed for reliable and positive identification of an individual beyond the traditional password and identification numbers, which may not be reliable. Among these biometric signs, fingerprint is the one, which has been researched for a long time and shows the most promising future in real-world applications. However, because of the complex distortions among the different impressions of the same finger, fingerprint recognition is still a challenging problem as the cover page demonstrates. This figure illustrates the distortions in the signature of the same index finger, under different conditions, of an individual taken with an electronic fingerprint sensor. Left to right and top to bottom images are: (1) baseline fingerprint image; (2) finger rotated counter-clockwise; (3) finger sheared towards right; (4) finger with moisture; (5) low resolution fingerprint, similar to the image in (9); (6) finger sheared towards tip; (7) finger translated; (8) finger applied lighter pressure on the sensor; (9) finger with dirt; and (10) low resolution fingerprint, similar to the image in (1).

In this book, we present an entire range of novel computational algorithms for fingerprint recognition. These include computer algorithms for feature extraction, indexing, matching, classification, identification, and performance prediction and validation. These algorithms have been compared with state-of-the-art algorithms and have been found to be effective and efficient on real-world data such as the NIST-4 database from National Institute of Standards and Technology (NIST). These algorithms will be of great interest to engineers, scientists, application developers and students involved in the basic and applied research in the emerging scientific field of *Biometrics*.

Specific algorithms addressed in this book include:

Learned templates-based minutiae extraction: Templates are learned from examples by optimizing a criterion function using Lagrange's method. To detect the presence of minutiae, templates are applied with appropriate orientations to the binarized fingerprints only at selected potential minutia locations. Experimental results show that learned templates can improve both the detection of features and the performance of an identification system.

Triplets of minutiae based fingerprint indexing: Novel features derived from triplets of minutiae are used as components for fingerprint indexing. Geometric constraints based on other characteristics of minutiae are used to eliminate false correspondences. Experimental results show that the indexing approach efficiently narrows down the number of candidate hypotheses in the presence of translation, rotation, scale, shear, occlusion and clutter. Comparisons between our approach and another prominent indexing approach show that the performance of our approach is much better.

Genetic algorithm based fingerprint matching: Genetic algorithm is used to find the optimized global transformation between two different fingerprints for verification. Fitness function is based on the local properties of each triplet of minutiae. Experimental results show

that the proposed approach can achieve good performance even when a large portion of fingerprints in the database is of poor quality and it is better than a mean-squared error estimation based approach.

Genetic Programming based feature learning for fingerprint classification: Genetic Programming (GP) is used in a novel manner to discover composite operators and associated features that are evolved from combinations of primitive operators and primitive features. Primitive features, based on the orientation image, allow the incorporation of domain knowledge into the evolutionary computational process. Experimental results, using a Bayesian classifier and the newly discovered features by genetic programming, show that the correct rates for four class (right loop, left loop, whorl, arch/tented arch) and five class (right loop, left loop, whorl, arch and tented arch) classification have advantages over the best results published to date.

Comparison of classification and indexing based approaches for identification: The traditional approach for fingerprint identification is based on classification of a fingerprint into one of the known classes such as right loop, left loop, etc. to be followed by verification. In this book we develop an alternate, model-based approach, that consists of indexing followed by verification. Extensive comparisons between indexing and classification based techniques for identification are performed. These comparisons show that, although the classification technique that we use has advantages over the other techniques published in the literature, the indexing approach performs better based on both the size of the search space and the identification results.

Fundamental matching performance prediction and validation: We develop statistical performance prediction models based on not only the positions and orientations of minutiae features but also the relations between different minutiae. Error rates of fingerprint matching, based on pairs or triplets of minutiae, that we obtain are significantly lower than that of previously published research. The fingerprint recognition rates are also compared with the iris biometric. These results

contribute towards making fingerprint matching a science and help settling the legal challenges to fingerprints.

This project started out as a commercial/security application of our research in automatic object recognition. The authors are grateful to a number of people who helped them with this effort. The support and inputs by Mr. Yoshihiro Tukamura, and Mr. William Saito were very helpful. The authors are grateful to Dr. Bob Herklotz for providing the critical support when it was needed. The authors are thankful to Dean Satish Tripathi for his support and encouragement over the years.

Various discussions with Mr. Grinnell Jones, Dr. Michael Boshra, Dr. Yingqiang Lin, Dr. Sohail Nadimi, Mr. Anlei Dong and Mr. Ju Han during the course of this work were very useful. The authors would like to thank Dr. Michael Boshra for providing comments on the manuscript. The authors would like to acknowledge the partial support received from Sony Corporation, I/O Software Inc., the Digital Media Initiative (DiMI) Program of the University of California and the AFOSR grants F49620-97-1-0184 and F49620-02-1-0315. Finally, the authors would like to thank NIST for the availability of fingerprint databases.

Bir Bhanu and Xuejun Tan
Riverside, California, USA

Chapter 1

INTRODUCTION

1.1 RESEARCH HISTORY

One of the most important human abilities is to recognize different objects. Recognition is defined as a procedure that involves perception and associates the information with certain memory. Computer vision and pattern recognition scientists and engineers develop computer systems to recognize objects in the real-world. However, in spite of significant advancements, 3D object recognition in the real-world with changing environmental conditions remains a challenging problem. Nevertheless, research in certain areas in this field, e.g. biometrics, shows promising results [35].

Biometric signs include fingerprint, iris, face, gait, palm, speech, signature, etc. [40][80][118][123]. Among them, fingerprint is the one that has been used for a long time [72]. The earliest dated prints of human hands and feet were made about 4,000 years ago during the pyramid-building era in Egypt [131]. Also, one small portion of palm print, not known to be human, has been found impressed in hardened mud at a 10,000-year old site in Egypt. It was common practice for Chinese to use inked fingerprints on official documents. The oldest

existing documents so endorsed dated from the 3rd century BC, and it was still an effective practice until recent times. Figure 1.1 shows a piece of ancient Chinese documents, which used a fingerprint as identity. Even though it is recorded that Chinese used their fingerprints to establish identity in courts, researchers fail to know whether Chinese were fully aware of the uniqueness of a fingerprint or whether the physical contact with documents had some spiritual significance [131].

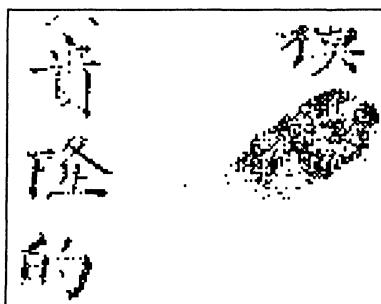


Figure 1.1. A piece of ancient Chinese document which used fingerprint as identity (courtesy of [131]).

Dr. Henry Faulds probably made the greatest advances in finger-print science in the late 19th and early 20th centuries. He became interested in fingerprints after 1874 while he was working at a hospital in Tokyo, Japan. After careful experiments and observations, he believed that superficial injury did not alter fingerprint patterns and as the injury healed, fingerprint patterns returned to their former patterns. In a letter written to Nature in October 1880, he described pattern formations on the fingers and stated how good sets of fingerprints may be obtained by using “a common slate or smooth board of any kind, or a sheet of tin, spread over very thinly with printer’s ink”. This technique, still in use today, appears to be a botanical technique called nature-printing. His most important conclusion was: 1) Fingerprints do not change; and 2) Fingermarks left on objects by bloody or greasy fingers “may lead to the scientific identification of criminals.”

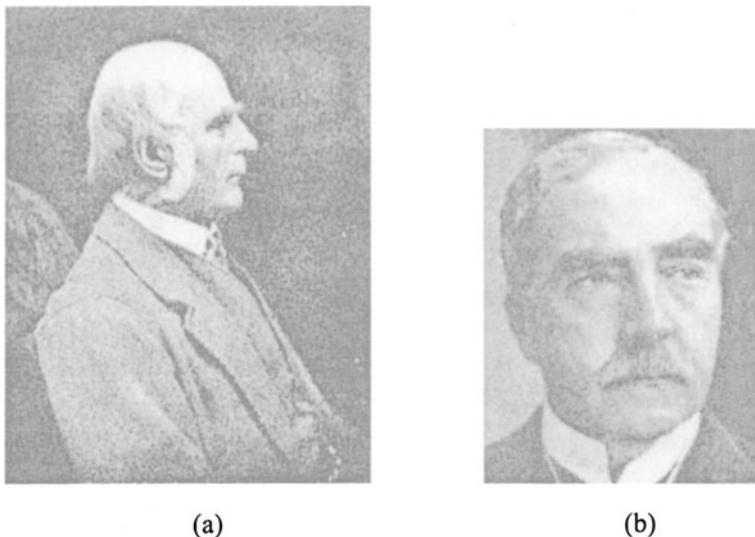


Figure 1.2. Fingerprint science pioneers: (a) Sir Francis Galton and (b) Sir Edward Henry (courtesy of [131]).

Sir William Herschel, an English administrator in India, commenced placing the inked palm and thumb impressions of some members of local population on contracts. These prints were used as a form of signature on the documents because of the high level of illiteracy in India and frequent attempts at forgery. Throughout his life, Herschel took his own fingerprints and noticed that no change had occurred in them in over 50 years. He also had a small collection of fingerprints and used his technique of handprinting to detect forgeries of legal documents. The fingerprints taken from prisoners were also of great interest to him, and he had the opportunity to see the same prisoners fingerprinted several times over some years with no change occurring in their fingerprints. However, Herschel never claimed that he had developed a method of registering and identifying criminals, nor did he foresee any crime scene application as Faulds had done.

In 1892, Sir Francis Galton (Figure 1.2(a)) published an accurate and in-depth study of fingerprint science [32], which included an attempt at a fingerprint classification system to facilitate the handling of

large collections of fingerprints. Although Galton's work proved to be sound and became the foundation of modern fingerprint science, his approach to classification was inadequate.

Juan Vucetich, an Argentinian police officer who corresponded with Galton, devised his own fingerprint classification system, which was put into practice in September 1891. In March 1892, Vucetich opened the first fingerprint bureau at San Nicholas, Buenos Aires, Argentina. In June 1892 at Necochea, Argentina, Francisca Rojas claimed that she had been brutally attacked and her two children had been murdered by a neighbor named Velasquez. However, Velasquez refused to confess to the murder of two children. Nine days after the crime, a search of the crime scene was carried out and a number of fingerprints in blood were found on a door post of the woman's hut. To people's surprise, Vucetich's fingerprint bureau found that these fingerprints were identical with the inked fingerprints of Rojas'. When confronted with this evidence, Rojas confessed to the murder of her children, and in July 1892 she was found guilty of their murder and sentenced to life imprisonment.

Sir Edward Henry (Figure 1.2(b)), who had been taught in fingerprints by Galton, established the famous Henry System, which is a systematic and effective method of classifying fingerprints. The classes he used are Right Loop (R), Left Loop (L), Whorl (W), Arch (A), and Tented Arch (T). Examples from each class are shown in Figure 1.3. Henry published his book *Classification and Uses of Fingerprints* in 1900. In 1901, he was appointed as Assistant Commissioner of Police at New Scotland Yard and began to introduce his fingerprint system into that institution. By the end of that year, the Fingerprint Office at New Scotland Yard was fully functional, and the first British court conviction by fingerprints was obtained in 1902. Approximately 10 years after the publication of Henry's book, his classification system was being used by police forces and prison authorities throughout the English-speaking world.

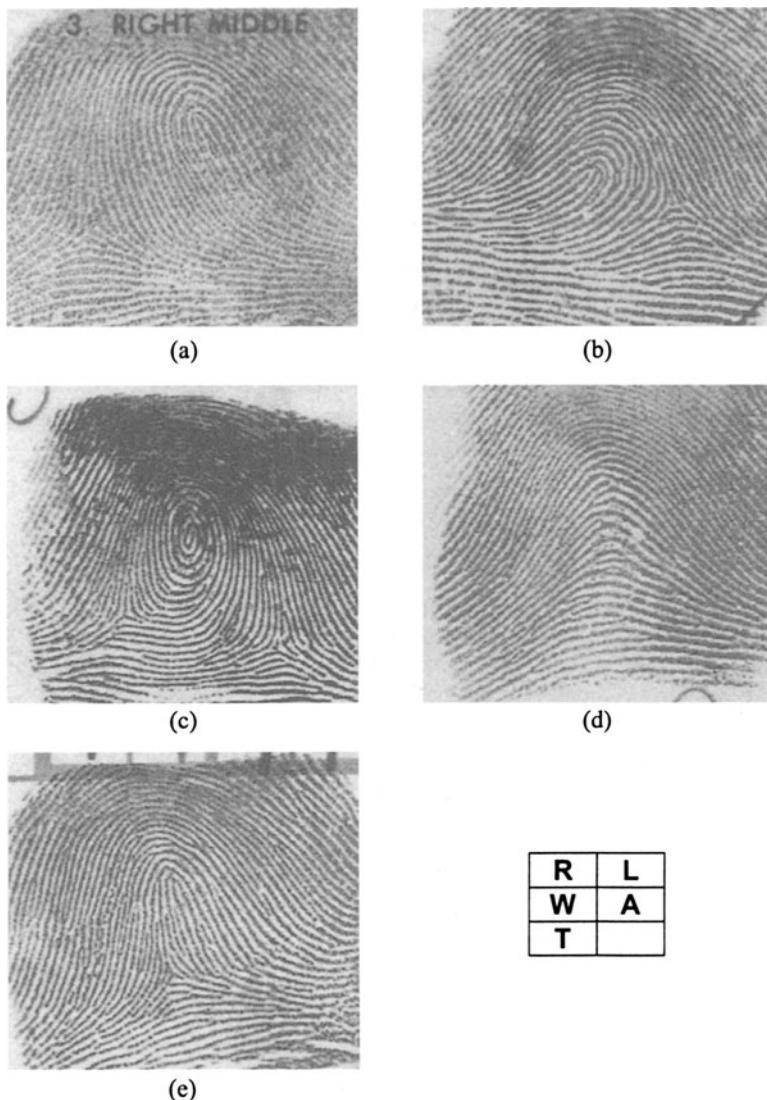


Figure 1.3. Examples of fingerprints from each class of *Henry System* for fingerprint classification: (a) Right Loop (R); (b) Left Loop (L); (c) Whorl (W); (d) Arch (A) and (e) Tented Arch (T).

After Galton and Henry, work on fingerprint recognition was extended and refined. In the early 20th century, fingerprints were formally used as valid signs of identity by law-enforcement agencies. However, manual fingerprint identification is tedious, time-consuming and expensive. Therefore, in 1960, Home Office (UK) and the Paris Police Department initiated studies on automatic fingerprint based biometric systems.

1.2 FINGERPRINT COLLECTION

Usually, there are two common methods to collect a fingerprint:

- Obtain an impression of an inked finger on paper and scan it by a scanner;
- Scan a fingerprint directly by a fingerprint sensor.

The first method has been widely used. However, with the development of fingerprint sensors, recently, more and more people use fingerprint sensors to collect fingerprints. There are some commercially available fingerprint sensors in the market, such as:

- DFR2080 and DFR2090 from Identix Inc. [125]
- AFS8500, AES3400 and AES2500 from AuthenTec Inc. [126]
- FIU710 from SONY Inc. [129]
- FPS110 and FPS200 from Veridicom Inc. [130]

Figure 1.4 shows an AFS8500 fingerprint sensor manufactured by AuthenTec Inc. More information about it can be found at [126].

1.3 FINGERPRINT FORMATION

A fingerprint is a group of associated curves. Figure 1.3 shows examples of fingerprint images. The white curves are called *valleys* while the dark curves are called *ridges*. In the research of Galton, he defined

minutiae by the crossing and ending of ridges, which include endpoint, bifurcation, forks, islands and enclosures. Minutiae that people usually use are only *endpoint* and *bifurcation* since other minutiae can be expressed by the combination of them. Figure 1.5 shows examples of endpoint and bifurcation. Singular points, including core and delta, shown in this figure are also used in fingerprint authentication.

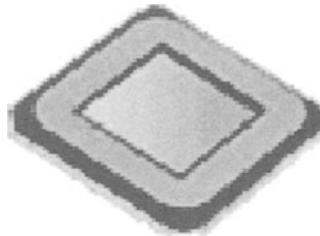


Figure 1.4. AFS8500 fingerprint sensor manufactured by AuthenTec Inc. (courtesy of [126]).

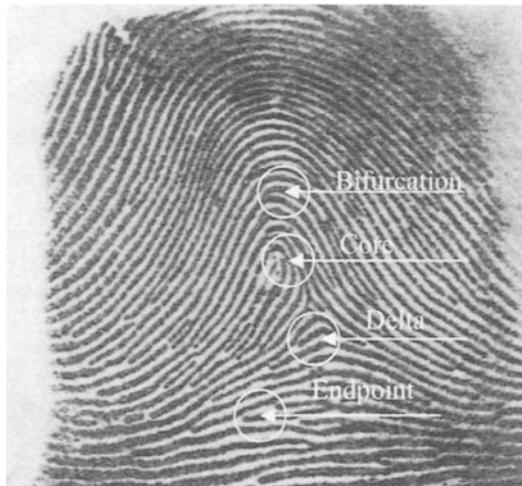


Figure 1.5. Examples of minutiae: endpoint and bifurcation. Singular points, core and delta, are also shown in this figure.

Both methods of fingerprint collection may generate highly distorted fingerprints. Following distortions between two impressions of the same finger are possible:

- *Translation* and *Rotation* because of different positions of the finger on the input device;
- *Scaling* and *Contrast Difference* because of different pressure;
- *Shear* because of different shear force;
- *Local Perturbations* because of non-uniform pressure and shear;
- *Occlusion* because of limited sensor area and dryness;
- *Clutter* because of scars, sweat and smudge.

1.4 FUNDAMENTAL CONCLUSIONS

Two fundamentally important conclusions were reached through extensive experiments on different age groups within different races:

- **Permanence:** A person's fingerprints will not naturally change their structure after about one year of his/her birth;
- **Uniqueness:** The fingerprints of individuals are unique. Even the fingerprints among twins are not the same (Jain et al. [46]). In real world, people have never found any two fingerprints, which are exactly the same.

Both conclusions were the building blocks of research in this field over the last century.

1.5 FUNDAMENTAL RECOGNITION SYSTEMS

In terms of utilizations, there are two kinds of fingerprint based biometric systems [35]:

- Automatic Fingerprint Identification Systems (AFIS).

- Automatic Fingerprint Authentication Systems (AFAS)¹;

Fingerprint based biometric systems are usually used as AFIS for criminal identification and police work. With the development of fingerprint science, AFAS is in a great demand for applications in access control and financial security. There are some commercial AFAS available in the market, which includes products from AuthenTec Inc. [126], I/O Software Inc. [127] and Digital Persona Inc. [128]. A list of them can be found at [132].

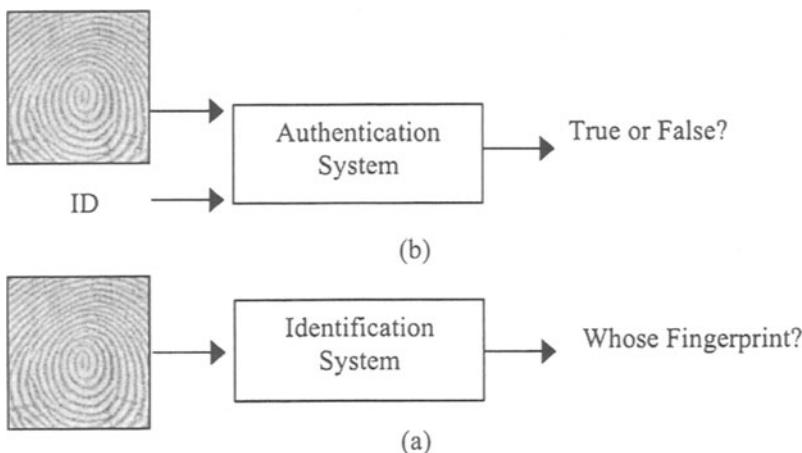


Figure 1.6. (a) Fingerprint authentication system (AFAS) and (b) Fingerprint identification system (AFIS).

In an AFAS, the input includes an identity (ID) and a fingerprint, and the output is an answer of yes or no, which indicates whether the input fingerprint is consistent with the ID. The system simply compares the input fingerprint with the one addressed by the ID in the database. However, in an AFIS, the input only includes a fingerprint and the output is a short list of persons who might have the fingerprint. The system compares the input fingerprint with many fingerprint records in the da-

¹ Besides *authentication*, researchers also use the terms *verification* or *matching* in fingerprint recognition field. The meaning of these words are the same in this book.

tabase. Figure 1.6 shows the structures of these two systems. We can think AFAS as a special kind of AFIS. The reason is that if the output of an AFIS is one person then the outputs of AFAS and AFIS are the same. Furthermore, AFAS is an important component of AFIS. Since the output of an AFIS is a short list of persons, usually, it needs an AFAS to verify the exact identity.

Since the distortions, which exist in different impressions of the same finger, may be quite complex and fingerprint image quality is not always good, reliable fingerprint recognition is still a challenging problem. People believe that a person can be identified with high confidence based on the minutiae of his/her fingerprints. Generally, minutiae based fingerprint verification is a kind of point matching approach. In order to improve the performance, additional characteristics, such as local orientation and core/delta point position are used. However, the distortions between two sets of minutiae extracted from the different impressions of the same finger may include translation, rotation, scale, shear, local perturbation, occlusion and clutter, which make it difficult to find the corresponding minutiae reliably.

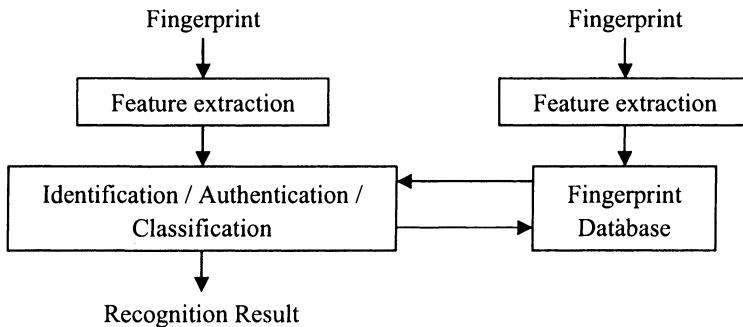


Figure 1.7. General block diagram of a fingerprint recognition system.

1.6 OUTLINE OF THE BOOK

Figure 1.7 shows a general block diagram of a fingerprint recognition system, including identification, authentication and classification. In the following chapter, a learned template based fingerprint minutiae extraction approach is introduced. Then, based on triplets of minutiae, an indexing and a verification approach are introduced in Chapter 3 and Chapter 4, respectively. In Chapter 5, a Genetic Programming (GP) based fingerprint classification is presented. The comparison of indexing and classification techniques for identification is given in Chapter 6. Following this, a framework for estimating the fundamental performance of fingerprint recognition is presented in Chapter 7. Finally, Chapter 8 provides a summary of the book.

Chapter 2

LEARNED TEMPLATES FOR MINUTIAE EXTRACTION

2.1 INTRODUCTION

Generally, image quality has a significant effect on gray-level features. Structural features, such as *Core* and *Delta* (Figure 1.5), can only be used to classify fingerprints into four or five classes. The most commonly used features in fingerprint recognition are minutiae (Figure 1.5). This representation reduces fingerprint recognition problem to a point matching problem. The key problem with this representation is that there is no *reliable* minutiae-based feature extraction algorithm. Figure 2.1 shows a block diagram of a minutiae-based feature extraction procedure, which is widely used in most fingerprint recognition systems.

Most current techniques for minutiae extraction in fingerprint images utilize complex preprocessing and postprocessing. In this Chapter, we present a new technique, based on the use of learned templates, which statistically characterize the minutiae. Templates are learned from examples by optimizing a criterion function using Lagrange's method. To detect the presence of minutiae in test images, templates

are applied with appropriate orientations to the binary image only at selected potential minutia locations. Several performance measures, which evaluate the quality and quantity of extracted features and their impact on identification, are used to evaluate the significance of learned templates. The performance of the proposed approach is evaluated on NIST special fingerprint database 4. The experimental results show that learned templates can improve both the features and the performance of the identification system.

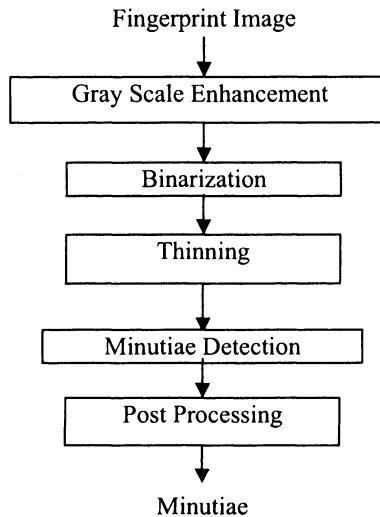


Figure 2.1. Block diagram for minutiae-based feature extraction.

2.2 RELATED RESEARCH AND CONTRIBUTIONS

2.2.1 Related Research

In Ratha et al. [86] and Jain et al. [43], fingerprints are taken as textured images and smoothed using a one-dimensional averaging mask on each line oriented along a direction orthogonal to the orientation

field of the window. A waveform projection-based ridge segmentation algorithm is used to locate ridges. Ridge skeleton is smoothed using morphological operators to detect the minutiae. A large number of spurious minutiae are removed by a postprocessing step.

In Mehtre [75], a method of computing directional image is presented. Contextual filters are designed based on directional image. The basic filter consists of linear combination of two filters, the averaging filter and the differentiating filter. A parallel iterative thinning algorithm is used to get the skeleton of fingerprint ridges. Minutiae are extracted by using the *Crossing Number (CN)*. CN at a point P is defined as: $CN = 0.5 \mid P_i - P_{i+1} \mid$, where P_i is the pixel value in the neighborhood of P . $P_i = 0$ or 1 , and i has a period of 8 , that is $P_9 = P_1$. CN has the following properties:

Table 2.1. Properties of *Crossing Number*.

CN	0	1	2	3	4
Property	Isolated point	End Point	Continuing Point	Bifurcation Point	Crossing Point

Endpoints and bifurcations are detected using properties of CN . Simple noise cleaning method is presented in the last step. Similar methods can be found in Kasaei et al. [56], Sun et al. [95], Jain et al. [41], Lin et al. [66], Jain et al. [42] and Amengual et al. [1].

In Lin et al. [67], a fast fingerprint enhancement algorithm is presented, which uses Gabor filters to adaptively improve the clarity of ridge and valley structures of input fingerprints based on the estimated local ridge orientation and frequency. The performance is evaluated by *Goodness Index (GI)* of the extracted minutiae and the accuracy of an online fingerprint verification system.

In Kamei and Mizoguchi [54], two distinct filters in Fourier domain are designed. One is a frequency filter corresponding to ridge frequencies, and the other is a directional filter corresponding to ridge direc-

tions on the basis of fingerprint ridge properties. An energy function for selecting features is defined by intensities of images obtained with the above filters and a measure of smoothness of the features. Using the image features, which minimize the energy function, an enhanced image is produced from the filtered images. Similarly, in Sherlock et al. [96], an enhancement algorithm, which is based on nonstationary directional Fourier domain filtering technique, is presented. Fingerprints are first smoothed using a directional filter, whose orientation matches local ridge orientation. The enhanced image is generated by thresholding. Other approaches, which also follow the procedure in Figure 2.1, are shown in Table 2.2.

Table 2.2. Other approaches for minutiae extraction.

System	Characteristics
O'Gorman et al. [77], 1989	Image is smoothed by application of oriented, matched filter masks
Hung [26], 1993	Filtering algorithms try to produce equally spaced ridge patterns, and the duality of foreground and background is used to purify ridge breaks
Fitz et al. [30], 1995	A hexagonal grid is used to smooth image and reduce noise
Wahab et al. [114], 1998	Local histogram equalization is used to modify ridge direction
Abutaleb et al. [1], 1999	Genetic algorithm is used to reconstruct ridge lines
Almansa et al. [3], 2000	Image is enhanced by shape adaptation of scale-space operators with automatic scale selection
Prabhakar et al. [82], 2000	A Learning Vector Quantizer is used to learn the characteristics
Wills et al. [115], 2001	A threshold FFT approach is used to smooth and enhance poor quality images and a wedge ring overlay minutiae detector is used to extract minutiae

The procedure in Figure 2.1 can be computationally expensive and requires advanced hardware to meet the response time requirements. Fuzzy logic is able to cope with complex data contained in images and

reduce the output information. It also requires less expensive hardware for similar performance. Minutiae extraction based on fuzzy approach can be found in Sagar and Alex [88] and Sagar et al. [89]. Another method, which is based on gray-level histogram decomposition, is proposed in Chang and Fan [21]. By decomposing the gray-level histogram, not only the ridge points can be located in complete fingerprint images, but also redundant ridge points can be eliminated according to some statistical parameters.

Minutiae extraction based on ridge line following is proposed in Maio and Maltoni [71]. From a mathematical point of view, a ridge line is defined as a set of points which are local maxima along one direction. Ridge line extraction algorithm attempts to locate, at each step, a local maximum relative to a section orthogonal to the ridge direction. By connecting the consecutive maxima, a polygonal approximation of the ridge line can be obtained. Minutiae are extracted directly from gray scale images. In Jiang et al. [53], an adaptive tracing algorithm has been proposed. In Domeniconi et al. [24], ridge is detected by gradient direction from gray scale image directly, which also avoids the time consuming procedure in Figure 2.1.

In Xiao and Raafat [119], the statistical and structural approaches are combined to represent fingerprint false minutia patterns, and a fingerprint image postprocessing algorithm is developed to eliminate the false minutiae. Similar postprocessing algorithm can be found in Tico and Kuosmanen [110].

Kovacs-Vajna et al. [61] address the problem of local average ridge distance computation. Two methods are considered: geometric and spectral. In the geometric approach the central points of ridges are estimated on a regular grid and straight lines passing through these points and parallel to the ridge directions are used. The second method is based on the computation of harmonic coefficients, which leads to an effective estimation of the average ridge period. Based on Kovacs-Vajna et al. [61], a set of algorithms for minutiae extraction from skeletonized binary fingerprints are proposed in Farina et al. [28].

2.2.2 Contributions

We present a template based approach for minutiae extraction, which is based on learning templates by Lagrange's method. The key contribution is the development of a new technique for learning templates for endpoints and bifurcations from examples and applying them adaptively to extract minutiae in fingerprints. Experimental results are evaluated using two measures on NIST-4 database to demonstrate the efficacy of the technique.

2.3 TECHNICAL APPROACH

Figure 2.2 shows the block diagram of our approach. For each fingerprint, first, the background is removed. Local orientation is computed in each local block. The fingerprint is adaptively smoothed according to local orientation and is then adaptively binarized and thinned. Potential minutiae are found using crossing number. Finally, learned templates are adaptively applied to purify the potential minutiae.

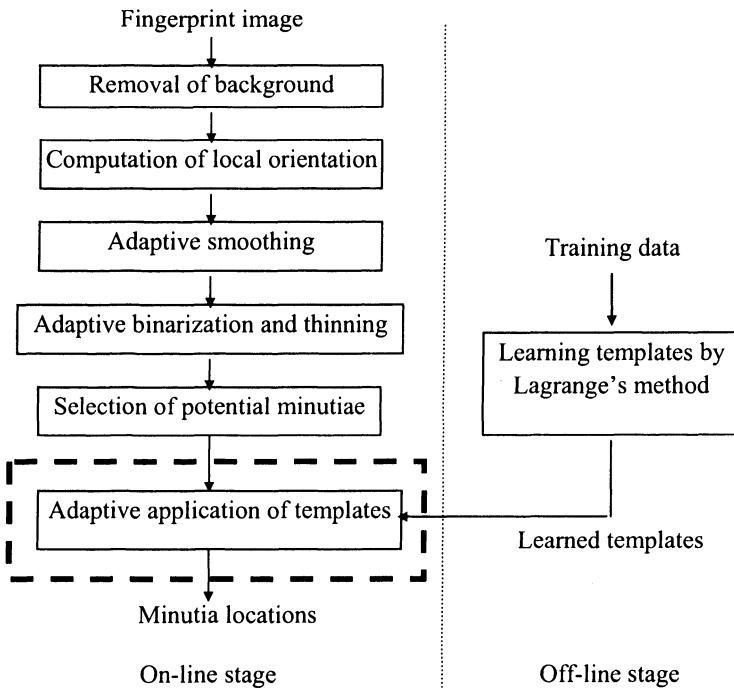


Figure 2.2. Block diagram of our approach for minutiae extraction.

2.3.1 Offline Learning of Templates

- **Template learning problem:** A template is a 2D filter that is concerned with detecting a minutia. Since a minutia can be an endpoint or a bifurcation, two templates are to be learned [14], one for each kind of feature. For simplicity, we use endpoints as the example to explain our learning approach. The template for bifurcations can be learned by a similar processing.

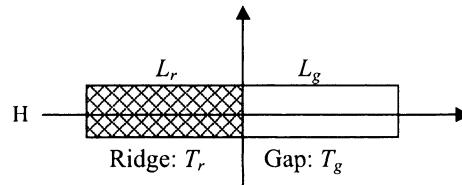


Figure 2.3. Illustration of an ideal endpoint template T .

Figure 2.3 shows an ideal endpoint template T that consists of two sub-templates, T_r (length L_r) and T_g (length L_g), which denote the template for ridge and gap, respectively. For simplicity, we assume $L_r = L_g$. H and L are the height and the length of the template T , and $L = L_r + L_g$. The value of each pixel in T_r and T_g are 1 and 0, respectively.

Suppose

- a ridge end E in a binary fingerprint is as ideal as the ideal template T ;
- the local orientation at the ridge end is θ_l ;
- the correlations between the template T and the ideal ridge end E with the orientation θ_l and $\theta_l + \pi$ are f_{θ_l} and $f_{\theta_l + \pi}$, respectively;
- the difference between f_{θ_l} and $f_{\theta_l + \pi}$ is Δ_{θ_l} ;

then

$$\Delta_{\theta_l} = f_{\theta_l} - f_{\theta_l + \pi} \quad (2.1)$$

The correlation between a template T and a ridge end E with the orientation θ_l is defined by:

$$f_{\theta_l} = \sum_{(h,l) \in (T \cap E_{\theta_l})} \{ T(h,l) \times E_{\theta_l}(h,l) \} \quad (2.2)$$

where $E_{\theta}(h, l)$ is the ridge with orientation θ_l and the template is applied along the ridge.

In real-world conditions, the ridge ends in a fingerprint are not ideal, so that the weight of each pixel in the template can not be measured by 1 or 0 alone. The problem of learning templates for minutiae extraction can be described as: given examples of endpoints and bifurcations, how to learn the templates that can be applied to minutiae extraction to avoid complex postprocessing?

- **Training data:** Suppose
 - a) Examples of endpoints and bifurcations are obtained from M fingerprint images FI_k , where $k = 1, 2, 3 \dots M$;
 - b) In the k th fingerprint image FI_k , there are N_k feature locations $(x_{k,i}, y_{k,i})$, where $i = 1, 2, 3 \dots N_k$;
 - c) in the local area around $(x_{k,i}, y_{k,i})$, $I_{k,i}(m, n)$ is the gray scale value at pixel (m, n) of the image FI_k , where $x_{k,i} - d_1 \leq m \leq x_{k,i} + d_1, y_{k,i} - d_2 \leq n \leq y_{k,i} + d_2$, d_1 and d_2 are constants;
 - d) $G = \{(x_{k,i}, y_{k,i})\}$.

Then, for each pixel in G , we perform the following steps:

- a) Estimate the local orientation $\theta_{k,i}$ at pixel $(x_{k,i}, y_{k,i})$ in the local area;
- b) Adaptively smooth $I_{k,i}(m, n)$ in the local area;
- c) Adaptively binarize $I_{k,i}(m, n)$ in the local area.

The details of these steps are the same as those in the run time minutiae extraction, which are given in Section 2.3.2.

- **Optimization for templates learning:** Suppose
 - a) The template is $T(h, l)$, where $1 \leq h \leq H$, $1 \leq l \leq L$, and $H = 2d_1 + 1$ and $L = 2d_2 + 1$;
 - b) $B_{k,i}(h, l)$ is the binary image of $I_{k,i}(m, n)$;
 - c) $B^{\theta_{k,i}}(h, l)$ is the rotated binary image of $B_{k,i}(h, l)$, rotation angle is $\theta_{k,i}$, which is the local orientation at pixel $(x_{k,i}, y_{k,i})$.

Then, according to (2.1) and (2.2), the objective of the learning algorithm can be defined as:

$$\arg \max_T \left\{ \sum_{k=1}^M \sum_{i=1}^{N_k} \sum_{h=1}^H \sum_{l=1}^L [T(h,l) \times Q_{k,i}(h,l)] \right\} \quad (2.3)$$

$$\text{where } Q_{k,i}(h,l) = B^{\theta_{k,i}}(h,l) - B^{\theta_{k,i}}(h, L-l).$$

If we normalize the template so that its energy is 1,

$$\sum_{h=1}^H \sum_{l=1}^L T^2(h,l) = 1 \quad (2.4)$$

Then, we can solve the optimization problem with Lagrange's method. Let

$$q(h,l) = \sum_{k=1}^M \sum_{i=1}^{N_k} Q_{k,i}(h,l) \quad (2.5)$$

Then, we have

$$\gamma = \sum_{h=1}^H \sum_{l=1}^L [T(h,l) \times q(h,l)] + \lambda \left[\sum_{h=1}^H \sum_{l=1}^L T^2(h,l) - 1 \right] \quad (2.6)$$

Let

$$\frac{\partial \gamma}{\partial T(h,l)} = q(h,l) + 2\lambda \times T(h,l) \quad (2.7)$$

$$\frac{\partial \gamma}{\partial T(h,l)} = 0 \quad (2.8)$$

Then, from (2.7) and (2.8), we have

$$\lambda = \frac{-q(h,l)}{2T(h,l)} \quad (2.9)$$

From (2.5) and (2.9), we have

$$\lambda = \frac{\sqrt{\sum_{h=1}^H \sum_{l=1}^L q^2(h,l)}}{2} \quad \text{and} \quad \frac{-\sqrt{\sum_{h=1}^H \sum_{l=1}^L q^2(h,l)}}{2} \quad (2.10)$$

Since

$$\frac{\partial^2 \gamma}{\partial T^2(h,l)} = 2\lambda \quad (2.11)$$

If $\lambda < 0$, (2.3) has a solution, otherwise it can not be satisfied. So,

$$\lambda = \frac{-\sqrt{\sum_{h=1}^H \sum_{l=1}^L q^2(h,l)}}{2} \quad (2.12)$$

and the template is

$$T(h,l) = \frac{q(h,l)}{\sqrt{\sum_{h=1}^H \sum_{l=1}^L q^2(h,l)}} \quad (2.13)$$

2.3.2 Run Time Feature Extraction

As shown in Figure 2.2, our approach consists of six steps: removal of background, computation of local orientation, adaptive smoothing, adaptive binarization and thinning, selection of potential minutiae and adaptive application of templates. These steps are summarized as follows:

1) Removal of Background: Since a fingerprint image usually includes some background that does not have any useful information, it is desired to eliminate it. We split a fingerprint into 16×16 blocks and compute the mean μ_s of the gray-scale value of the pixels in each block. We assume that if the mean μ_s is greater than a threshold, then the block belongs to background.

2) Computation of Local Orientation: The input fingerprint is first smoothed using a 5×5 Gaussian filter of $\mu = 0$ and $\sigma = 1$. Sobel operators are then applied to the smoothed image to estimate the gradient magnitude. After that, the fingerprint is split into 16×16 blocks with 8 pixels overlap along each dimension. For each block, the local orientation of the ridge pattern θ is obtained through Mean Square Error (*MSE*) (Bazen and Gerez [7]) estimate:

$$\theta = \frac{1}{2} \tan^{-1} \left(\frac{\sum_{i=1}^{16} \sum_{j=1}^{16} 2G_x(i,j)G_y(i,j)}{\sum_{i=1}^{16} \sum_{j=1}^{16} (G_x^2(i,j) - G_y^2(i,j))} \right) \quad (2.14)$$

where G_x and G_y are the gradient magnitudes of Sobel operators in x and y directions, respectively.

3) Adaptive Smoothing: The fingerprint obtained after background removal is adaptively smoothed using guidance from the local orientation. The purpose of this processing is to eliminate most fine details such as islands and pores. We perform uniform smoothing along the local ridge orientation and Gaussian smoothing normal to it. The kernel of the smoothing filter is the normalized product of a 5×1 uniform kernel and a 1×3 Gaussian kernel of $\mu = 0$ and $\sigma = 1$. Possible orientations of the smoothing filters are discretized into 16 values. An appropriate filter is selected according to the local orientation and applied to each pixel.

4) Adaptive Binarization and Thinning: The smoothed fingerprint is split into 16×16 blocks with 8 pixels overlap along each dimension. For each block, we perform histogram equalization, and binarize the block by

$$I_b(i, j) = \begin{cases} 1 & \text{if } I_h(i, j) > \delta_b \\ 0 & \text{otherwise} \end{cases} \quad (2.15)$$

where I_h is the gray-scale value of pixel (i, j) in the histogram equalized block, I_b is the binary value of pixel (i, j) in the binary image, and δ_b is the threshold for binarization. Thinned ridges are obtained by thinning the binary image.

5) Selection of Potential Minutiae: The initial potential minutiae are selected by CN at each pixel in the thinned image, which is defined in Table 2.1. Generally, the initial potential minutiae are very noisy because of binarization, thinning and error in estimating local orientation. Two simple criteria we use to filter the initial potential minutiae are:

- a) In a small local area, if an endpoint and a bifurcation are chosen as the initial potential minutiae, then ignore both of them;

- b) In a small local area, if more than one endpoint or one bifurcation are chosen as the initial potential minutiae, then ignore all these minutiae.

The result is a relatively good set of potential minutiae.

6) Adaptive Application of Templates: At this step, the learned templates are adaptively applied to the potential minutia locations obtained above. Suppose the local orientation at a potential minutia location is θ , then we rotate the learned template by θ and compute the correlation using (2.1) and (2.2). In order to compensate the error in estimating the local orientation, the correlations of the fingerprint are computed with five templates which are the learned templates rotated with θ , $\theta \pm 5^\circ$ and $\theta \pm 10^\circ$. The largest of these values is taken as the correlation at this location. Let

- a) The number of potential minutiae in an image is N_a ;
- b) The correlation of potential minutiae are $V = \{ v_i \}$, where $i = 1, 2, 3 \dots N_a$;
- c) The mean and the standard deviation of V are μ_v and σ_v , respectively;
- d) k_r is a constant for adjusting the threshold of rejecting false minutiae and it could be positive or negative.

If $v_i > (\mu_v + k_r \times \sigma_v)$, we choose the i th potential minutia as a true minutia, otherwise it is a false minutia.

2.4 EXPERIMENTS

2.4.1 Database

We use NIST special fingerprint database 4 (NIST-4) [115] in our experiments. Since fingerprint images in NIST-4 are captured by an ink based method, most fingerprints are of bad quality. We choose 400 pairs of reasonable quality from the first 1000 pairs of fingerprints in

NIST-4 (f0001_01~f01000_07) to test our minutiae extraction approach. These fingerprints are chosen visually based on the size of overlapped areas, the number of scars, translation, rotation and scale between two fingerprints. The size of these images is 480×512 pixels with the resolution of 500 DPI. Sample fingerprints in NIST-4 are shown in Figure 2.4.

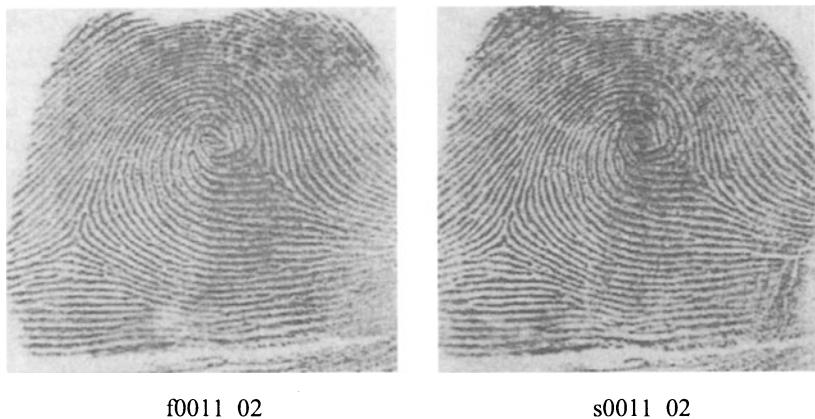


Figure 2.4. Sample fingerprints in NIST-4.

2.4.2 Learned Templates

Training data is manually obtained from 30 fingerprints based on the quality and the location of the minutiae. Totally, there are 85 endpoints and 86 bifurcations. Figure 2.5 shows 5 examples of the training data for endpoints and bifurcations. Figure 2.6 shows the binarized examples in Figure 2.5. The templates for endpoint and bifurcation are learned from these binarized examples. Figure 2.7 shows the learned templates that are used to extract minutiae. Note that, in order to show the structure of the templates clearly, the templates are normalized such that the minimum and maximum values map to black (0) and white (1), respectively.

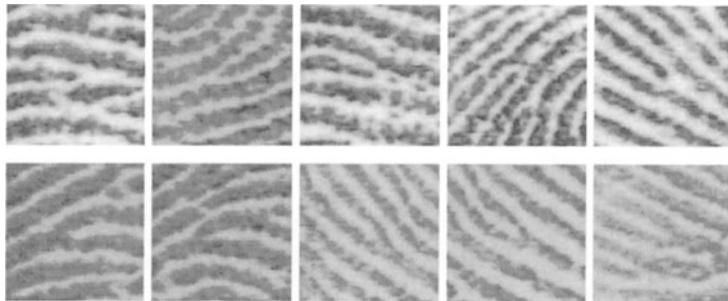


Figure 2.5. Examples of training data: endpoint (first row) and bifurcation (second row).



Figure 2.6. Binarized examples of training data in Figure 2.5.

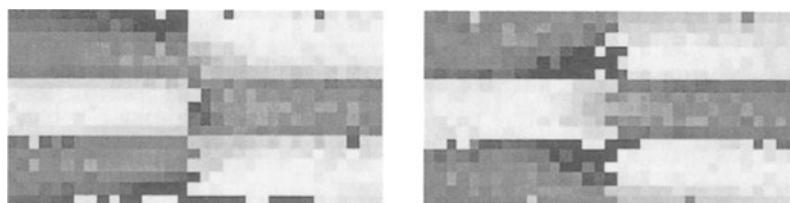


Figure 2.7. Learned templates: endpoint (left) and bifurcation (right).

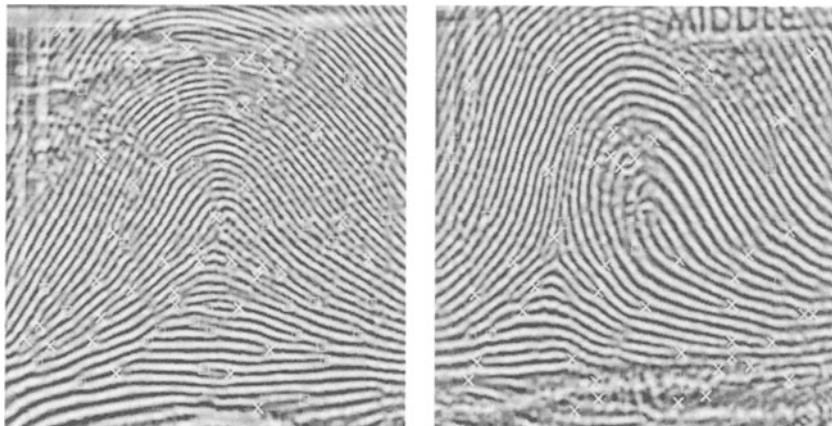


Figure 2.8. Examples of minutiae extraction (\square : endpoint, \times : bifurcation).

2.4.3 Results

Figure 2.8 shows two examples of minutiae extraction results.

- **Evaluation by Goodness Value:** Suppose $M_e = \{e_i, i = 1, 2, 3 \dots n\}$ is the set of n minutiae extracted by a feature extraction algorithm and $M_g = \{g_j, j = 1, 2, 3 \dots m\}$ is the set of m minutiae extracted by an expert in a fingerprint. We define the following terms:

- *Matched* minutiae: if minutia e_i is located in an uncertainty region centered around minutia g_j , e_i and g_j are matched minutiae;
- *Occluded* minutia: if minutia g_j is not in an uncertainty region of any minutia e_i , then g_j is an occluded minutia;
- *Clutter* minutia: if e_i is not in an uncertainty region of any minutia g_j , then e_i is a clutter minutia.

In our experiments, the size of the uncertainty region is 8×8 . Goodness Value (GV) of extracted feature is defined as:

$$GV = \frac{n_m}{n_m + n_o + n_c} \quad (2.16)$$

where n_m , n_o and n_c are the number of matched, occluded and clutter minutiae, respectively. The rationale behind GV is that it focuses on the number of features that can actually match in the presence of noise and uncertainty.

Figure 2.9 shows Goodness Value of fifteen fingerprints. From this figure, we find that the learned templates work better than the fixed templates described in [12]. For example, the mean of GV on these fifteen images is 0.66 for the learned templates, and 0.57 for the fixed templates, which amounts to an improvement of 15.7%.

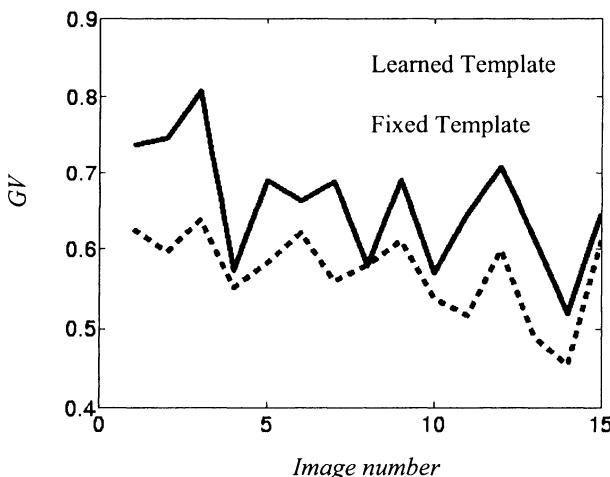


Figure 2.9. Goodness Value of fifteen fingerprints.

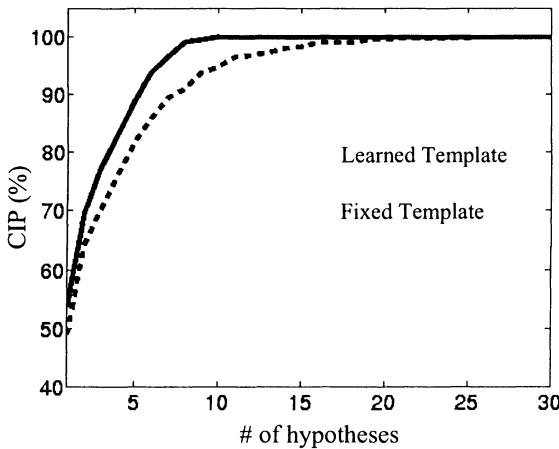


Figure 2.10. Comparison of experimental results.

- **Evaluation by indexing performance:** A query fingerprint, which has a corresponding fingerprint in the database, is said to be correctly indexed if it has enough corresponding triangles in the model database and the correct corresponding fingerprint appears in a short list of hypotheses obtained by the indexing approach. We define Correct Index Power (CIP) as:

$$CIP = \frac{N_{ci}}{N_d} \times 100\% \quad (2.17)$$

where N_{ci} is the number of correctly indexed fingerprints and N_d is the number of images in the database. In our experiments, the total number of images in the database, $N_d = 400$.

Figure 2.10 shows the comparison of CIP for fixed and learned templates. We observe that the performance of the learned templates is better than that of the fixed templates [12]. CIP for the top 1 hypothesis increases by 2.8%, and by 6.5% and 5.2% when we consider the top 5 and top 10 hypotheses, respectively. Using the fixed templates, the CIP

reaches 100% only when we consider the top 26 hypotheses, while for learned templates, we only need to consider top 10 hypotheses.

2.5 CONCLUSIONS

We have presented a new and effective fingerprint minutiae extraction approach, which is based on using learned templates. Using this approach, we can avoid the complex preprocessing and postprocessing algorithms, which are necessary in most minutiae extraction approaches. The performance of the approach is evaluated by the Goodness Value on typical images and the indexing performance of an identification system. Experimental results show that our minutiae extraction approach is capable of improving both the Goodness Value and the indexing performance. Compared with a fixed templates based minutiae extraction approach, our approach improves CIP of the top 5 and top 10 hypotheses by 6.5% and 5.2% for fingerprints in NIST-4 database.

Chapter 3

FINGERPRINT INDEXING

3.1 INTRODUCTION

In this Chapter we present a model-based approach, which efficiently retrieves correct hypotheses using novel features of triangles formed by the triplets of minutiae as the basic representation unit. The triangle features that we use are its angles, handedness, type, direction and maximum side. Geometric constraints based on other characteristics of minutiae are used to eliminate false correspondences. Experimental results on NIST special database 4 show that our indexing approach efficiently narrows down the number of candidate hypotheses in the presence of translation, rotation, scale, shear, occlusion and clutter. We also perform scientific experiments to compare the performance of our approach with another prominent indexing approach and show that the performance of our approach is much better.

There are three kinds of approaches to solve the fingerprint identification problem:

- 1) Repeat the verification procedure for each fingerprint in the database and select the best match;

- 2) Fingerprint classification followed by verification;
- 3) Fingerprint indexing followed by verification.

Figure 3.1 shows the block diagram of these approaches. Detailed explanation of the difference among these approaches is given in Chapter 6. In this Chapter, we are focusing on the indexing problem.

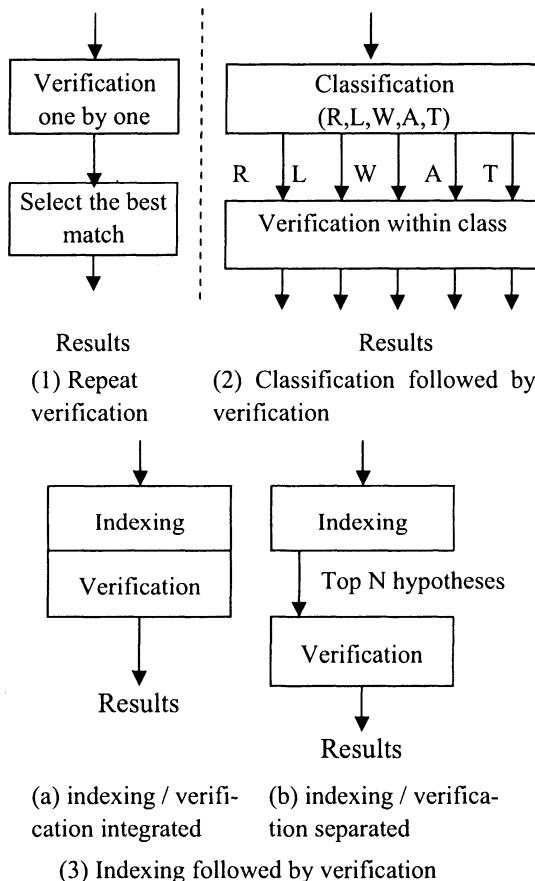


Figure 3.1. Block diagram of different approaches to solve identification problem.

3.2 RELATED RESEARCH AND CONTRIBUTIONS

3.2.1 Related Research

A prominent approach for fingerprint identification is by Germain et al. [33], which integrates the indexing and verification in their approach (Figure 3.1(3)(a)). They use the triplets of minutiae in their identification procedure. The features they use as the indexing components are: the length of each side, the angles that the ridges make with respect to the X-axis of the reference frame, and the ridge count between each pair of vertices. The problems with their approach are:

- a) the length changes are not insignificant under scale and shear;
- b) the ridge angles change greatly with different quality images of the same finger;
- c) uncertainty of minutiae locations is *not* modeled explicitly.

As a result, large size bins have to be used to handle distortions, which increases the probability of collisions and degrades the performance of their algorithm.

Our technique follows their work in that we also use the triplets of minutiae. However, the features that we use are quite different from theirs [16]. The features that we use are: triangle's angles, handedness, type, direction, and maximum side. These features are different, new and more robust than the features used by Germain et al. Table 3.1 shows the substantive differences between these two approaches.

Table 3.1. Comparisons between Germain et al.'s [33] and our approach.

Characteristics of the approach	Germain et al.'s approach	Bhanu and Tan's approach	Comments
Indexing components	Length of each side	Maximum length of three sides	Max. side is less sensitive to shear
	Three angles at triangle's vertices that ridges make with respect to the X-axis of the reference frame (local orientation)	Median and minimum angles of the triangles formed by the triplets of minutiae	Our angles are invariant to translation, rotation, and scale, and robust to shear. The errors in estimating local orientation angles in Germain et al.'s approach could be significant
	Ridge counts between every 2 vertices of a triangle	Triangle's orientation, type, and direction	Ridge count is sensitive to image quality
Indexing scheme	Flash	Hashing	
Geometric constraints	None	Yes	Constraints limit the size of the search space in our approach
Equivalent scale, bin size and data distortion	No explicit model, quantization for three sides is used	Only maximum side is affected by quantization	Germain et al. can not effectively deal with significant distortion by quantization alone
Shear	No explicit model, quantization is used	Threshold for angles	Shear not only affects side length, but also local orientation
Image database size	97492	4000 (NIST-4)	Germain et al.'s database is proprietary, NIST-4 is public
# of test images	657 (<1% of database)	2000 (100% of NIST-4) (This Chapter)	Germain et al. do not mention anything about the image quality they used in experiments.

3.2.2 Contributions

Our major contributions in this work are:

- An indexing algorithm, based on novel features formed by the triplets of minutiae, and associated performance analysis are presented;
- The indexing performance is demonstrated in a principled manner by using triplets as the basic representation unit;
- Unlike the previously published research (Table 3.2), where ad hoc personalized criteria, partial data or handcrafted pre-processing are used for the selection of images to demonstrate the results, in this Chapter the entire NIST-4 database is processed and analyzed in an automated manner in a black-box approach²;
- Comparisons of the performance of our approach with Germain et al.'s approach [33] are carried out, which show that our approach has performed better for the NIST-4 database.

² Black-box approach means no specific data tuning within the black-box where the algorithm processes the entire input image.

Table 3.2 NIST-4 database used in published fingerprint recognition research.

Approach	Technique	Info. on images	# of Classes	Comments
Jain et al. 1999 [45]	classification	1972 (28 rejected) training and 1965 (35 rejected) testing	4 or 5	No specific information on rejected images
Cappelli et al. 1999 [18]	classification	1204 pairs selected per occurrence of 5 classes	5	Compare continuous and exclusive classifications
Halici et al. 1996 [36]	classification	The first 500 pairs: 500 training and 500 testing	At most 15	Results depend on the number of classes and the size of search space
Kovacs-Vajna 2000 [60]	verification	Manually rejected about 6%	N/A	No specific information on rejected images
Senior 2001 [94]	classification	First 1000 pairs training and second 1000 pairs testing	4	Shows test results for the second 1000 pairs only
Karu and Jain 1996 [57]	classification	2000 pairs	4 or 5	40 pixels along the borders of the direction image are removed
This work	Indexing	2000 test images are completely processed without any handcrafting of the data	N/A	Rank ordered top N hypotheses are considered as a possible match to a query fingerprint

3.3 TECHNICAL APPROACH

Our system for fingerprint identification is composed of two stages: an off-line stage and an on-line stage. The model database and indexing structure are constructed during the off-line stage, and identification is carried out during the on-line stage. During the off-line stage, fingerprints in the database are processed one-by-one by minutiae extraction procedure, which is introduced in Chapter 2, to construct model database. During the on-line stage, the query image is processed by the same procedure to extract minutiae. Indexing components are derived from the triplets of minutiae locations and used to map the points in the feature space to the points in the indexing space. The potential correspondences between the query image and images in the database are searched in a local area in the indexing space. An indexing score is computed based on the number of triangle correspon-

dences and candidate hypotheses are generated. The top N ranked hypotheses are the result of our indexing algorithm.

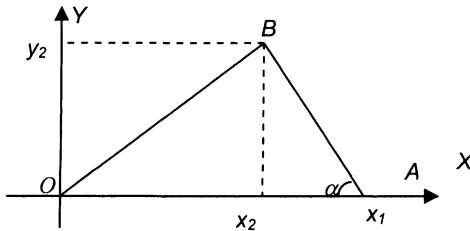


Figure 3.2. Illustration of variables.

3.3.1 Indexing Components

Figure 3.2 shows a triangle. Without loss of generality, we assume that one vertex, O , of the triangle is $(0, 0)$, and it does not change under distortions. Since distance is invariant under translation and rotation and relatively invariant under scale, and angles are defined in terms of the ratio of distance, it can be proved that angles are invariant under these transformations. However, in fingerprint recognition, because of the uncertainty of minutiae locations, which is associated with feature extraction and shear, the location of each minutia translates in a small local area independently in a random manner. Suppose the locations of points A and B are $(x_1, 0)$ and (x_2, y_2) , $x_1 > 0$, $y_2 > 0$ and $x_2 \in (-\infty, +\infty)$. Because of the uncertainty of minutiae locations, A and B move to $A' (x_1 + \Delta x_1, 0)$ and $B' (x_2 + \Delta x_2, y_2 + \Delta y_2)$, respectively, and α changes to $\alpha + \Delta\alpha$. Then

$$\tan \Delta\alpha = \frac{(x_1 - x_2)\Delta y_2 - y_2(\Delta x_1 - \Delta x_2)}{(x_1 - x_2)^2 + (x_1 - x_2)(\Delta x_1 - \Delta x_2) + y_2^2 + y_2\Delta y_2} \quad (3.1)$$

Suppose Δx_1 , Δx_2 , and Δy_2 are independent, and $-4 \leq \Delta x_i, \Delta y_2 \leq 4$, $i = 1, 2$, and Δx_i and Δy_2 are all integers, then

$$g(x_1, x_2, y_2) = E\{\Delta\alpha\} \approx \sum_{-4}^4 \sum_{-4}^4 \sum_{-4}^4 (\tan \Delta\alpha \times p(\Delta x_1) p(\Delta x_2) p(\Delta y_2)) \quad (3.2)$$

Suppose $p(\Delta x_1)$, $p(\Delta x_2)$ and $p(\Delta y_2)$ are discrete uniform distributions in $[-4, +4]$. Let $0 < x_1, y_2, |x_2| < L$, where L is the maximum value of these variables in the image. We compute $g(x_1, x_2, y_2)$ at each point (x_1, x_2, y_2) based on whether α is the minimum, median or maximum angle in the triangle. Notice that, if $\alpha_{min} < \delta_\alpha$ or $\tau < \delta_\tau$, then the uncertainty of minutiae locations may have more effect on α_{min} and α_{med} , so we do not use these triangles in the model-base, where τ is the minimum length of the sides in a triangle.

Table 3.3. Expectation of the percentage of angle changes less than various thresholds.

Angle's Type	Angle Change Threshold					
	$\pm 1^\circ$	$\pm 2^\circ$	$\pm 3^\circ$	$\pm 4^\circ$	$\pm 5^\circ$	$\pm 6^\circ$
α_{min}	51.6	93.2	98.5	99.6	99.9	100.0
α_{med}	56.6	87.3	94.5	97.3	98.7	99.4
α_{max}	1.0	67.7	87.3	94.2	97.2	98.7

From Table 3.3, we observe:

- 1) The minimum and the median angles α_{min} and α_{med} are more robust than the maximum angle α_{max} and they can be used to find the correspondences;
- 2) 2° - 4° change in α_{min} and α_{med} can accommodate the uncertainty of most distortions and keep the size of the search space as small as possible. Using other distributions for $p(\Delta x_1)$, $p(\Delta x_2)$ and $p(\Delta y_2)$, we find the results similar to that in Table 3.3.

Thus, minimum angle and median angle in a triangle formed by the triplets of minutiae can be taken as components of the index to construct a model database for fingerprint identification.

The features we use to find potential corresponding triangles are defined as:

- **Angles α_{min} and α_{med} :** Suppose α_i are three angles in the triangle, $i = 1, 2, 3$. Let $\alpha_{max} = \max\{\alpha_i\}$, $\alpha_{min} = \min\{\alpha_i\}$, $\alpha_{med} = 180^\circ - \alpha_{max} - \alpha_{min}$, then the label of the triplets in this triangle is: if the minutia is the vertex of angle α_{max} , we label this point as P_1 ; if the minutia is the vertex of angle α_{min} , we label it as P_2 ; the last minutia is labeled as P_3 . Figure 3.3 shows an example of this definition.

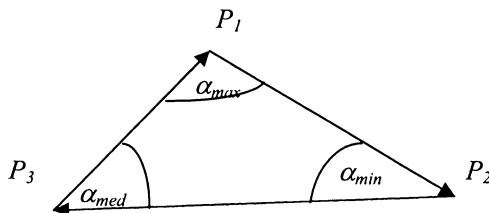


Figure 3.3. Definition of feature points' labels.

- **Triangle Handedness ϕ :** Let $Z_i = x_i + jy_i$ be the complex number ($j = \sqrt{-1}$) corresponding to the coordinates (x_i, y_i) of point P_i , $i = 1, 2, 3$. Define $Z_{21} = Z_2 - Z_1$, $Z_{32} = Z_3 - Z_2$, and $Z_{13} = Z_1 - Z_3$. Let $\phi = \text{sign}(Z_{21} \times Z_{32})$, where sign is the signum function and \times is the cross product of two complex numbers. Figure 3.4 shows two triangles that have the same α_{min} and α_{med} but different ϕ .
- **Triangle Type γ** — Each minutia is either an endpoint or a bifurcation, we define triangle type based on the types of minutiae that form the triangle. Let $\gamma = 4\gamma_1 + 2\gamma_2 + \gamma_3$, where γ_i is the feature type of point P_i , $i = 1, 2, 3$. If point P_i is an endpoint, $\gamma_i = 1$, else $\gamma_i = 0$. $0 \leq \gamma \leq 7$.

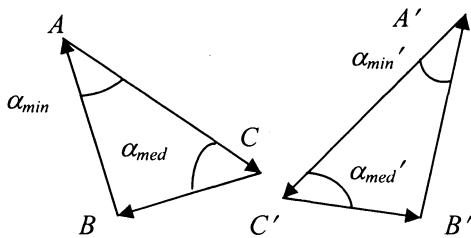


Figure 3.4. $\Delta_{A'BC'}$ and Δ_{ABC} have the same angles but different handedness.

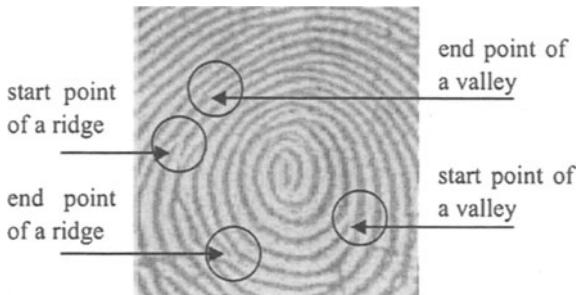


Figure 3.5. Examples of minutiae with different v values.

- **Triangle Direction η :** Search the minutia from top to bottom and left to right in the fingerprint, if the minutia is the start point of a ridge or valley, then $v = 1$, else $v = 0$. η is the combination of v_i , v_i is the v value of point P_i , $i = 1, 2, 3$. Figure 3.5 shows examples of minutiae with different v value.
- **Maximum Side λ :** Let $\lambda = \max\{L_i\}$, where $L_1 = |Z_{21}|$, $L_2 = |Z_{32}|$, and $L_3 = |Z_{13}|$.

3.3.2 Geometric Constraints

Geometric constraints are used to reduce the number of false correspondences obtained from querying the lookup table by the index.

- **Relative local orientation at mid points** — Let points P_{21} , P_{32} , and P_{13} be the midpoint of line P_2P_1 , P_3P_2 and P_1P_3 , respectively, and point P_{123} be the centroid of the triangle $\Delta P_1P_2P_3$. Let $\varphi_{21} = \psi_{21} - \psi_{123}$, $\varphi_{32} = \psi_{32} - \psi_{123}$, and $\varphi_{13} = \psi_{13} - \psi_{123}$, where ψ_{21} , ψ_{32} , ψ_{13} and ψ_{123} are the local orientations in the image at points P_{21} , P_{32} , P_{13} and P_{123} , respectively. We assume that relative local orientations φ_{21} , φ_{32} , and φ_{13} will not change much in different impressions. So, $|\varphi - \varphi'| < \delta_m$, where φ and φ' are φ_{21} , φ_{32} or φ_{13} in two different impressions.
- **Relative local orientation at vertices** — Let ψ_i be the local orientation of point P_i , and $\omega_i = \psi_i - \psi_{123}$, we have $|\omega - \omega'| < \delta_l$, where $i = 1, 2, 3$, and ω and ω' are ω_1 , ω_2 or ω_3 in two different impressions of the same finger.
- **Relative translation** — Let $Z_c = (Z_1 + Z_2 + Z_3) / 3$, we have $|Z - Z'| < \delta_t$, where Z and Z' are the Z_c in two different impressions of the same finger. $|Z - Z'|$ is the translation between the centroids of these two triangles.
- **Relative rotation** — Let $\theta_{21} = \text{angle}(Z_{21})$, $\theta_{32} = \text{angle}(Z_{32})$, and $\theta_{13} = \text{angle}(Z_{13})$, where $\text{angle}(Z)$ is the phase angle of Z . Let $|\theta - \theta'| < \delta_r$, where θ and θ' are θ_{21} , θ_{32} or θ_{13} in two different impressions of the same finger.

3.3.3 Indexing Score

Suppose

- a) I is the query image and I_i are the images in the database, $i = 1, 2, \dots, N_d$, where N_d is the number of images in the database;
- b) M and M_i are the sets of minutiae in I and I_i , respectively;
- c) m is a minutia, and $m \in (M \cap M_i)$;
- d) N_i is the number of matched triangles between I and I_i , $C_3^{n_i-1} \leq N_i \leq C_3^{n_i}$, and n_i is an integer, which is the number of potential corresponding minutiae in each image;
- e) r is the number of triangles in I_i which include m .

Then, we can compute the posterior probability $PM_i = P\{I_i = I | m \in (M \cap M_i)\} = c \cdot r$, where c is a constant factor that makes PM_i to be the correct posterior probability. We sort PM_i for each m , and find the n_i largest probabilities, suppose they are p_k , where $k = 1, 2, \dots, n_i$. We define the index score of image I_i as:

$$S_i = \sum_{k=1}^{n_i} p_k \quad (3.3)$$

3.3.4 Algorithms

Figure 3.6 shows the algorithms used in our experiments.

- (1) **For** $i = 1$ to N , **do** 2 and 3
- (2) Compute local orientation and extract minutiae locations in image I_i using feature extraction procedure.
- (3) **For** each triplets in image I_i , compute α_{min} , α_{med} , ϕ , γ , η , and λ for the triangle formed by the triplets, and add this model together with the information required by the constraints for this model into the model database.

(a) Algorithm for constructing the model database (off-line stage)

Suppose α_1 and α_2 are thresholds for α_{min} and α_{med} to deal with the changes in these angles.

- (1) Compute local orientation and extract minutiae locations from the test image I .
- (2) **For** each triplet in image I , **do** 3 and 4.
- (3) Compute α_{min} , α_{med} , ϕ , γ , η , and λ for the triangle formed by the triplet.
- (4) **For** $k_1 = -T_{\alpha_1}$ to $+T_{\alpha_1}$ **step** 1, **do**
For $k_2 = -T_{\alpha_2}$ to $+T_{\alpha_2}$ **step** 1, **do**
 - (4.1) Search the index space using $(\alpha_{min} + k_1)$, $(\alpha_{med} + k_2)$, ϕ , γ , η , and λ as the elements of index.
 - (4.2) **If** the triangle satisfies the subsequent geometric constraints, **then** take it as a successful correspondence of the triangle in the database.**End loop** for k_1
End loop for k_2
- (5) Suppose there are $\sum_{i=1}^{N_d} M_i$ corresponding triangles, where M_i triangles belong to the same image I_i , $i = 1, 2, \dots, N_d$, and N_d is the number of images in the database. Let $M_{max} = \max\{M_i\}$.
- (6) **If** $M_{max} < T$, **then** reject the test image, where T is the threshold for rejecting a test image (see Section 3.2.5 for estimation of T), Otherwise do following steps.
- (7) Compute index score S_i based on M_i for I_i .
- (8) Sort S_i in a descending order, output top N hypotheses.

(b) Algorithm for run-time identification (on-line stage)

Figure 3.6. Algorithms used in our experiments.

3.3.5 Probability of False Indexing

Suppose:

- a) S is the size of the index space;
- b) f_k is the number of triangles in the model database for image I_k , and these triangles are uniformly distributed in the indexing space;
- c) b is the search redundancy for each triangle in the query image;
- d) v_k is the number of corresponding triangles between image I and I_k ;
- e) f_t is the number of triangles for the query image [70].

Then the value of v_k that is greater than a threshold T can be approximated by the Poisson distribution:

$$P\{v_k > T\} \approx 1 - e^{-\xi} \sum_{i=0}^T (\xi^i / i!) \quad (3.4)$$

where $\xi = f_t \times p_1$, $p_1 \approx b p_0$, $p_0 = f_k / S$. In our approach, in a triangle if $\alpha_{min} < \delta_\alpha$ or $\tau < \delta_\tau$, where τ is the minimum side of the triangle, then we do not use this triangle to build the model. We use 0.5° as the bin size for angles α_{min} and α_{med} , δ_λ for λ , and we search the indexing space with the uncertainty of $\pm 2^\circ$. Hence, $\xi \approx 15.885$. Figure 3.7 shows the curve of $P\{v_k > T\}$ with respect to T . When $T = 25$, $P\{v_k > T\} = 0.0121$. That is, if there is no image in the database corresponding to the test image, the probability of finding 25 corresponding triangles between the test image and any of the images in the database is about 0.0121. We can use $T = 25$ as the threshold to reject a test image which has no corresponding image in the database. Note that while the triangles are not uniformly distributed in the model database, since we apply geometric constraints, T can be less than 25.

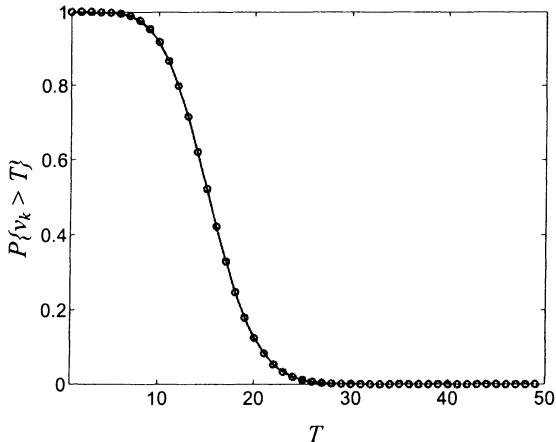


Figure 3.7. $P\{v_k > T\}$ with respect to T .

3.4 EXPERIMENTS

3.4.1 Database

We use NIST special fingerprint database 4 (NIST-4) [115] in our experiments, which is a publicly available fingerprint database. Since the fingerprints in NIST-4 are collected by ink-based method, a large portion of fingerprints are of poor quality and contains certain other objects, such as characters and handwritten lines. The size of the fingerprint images is 480×512 pixels with the resolution of 500 DPI. NIST-4 contains 2000 pairs of fingerprints. Each pair is a different impression of the same finger. One pair of fingerprints is shown in Figure 3.8. Parameters we use are: $\delta_\alpha = 10^\circ$, $\delta_r = 40$ pixels, $\delta_\lambda = 20$ pixels, $\delta_m = 30^\circ$, $\delta_l = 30^\circ$, $\delta_r = 30^\circ$, $\delta_t = 50$ pixels, $T_{\alpha l} = 2^\circ$, $T_{\alpha 2} = 2^\circ$, and $T = 20$.

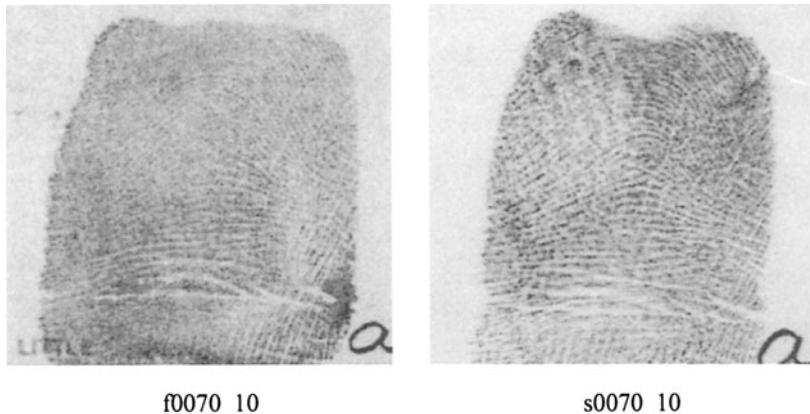


Figure 3.8. Sample fingerprints in NIST-4 database.

3.4.2 Performance Evaluation Measures for Indexing

False Positive Rate (*FPR*) and False Negative Rate (*FNR*) are used to evaluate the performance of a verification algorithm [113]. However, the goal of the indexing method presented in this Chapter is to narrow down the number of hypotheses which need to be considered for subsequent verification. The output of an indexing algorithm is the set of top N hypotheses. If the corresponding fingerprint is in the list of top N hypotheses, we should take the indexing result as a correct result. Hence, *FPR* and *FNR* are not suitable for evaluating the results of an indexing algorithm. We define Correct Index Power (CIP) as the performance evaluation measures for indexing:

$$CIP = \frac{N_{ci}}{N_d} \times 100\% \quad (3.5)$$

where N_{ci} is the number of correctly indexed images, N_d is the number of images in the database.

3.4.3 Indexing Results

Figure 3.9 shows the number of corresponding triangles for each query fingerprint. Note that 32 images do not have any corresponding triangles. That is why *CIP* can not reach 100% as the number of hypotheses increases. Figure 3.10 shows the distribution of the number of corresponding triangles among those 2000 query images on a log scale. Because of bad quality, for some queries the number of corresponding triangles is quite small. Figure 3.11 shows how *CIP* performance varies with the number of corresponding triangles for different values of threshold T . Approximately 10 good features lead to good indexing results.

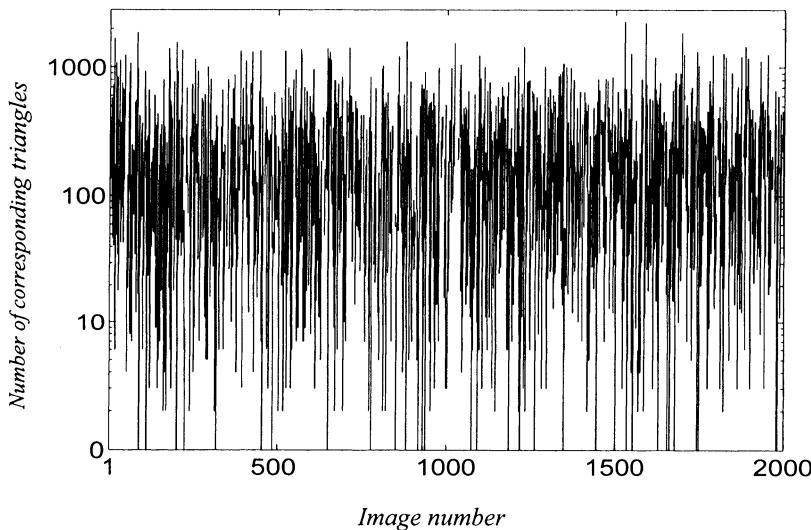


Figure 3.9. Number of corresponding triangles of the 2000 query fingerprints.

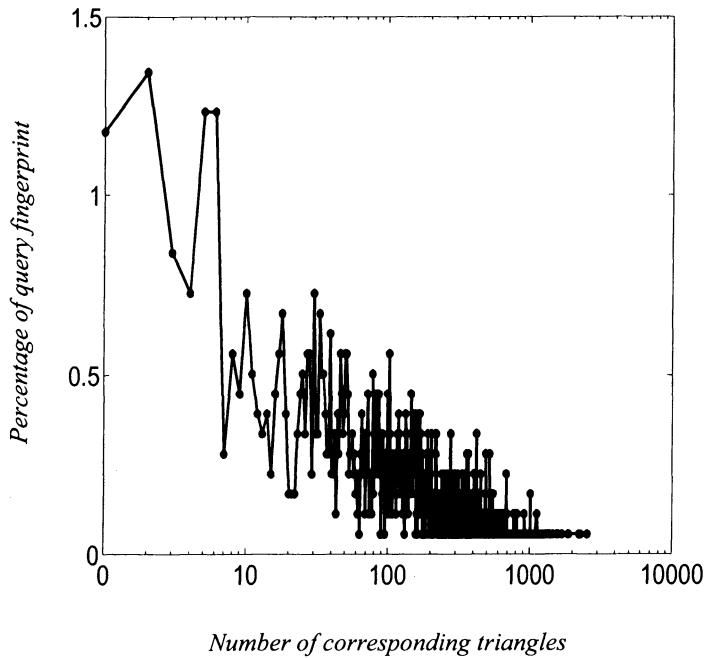


Figure 3.10. Distribution of corresponding triangles among 2000 query fingerprints.

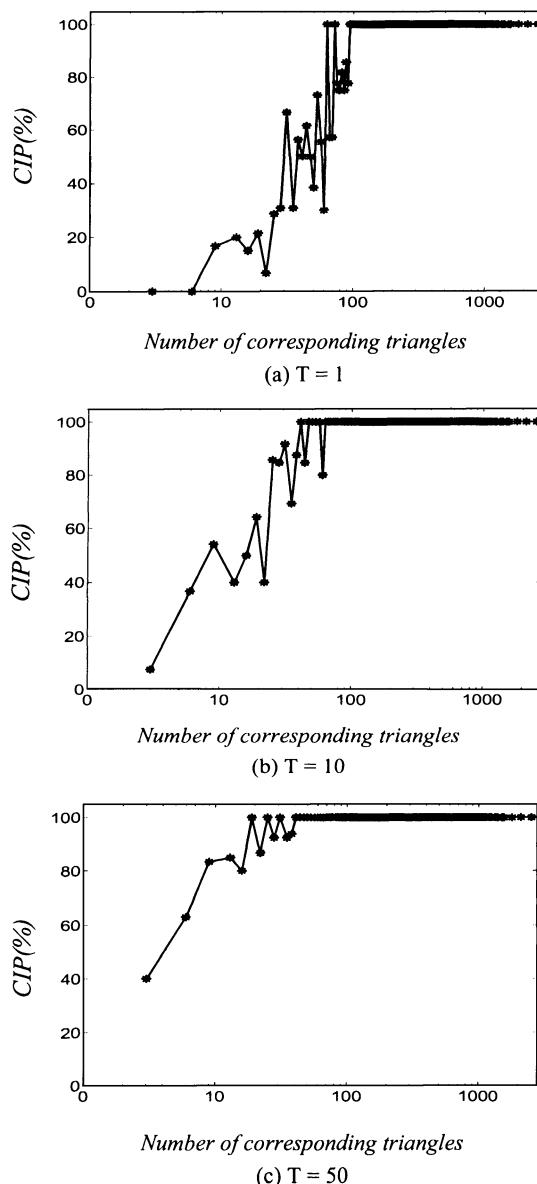


Figure 3.11. CIP performance varies with number of corresponding triangles.

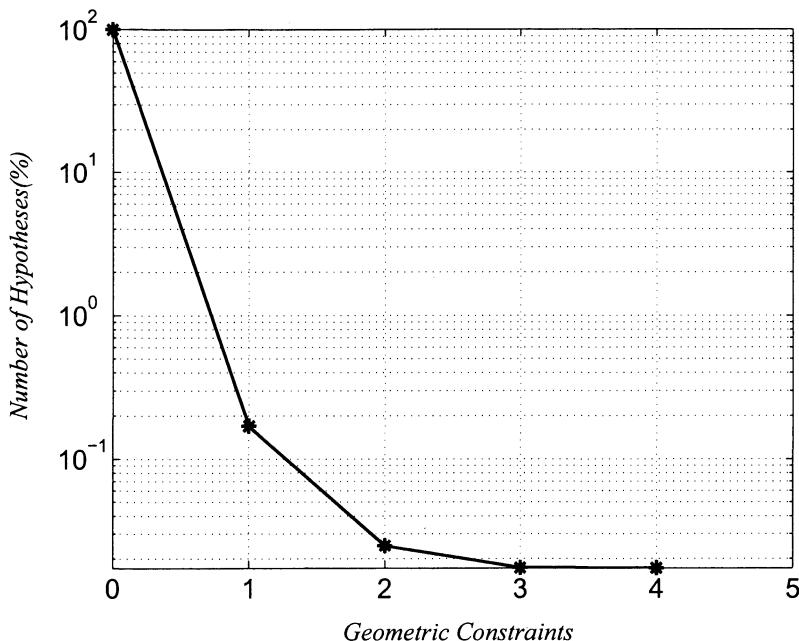


Figure 3.12. Effect of constraints.

3.4.4 Effect of Geometric Constraints

Figure 3.12 shows the effect of geometric constraints in reducing the average percentage of hypotheses that need to be considered for indexing. We observe that four geometric constraints provide a reduction by a factor of 589.487, 6.869, 1.428, and 1.007, sequentially. Note that the order of the constraints is the same as that in Section 3.3.2.

3.4.5 Extrapolation of Indexing Performance

There exists no general theory in the computer vision and pattern recognition field to predict the performance of model-based indexing and matching algorithms under arbitrary transformations and arbitrary size of databases. Initial attempts have been made in [10][11], which takes into consideration of uncertainty in features, occlusion, clutter

and similarity of object models. In the absence of a general theory, we perform extrapolation of the results obtained on NIST-4 database to estimate the scalability of our approach. Let N , the number of hypotheses, be 10% of M , where M is the size of the database, but if $N > 100$, then let $N = 100$, so that the maximum number of hypotheses need to be considered is 100. Figure 3.13 shows the extrapolated performance of our approach on databases of different size, which uses a linear regression model. Figure 3.14 shows the extrapolation with large M . As M increases, the performance will decrease and the 95% confidence interval of the performance will increase. However, these results indicate that a CIP of 50% could be achieved with a short list of 100 hypotheses, which would be only 0.33% of a 30,000-image database, which is really a good performance. This extrapolation from the results of 2000 images to 30,000 needs to be taken with caution. It is dependent on the quality of input images and, as our analysis shows, NIST-4 is a difficult database.

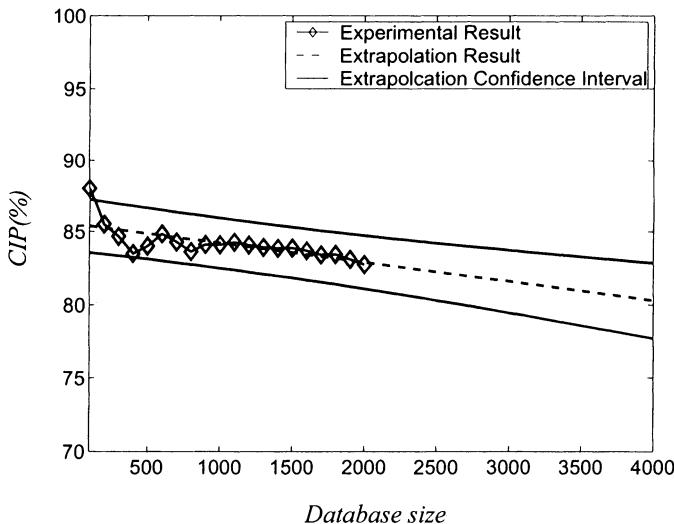


Figure 3.13. Performance on NIST-4.

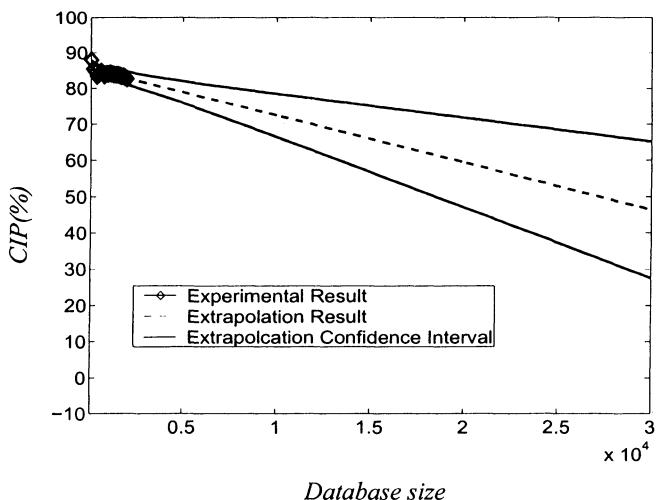


Figure 3.14. Performance with large M .

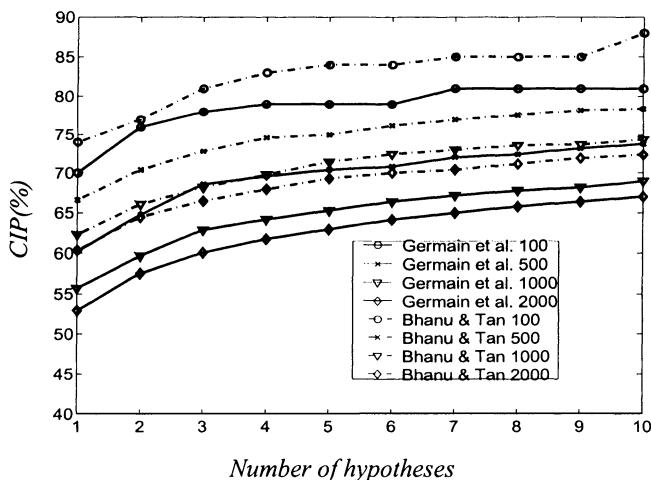


Figure 3.15. Comparison of Germain et al.'s and our approaches on NIST-4 dataset.

3.4.6 Comparison of Approaches

We have done a direct comparison with Germain et al.'s approach. We implemented Germain et al.'s approach and compared the performance of our indexing algorithm with it. Figure 3.15 shows the comparisons of the two approaches on four subsets, first 100, 500, 1000, and 2000 fingerprints, of NIST-4 database. Our approach has performed better than that of Germain et al.'s approach. When the entire NIST-4 is used, although the CIP of Germain et al.'s approach increases from 53.0% to 67.0%, the CIP of our approach increases from 60.4% to 72.4% as the number of hypotheses increases from top 1 (0.05% of the database) to top 10 (0.5% of the database). Figure 3.16 shows that the indexing performance increases as the percentage of the database size increases. When it is 10%, our approach is still better, the CIP of these two approaches are 85.5% and 83.7%, respectively.

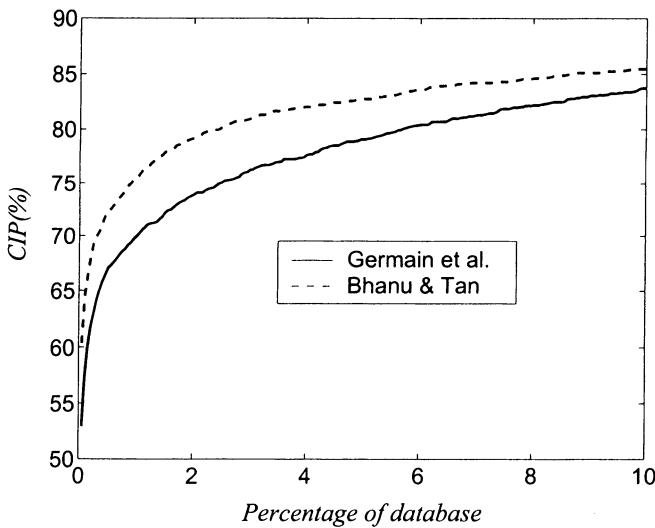


Figure 3.16. Comparison of Germain et al.'s approach and our approach on NIST-4.

A direct comparison with traditional fingerprint classification approaches can not be done for the following reasons: our approach for

indexing is different from classification techniques. They are classifying each testing image into 4 or 5 classes, while we are generating the top N hypotheses for an input fingerprint query. The outputs of these two systems are different. If we take our approach and traditional classification approaches as a filtering step for data reduction before detailed verification, then it is possible to compare the two approaches indirectly. The results of our technique can be evaluated in term of the number of hypotheses that need to be considered for detailed verification. We do not use error/efficiency analysis since it has its own limitations as pointed out by Senior [94]. Figure 3.12 and Figure 3.13 show the percentage of the NIST-4 database that needs to be examined in subsequent verification. Rather than reducing the computation for verification to about 1/3 of the database by a classification approach, our approach reduces the computation to about 10% of the database for the CIP to be 85.5%. So, on the NIST-4 database, our approach is better than traditional classification approach. For a database of 30,000, for our approach, a CIP of 50% could be achieved with only 100 hypotheses (0.33% of the database). The performance of the traditional approach will be quite low, because the number of classes is limited (4 or 5) and quite likely there will be more than 10% misclassification. A thorough comparison of these two kinds of approaches (indexing and classification followed by verification) is presented in Chapter 6. Note that a low value for CIP is not a fatal flaw with our approach for large database where a low miss rate is required, because indexing can produce an ordered list of hypotheses of the entire database, which will, on average, be much more efficient for verification than a blind exhaustive search.

3.5 CONCLUSIONS

Our approach, based on triplets of minutiae, is promising for indexing fingerprints under translation, rotation, scale, shear, occlusion and clutter. Experimental results show that it can greatly reduce the number of candidate hypotheses for further verification. In various comparisons with the prominent indexing approach developed by Germain

et al., our approach has performed better. We have performed the analysis of the entire NIST-4 database in a systematic manner and characterized indexing performance in terms of the number of corresponding triangles. This will allow the comparison of our results by others on the same publicly available database.

Chapter 4

FINGERPRINT MATCHING BY GENETIC ALGORITHMS

4.1 INTRODUCTION

The input to a fingerprint verification/authentication system is a query fingerprint and an identity (ID), and the system verifies whether the ID is consistent with the fingerprint (see Figure 1.6(b)). The output is an answer of yes or no. In this chapter, we are dealing with the verification problem.

Fingerprint matching is still a challenging problem for reliable person authentication because of the complex distortions involved in two impressions of the same finger. In this Chapter, we present a finger-print matching approach based on Genetic Algorithms (GA), which finds the optimal global transformation between two different fingerprints. In order to deal with low quality fingerprint images, which introduce significant occlusion and clutter of minutiae features, we design the fitness function based on the local properties of each triplet of minutiae. The experimental results on National Institute of Standards and Technology fingerprint database, NIST-4, not only show that the proposed approach can achieve good performance even when a large

portion of fingerprints in the database is of poor quality, but also show that the proposed approach is better than another approach, which is based on mean-squared error estimation.

4.2 RELATED RESEARCH AND CONTRIBUTIONS

4.2.1 Related Research

Generally, fingerprint matching algorithms have two steps: 1) align the fingerprints and 2) find the correspondences between two fingerprints. The approach, proposed in Jain et al. [43], is capable of compensating for some of the nonlinear deformations and finding the correspondences. However, since the ridges associated with the minutiae are used to estimate the alignment parameters, the size of the templates has to be large, which takes much memory and computation, otherwise, the alignment will be inaccurate. Jiang and Yau [52] use the local and global structures of minutiae in their approach. The local structure of a minutia describes a rotation and translation invariant feature of the minutia in its neighborhood, and the global structure tries to determine the uniqueness of a fingerprint. The problem with this technique is that it can not compensate for real-world distortions of a 3D elastic finger. These distortions can be considered equivalent to a space variant scale distortion. Furthermore, the weight vector that is associated with each component of the feature vector, such as distances, directions, relative local orientations, etc., has to be empirically determined. Another prominent matching algorithm, which is proposed by Kovacs-Vajna [59], uses triangular matching to deal with the deformations of fingerprints. However, the final results of matching have to be validated by a Dynamic Time Warping (DTW) algorithm. Without DTW for further verification, the result is not acceptable. Tan and Bhanu [104][105] develop a fingerprint identification approach, which is based on the local optimization of the corresponding triangles to perform verification between two fingerprints. This approach is compared with the approach developed in this Chapter.

Besides minutiae, researchers have also used other features for fingerprint matching. Saleh and Adlhami [91] proposed an approach, which transforms fingerprint images into a sequence of points in the Angle-Curvature domain. The matching between a query fingerprint and a template fingerprint is based on the least square error of the Euclidean distance between corresponding points in the Angle-Curvature domain. Jain et al. [48] presented a filter-based algorithm, which uses a bank of Gabor filters to capture both local and global details in a fingerprint as a compact fixed length FingerCode. The authors reported that the FingerCode based system performs better than a state-of-the-art minutiae-based system when the performance requirement of the application system does not demand a very low false acceptance rate.

The combinations of different kind of features are also used in fingerprint matching. Jain et al. [47] present a hybrid matching algorithm that uses both minutiae and texture information. Ceguerra and Koprin-ska [20] propose an approach that uses matched minutiae as the reference axis to generate shape signature of the fingerprint. Shape signature is then used to form a feature vector describing the fingerprint. A Linear Vector Quantization (LVQ) neural network is trained by the feature vectors to match fingerprints. Improved matching results are reported in both approaches.

4.2.2 Contributions

We have developed a Genetic Algorithms (GA) based approach to achieve a globally optimized solution for the transformation between two sets of minutiae extracted from two different fingerprints. The fitness function is based on the local properties of each triplet of minutiae, which include angles, triangle handedness, triangle direction, maximum side, minutiae density and ridges counts. The performance of our approach on the NIST-4 database, which has a large portion of fingerprints of poor quality, shows that our approach can tolerate highly nonlinear deformations. The comparison of the proposed ap-

proach with our work in [104] shows the advantage of GA based verification.

4.3 TECHNICAL APPROACH

4.3.1 Fingerprint Matching Problem

Suppose the sets of minutiae in the template and the query fingerprints are $\{(x_{n,1}, x_{n,2})\}$ and $\{(y_{m,1}, y_{m,2})\}$ respectively, where $n = 1, 2, 3, \dots, N$, $m = 1, 2, 3, \dots, M$. The number of minutiae in the template and the query fingerprints are N and M , respectively. The transformation $Y_i = F(X_i)$ between X_i and Y_i can be simplified as:

$$Y_i = s \cdot R \cdot X_i + T \quad (4.1)$$

where s ($s \approx 1$) is the scale factor, $R = \begin{bmatrix} \cos \theta & -\sin \theta \\ \sin \theta & \cos \theta \end{bmatrix}$, θ is the angle of rotation between two fingerprints, and $T = \begin{bmatrix} t_x \\ t_y \end{bmatrix}$ is the vector of translation.

Thus, the matching problem can be defined as to find the optimized transformation, which can map as many as possible minutiae in the template fingerprint to the minutiae in the query fingerprint.

4.3.2 Selection of an Optimization Technique

We have reviewed many techniques commonly used for optimization to determine their usefulness for fingerprint recognition. The drawbacks of each of these methodologies are as follows:

- **Exhaustive techniques (Random walk, depth first, breadth first, enumerative):** Able to find global maximum but computationally prohibitive because of the size of the search space;

- **Calculus-Based techniques (Gradient methods, solving systems of equations):** No closed form mathematical representation of the objective function is available. Discontinuities and multimodal complexities are present in the objective function;
- **Partial Knowledge techniques (Hill climbing, beam search, best first, branch and bound, dynamic programming, A^{*}):** Hill climbing is plagued by the foothill, plateau, and ridge problems. Beam, best first, and A^{*} search techniques have no available measure of goal distance. Branch and bound requires too many search points while dynamic programming suffers from the curse of dimensionality and is expensive computationally;
- **Knowledge-Based techniques (Production rule systems, heuristic methods):** These systems have a limited domain of rule applicability, tend to be brittle, and are usually difficult to formulate. Further, the visual knowledge required by these systems may not be representable in knowledge-based formats;
- **Hierarchical techniques:** Generally, a coarse resolution is employed first to find a narrow range of the solution. Then a fine resolution search in the narrow range is performed to find the optimal solution. However, just like hill climbing techniques, hierarchical techniques may not always find the optimal solution.

Genetic algorithms are able to overcome the problems mentioned in the above. They search from a population of individuals, which make them ideal candidates for parallel implementation, and much more efficient than exhaustive techniques. Since they use simple rules to generate new individuals, they do not require domain specific knowledge or measures of goal distance.

4.3.3 Optimization Based on GA

Genetic algorithms (GA), introduced by Holland [37], provide an approach to learning that is loosely based on simulated evolution. The search for an appropriate hypothesis begins with a population of initial hypotheses. Members of the current population generate the new gen-

eration by means of selection, crossover and mutation, which are patterned after processes in biological evolution. At each step, hypotheses in the current population are evaluated by a fitness function, with the better fit hypotheses selected probabilistically for generating the next population. A detailed introduction to GA can be found in Back et al. [5].

GA have been widely used for optimization. Johnson [50] considers the design of a switched beam linear array in which two beams with specified shapes are to be produced. GA based optimizations are used for the design task. It is discovered that much better results are obtained by simultaneous, multi-objective optimization based design using GA. Lam et al. [68] propose the stability analysis of fuzzy model-based nonlinear control systems, and the design of nonlinear gains and feedback gains of the nonlinear controller using GA. The solution of the stability conditions is also determined by GA. An application example of stabilizing a cart-pole type inverted pendulum system is given to show the stabilizability of the nonlinear controller. Other researches, which use GA for optimization, can be found in Johnson and Samii [49], Mandal et al. [74], Oliveira and Lorena [77] and Sareni et al. [93].

GA has also been used in applications of object recognition. Bebis et al. [8] use GA to recognize 2D or 3D objects from 2D intensity images. The approach is model-based, while the recognition strategy uses the theory of algebraic functions of views. Tsang [101] presents a GA based technique for searching the best alignment between contours of near-planar objects. The method is more efficient and robust than the dominant point approaches. Similar idea has been used in Toet and Hajaema [111]. Ozcan and Mohan [79] apply GA to the partial shape matching problem. The quality of matching is evaluated by a measure derived from attributed shape grammars. Kawaguchi and Nagao [63] and Zaki et al. [122] also use similar ideas in their systems. GA has also been used in handwriting recognition systems (Cho [17] and Yeuung et al. [121]) and stereo matching of images (Saito and Mori [90]).

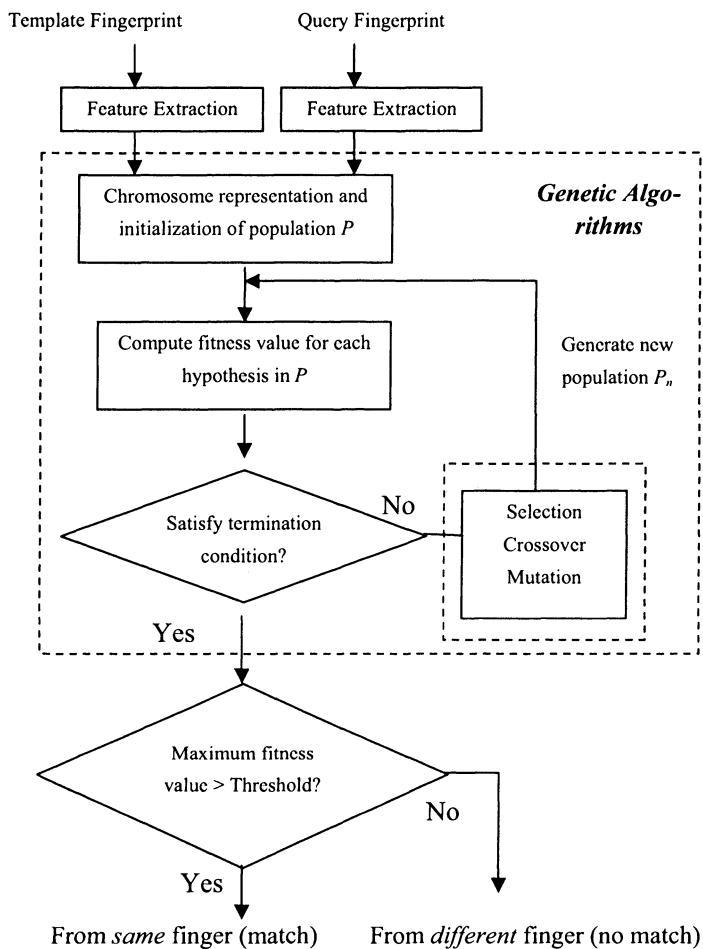


Figure 4.1. Block diagram of GA based approach.

4.3.4 GA Based Fingerprint Recognition

Figure 4.1 shows the block diagram of our approach. First, the minutiae extraction procedure, which is introduced in Chapter 2, is applied to both the template and query fingerprints. The extracted features are the input to GA, which are used to find the optimized parameters and the maximum fitness value. If the maximum fitness value is greater than a given threshold, then the input fingerprint is from the same finger. Otherwise, it is not. In the following, we introduce our GA based approach in detail.

- **Chromosome representation and initialization:** As shown in Section 4.3.1, the parameters that need to be optimized are s , θ , t_x and t_y . According to our experimental results, which are explained in detail in Section 4.4, the range of these parameters are: $0.9 \leq s \leq 1.1$, $-30^\circ \leq \theta \leq 30^\circ$, $-128 \leq t_x \leq 128$, $-128 \leq t_y \leq 128$. The resolutions of these parameters are 0.01 , 1° , 1 pixel and 1 pixel, respectively. Thus, the number of bits to represent s , θ , t_x and t_y are 5, 6, 8 and 8, respectively. The length of chromosome representation is 27 bits. The size of the entire search space is about $2^{27} \approx 1.34 \times 10^8$. The bit string is initialized randomly.
- **Fitness Function:** Fitness function is critical to the performance of GA. In our approach, fitness function is defined by a two-step process. During the first step, the optimized transformation is used to check the global consistency between two sets of minutiae. In the second step, local properties of the minutiae are used to verify the detailed matching.

- 1) **Step 1:** Suppose the optimized transformation is $\hat{F}_{\hat{e}}(\bullet)$, where $\hat{e} = (\hat{s}, \hat{\theta}, \hat{t}_1, \hat{t}_2)$, $\forall j, j = 1, 2, 3, \dots, N$. Let

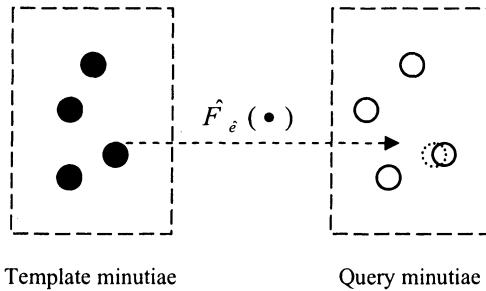


Figure 4.2. Illustration of $\hat{F}_{\hat{e}}(\bullet)$, where $\hat{e} = (\hat{s}, \hat{\theta}, \hat{t}_1, \hat{t}_2)$.

$$d_j = \min_k \left\{ \left| \hat{F}_{\hat{e}} \begin{pmatrix} x_{j,1} \\ x_{j,2} \end{pmatrix} - \begin{pmatrix} y_{k,1} \\ y_{k,2} \end{pmatrix} \right| \right\} \quad (4.2)$$

If d_j is less than a threshold T_d , then we define the points $\begin{bmatrix} x_{j,1} \\ x_{j,2} \end{bmatrix}$ and $\begin{bmatrix} y_{j,1} \\ y_{j,2} \end{bmatrix}$ as potential corresponding points. Figure 4.2 shows the illustration of $\hat{F}_{\hat{e}}(\bullet)$. If n_c , the number of potential corresponding points based on $\hat{F}_{\hat{e}}(\bullet)$, is less than a threshold T_n , then let the fitness value for the transformation $\hat{F}_{\hat{e}}(\bullet)$ be $FV(\hat{F}_{\hat{e}}) = n_c$. In this case, it does not make sense to further evaluate the matching. Otherwise, we check the local properties of the triplets of minutiae in step 2.

2) Step 2: Each noncolinear triplet of potential corresponding points can form a triangle, the local properties associated with the triangle include³:

³ Some figures that explain the triangle's local properties are shown in Section 3.3.1.

- **Angles α_{min} and α_{med} :** Suppose α_i are three angles in the triangle, $i = 1, 2, 3$. Let $\alpha_{max} = \max\{\alpha_i\}$, $\alpha_{min} = \min\{\alpha_i\}$, $\alpha_{med} = 180^\circ - \alpha_{max} - \alpha_{min}$, then the label of the triplets in this triangle is such that if the minutia is the vertex of angle α_{max} , we label this point as P_1 ; if the minutia is the vertex of angle α_{min} , we label it as P_2 ; the last minutia is labeled as P_3 .

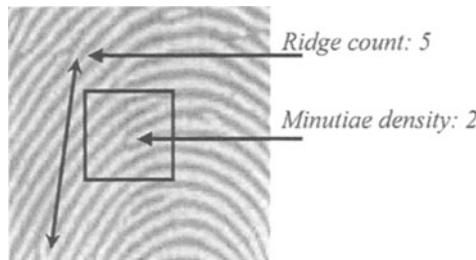


Figure 4.3. Examples of ridge count and minutiae density.

- **Triangle handedness ϕ :** Let $Z_i = x_i + jy_i$ be the complex number ($j = \sqrt{-1}$) corresponding to the coordinates (x_i, y_i) of point P_i , $i = 1, 2, 3$. Define $Z_{21} = Z_2 - Z_1$, $Z_{32} = Z_3 - Z_2$, and $Z_{13} = Z_1 - Z_3$. Let $\phi = \text{sign}(Z_{21} \times Z_{32})$, where sign is the signum function and \times is the cross product of two complex numbers.
- **Triangle Direction η :** Search the minutia from top to bottom and left to right in the fingerprint, if the minutia is the start point of a ridge or valley, then $v = 1$, else $v = 0$. Let $\eta = 4v_1 + 2v_2 + v_3$, where v_i is the v value of point P_i , $i = 1, 2, 3$.
- **Maximum Side λ :** Let $\lambda = \max\{L_i\}$, where $L_1 = |Z_{21}|$, $L_2 = |Z_{32}|$, and $L_3 = |Z_{13}|$.
- **Minutiae Density χ :** In a local area centered at minutiae P_i , if there exists n_χ minutiae, then minutiae density for P_i is $\chi_i = n_\chi$. Minutiae density χ is a vector consisting of all χ_i 's.
- **Ridge Counts ξ :** Let ξ_1 , ξ_2 and ξ_3 be the ridge counts of sides P_1P_2 , P_2P_3 and P_3P_1 , respectively, then ξ is a vector consisting

of all ξ_i 's. Figure 4.3 shows the examples of ridge count and minutiae density.

If two triangles from two different fingerprints satisfy the following criteria, then they are potential corresponding triangles, and the fitness value of the hypothesis $FV(\hat{F}) = n_t$, where n_t is the number of potential corresponding triangles. The criteria are:

$$\begin{aligned} |\alpha'_{\min} - \alpha''_{\min}| &\leq T_{\alpha_{\min}}, |\alpha'_{\text{med}} - \alpha''_{\text{med}}| &\leq T_{\alpha_{\text{med}}} \\ \phi' = \phi'', \eta' = \eta'', |\lambda' - \lambda''| &\leq T_\lambda \\ |\chi'_i - \chi''_i| &\leq T_\chi, i = 1, 2, 3, |\xi'_i - \xi''_i| &\leq T_\xi, i = 1, 2, 3 \end{aligned} \quad (4.3)$$

where $(\alpha'_{\min}, \alpha'_{\text{med}}, \phi', \eta', \lambda', \chi'_i, \xi'_i)$ and $(\alpha''_{\min}, \alpha''_{\text{med}}, \phi'', \eta'', \lambda'', \chi''_i, \xi''_i)$ are the local properties of the triangle in different fingerprints; $T_{\alpha_{\min}}$, $T_{\alpha_{\text{med}}}$, T_λ , T_χ , and T_ξ are thresholds to deal with the local distortions.

Thus, the fitness function is defined as:

$$FV(\hat{F}) = \begin{cases} n_c, & \text{if } n_c < T_n \\ n_t, & \text{if } n_c \geq T_n \end{cases} \quad (4.4)$$

where n_c is the number of potential corresponding points, n_t is the number of potential corresponding triangles, and T_n is the threshold. Generally, the larger the T_n , the longer the evolution of GA takes.

- **Population generation:** Suppose

- a) The size of population P is N_p ;
- b) Hypotheses of the transformation are $\hat{F}_i, i = 1, 2, 3 \dots N_p$;
- c) Crossover rate is p_s ;
- d) Mutation rate is p_m ;
- e) Hypotheses are ordered in the descending order of their fitness values;

- f) The fitness value of a hypothesis \hat{F}_i is $FV(\hat{F}_i)$.

Then, a new generation P_n is generated by

- **Selection:** Probabilistically select the first $p_s \times N_p$ hypotheses from P and add them to P_n .
- **Crossover:** Probabilistically select $\frac{(1 - p_s) \times N_p}{2}$ pairs of hypotheses from P according to $Pr(\hat{F}_i)$. For each pair of hypotheses, generate two children by applying the crossover operator and add them to P_n . The probability $Pr(\hat{F}_i)$ is defined by

$$Pr(\hat{F}_i) = \frac{FV(\hat{F}_i)}{\sum_{j=1}^{N_p} FV(\hat{F}_j)} \quad (4.5)$$

- **Mutation:** Choose $p_m \times N_p$ hypotheses from P with uniform probability. For each hypothesis, invert one randomly selected bit.

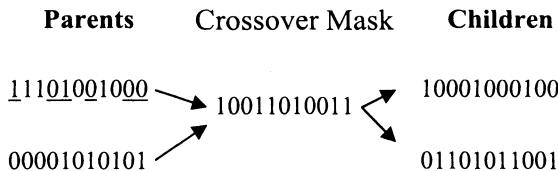


Figure 4.4. An example of uniform crossover operator in GA.

Crossover operators include single-point crossover, two-point crossover, and uniform crossover. In uniform crossover, the crossover mask is generated as a random bit string with each bit chosen at random and independent of the others. Figure 4.4 shows an example of uniform crossover, which is used in our approach.

- **Termination conditions:** Termination conditions decide the length of the learning time and possibly how good the solution is. The termination conditions we use are:
 - 1) Terminate GA if the maximum fitness value does not change in N_t generations;
 - 2) Terminate GA if the fitness value of matching is greater than 100. We have found that a fitness value greater than 100 provides a genuine matching and it is unnecessary to continue the evolution.
- **Computation time reduction:** One problem with GA is that the computation time of the evolution can be long. In order to reduce it, we use the termination condition 2) that is defined above. In addition, during the evolution, the first $p_s \times N_p$ hypotheses are selected and added to the next generation. Only mutation can change the fitness value of these hypotheses. If no mutation happens, it is unnecessary to compute the fitness value again.

4.4 EXPERIMENTS

4.4.1 Database

We use NIST Special Database 4 (NIST-4) [115] in our experiments. Since the fingerprints in NIST-4 are collected by an ink-based method, a large portion of the fingerprints are of poor quality and contain certain other objects, such as characters and handwritten lines. The size of the fingerprint images is 480×512 pixels with a resolution of 500 DPI. NIST-4 contains 2000 pairs of fingerprints. Each pair is a different impression of the same finger. The fingerprint is coded as a f or s followed by 6 numbers, which means the fingerprint image is the first or second impression of certain finger. Some sample fingerprints are shown in Figure 4.5.

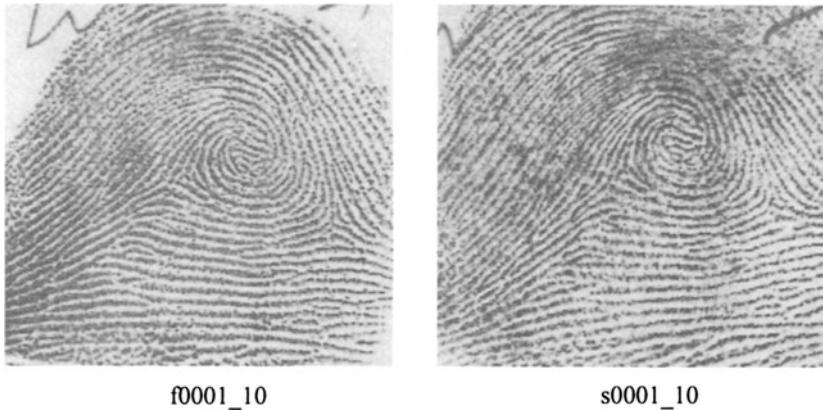


Figure 4.5. Sample fingerprints in NIST-4 database. This pair of fingerprints are quite challenging to match.

4.4.2 Estimation of Parameters for GA

Parameters that need to be optimized are s , θ , t_x and t_y . The data ranges of these parameters are critical to the performance of the approach. On one side, if the data range is too small, we may miss the optimal solution. On the other side, if the data range is too large, GA may take a much longer time to finish. In addition, non-matches can get higher scores. The estimation of the data range is based on the experiments of the first 100 pairs of fingerprints of NIST-4.

For the first fingerprint in each pair of fingerprints, we manually choose minutiae features, and then find the correspondences in the second fingerprint. Using the pairs of correspondences of minutiae features in each pairs of fingerprint, we estimate the transformation parameters by Mean Squared Error [104]. Figure 4.6 shows the distributions of s , θ , t_x and t_y . The data range that we choose for these parameters are: $0.9 \leq s \leq 1.1$, $-30^\circ \leq \theta \leq 30^\circ$, $-128 \leq t_x \leq 128$, $-128 \leq t_y \leq 128$.

128. Figure 4.6 shows that the transformation parameters of the first 100 pairs of fingerprints are all in these data ranges.

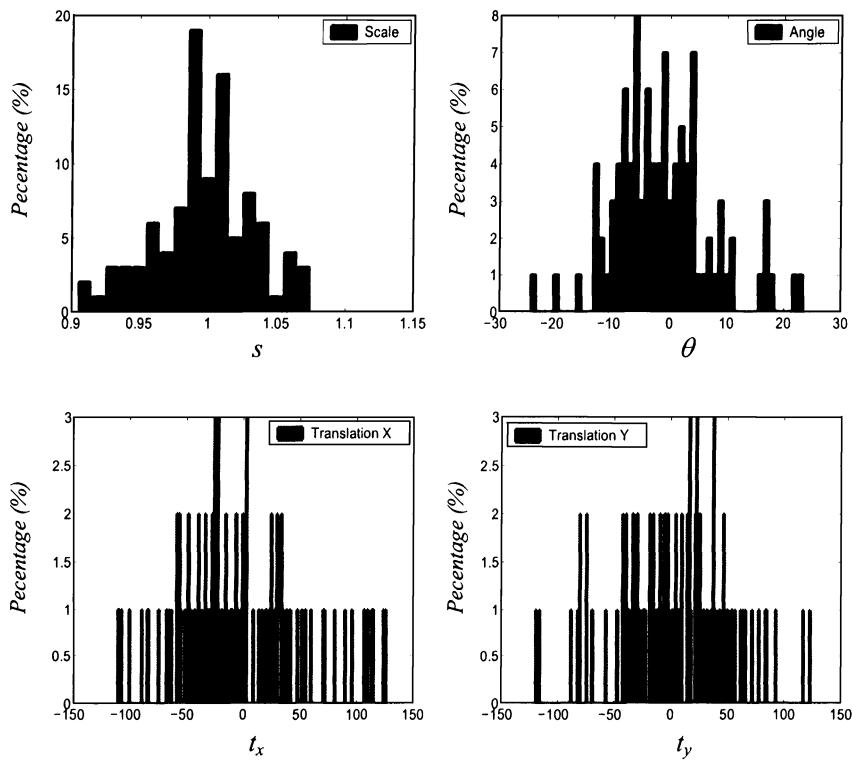


Figure 4.6. Histograms of transformation parameters (first 100 pairs of fingerprints in NIST-4).

According to the analysis in Section 3.3.1, we have $T_{\alpha_{min}} = 2^\circ$ and $T_{\alpha_{med}} = 2^\circ$. Other parameters are decided based on experimental evaluation. The parameters used in our experiments are shown in Table 4.1.

Table 4.1. Parameters in experiments.

Parameters	Value	Parameters	Value	Parameters	Value
T_d	12	T_n	5	p_s	0.2
$T_{\alpha_{min}}$	2°	$T_{\alpha_{med}}$	2°	p_m	0.005
T_λ	20	T_χ	2	N_t	15
T_ζ	2	N_p	100		

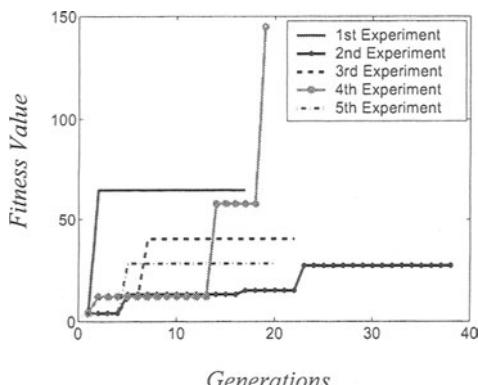


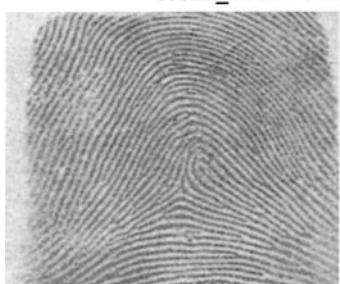
Figure 4.7. Fitness value changes for matching between the fingerprints in Figure 4.5.



f0032_03 and s0032_03, fitness value = 75



f0025_06 and s0025_06, fitness value = 162



f0023_04 and s0023_04, fitness value = 290

Figure 4.8. Examples of maximum fitness value between two fingerprints which are from the *same* fingers (genuine matching).



f0848_01 and s0234_10, fitness value = 0



f1174_07 and s0310_05, fitness value = 1



f1295_07 and s0814_01, fitness value = 0

Figure 4.9. Examples of maximum fitness value between two fingerprints which are from the *different* fingers (impostor matching).

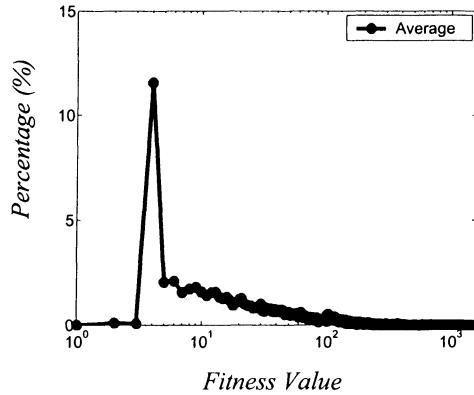


Figure 4.10. PDF of fitness value for genuine matchings.

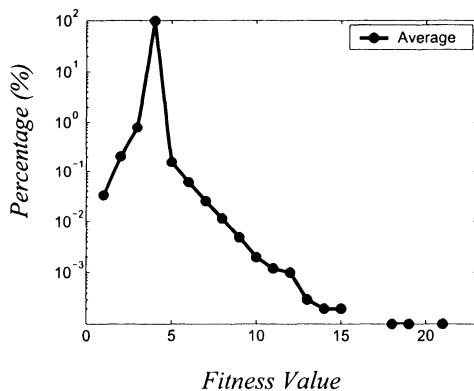


Figure 4.11. PDF of fitness value for imposter matchings.

4.4.3 Results

A total of 2000 matchings, between consistent pairs, are performed to estimate the genuine distribution. We also perform 200,000 matchings out of about 4,000,000 possibilities among inconsistent pairs to estimate the imposter distribution, where for each matching we randomly select two fingerprints from NIST-4 that are the impressions of different fingers. Figure 4.7 shows five evolutions of the fitness value for the matching between the fingerprints shown in Figure 4.5. As noted earlier, if the fitness value becomes greater than 100, then the evolution stops. We observe that the evolutions terminate within 40 generations. Considering the size of the search space is 2^{27} , the efficiency of GA is obvious. We repeat experiments of genuine matchings and imposter matchings five times, respectively. Figure 4.8 shows three pairs of fingerprints, which are different impressions of the same fingers, and their corresponding maximum fitness values. Figure 4.9 shows three pairs of fingerprints, which are different impressions of different fingers, and their corresponding maximum fitness values. We observe that the more similar the two fingerprints, the larger the fitness value. Since the fingerprints in Figure 4.9 are not from the same fingers, their fitness values are small. Figure 4.10 and Figure 4.11 show the average PDF of 10 experiments for genuine and imposter matchings, respectively. Note that the fitness value is shown on a logarithmic scale in Figure 4.10 and on a linear scale in Figure 4.11. Also, the vertical axis in Figure 4.11 is shown on a logarithmic scale. On the average, 99.1% imposter matchings (Figure 4.11) and 11.5% genuine matchings (Figure 4.10) have a fitness value of 4.

Based on genuine and imposter distributions, the Receiver Operating Characteristic (ROC) curve is defined as the plot of Genuine Acceptance Rate (GAR) against False Acceptance Rate (FAR). Figure 4.12 shows the comparison of the average ROC curve of our GA based approach and the ROC curve obtained using the mean-squared error (MSE) based approach [104]. It also shows the lower and upper bounds of GA based approach, which are estimated by repeating the experiments 10 times. The advantage of GA based approach is about

3.0%. When FAR is small, i.e. less than 0.02%, the advantage is about 2.0%. The main reason is that GA based approach tries to find global transformation and uses the local properties to verify it, which is better than finding local transformation only. Note that we have shown the results on the entire NIST-4 database. Verification results on NIST-4 are reported in Kovacs-Vajna [59]. However, in [59] 6.0% data are rejected manually by the author because of bad quality. Without Dynamic Time Warping (DTW) for the detailed verification, the FAR is 10.0%, which is unacceptable, although the GAR is 85.0%. With DTW, at FAR of 0.002% the GAR is 85% and for FAR of 0.05%, the corresponding GAR is 80%. It makes no sense for us to compare the performance reported in [59] with DTW since the author has not used the entire database, and we do not know which of the fingerprints have been rejected manually.

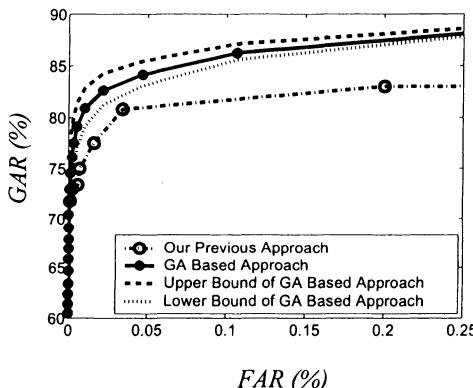


Figure 4.12. Comparison of ROC curves of two approaches using entire NIST-4 database (2000 pairs of fingerprints).

Examining the results, we find that:

- a) The low fitness values for most genuine matchings are due to the poor quality of fingerprints. There is not enough overlapped areas from which the feature extraction procedure can extract enough good minutiae;

- b) The nonzero fitness value for most imposter matchings are due to the similar structures and clutter features in two different fingerprint images. Figure 4.13 shows some low quality fingerprint pairs. In each case, the maximum fitness value is 4.

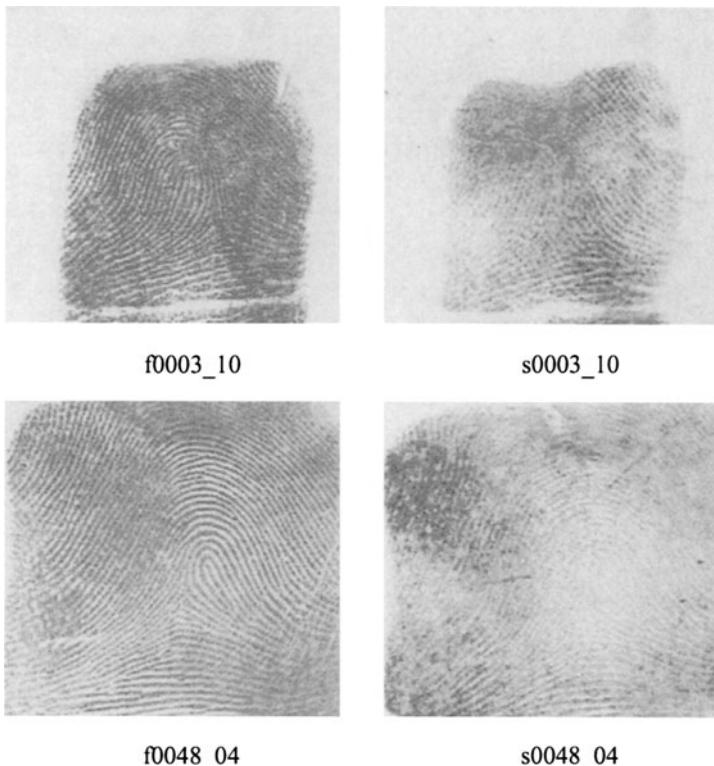


Figure 4.13. Sample fingerprints with low fitness value in genuine matching.

4.4.4 Effectiveness of Selection and Crossover Operators

In order to demonstrate the effectiveness of selection and crossover operators, we compared the performance of the pure GA and that of two other variations of GA. Two variations of the pure GA are:

- Instead of selecting hypotheses from parental population according to their fitness value, the first variation selects the hypotheses randomly for further evolution. The only restriction is that any hypothesis can only be selected once.
- The second variation simply skips the crossover. In order to generate the same number of children as the pure GA, the mutation rate of this variation is increased to 0.8, which is the same as the crossover rate.

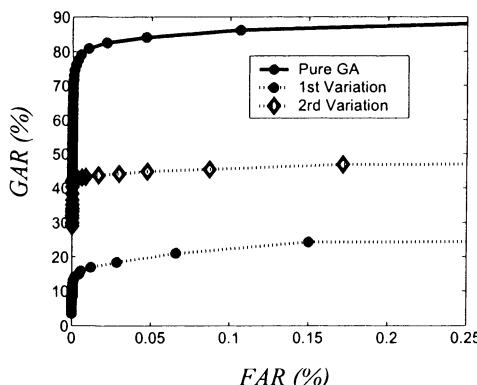


Figure 4.14. Comparison of the average performance of pure GA and its two other variations.

We perform the same experiments, which are explained in Section 4.4.3, to test the performance of both variations. Figure 4.14 shows the comparison of the average performance of the pure GA and other two variations. We observe that the performance of pure GA is much better than that of the other two variations. This demonstrates that the selection and crossover are critical to the success of GA.

4.4.5 Computation Time

On a SUN Ultra II workstation, which has a 200MHZ cpu, the average computation time for a genuine matching and an imposter

matching are 15 and 8 seconds, respectively. The difference of run-time between two kinds of matching is because GA needs to check more detailed local properties of the fingerprints for the genuine matchings, while it does not need do this for imposter matchings. The run-time may not satisfy the real-time requirement of the real-world applications. However, the computation time may be reduced by:

- Faster CPU, i.e. the 3GHZ CPU is available in the PC market now, and following Moore's law, faster computer will be available in the future;
- Parallel computation. Computation time may be reduced by parallel hardware.

4.5 CONCLUSIONS

In this Chapter, we have proposed a fingerprint matching approach, which is based on GA to find the globally optimized transformation. The local properties of each triplet of minutiae are used to find potential corresponding triangles and tolerate reasonable distortions, including translation, rotation, scale, shear, local perturbation, occlusion and clutter. We have explicitly modeled scale, rotation and translation, and other distortions are taken care of by local properties of the triangle. We achieve promising experimental results on the NIST-4 database, which has a large portion of poor quality fingerprints. The comparison shows the advantage of the proposed approach over the mean-squared error based approach.

Chapter 5

GENETIC PROGRAMMING FOR FINGERPRINT CLASSIFICATION

5.1 INTRODUCTION

Henry System is a systematic method for classifying fingerprints into five classes: Right Loop (R), Left Loop (L), Whorl (W), Arch (A), and Tented Arch (T). Figure 5.1 shows an example for each class. This system of fingerprint classification is commonly used by most of the developers and users, although the scheme adopted by the FBI defines eight classes [28].

In this Chapter, we present a fingerprint classification approach based on a novel feature-learning algorithm. Unlike current research for fingerprint classification that generally uses visually meaningful features, our approach is based on Genetic Programming (GP), which learns to discover composite operators and features that are evolved from combinations of primitive image processing operations. The experimental results show that our approach can find good composite operators to effectively extract useful features. Using a Bayesian classifier, without rejecting any fingerprints from NIST-4 database, the correct rates for 4 and 5-class classification are 93.3% and 91.2% re-

spectively, which compare favorably and have advantages over the best results published to date.

5.2 RELATED RESEARCH AND CONTRIBUTIONS

5.2.1 Related Research

The most widely used approaches for fingerprint classification are based on the number and relations of the singular points (SPs), including *Core* and *Delta*, which are defined as the points where a fingerprint's orientation field is discontinuous. *Core* is the topmost point on the innermost recurring ridge and *Delta* is the center of a triangular region where three different direction flows meet. Figure 5.2 shows an example of core and delta in a fingerprint.

Hsieh et al. [39] present methods for constructing a direction matrix, which is used to find out counts and locations from the fingerprint SPs. Besides directional image, curvature (Koo and Kot [59]), histogram of directional image (Srinivasan and Murthy [97]) and multi-resolution schemes (Tico and Kuosmanen [109], Drets and Liljenstrom [27]) are also used in finding fingerprint SPs.

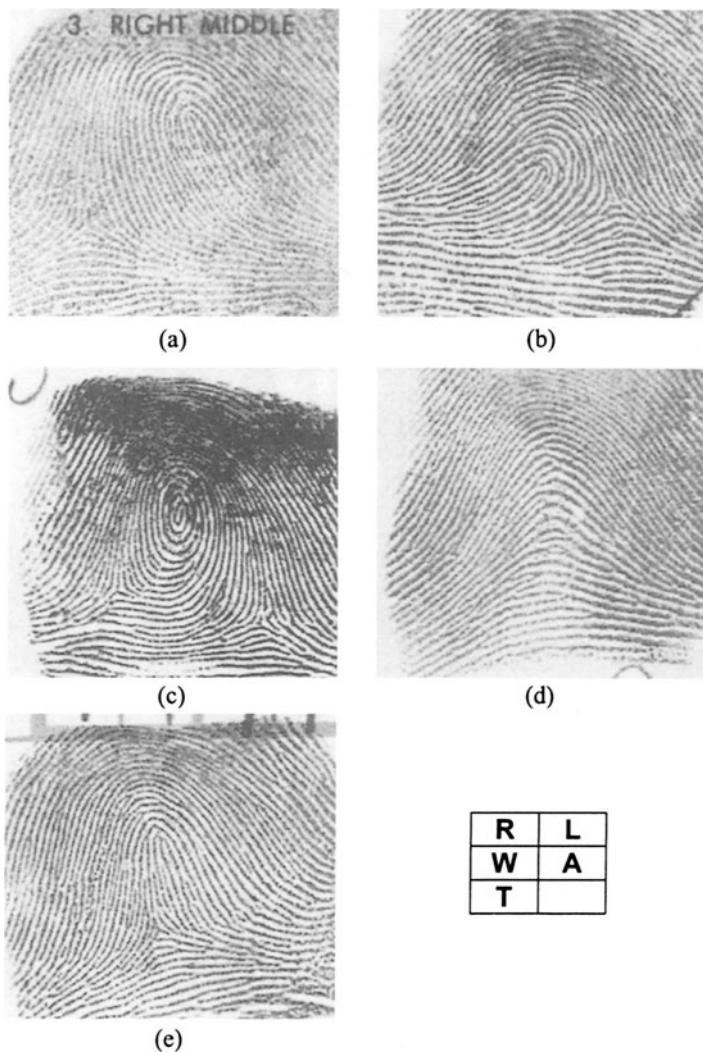


Figure 5.1. Examples of fingerprints from each class of *Henry System* for fingerprint classification: (a) Right Loop; (b) Left Loop; (c) Whorl; (d) Arch and (d) Tented Arch.⁴

⁴ Same Figure has been shown in Figure 1.3.



Figure 5.2. Examples of *Core* and *Delta*.

Using SPs as reference points, Karu and Jain [57] present a classification approach based on the structural information around SPs. Most other research uses a similar method: first, find the SPs and then use a classification algorithm to find the difference in areas, which are around the SPs for different classes. Several representations based on principal components analysis (PCA) (Candela et al. [17]), self-organizing map (SOM) (Halici and Ongun [36]) and Gabor filters (Jain et al. [45]) are used. The problems with these approaches are:

- It is not easy to detect the SPs and some fingerprints do not have SPs;
- Uncertainty in the location of SPs is large. This has a great effect on the classification performance since the features around the SPs are used.

Cappelli et al. [18] present a structural analysis of a fingerprint's orientation field. Jain and Minut [44] propose a classification algorithm based on finding the kernel that best fits the flow field of the given fingerprint. Both approaches do not need SPs. Researchers have also tried different methods to combine different classifiers to improve the classification performance. Senior [94] combines hidden Markov models (HMM), decision trees and PCASYS (a standard fingerprint

classification algorithm, Wilson et al. [117]). Yao et al. [120] present new fingerprint classification algorithms based on two machine learning approaches: support vector machines (SVMs) and recursive neural networks (RNNs). Table 5.1 summarizes representative fingerprint classification approaches. The features used in these approaches are well-defined conventional known features. It is clear that most current approaches in fingerprint classification are based on the extraction of reference points and conventional transforms.

Some researchers use learning algorithms to extract minutiae from fingerprints. Prabhakar et al. [82] propose a feedback system, which learns the characteristics of minutiae in gray scale images and can be used to verify each detected minutia. They show that a minutiae verification stage, which is based on reexamining the gray-scale profile in a detected minutia's spatial neighborhood in a fingerprint images, can improve the matching performance. Chapter 2 presented a learned templates based algorithm for minutiae extraction. Templates are learned from examples by optimizing a criterion function using Lagrange's method [14]. To detect the presence of minutiae in fingerprints, templates are applied with appropriate orientations to the binary fingerprints only at selected potential minutia locations. However, the above two approaches are for learning minutiae, which are well-defined structural features in fingerprints and are commonly used in fingerprint verification. To the best of our knowledge, unconventional features discovered by the computer are never used in fingerprint classification.

Table 5.1. Representative fingerprint classification approaches.

Approaches	Characteristics
Kamijo [56], 1993	A four-layered neural network integrated in a two-step learning method
Candela et al. [17], 1995	Probabilistic neural network (PNN)
Karu and Jain [57], 1996	Rule-based classification
Halici and Ongun [36], 1996	Neural network based on self organizing feature maps (SOM)
Fitz et al. [31], 1996	Hexagonal FFT
Chong et al. [23], 1997	The framework employs both a geometric grouping and a global geometric shape analysis of fingerprint ridges.
Ballan et al. [6], 1998	Singular point is obtained from the directional histograms of a fingerprint
Uchida et al. [112], 1998	Classification is based on a ridge structure analysis and a direction-based Neural Networks. Also, core-delta distance is used.
Kamei et al. [55], 1998	A new distance measure, which is defined by likelihood ratio of error distribution of feature vectors to the whole distribution of feature vector differences, is used in classification
Qi et al. [85], 1998	Probabilistic neural network based on Genetic Algorithm (GA) and feedback mechanism
Jain et al. [45], 1999	K-nearest neighbor + Neural network based on Gabor features (FingerCode)
Cappelli et al.[18], 1999; Lumini [69], 1999	Directional image is partitioned into homogeneous connected regions according to the fingerprint topology, thus giving a synthetic representation, which can be exploited as a basis for the classification.
Su et al. [99], 2000	Fractal analysis
Cappelli et al. [19], 2000	Several ways of combining the MASKS and MKL-based classifiers are discussed.
Shalash et al. [95], 2000	Multilayer SOM Neural Networks
Pattichis et al. [83], 2001	Probabilistic neural network + AM-FM representation for fingerprints
Bernard et al. [8], 2001	Kohonen topologic map
Senior [94], 2001	Hidden Markov model + Decision tree + PCASYS
Jain and Minut [44], 2002	Model-based method based on hierarchical kernel fitting
Mohamed and Nyongesa [76], 2002	Fuzzy neural network
Yao et al. [120], 2003	Support vector machine + Recursive neural network based on FingerCode

In most imaging applications, the approach, which is used to extract feature vectors from images, can often be dissected into some primitive operations on a set of selected feature images. Generally, the task of finding a good feature is equivalent to finding a good point in the search space of *composite operators*, where a composite operator consists of *primitive operators* and it can be viewed as a selected combination of primitive operations applied on *primitive feature* images. Our Genetic Programming (GP) based approach may try many unconventional ways of combining primitive operations that may never be imagined by humans and may yield exceptionally good results. The parallelism of GP and the speed of computers allow the search space explored by GP to be much larger than that by human experts. As the search goes on, GP gradually shifts the population of composite operators to the portion of the space containing good composite operators.

Genetic programming, an extension of genetic algorithm, was first proposed by Koza in [62]. In GP, the individuals can be binary trees, graphs or some other complicated structures of dynamically varying size. Poli [84] used GP to develop effective image filters to enhance and detect features of interest or to build pixel-classification-based segmentation algorithms. Stanhope and Daida [98] used GP paradigm for the generation of rules for target/clutter classification and rules for the identification of objects. To perform these tasks, previously defined feature sets are generated on various images and GP is used to select relevant features and methods for analyzing these features. Howard et al. [38] applied GP to automatic detection of ships in low-resolution SAR imagery using an approach that evolves detectors. Roberts and Howard [87] used GP to develop automatic object detectors in infrared images.

5.2.2 Contributions

- An approach that learns composite operators based on primitive features automatically. It helps to find some useful unconventional features, which are beyond the imagination of humans. The primitive operators and features defined in our approach

are very basic and easy to compute. The primitive features are based on orientation field image of a fingerprint and they allow the incorporation of domain knowledge into evolutionary process;

- Primitive operators are separated into computation operators and feature generation operators. Features are computed wherever feature generation operators are used. These features are used to form a feature vector that represents a particular fingerprint image and it is used for subsequent fingerprint classification;
- Results are shown on the entire NIST-4 fingerprint database and they are compared with the other published research.

5.3 TECHNICAL APPROACH

Figure 5.3 shows the block diagram of our approach. During the training, GP is used to generate composite operators, which are applied to the primitive features generated from the original orientation field. Feature vectors used for fingerprint classification are generated by composite operators. A Bayesian classifier is used for classification. During training, fitness value is computed according to the classification result and is monitored during evolution. During testing, the learned composite operator is applied directly to generate feature vectors. Note that, in our approach, we do not need to find the reference points.

In our GP based approach, individuals are composite operators, which are represented by binary trees [106]. The search space of GP is the space of all possible composite operators. This space is very large. In order to illustrate this, consider only a special kind of binary tree, where each tree has exactly 30 internal nodes and one leaf node and each internal node has only one child. For 17 primitive operators and only one primitive feature image, the total number of such trees is 17^{30} . It is extremely difficult to find good operators from this vast space unless one has a smart search strategy.

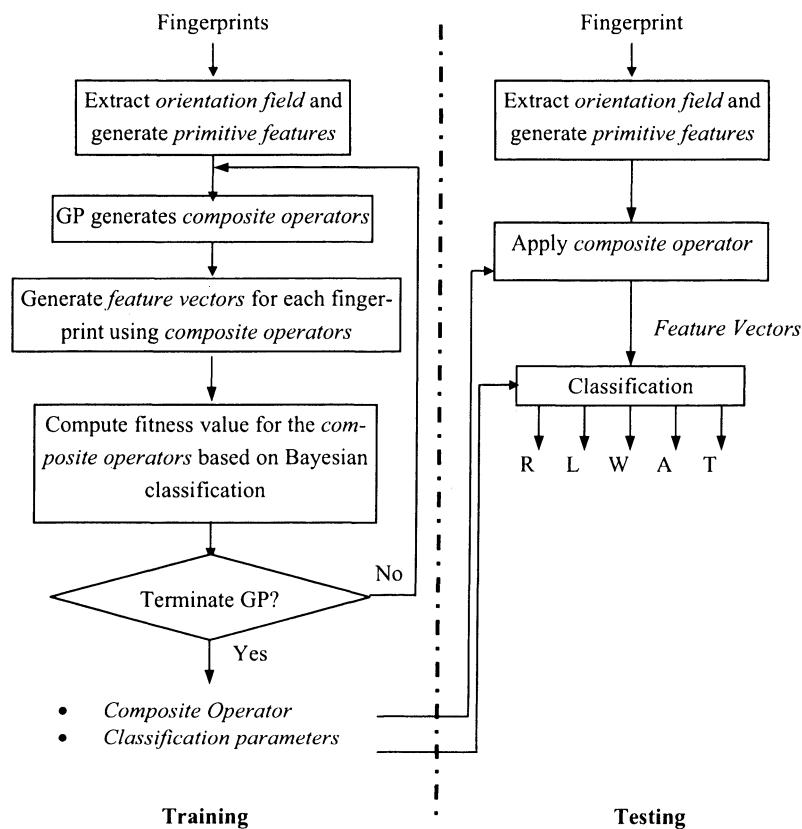


Figure 5.3. Block diagram of our approach.

5.3.1 Design Considerations

The major design considerations of GP are explained in the following:

- **The Set of Terminals:** For a fingerprint, we can estimate the orientation field (Bazen and Gerez [7]):

$$\theta = \frac{1}{2} \tan^{-1} \left(\frac{\sum_{i=1}^m \sum_{j=1}^m 2G_x(i, j)G_y(i, j)}{\sum_{i=1}^m \sum_{j=1}^m (G_x^2(i, j) - G_y^2(i, j))} \right) \quad (5.1)$$

where G_x and G_y are the gradient magnitudes of sobel operators in x and y directions, respectively. The block size, $m = 32$ in our experiments and $\theta \in [0, 180]$ which is measured in clockwise direction.

The set of terminals used in our approach are called *primitive features*, which are generated from the orientation field. Primitive features used in our experiments are:

- Original orientation image (primitive feature 1). Orientation image contains important structural information about a fingerprint.
- Mean, standard deviation, min, max and median images obtained by applying 3×3 and 5×5 templates on orientation image (primitive feature 2-11). These images contain information in the neighborhood of every pixel in orientation image.
- Edge images obtained by applying sobel filters along horizontal and vertical directions on orientation image (primitive feature 12-13). Both images contain information of the changes of orientation along different directions.
- Binary image obtained by thresholding the orientation image with a threshold of 90 (primitive feature 14). Since $\theta \in [0, 180]$, the threshold is chosen as 90. If the pixel value in orientation

image is greater than 90, the corresponding pixel in binary image is set to 1, otherwise, it is set to 0.

- Images obtained by applying *sine* and *cosine* operations on the orientation image (primitive feature 15-16). Both images contain information about the changes in orientation.

These 16 images are input to the composite operators. The size of these images is 12×13 . GP determines which operations are applied on them and how to combine the results. Figure 5.4 shows an example of a fingerprint image from NIST-4 fingerprint database and its corresponding primitive feature images. Note that, in order to show primitive feature images clearly, in each primitive image, maximum and minimum values in the image are mapped to 255 and 0, respectively, and other values are linearly mapped to a value between 0 and 255.

- **The Set of Primitive Operators:** A primitive operator takes one or two input images, performs a primitive operation on them and outputs a resultant image. Suppose

- 1) A and B are images of the same size and c is a constant of real number, $c \in [-100, +100]$;
- 2) for operators, which take two images as input, the operations are performed on the pixel-by-pixel basis.

Currently, there are two kinds of primitive operators in our approach: computation operators and feature generation operators. Table 5.2 explains the meaning of these operators in detail. For computation operators, the output is an image, which is generated by applying the corresponding operations on the input image. However, for feature generation operators, the output includes an image and a real number or vector. The output image is the same as the input image and passed as the input image to the next node in the composite operator. The real numbers or the vectors so generated are the elements of the feature vector, which is used for classification. Thus, the size of the feature vectors depends on the number of the feature generation operators that are a part of the composite operator. Figure 5.5 shows an example of a composite operator, which includes three computation operators and

three feature generation operators. Computation operators do computation and feature vector is generated by feature generation operators: SPE_MAX_OP, SPE_U3_OP and SPE_STD_OP.

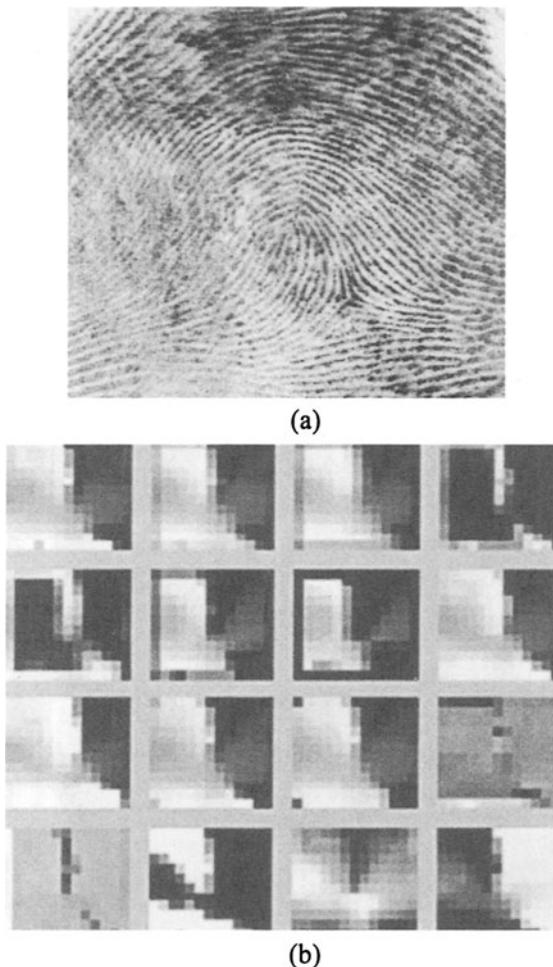


Figure 5.4. An example of fingerprint image from NIST-4 fingerprint database and the primitive feature images derived from the original image: (a) original image, f0760_06; and (b) primitive feature images. Note that 16 primitive feature images are sorted from left to right and top to bottom.

Table 5.2. Primitive operators used in our approach. The first nine rows are computation operators and the last five rows are feature generation operators.

Primitive Operator	Meaning
ADD_OP, SUB_OP, MUL_OP and DIV_OP	$A+B$, $A-B$, $A \times B$ and A/B . If the pixel in B has value 0, the corresponding pixel in A/B takes the maximum pixel value in A.
MAX2_OP and MIN2_OP	$\max(A,B)$ and $\min(A,B)$
ADD_CONST_OP, SUB_CONST_OP, MUL_CONST_OP and DIV_CONST_OP	$A+c$, $A-c$, $A \times c$ and A/c
SQRT_OP and LOG_OP	$\text{sign}(A) \times \sqrt{ A }$ and $\text{sign}(A) \times \log(A)$.
MAX_OP, MIN_OP, MED_OP, MEAN_OP and STD_OP	$\max(A)$, $\min(A)$, $\text{med}(A)$, $\text{mean}(A)$ and $\text{std}(A)$, replace the pixel value by the maximum, minimum, median, mean or standard deviation in a 3×3 block
BINARY_ZERO_OP and BINARY_MEAN_OP	threshold/binarize A by zero or mean of A
NEGATIVE_OP	-A
LEFT_OP, RIGHT_OP, UP_OP and DOWN_OP	left(A), right(A), up(A) and down(A). Move A to the left, right, up or down by 1 pixel. The border is padded by zeros
HF_DERIVATIVE_OP and VF_DERIVATIVE_OP	HF(A) and VF(A). Sobel filters along horizontal and vertical directions
SPE_MAX_OP, SPE_MIN_OP, SPE_MEAN_OP, SPE_ABS_MEAN_OP and SPE_STD_OP	$\max2(A)$, $\min2(A)$, $\text{mean2}(A)$, $\text{mean2}(A)$ and $\text{std2}(A)$
SPE_U3_OP and SPE_U4_OP	$\mu_3(A)$ and $\mu_4(A)$. Skewness and kurtosis of the histogram of A
SPE_CENTER_MOMENT11_OP	$\mu_{11}(A)$. First order central moments of A
SPE_ENTROPY_OP	$H(A)$. Entropy of A
SPE_MEAN_VECTOR_OP and SPE_STD_VECTOR_OP	$\text{mean_vector}(A)$ and $\text{std_vector}(A)$. A vector contains the mean or standard deviation value of each row/column of A

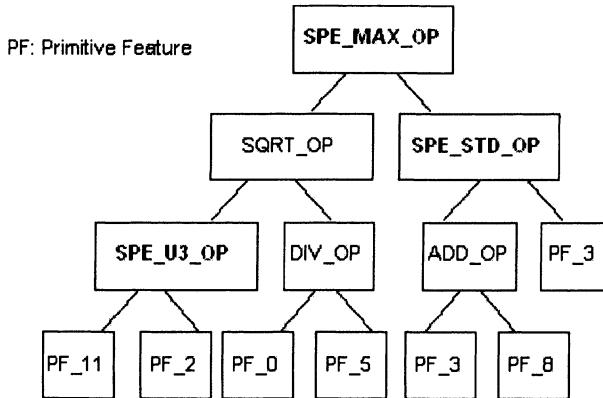


Figure 5.5. An example of a composite operator, which includes three computation operators and three feature generation operators. PFs are primitive features.

- **The Fitness Measure:** During training, at every generation for each composite operator proposed by GP, we compute the feature vector and estimate Probability Distribution Function (PDF) for each class using all the available feature vectors. For simplicity, we assume feature vectors for each class have normal distribution, $v_{i,j}$, where $i = 1, 2, 3, 4, 5$ and $j=1, 2, \dots, n_i$, n_i is the number of feature vectors in the training for class i , ω_i . Then, for each i , we estimate the mean μ_i and covariance matrix Σ_i by all $v_{i,j}$:

$$\mu_i = E[x], \quad \Sigma_i = E[(x - \mu_i)(x - \mu_i)^T] \quad (5.2)$$

where $x \in \{v_{i,1} v_{i,2} \dots v_{i,n_i}\}$.

Thus, PDF of ω_i can be expressed as:

$$p(x|\omega_i) = \frac{1}{(2\pi)^{n_i/2} |\Sigma_i|^{1/2}} \exp(-\frac{1}{2}(x - \mu_i)^T \Sigma_i^{-1} (x - \mu_i)) \quad (5.3)$$

According to Bayesian theory, we have

$$v \in \omega_k, \text{ iff. } p(v|\omega_k) \cdot p(\omega_k) = \max_{i=1,2,3,4,5} (p(v|\omega_i) \cdot p(\omega_i)) \quad (5.4)$$

where n is the size of feature vector and v is a feature vector for classification.

During training, we estimate $p(x | \omega_i)$, then use the entire training set to do the classification. Percentage of Correct Classification (PCC) is taken as the fitness value of the composite operator.

$$\text{Fitness Value} = \frac{n_c}{n_s} \times 100\% \quad (5.5)$$

where n_c is the number of correctly classified fingerprints in training set and n_s is the size of training set. Note that, if $|\Sigma_i| = 0$ for ω_i in equation (5.3), we simply let the fitness value of the composite operator be 0. During testing, we still use equation (5.4) to obtain the classification results on the testing set, however, none of the testing fingerprints is used in the training.

- **Parameters and Termination:** The key parameters are maximum size of composite operator (150), population size (100), number of generations (100), crossover rate (0.6), and mutation rate (0.05). GP stops whenever it finishes the pre-specified number of generations.

5.3.2 Reproduction, Crossover and Mutation

GP searches through the space of composite operators to generate new operators, which may be better than the previous ones. By searching through the composite operator space, GP gradually adapts the population of composite operators from generation to generation and improves the overall fitness of the whole population. More importantly, GP may find an exceptionally good operator during the search. The search is done by performing reproduction, crossover and muta-

tion operations. The initial population is randomly generated and the fitness of each individual is evaluated.

- **Reproduction:** Reproduction involves selecting a composite operator from the current population. We use tournament selection, where a number of individuals are randomly selected from the current population and the one with the highest fitness value is copied into the new population.
- **Crossover:** To perform crossover, two composite operators are selected on the basis of their fitness values. These two composite operators are called parents. One internal node in each of these two parents is randomly selected, and the two subtrees with these two nodes as root are exchanged between the parents. In this way, two new composite operators, called offspring, are created. Figure 5.6 shows an example of crossover between two composite operators.

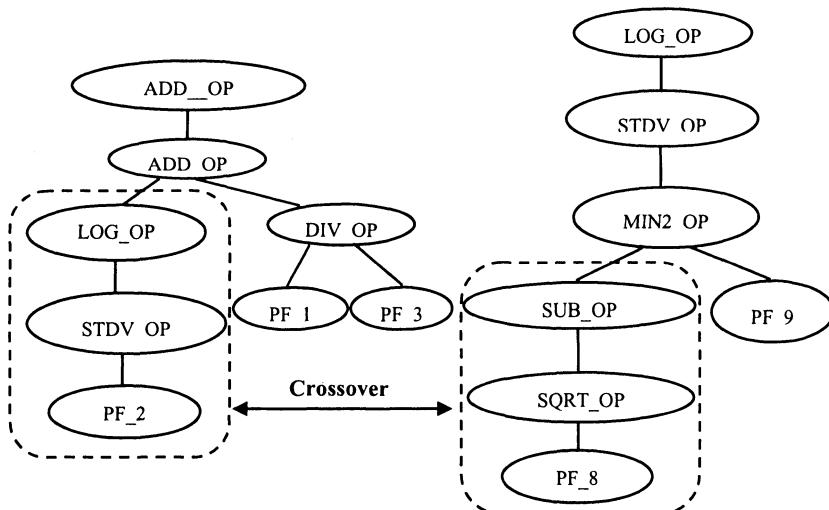


Figure 5.6. An example of crossover between two composite operators.

- **Mutation:** In order to avoid premature convergence, mutation is introduced to randomly change the structure of some of the individuals to help maintain the diversity of the population. Once a composite operator is selected to perform mutation operation, an internal node of the binary tree representing this operator is randomly selected, then the subtree rooted at this node is deleted, including the node selected. Another binary tree is randomly generated and this tree replaces the previously deleted subtree. The resulting new binary tree represents a new composite operator. This new composite operator replaces the old one in the population. Figure 5.7 shows an example of mutation of a composite operator.

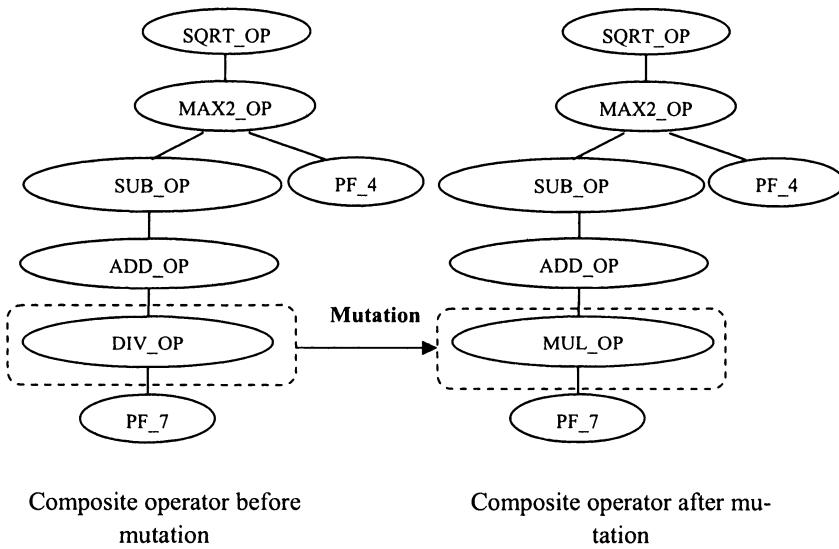


Figure 5.7. An example of mutation.

5.3.3 Steady-State and Generational Genetic Programming

In *steady-state* GP, two parental composite operators are selected on the basis of their fitness for crossover. The children of this crossover, perhaps mutated, replace a pair of composite operators with the smallest fitness values. The two children are executed immediately and their fitness values are recorded. Then another two parental composite operators are selected for crossover. This process is repeated until crossover rate is satisfied. In *generational* GP, two composite operators are selected on the basis of their fitness values for crossover. Then, two composite operators with the smallest fitness values, among those that have not been selected for replacement, are selected. They will be replaced by the children of the crossover. At this time, the replacement has not occurred. The above process is repeated until crossover rate is satisfied. A composite operator may be repeatedly selected for crossover, but it cannot be repeatedly selected for replacement. After crossover operations are finished, all the children resulted from the crossover operations replace all the composite operators selected for replacement at once. In addition, we adopt an elitism replacement method that copies the best composite operator from generation to generation. The steady state and generational genetic programming algorithms are given in Figure 5.8 and Figure 5.9, respectively. For simplicity, we use *steady-state* GP in our experiments.

Steady-state Genetic Programming:

- 1) randomly generate population P and evaluate each composite operator in P.
- 2) **for** gen = 1 **to** generation_num **do**
- 3) keep the best composite operator in P and perform reproduction to generate population P'.
- 4) number_of_crossover = population_size * cross_over_rate / 2.
- 5) **for** i = 1 **to** number_of_crossover **do**
- 6) select 2 composite operators from P' based on their fitness values for crossover.
- 7) select 2 composite operators with the lowest fitness values in P' for replacement.
- 8) perform crossover operation and let the 2 offspring composite operators replace the 2 composite operators selected for replacement.
- 9) **if** mutation is performed on the composite operators from the crossover **then** perform mutation on the 2 offspring operators with probability mutation_rate.
- 10) execute the 2 offspring composite operators and evaluate their fitness values.
- 11) **end** // loop 5
- 12) **if** mutation is performed on composite operators from the whole population P' **then**
- 13) perform mutation on each composite operator with probability mutation_rate.
- 14) execute and evaluate mutated composite operators.
- 15) **end**
- 16) replace the worst composite operator in P' by the best composite operator in P and **let** P = P'.
- 17) **if** fitness value of the best composite operator in P is above fitness threshold **then** stop.
- 18) **end** // loop 2

Figure 5.8. Steady-state Genetic Programming.

Generational Genetic Programming:

- 1) randomly generate population P and evaluate each composite operator in P.
- 2) **for** gen = 1 **to** generation_num **do**
- 3) keep the best composite operator in P
- 4) perform reproduction to generate population P' from P.
- 5) number_of_crossover = population_size * crossover_rate / 2.
- 6) perform crossover number_of_crossover times and record 2 * number_of_crossover composite operators to be replaced.
- 7) perform mutation on the composite operators generated from crossover or on the composite operators from the whole population.
- 8) **If** a composite operator is mutated **then** record it for later execution.
- 9) execute offspring composite operators from crossover and the mutated composite operators and evaluate their fitness values.
- 10) put offspring composite operators from crossover in P' and remove the composite operators selected for replacement from P'.
- 11) replace the worst composite operator in P' by the best composite operator in P.
- 12) **let** P = P'
- 13) **if** the fitness value of the best composite operator in P is above fitness threshold value **then** stop.
- 14) **end** // loop 2

Figure 5.9. Generational Genetic Programming.

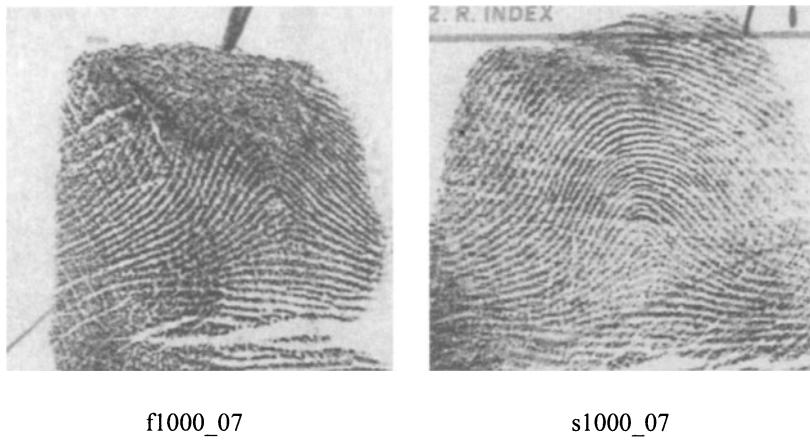


Figure 5.10. Sample fingerprints from NIST-4.

5.4 EXPERIMENTS

5.4.1 Database

We use NIST special fingerprint database 4 (NIST-4) [115] in our experiments. The size of the fingerprint images is 480×512 pixels with a resolution of 500 DPI. Since fingerprints' borders do not have much useful information, we only use the 384×416 pixels around the center of fingerprints. Thus, the size of primitive feature images is 12×13 pixels. Every pixel represents the orientation value in a local 32×32 block. NIST-4 contains 2000 pairs of fingerprints. One pair of sample fingerprints is shown in Figure 5.10. We use the first 1000 pairs of fingerprints for training and the second 1000 pairs of fingerprints for testing. In order to reduce the effect of overfitting, for the 1000 pairs of fingerprints in training set, we use the first 500 pairs to estimate the parameters for each class and use the entire training set to evaluate the training results. Note that the second 500 pairs in training set are not used in the estimation of distribution parameters for each class.

5.4.2 Results

We performed the experiments 10 times and took the best result as the learned composite operator. Figure 5.11 shows the fitness values based on the number of generations in GP. Since NIST-4 fingerprint database is a difficult database and includes many bad quality fingerprints, even in the training, the classification performance can not reach 100%. Figure 5.12 shows the best composite operators for 5-class and 4-class classifications, respectively. For 5-class classification, the composite operator's size is 61, out of which there are 21 feature generation operators and the length of the feature vector is 87. The tree, which represents this composite operator, is shown Figure 5.13. For 4-class classification, the composite operator's size is 149, out of which there are 23 feature generation operators and the length of the feature vector is 102. Obviously, these composite operators are not easy to be constructed by humans. Note that it is possible to perform feature selection to reduce the size of feature vectors by using Genetic Algorithms (GAs) or carry out an analysis like Principal Component Analysis (PCA) or Factor Analysis (FA). Figure 5.14 and Figure 5.15 show the output images of each node for the composite operators in Figure 5.12, respectively. The input image is s1000_07 shown in Figure 5.10. Note that, those images are sorted from left to right and top to bottom according to the preorder traversal of the composite operator. Accordingly, the feature vectors extracted by composite operators are shown in Figure 5.16. Note that feature vectors are multidimensional vectors, for simplicity, we show them as signal sequences.

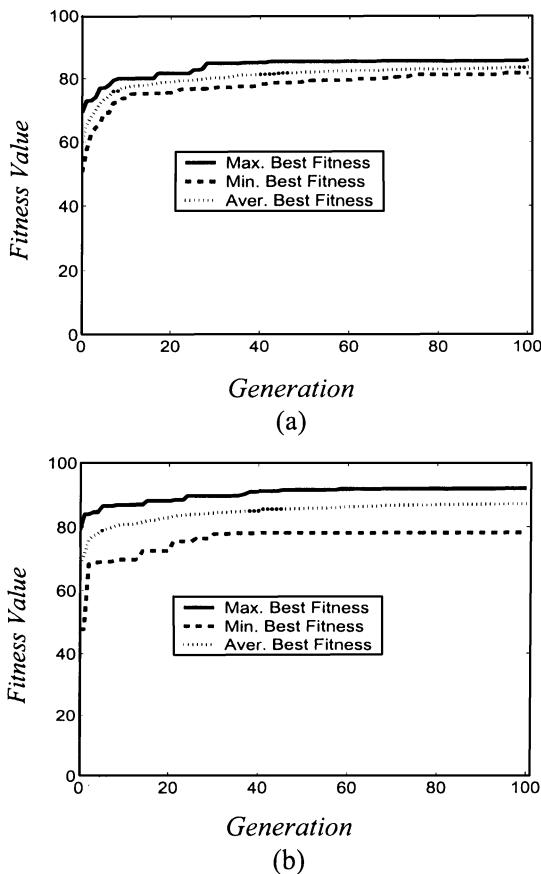


Figure 5.11. Average fitness values based on the number of generations: (a) 5-class and (b) 4-class.

(a) Composite Operator for 5-class classification, size 61:

```
((SUB_OP)((MIN_OP)((HF_DERIVATIVE_OP)((HF_DERIVATIVE_OP)((ADD_CONST_OP)((
(MUL_OP)((SPE_STD_VECTOR_OP)((STDV_OP)((SPE_CENTER_MOMENT11_OP)((
(SQRT_OP)((SUB_CONST_OP)((VF_DERIVATIVE_OP)((MEAN_OP)((PF_OP: 0)))))))))))(
(SUB_CONST_OP)((HF_DERIVATIVE_OP)((SUB_CONST_OP)((HF_DERIVATIVE_OP)((
(ADD_CONST_OP)((SUB_CONST_OP)((ADD_CONST_OP)((MUL_OP)((SPE_STD_OP)((
(MEAN_OP)((LOG_OP)((SPE_MEAN_VECTOR_OP)((SQRT_OP)((RIGHT_OP)((
(SPE_MIN_OP)((ABS_OP)((MEAN_OP)((PF_OP: 0))))))))))))((SUB_CONST_OP)((
(SPE_MEAN_VECTOR_OP)((SPE_STD_VECTOR_OP)((SPE_MIN_OP)((STDV_OP)((
(SPE_CENTER_MOMENT11_OP)((SPE_U3_OP)((SPE_STD_VECTOR_OP)((SPE_MIN_OP)((
(STDV_OP)((SPE_CENTER_MOMENT11_OP)((SPE_U3_OP)((UP_OP)((SPE_MEAN_OP)((
(PF_OP: 1))))))))))))))))))))((SUB_CONST_OP)((SPE_MEAN_OP)((
(SQRT_OP)((SUB_CONST_OP)((SPE_U3_OP)((SPE_U4_OP)((SPE_STD_VECTOR_OP)((
(SPE_MIN_OP)((STDV_OP)((SPE_CENTER_MOMENT11_OP)((SQRT_OP)((SUB_CONST_OP)((
(VF_DERIVATIVE_OP)((PF_OP: 13))))))))))))))))))
```

(b) Composite Operator for 4-class classification, size 149:

```
((SPE_STD_OP)((SPE_U4_OP)((ADD_CONST_OP)((ADD_OP)((BINARY_MEAN_OP)((
(MUL_OP)((SPE_MEAN_VECTOR_OP)((NEGATIVE_OP)((BINARY_ZERO_OP)((
(ADD_CONST_OP)((SPE_MAX_OP)((SPE_U3_OP)((ADD_CONST_OP)((SPE_STD_OP)((
(ADD_OP)((SPE_STD_OP)((PF_OP: 9))))((SPE_STD_VECTOR_OP)((ADD_CONST_OP)((
(LOG_OP)((PF_OP: 6))))))))))))((ADD_CONST_OP)((SPE_ABS_MEAN_OP)((
(SPE_CENTER_MOMENT11_OP)((MUL_CONST_OP)((ADD_CONST_OP)((LEFT_OP)((
(MAX2_OP)((SUB_CONST_OP)((SUB_CONST_OP)((MAX_OP)((MAX_OP)((UP_OP)((
(PF_OP: 1))))))))((SPE_MEAN_VECTOR_OP)((BINARY_ZERO_OP)((LOG_OP)((PF_OP: 6))(
))))))))((SPE_MEAN_VECTOR_OP)((SPE_U4_OP)((MAX2_OP)((MAX_OP)((UP_OP)((
(ADD_OP)((PF_OP: 4))((SPE_CENTER_MOMENT11_OP)((LOG_OP)((MAX2_OP)((
(MAX2_OP)((MAX_OP)((UP_OP)((MAX_OP)((UP_OP)((SPE_U4_OP)((MAX_OP)((
(DIV_CONST_OP)((SUB_OP)((DIV_CONST_OP)((SUB_OP)((SPE_CENTER_MOMENT11_OP)((
((ABS_OP)((PF_OP: 12))))((MED_OP)((MIN2_OP)((SPE_MEAN_VECTOR_OP)((SUB_OP)((
(MAX2_OP)((MAX_OP)((ABS_OP)((PF_OP: 12))))((BINARY_ZERO_OP)((PF_OP: 6))))((
(SPE_MEAN_VECTOR_OP)((SPE_U4_OP)((MAX2_OP)((MAX2_OP)((MAX_OP)((UP_OP)((
(ADD_OP)((PF_OP: 4))((BINARY_ZERO_OP)((LEFT_OP)((PF_OP: 15))))))))((LEFT_OP)((
(MED_OP)((LOG_OP)((PF_OP: 6))))))))((LEFT_OP)((MED_OP)((MAX_OP)((NEGATIVE_OP)((PF_OP: 8))))))))((SUB_CONST_OP)((MAX2_OP)((MAX_OP)((ABS_OP)((PF_OP: 12))))((SUB_CONST_OP)((MAX2_OP)((MAX_OP)((ABS_OP)((PF_OP: 12))))((MUL_OP)((MAX_OP)((NEGATIVE_OP)((PF_OP: 8))))((PF_OP: 4))))))))((RIGHT_OP)((MED_OP)((MIN2_OP)((SPE_MEAN_VECTOR_OP)((SUB_OP)((SPE_U4_OP)((
(MAX2_OP)((MAX_OP)((UP_OP)((ADD_OP)((PF_OP: 4))((SPE_CENTER_MOMENT11_OP)((
(SQRT_OP)((SPE_U4_OP)((ADD_CONST_OP)((LOG_OP)((PF_OP: 6))))))))((LEFT_OP)((
((MED_OP)((NEGATIVE_OP)((MED_OP)((NEGATIVE_OP)((PF_OP: 8))))))))((PF_OP: 10))))((SUB_CONST_OP)((PF_OP: 4))))))))((BINARY_MEAN_OP)((MUL_OP)((
(MAX_OP)((NEGATIVE_OP)((PF_OP: 8))))((PF_OP: 4))))((BINARY_MEAN_OP)((
(MUL_OP)((MAX_OP)((NEGATIVE_OP)((PF_OP: 8))))((PF_OP: 4))))((LEFT_OP)((
(ADD_CONST_OP)((LOG_OP)((PF_OP: 4))))))))))))
```

Figure 5.12. Learned composite operators: (a) 5-class, size is 61; and (b) 4-class, size is 149.

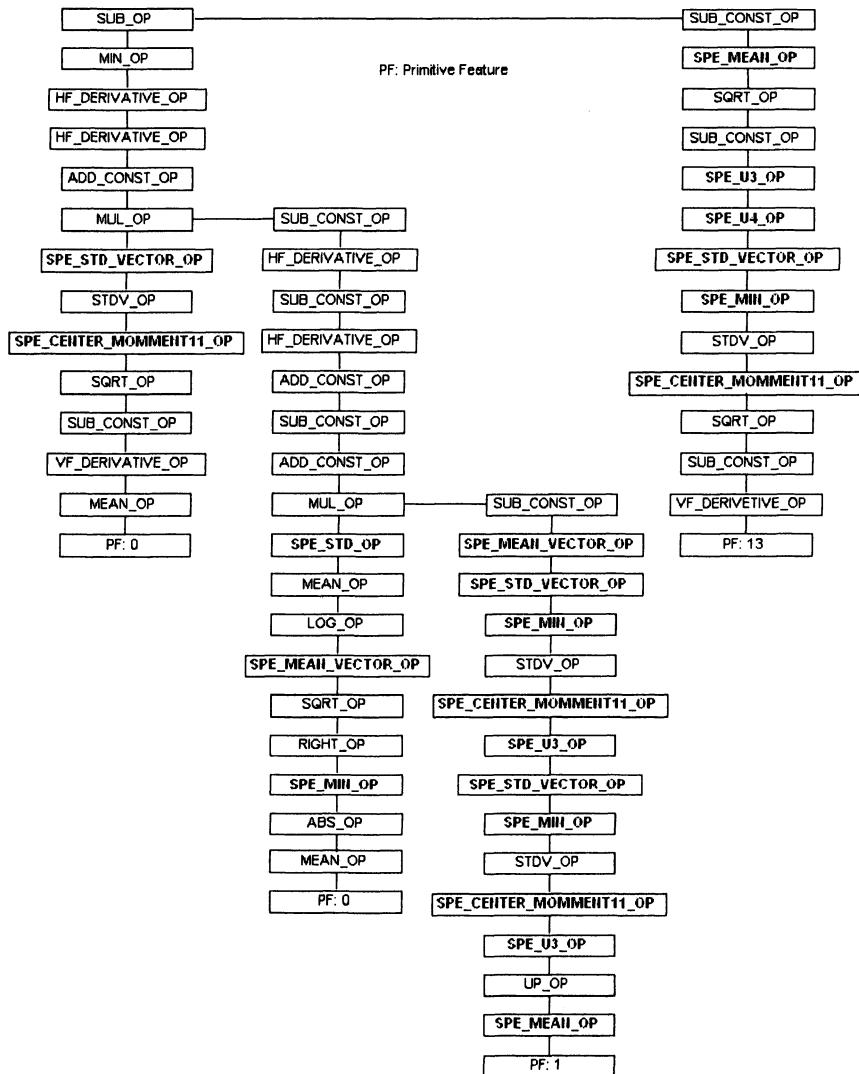


Figure 5.13. Tree structure for the composite operator in Figure 5.12(a). All these operators are defined in Table 5.2, and the size of the tree is 61. Feature generation operators are shown in black font and start with SPE, and all the others are computation operators. PFs are primitive features.

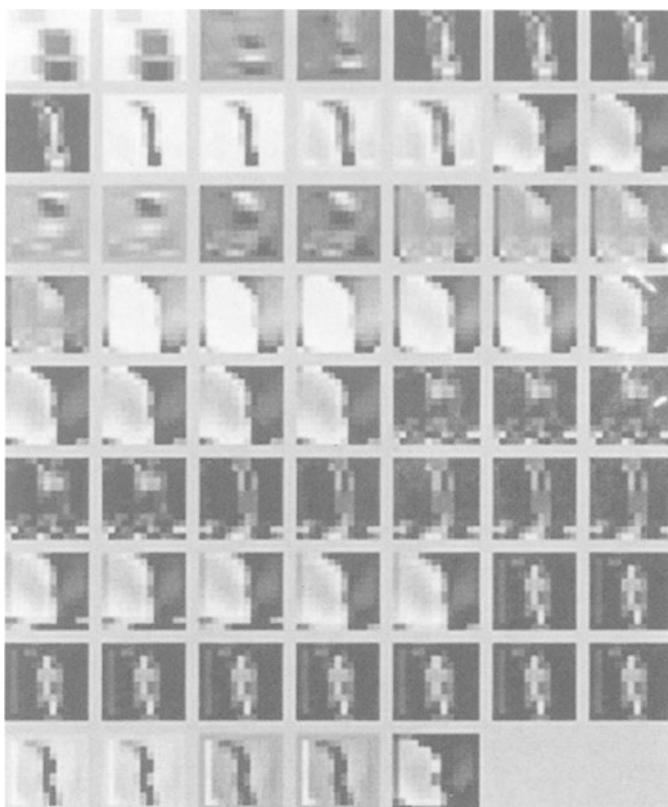


Figure 5.14. Output images of each node for the composite operator in Figure 5.12(a). The input fingerprint is s1000_07 shown in Figure 5.10. Those images are sorted from left to right and top to bottom according to the preorder traversal of the composite operator. The size of the composite operator is 61.

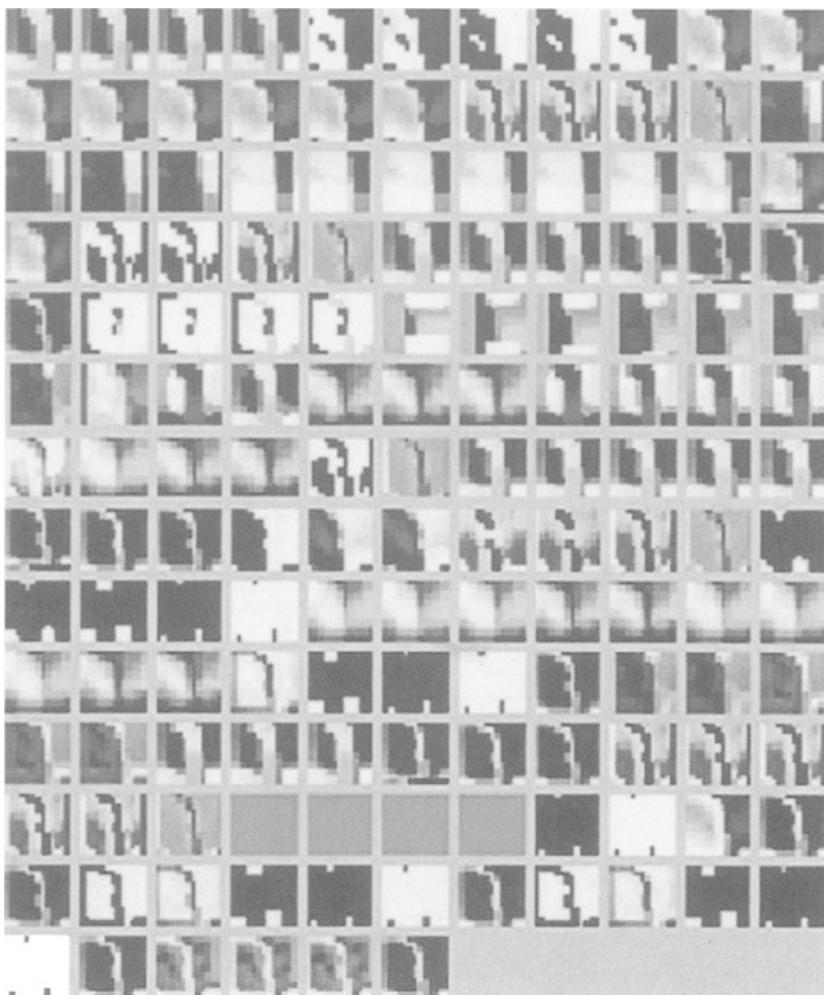


Figure 5.15. Output images of each node for the composite operator in Figure 5.12(b). The input fingerprint is s1000_07 shown in Figure 5.10. Those images are sorted from left to right and top to bottom according to the preorder traversal of the composite operator. The size of the composite operator is 149.

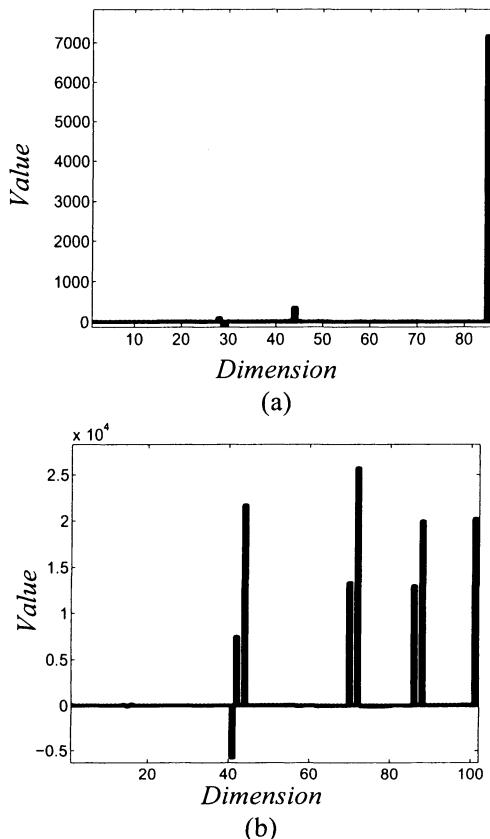


Figure 5.16. Feature vectors generated by the composite operators shown in Figure 5.12 on fingerprint s1000_07: (a) 5-class, length is 87 and (b) 4-class, length is 102. Note that, vertical scales are different in (a) and (b).

Table 5.3. Confusion matrix of the testing results for 5 and 4-class classifications.

	R	L	W	A	T
R	356	4	16	3	10
L	12	372	15	3	24
W	4	6	369	0	2
A	8	9	0	416	19
T	23	3	1	8	337

	R	L	W	A/T
R	381	4	11	23
L	11	375	5	40
W	1	4	382	1
A/T	20	12	3	741



f0008_10



s0008_10

Figure 5.17. Fingerprints with two ground-truth labels: T and L.

During training step, since we use GP, the experiments run slowly. Usually, it takes about 60 minutes to evolve one generation. However, once training is finished, applying composite operator is simple and it runs fast. On a SUN Ultra II workstation with a 200MHZ CPU, without any code optimization, the average run-times, for one test for 5-class and 4-class classification, are 40ms and 71ms, respectively.

Table 5.3 shows the confusion matrix of our testing results on the second 1000 pairs of fingerprint in NIST-4. Note that, because of bad quality, the ground-truths of some fingerprints provided by NIST-4 fingerprint database contain 2 classes. For example, the ground-truths

of f0008_10 and s0008_10 include class T and L, Figure 5.17 shows these two images together. As performed by other researchers in their experiments, we only use the first ground-truth label to estimate the parameters of the classifier. However, in testing, we use all the ground-truth labels and consider a test as correctly classified if the output of the system matches to one of the ground-truths. PCC is 93.2% and 91.2% for 4 and 5-class classifications respectively. Because bad quality image areas do not provide any useful information, they result in misclassifications.

Some examples of misclassifications are shown in Figure 5.18. Figures 5.19 and 5.20 show all the fingerprints that are misclassified in our approach for 5-class and 4-class, respectively. We show these results so that people can compare them with their work in the future.

Classes R, L, W, A and T are uniformly distributed in NIST-4. However, in nature, the frequencies of their occurrence are 31.7%, 33.8%, 27.9%, 3.7% and 2.9%, respectively. From Table 5.3, we observe that most of the classification errors are related to classes A and T. Considering that A and T occur less frequently in nature, our approach is expected to perform better in real world.

Table 5.4 shows the results on NIST-4 database reported by other researchers. Considering that we have not rejected any fingerprints from NIST-4, our result is one of the best. For the 5-class classification, our result has 1.6% advantage over the result shown in Jain et al. [45], although in [45] the reject rate is 1.8%.

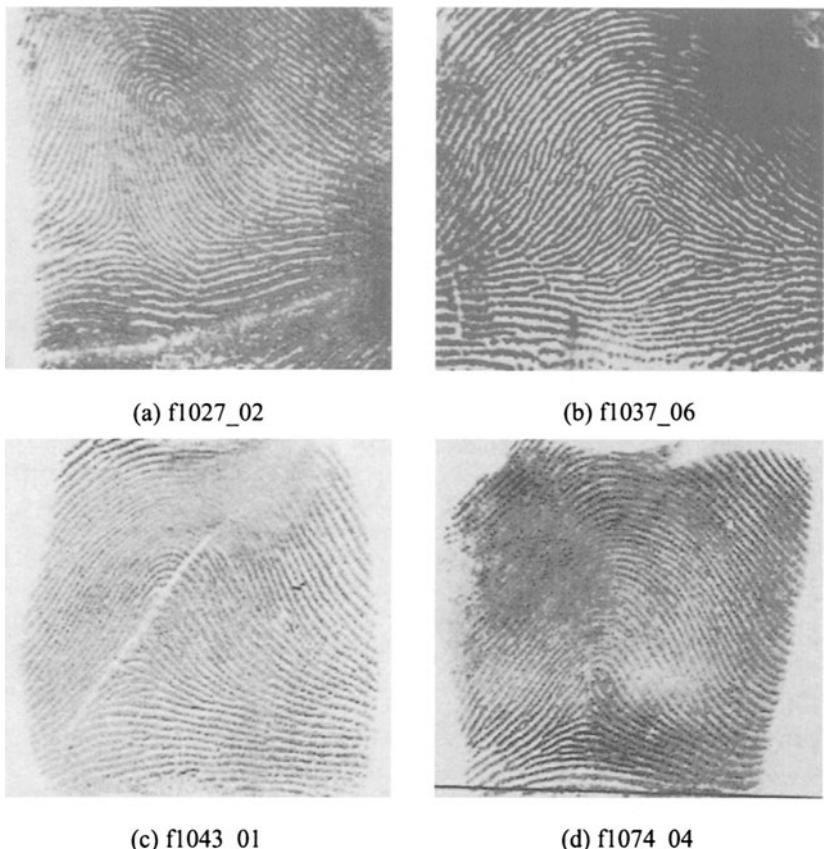


Figure 5.18. Errors in classification: (a) ground-truth: R, classification: W; (b) ground-truth: T, classification: A; (c) ground-truth: R, classification: T; (d) ground-truth: T/R, classification: A.

```

f1027_02( 2, 3 ), f1037_06( 1, 0 ), f1043_01( 0, 3 ), f1061_01( 2, 3 ),
f1070_07( 1, 0 ), f1074_04( 1, 0 & 3 ), f1076_09( 0, 1 ), f1130_09( 3, 4 ),
f1131_02( 4, 0 ), f1142_03( 4, 0 ), f1173_08( 4, 0 ), f1199_08( 4, 0 ), f1202_09( 2,
4 ), f1204_06( 1, 4 ), f1217_05( 3, 2 ), f1237_09( 4, 2 ), f1259_08( 4, 0 ),
f1260_04( 3, 2 ), f1266_08( 3, 4 ), f1268_02( 1, 0 & 3 ), f1269_04( 4, 3 ),
f1290_10( 4, 2 ), f1292_07( 1, 3 & 0 ), f1297_09( 2, 4 ), f1347_09( 2, 4 ),
f1349_04( 0, 1 ), f1362_01( 3, 2 ), f1376_08( 4, 2 ), f1401_06( 4, 1 ), f1405_07( 3,
0 ), f1417_02( 0, 3 ), f1437_07( 1, 0 & 4 ), f1438_09( 4, 2 ), f1440_05( 4, 3 ),
f1441_02( 0, 3 ), f1459_09( 4, 2 ), f1462_09( 3, 2 ), f1473_08( 0, 1 ), f1475_09( 4,
3 ), f1480_02( 4, 0 ), f1492_09( 0, 4 ), f1502_04( 4, 0 & 3 ), f1521_02( 4, 0 ),
f1522_03( 4, 2 ), f1527_02( 3, 2 ), f1529_03( 3, 0 ), f1545_03( 3, 2 ), f1552_04( 0,
3 ), f1572_07( 1, 0 ), f1576_06( 1, 0 & 4 ), f1580_09( 4, 2 ), f1653_07( 4, 0 ),
f1659_04( 0, 3 ), f1668_07( 0, 3 ), f1700_07( 4, 1 ), f1702_03( 1, 0 & 3 ),
f1706_07( 4, 0 ), f1711_02( 1, 0 ), f1724_01( 4, 2 ), f1727_02( 4, 0 ), f1745_09( 1,
0 & 4 ), f1746_09( 1, 4 ), f1755_04( 0, 3 ), f1757_08( 4, 3 ), f1774_03( 2, 3 ),
f1780_03( 3, 0 ), f1785_10( 4, 0 ), f1791_03( 4, 0 & 3 ), f1802_08( 4, 0 ),
f1812_09( 3, 4 ), f1822_03( 1, 3 ), f1833_05( 3, 2 ), f1852_01( 3, 2 ), f1863_04( 3,
2 ), f1889_04( 4, 3 ), f1897_07( 1, 0 & 4 ), f1902_04( 0, 3 ), f1907_04( 0, 3 ),
f1915_07( 4, 2 ), f1921_04( 1, 3 ), f1930_03( 3, 0 ), f1953_03( 3, 0 & 1 ),
f1966_07( 1, 0 & 3 ), f1980_04( 3, 2 ), f1998_07( 0, 1 ), s1009_10( 4, 2 ),
s1024_02( 2, 0 ), s1037_06( 1, 0 ), s1057_06( 4, 2 ), s1093_05( 4, 3 ), s1100_09(
0, 4 ), s1131_02( 4, 0 ), s1172_01( 3, 1 ), s1193_05( 0, 3 ), s1199_08( 4, 0 ),
s1210_08( 4, 0 ), s1217_05( 3, 2 ), s1242_08( 4, 0 ), s1259_08( 4, 0 ), s1269_04(
0, 3 ), s1278_06( 4, 2 ), s1299_07( 1, 0 ), s1338_03( 0, 3 ), s1363_08( 2, 0 & 3 ),
s1371_08( 4, 0 & 3 ), s1372_04( 4, 2 ), s1391_10( 2, 4 ), s1405_07( 4, 0 ),
s1441_02( 0, 3 ), s1443_03( 3, 2 ), s1459_09( 4, 2 ), s1462_09( 3, 2 ), s1486_03(
1, 0 & 3 ), s1516_02( 0, 3 ), s1521_02( 3, 0 ), s1527_02( 3, 2 ), s1528_09( 2, 4 ),
s1529_03( 3, 0 ), s1546_08( 2, 4 ), s1568_08( 0, 1 ), s1583_04( 0, 3 ), s1594_07(
4, 2 ), s1595_08( 0, 4 ), s1611_07( 4, 0 ), s1625_03( 3, 0 ), s1646_04( 0, 3 ),
s1668_07( 0, 3 ), s1711_02( 1, 0 ), s1712_04( 4, 3 ), s1727_02( 3, 0 ), s1740_07(
4, 1 & 0 ), s1745_09( 1, 0 & 4 ), s1746_09( 1, 4 ), s1755_04( 0, 3 ), s1759_09( 1,
0 & 4 ), s1760_04( 0, 3 ), s1772_04( 0, 2 & 3 ), s1783_04( 0, 3 ), s1785_10( 4, 0 ),
s1822_03( 0, 3 ), s1850_04( 4, 3 ), s1855_02( 3, 1 ), s1863_04( 3, 2 ), s1868_02(
3, 0 & 4 ), s1873_05( 4, 3 ), s1921_04( 0, 3 ), s1948_05( 0, 1 ), s1956_10( 0, 1 ),
s1957_05( 3, 2 ), s1998_07( 0, 1 )

```

Figure 5.19. Fingerprints that are misclassified in our approach for 5-class classification: Fingerprint_ID (class, ground-truth). 0, 1, 2, 3 and 4 represent classes T, A, W, R and L, respectively.

f1018_05(0, 2), f1024_02(2, 0), f1049_08(3, 0), f1081_05(0, 2), f1117_02(2, 0), f1129_06(3, 0), f1131_02(3, 0), f1142_03(3, 0), f1144_05(0, 2), f1172_01(2, 0), f1199_08(3, 0), f1202_09(1, 3), f1225_04(2, 1), f1241_07(3, 2), f1259_08(3, 0), f1284_05(0, 2), f1290_10(3, 1), f1316_09(2, 3 & 0), f1347_09(0, 3), f1362_01(2, 1), f1393_01(3, 1), f1401_06(3, 0), f1405_07(3, 0), f1406_03(2, 0), f1424_10(0, 3), f1440_05(3, 2), f1462_09(2, 1), f1480_02(2, 0), f1486_03(3, 0 & 2), f1492_09(0, 3), f1494_04(2, 0), f1510_06(3, 0), f1521_02(3, 0), f1527_02(2, 1), f1529_03(2, 0), f1552_04(3, 2), f1554_10(0, 3), f1611_07(3, 0), f1623_04(0, 2), f1655_07(3, 0), f1659_04(0, 2), f1666_10(0, 3), f1668_07(0, 2), f1693_08(2, 0 & 3), f1700_07(3, 0), f1727_02(3, 0), f1741_06(1, 3), f1746_09(0, 3), f1772_04(0, 1 & 2), f1794_08(0, 2), f1807_10(0, 1), f1822_03(0, 2), f1835_02(3, 0), f1841_05(2, 0), f1863_04(2, 1), f1868_02(2, 0 & 3), f1907_04(0, 2), f1930_03(2, 0), f1957_05(2, 1), f1980_04(2, 1), f1997_04(2, 0), s1024_02(3, 0), s1025_09(0, 3), s1057_06(3, 1), s1064_06(3, 1), s1109_04(0, 2), s1129_06(3, 0), s1131_02(3, 0), s1142_03(2, 0), s1172_01(2, 0), s1175_08(0, 3), s1185_03(3, 0 & 2), s1193_05(0, 2), s1199_08(3, 0), s1206_02(1, 0 & 2), s1210_08(3, 0), s1211_10(0, 3), s1255_07(3, 0 & 2), s1259_08(3, 0), s1260_04(2, 1), s1290_10(3, 1), s1296_07(2, 0), s1346_05(0, 2), s1371_08(3, 0 & 2), s1405_07(3, 0), s1406_03(2, 0), s1427_10(3, 0), s1433_07(3, 0 & 2), s1443_03(2, 1), s1462_09(0, 1), s1480_02(3, 0), s1492_09(0, 3), s1506_02(2, 0), s1521_02(2, 0), s1527_02(2, 1), s1528_09(1, 3), s1529_03(2, 0), s1538_04(0, 2), s1549_08(3, 0), s1557_10(1, 3), s1588_07(3, 0 & 2), s1611_07(3, 0), s1659_04(0, 2), s1668_07(0, 2), s1706_07(3, 0), s1722_10(0, 3), s1727_02(2, 0), s1746_09(0, 3), s1751_08(3, 0 & 0), s1755_04(0, 2), s1769_07(3, 0), s1785_10(3, 0), s1794_08(0, 2), s1817_07(3, 0 & 2), s1862_09(2, 3 & 0), s1863_04(2, 1), s1902_04(3, 2), s1921_04(0, 2), s1923_06(3, 0), s1930_03(2, 0), s1955_08(3, 0)
--

Figure 5.20. Fingerprints that are misclassified in our approach for 4-class classification: Fingerprint_ID (class, ground-truth). 0, 1, 2 and 3 represent classes T/A, W, R and L, respectively.

Table 5.4. Classification results on NIST-4.

Approaches	Class #	Error rate %	Reject rate %	Dataset	Comments
Karu and Jain 1996 [57]	5	14.6	zero	4000 images, no training	Decision based on topological information
	4	8.6			
Jain and Minut 2002 [44]	4	8.7	zero	Same as above	Hierarchical kernel fitting
Jain et al. 1999 [45]	5	14.6	1.8	Training: First 2000 images	KNN
	4	8.5			
	5	13.6		Testing: Second 2000 images	Neural Network
	4	7.9			KNN+NN, two stage classifier
	5	10.0			
	4	5.2			
Senior 2001 [94]	4	Average 8.5	zero	Same as above	Neural Network fusion with priors
Yao et al. [120]	5	10.0	1.8	Same as above	SVM+RNN
	4	5.3			
Proposed approach	5	8.4	zero	Same as above	GP based learned features + Bayesian classifier
	4	6.7			

5.5 CONCLUSIONS

In this Chapter, we have proposed a learning algorithm for fingerprint classification based on GP. Our experimental results show that the primitive operators selected by us are effective and GP can find good composite operators, which are beyond humans' imagination, to extract the feature vectors for fingerprint classification. Experimental results on NIST-4 fingerprint database show that our approach is one of the best approaches. Without rejecting any fingerprints, the experimental results show that our approach is efficient and promising, and has advantages over the best results reported in the literature.

Chapter 6

CLASSIFICATION AND INDEXING APPROACHES FOR IDENTIFICATION

6.1 INTRODUCTION

As shown in Figure 1.6(a), in an identification system, the input is a query fingerprint, the system tries to answer the question: are there any fingerprints in the database, which resemble the query fingerprint? In this chapter, we are dealing with the identification problem. We present a comparison of two key approaches for fingerprint identification. These approaches are based on:

- 1) Classification followed by verification;
- 2) Indexing followed by verification.

The fingerprint classification approach is based on a novel feature-learning algorithm (Chapter 5). It learns to discover composite operators and features that are evolved from combinations of primitive image processing operations. These features are then used for classification of fingerprint into five classes. The indexing approach is based on novel features derived from triplets of minutiae. The same verification algorithm, which is based on least square minimization over each of

the possible triplets of minutiae pair, is used for identification in both cases. On the NIST-4 fingerprint database, the comparison shows that, although correct classification rate can be as high as 92.8% for 5-class problems, the indexing based approach performs better based on both the size of search space that is examined and the identification results.

6.2 RELATED RESEARCH AND CONTRIBUTIONS

6.2.1 Related Research

There are three approaches to solve the fingerprint identification problem: 1) repeat the verification procedure for each fingerprint in the database and select the best match; 2) fingerprint classification followed by verification; 3) fingerprint indexing followed by verification. Figure 6.1 shows the block diagram of these three kinds of approaches.

- The *first* approach is based on verification only. Techniques for fingerprint verification can be found in Jain et al. [43], Jiang and Yau [52] and Kovacs-Vajna [59]. Chapter 4 provided a detailed discussion of matching techniques. However, if the size of the database is large, this approach will be time-consuming and it is not practical for real-world applications.

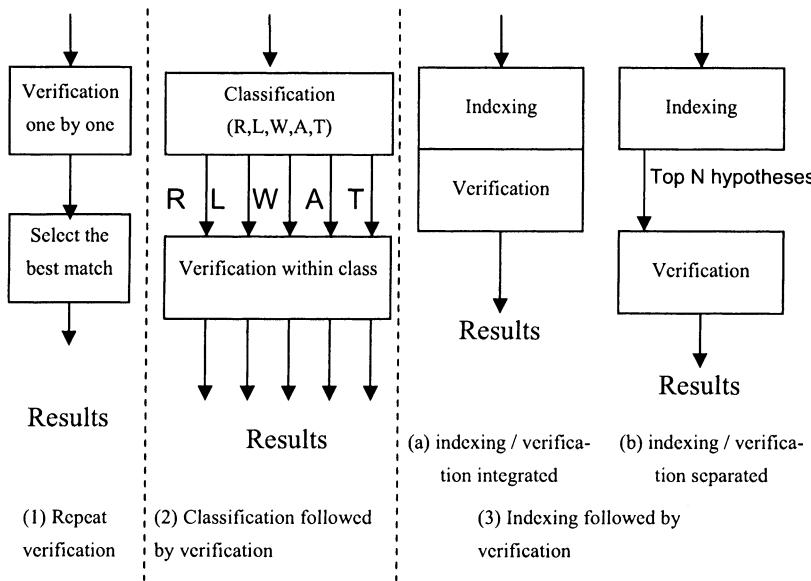


Figure 6.1. Block diagram of different approaches to solve identification problem.

- The *second* approach is based on classification. Traditional classification techniques attempt to classify fingerprints into five classes: Right Loop (R), Left Loop (L), Whorl (W), Arch (A), and Tented Arch (T). Figure 1.3 shows an example for each class. The most widely used approaches for fingerprint classification are based on the number and relations of the Singular Points (SPs), which are defined as the points where a fingerprint's orientation field is discontinuous. Using SPs as the reference points, Karu and Jain [57] present a classification approach based on the structural information around SPs. Other research works, which use SPs as reference points, include Candela et al. [17], Halici and Ongun [36], and Jain et al. [45]. The problem with this kind of approach is that it is not easy to detect SPs and some fingerprints do not have SPs. The worst thing is that the uncertainty in the locations of SPs could be large, which has great effects on the classification results. Cap-

orientation field, which does not need SPs. Researchers also present methods to combine different classifiers to improve the classification performance, e.g. Senior [94], Marcialis et al. [73] and Yao et al. [120]. The most important problem associated with the classification technique for identification is that the number of principal classes is small and the fingerprints are unevenly distributed (31.7%, 33.8%, 27.9%, 3.7%, and 2.9% for classes R, L, W, A, and T, respectively [115]). The classification approach does not narrow down the search enough in the database for efficient identification of a given fingerprint.

- The goal of the *third* approach is to significantly reduce the number of candidate hypotheses to be considered by the verification algorithm. These approaches are called indexing techniques in the fingerprint recognition area. A prominent approach for fingerprint identification is by Germain et al. [33], which integrates the indexing and verification in their approach (Figure 6.1(3)(a)). They use the triplets of minutiae in their identification procedure. The features they use are: the length of each side, the angles that the ridges make with respect to the X-axis of the reference frame, and the ridge count between each pair of vertices. The problems with their approach are:
 - a) The length changes are not insignificant under scale and shear;
 - b) The ridge angles change greatly with different quality images of the same finger;
 - c) Uncertainty of minutiae locations is not modeled explicitly.
 As a result, large size bins have to be used to handle distortions, which increases the probability of collisions and degrades the performance of their algorithm.

Our indexing approach follows Germain et al. [33] in that we also use the triplets of minutiae and ridge counts. However, the indexing and verification in our approach for identification are separated (Figure 6.1(3)(b)). Firstly, we apply indexing techniques to find top N hypotheses, and then apply a verification technique to verify hypotheses. Furthermore, most features that we use are quite different from theirs.

The features that we use are: triangle's angles, handedness, direction; maximum side; minutiae density and ridge counts. These features are different, new and more robust than the features used by Germain et al. We also use the constraints of the transformation to eliminate the false corresponding triangles. Their approach is basically an indexing method where the top hypothesis is taken as the identification result [51].

Figure 6.2 shows the block diagram of our identification approach [108]. Minutiae are extracted by the approach introduced in Chapter 2. During the offline processing for indexing, template fingerprints are processed to construct the model database. During the online processing, features of the query fingerprint based on the triplets of minutiae are used to find the potential corresponding triangles. Then, top N hypotheses are generated according to the number of potential corresponding triangles. The transformation between each pair of potential corresponding triangles is estimated using Mean Square Error. Finally, the constraints of the transformation are applied to eliminate the false corresponding triangles. The identification score is computed based on the number of corresponding triangles.

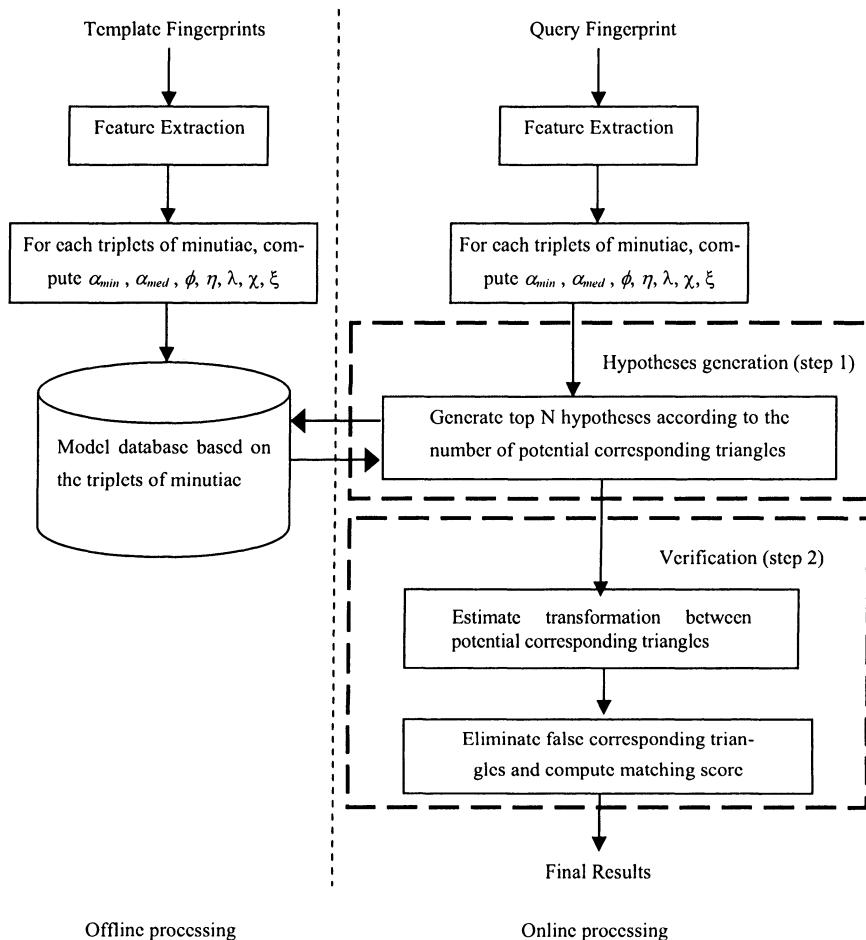


Figure 6.2. Two step approach for fingerprint identification.

6.2.2 Contributions

The major contributions are:

- We present a robust fingerprint identification approach, which can tolerate highly nonlinear deformations;
- We integrate the classification technique into the fingerprint identification system;
- Extensive comparison of identification based on classification and indexing techniques is performed;
- All experiments are carried out on NIST-4, a standard fingerprint database.

6.3 TECHNICAL APPROACH

The classification approach has already been introduced in Chapter 5. In the following, we present a new indexing approach, which is slightly different from that has been presented in Chapter 3. Verification follows classification and indexing. For indexing, verification is simple, since after indexing, we know the potential corresponding triangles and we may use this information in the verification directly. However, for classification, we only know the class information. So, we have to find the potential corresponding triangles between the query fingerprint and each template fingerprint that belongs to the same class. Then we can apply the verification approach to each template from that class for identification.

6.3.1 Indexing

For a triangle that is based on a triplet of minutiae, in Section 3.3 we have shown that: 1) the minimum and the median angles α_{min} and α_{med} are more robust than the maximum angle α_{max} and they can be used to find the correspondences; 2) 2° - 4° change in α_{min} and α_{med} can accommodate the uncertainty of most distortions and keep the size of the search space as small as possible.

The features we use to find potential corresponding triangles are defined as follows. Note that these features are not exactly the same as we introduced in Chapter 3. We use minutiae density and ridge counts instead of triangle type.

- **Angles α_{min} and α_{med} :** Suppose α_i are three angles in the triangle, $i = 1, 2, 3$. Let $\alpha_{max} = \max\{\alpha_i\}$, $\alpha_{min} = \min\{\alpha_i\}$, $\alpha_{med} = 180^\circ - \alpha_{max} - \alpha_{min}$, then the label of the triplets in this triangle is: if the minutia is the vertex of angle α_{max} , we label this point as P_1 ; if the minutia is the vertex of angle α_{min} , we label it as P_2 ; the last minutia is labeled as P_3 .
- **Triangle Handedness ϕ :** Let $Z_i = x_i + jy_i$ be the complex number ($j = \sqrt{-1}$) corresponding to the coordinates (x_i, y_i) of point P_i , $i = 1, 2, 3$. Define $Z_{21} = Z_2 - Z_1$, $Z_{32} = Z_3 - Z_2$, and $Z_{13} = Z_1 - Z_3$. Let $\phi = \text{sign}(Z_{21} \times Z_{32})$, where sign is the signum function and \times is the cross product of two complex numbers.
- **Triangle Direction η :** Search the minutia from top to bottom and left to right in the fingerprint, if the minutia is the start point of a ridge or valley, then $v = 1$, else $v = 0$. η is the combination of v_i , v_i is the v value of point P_i , $i = 1, 2, 3$.
- **Maximum Side λ :** Let $\lambda = \max\{L_i\}$, where $L_1 = |Z_{21}|$, $L_2 = |Z_{32}|$, and $L_3 = |Z_{13}|$.
- **Minutiae Density χ :** In a local area centered at the minutiae P_i , if there exists n_χ minutiae, then minutiae density $\chi_i = n_\chi$. χ is a vector consisting of all χ_i 's.
- **Ridge Counts ξ :** ξ_1 is the ridge count of the side P_1P_2 , ξ_2 is the ridge count of the side P_2P_3 , and ξ_3 is the ridge count of the side P_3P_1 . ξ is a vector consisting of all ξ_i 's.

If two triangles from two different fingerprints satisfy the following criteria, then they are potential corresponding triangles:

$$\begin{aligned} |\alpha'_{min} - \alpha''_{min}| &\leq T_{\alpha_{min}} \\ |\alpha'_{med} - \alpha''_{med}| &\leq T_{\alpha_{med}} \\ \phi' &= \phi'' \end{aligned}$$

$$\begin{aligned}
 \eta' &= \eta'' \\
 |\lambda' - \lambda''| &\leq T_\lambda \\
 |\chi'_i - \chi''_i| &\leq T_\chi, i = 1, 2, 3 \\
 |\xi'_i - \xi''_i| &\leq T_\xi, i = 1, 2, 3
 \end{aligned} \tag{6.1}$$

where $(\alpha'_{\min}, \alpha'_{med}, \phi', \eta', \lambda', \chi'_i, \xi'_i)$ and $(\alpha''_{\min}, \alpha''_{med}, \phi'', \eta'', \lambda'', \chi''_i, \xi''_i)$ are the local properties of the triangle in different fingerprints; $T_{\alpha_{\min}}$, $T_{\alpha_{med}}$, T_λ , T_χ , and T_ξ are thresholds to deal with the local distortions.

6.3.2 Verification

Suppose the sets of minutiae in the template and the query fingerprints are $\{(t_{m,1}, t_{m,2})\}$ and $\{(q_{n,1}, q_{n,2})\}$ respectively, where $m = 1, 2, 3, \dots, M$, $n = 1, 2, 3, \dots, N$, M and N are the number of minutiae in the template and the query fingerprints respectively. Let Δ_t and Δ_q be two potential corresponding triangles in the template and the query fingerprints, respectively. The coordinates of the vertices of Δ_t and Δ_q are $(x_{i,1}, x_{i,2})$ and $(y_{i,1}, y_{i,2})$, respectively, and $i = 1, 2, 3$.

Suppose $X_i = [x_{i,1}, x_{i,2}]'$, $Y_i = [y_{i,1}, y_{i,2}]'$, and the transformation $Y_i = F(X_i)$ between X_i and Y_i can be expressed as:

$$Y_i = \begin{bmatrix} 1 & \delta h_x \\ \delta h_y & 1 \end{bmatrix} \begin{bmatrix} 1 + \delta s_x & 0 \\ 0 & 1 + \delta s_y \end{bmatrix} R \cdot X_i + T \tag{6.2}$$

where $(\delta h_x, \delta h_y)$ and $(1 + \delta s_x, 1 + \delta s_y)$ are the shear and scale parameters; $R = \begin{bmatrix} \cos \theta & -\sin \theta \\ \sin \theta & \cos \theta \end{bmatrix}$, θ is the angle of rotation between two fingerprints; and $T = \begin{bmatrix} t_1 \\ t_2 \end{bmatrix}$ is the vector of translation.

Under the assumption that $\delta h_x \ll 1$, $\delta h_y \ll 1$, and $\delta s_x \approx \delta s_y$, we can simplify equation (6.2) to

$$Y_i = s \cdot R \cdot X_i + T \quad (6.3)$$

where s is the scaling factor.

We estimate the transformation parameters by minimizing the sum of the squared distances between the transformed query points and their corresponding template points. That is,

$$\text{error} = \arg \min_{(\hat{s}, \hat{R}, \hat{T})} \{\varepsilon^2\} \quad (6.4)$$

where $\varepsilon^2 = \sum_{i=1}^3 \|Y_i - (\hat{s} \cdot \hat{R} \cdot X_i + \hat{T})\|_V^2$, $\|V\|$ is the L_2 norm of vector V .

The solution of equation (6.3) is

$$\begin{aligned} \hat{\theta} &= \arctan\left(\frac{B}{A}\right), \quad \hat{s} = \frac{\sum_{i=1}^3 \{(X_i - \bar{X})' \hat{R}' (Y_i - \bar{Y})\}}{\sum_{i=1}^3 \{(X_i - \bar{X})' (Y_i - \bar{Y})\}}, \\ \hat{T} &= \bar{Y} - \hat{s} \cdot \hat{R} \cdot \bar{X} \end{aligned} \quad (6.5)$$

where

$$\begin{aligned} A &= \sum_{i=1}^3 \{(\bar{x}_1 - x_{i,1})(y_{i,1} - \bar{y}_1) + (\bar{x}_2 - x_{i,2})(y_{i,2} - \bar{y}_2)\}, \\ B &= \sum_{i=1}^3 \{(\bar{x}_1 - x_{i,1})(y_{i,2} - \bar{y}_2) - (\bar{x}_2 - x_{i,2})(y_{i,1} - \bar{y}_1)\}, \\ \bar{X} &= \begin{bmatrix} \bar{x}_1 \\ \bar{x}_2 \end{bmatrix} = \sum_{i=1}^3 \bar{X}_i, \quad \bar{Y} = \begin{bmatrix} \bar{y}_1 \\ \bar{y}_2 \end{bmatrix} = \sum_{i=1}^3 \bar{Y}_i \end{aligned}$$

$$\hat{R} = \begin{bmatrix} \cos \hat{\theta} & -\sin \hat{\theta} \\ \sin \hat{\theta} & \cos \hat{\theta} \end{bmatrix}, \quad \hat{T} = \begin{bmatrix} \hat{t}_1 \\ \hat{t}_2 \end{bmatrix} \quad (6.6)$$

If \hat{s} , $\hat{\theta}$, \hat{t}_1 and \hat{t}_2 are less than certain thresholds, then we take them as the parameters of the transformation between two potential corresponding triangles Δ_t and Δ_q . Otherwise, they are false correspondences. Based on the transformation $\hat{F}_{\hat{e}}(\bullet)$, where $\hat{e} = (\hat{s}, \hat{\theta}, \hat{t}_1, \hat{t}_2)$, $\forall j, j = 1, 2, 3, \dots, M$, we compute the number of corresponding points by

$$d = \arg \min_k \left\{ \left\| \hat{F}\begin{bmatrix} t_{j,1} \\ t_{j,2} \end{bmatrix} - \begin{bmatrix} q_{k,1} \\ q_{k,2} \end{bmatrix} \right\| \right\} \quad (6.7)$$

If d is less than a threshold T_d , then we define the points $[t_{j,1}, t_{j,2}]'$ and $[q_{k,1}, q_{k,2}]'$ as corresponding points. If the number of corresponding points based on $\hat{F}_{\hat{e}}(\bullet)$ is greater than a threshold T_n , then we define Δ_t and Δ_q as the corresponding triangles between the template and the query fingerprints. The identification score is simply the number of corresponding triangles.

6.4 EXPERIMENTS

We use NIST special fingerprint database 4 (NIST-4) [115] in our experiments, in which a large portion of the fingerprints is of poor quality and contain certain other marks. The size of the fingerprint images is 480×512 pixels with a resolution of 500 DPI. NIST-4 contains 2000 pairs of fingerprints. Some sample fingerprints in NIST-4 are shown in previous Chapters, e.g. Figures 2.4, 3.8, 4.5 and 5.10.

6.4.1 Classification results

We use the first 1000 pairs of fingerprints for training. In order to reduce the effect of overfitting, we use only the first 500 pairs to estimate the parameters for each class and use the entire training set to validate the training results. Since we want to compare the results of classification and indexing, we only test the second impression of the second 1000 pairs of fingerprints. The first impressions of the second 1000 pairs of fingerprints are used as templates in verification.

Table 6.1. Confusion matrix for 5-class classifications.

	R	L	W	A	T
R	180	1	6	2	5
L	5	188	6	1	10
W	1	3	187	0	2
A	1	3	0	208	6
T	14	2	1	4	172

We performed the experiments 10 times and took the best result as the learned composite operator. Table 6.1 shows the confusion matrix of our testing results on the second 1000 pairs of fingerprint in NIST-4. Note that, because of bad quality, the ground-truths of some fingerprints provided by NIST-4 contain 2 classes, e.g. the ground-truth labels of f0008_10 include class T and L. We only use the first ground-truth label to estimate the parameters of classifiers. However, in testing, we use all the ground-truth labels and consider it as a correct classification, if the classification result matches to one of the ground-truth labels. PCC is 92.8% for 5-class classifications. Table 6.2 shows the results on NIST-4 database reported in the literature. Considering that we have not rejected any fingerprints from NIST-4, our result is one of the best.

Table 6.2. Classification results on NIST-4.

Approaches	Class #	Error rate %	Reject rate %	Dataset	Comments
Karu and Jain 1996 [57]	5	14.6	zero	4000 images, no training	Decision based on topological information
Jain et al. 1999 [45]	5	14.6	1.8	Training: First 1000 pairs of images; Testing: Second 1000 pairs of images	KNN
	5	13.6			Neural Network
	5	10.0			KNN+NN, two stage classifier
Senior 2001 [94]	4	8.5	zero	Same as above	Neural Network fusion with priors
Yao et al. [120]	5	10.0	1.8	Same as above	SVM+RNN
Our approach	5	7.2	zero	Training: First 1000 pairs of images; Testing: Second impression of the second 1000 pairs of images	GP based learned features + Bayesian classifier

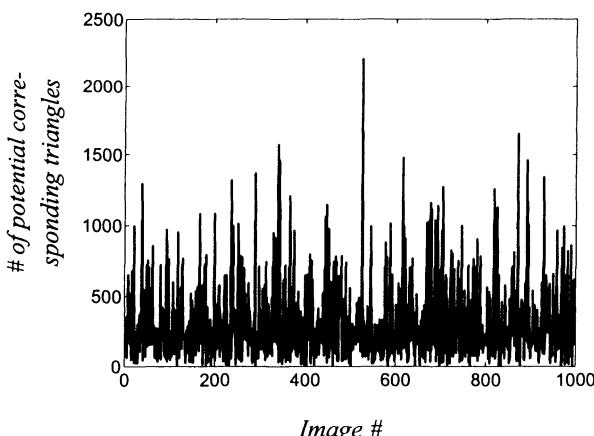


Figure 6.3. Number of potential corresponding triangles between each pair of genuine indexing.

6.4.2 Indexing Results

In order to compare the results between indexing and classification, we do the indexing experiments only on the second impressions of the second 1000 pairs of fingerprints.

Figure 6.3 shows the number of the potential corresponding triangles between each pair of genuine indexing in our experiment. The mean of this number is 299. Figure 6.4 shows the Correct Indexing Power (CIP), which is defined as the percentage of correctly indexed queries based on the percentage of hypotheses that need to be searched in the matching step. We observe that CIP increases as p , the percentage of the database searched, increases. The CIP are 83.3%, 88.1%, 91.1%, and 92.6% for p are 5%, 10%, 15%, and 20%, respectively. As p reached about 60%, the relation between CIP and p becomes linear.

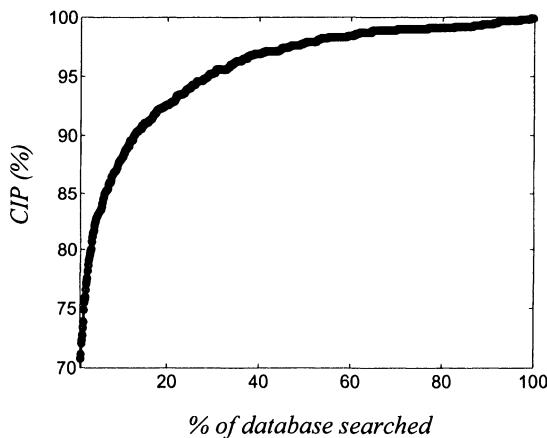


Figure 6.4. Indexing performance.

6.4.3 Identification Results

For classification, since the number of classes in fingerprint is small, we have to check more hypotheses in matching. For example,

the classification result of our approach is one of the best results reported in published papers, however, we can only classify fingerprints into 5 classes. Since each class is uniformly distributed in NIST-4, after classification, about 200 hypotheses need to be considered in verification. And, this number can not be tuned. As for indexing, since CIP varies according to the size of the search space, we have different performances of identification by indexing approach depending on the percentage of the database that is searched. Conceptually, each fingerprint as a query is verified against all the stored fingerprint templates. That is 1,000,000 verifications. Among them, 999,000 verifications are estimating False Acceptance Rate (FAR) and 1,000 verifications are for estimating Genuine Acceptance Rate (GAR). The Receiver Operating Characteristic (ROC) curve is defined as the plot of GAR against FAR. Based on different CIP, we have different ROCs for identification results for indexing based approach and only one ROC for classification based approach.

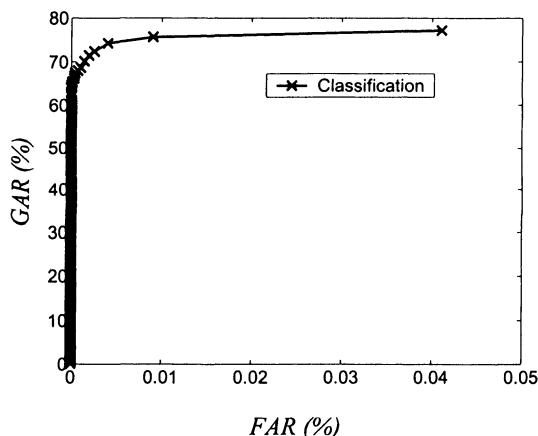


Figure 6.5. Identification results using classification based approach.

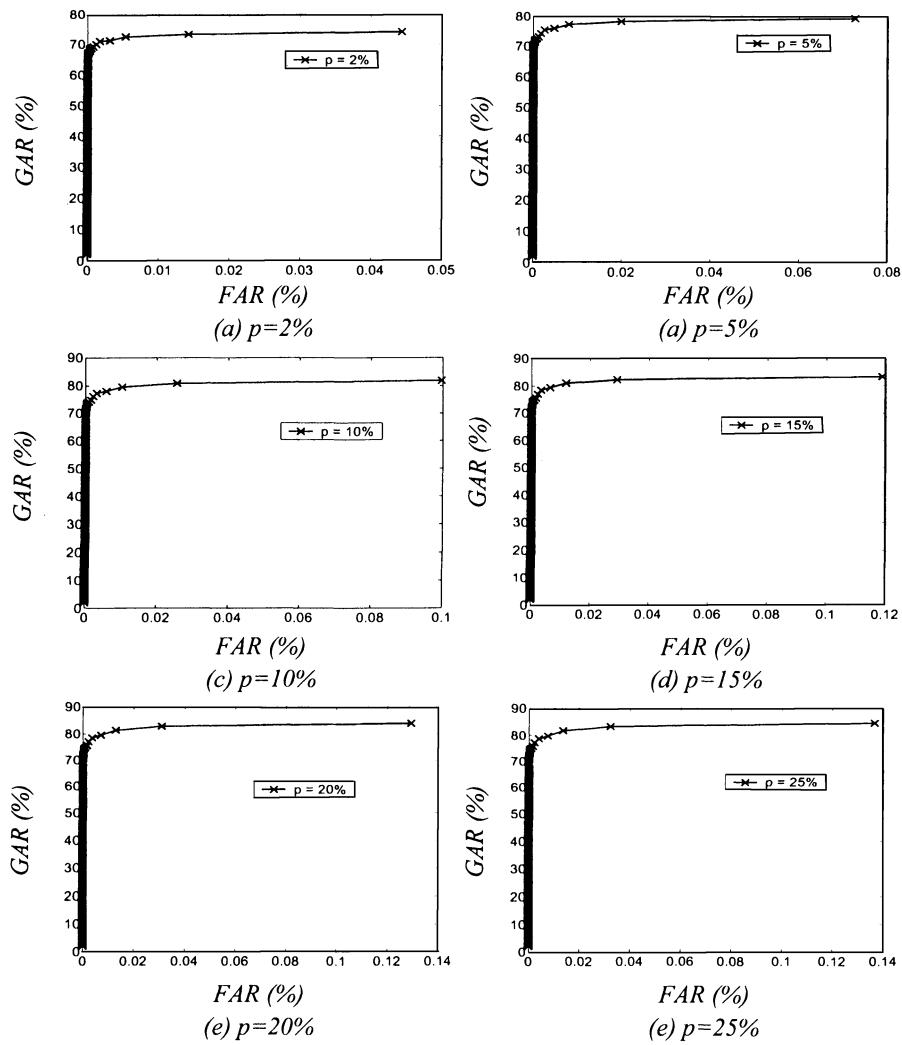


Figure 6.6. Identification results using indexing based approach.

Figure 6.5 and Figure 6.6 show identification results based on classification and indexing based approaches, respectively. An interesting aspect of the comparison is the computational cost of each of them. Using the classification based approach, GAR is 77.2% when FAR is $4.1 \times 10^{-2}\%$, while using the indexing based approach with $p=5\%$, GAR is 77.2% and FAR is $8.0 \times 10^{-3}\%$. It shows that in order to achieve similar GAR in identification, we only need to search 5% of the database by indexing based approach for identification, while classification based approach for identification may need to search 20% of the entire search space. FAR for indexing based approach is much less than that for classification based approach. The classes R, L, W, A and T are uniformly distributed in NIST-4. However, in nature, the frequencies of their occurrence are 31.7%, 33.8%, 27.9%, 3.7% and 2.9%, respectively. So, using the classification based approach the search space that need to be searched will be more than 30.0%, since there are fewer fingerprints that belong to A and T classes in nature than to other classes.

6.5 CONCLUSIONS

In this Chapter, we have compared the performance of two approaches for identification. One is the traditional approach that first classifies a fingerprint into one of the five classes (R, L, W, A, T) and then performs matching. The alternative approach is based on indexing followed by matching. Using state of the art highly competitive approach for classification, indexing and verification, we compared the performance of the two approaches for identification using NIST-4 fingerprint database. We find that indexing technique performs better considering the size of search space (5% vs. 20%) that needs to be searched. Also for the same GAR, FAR performance ($8.0 \times 10^{-3}\%$ vs. $4.1 \times 10^{-2}\%$) of indexing based approach is lower. Thus, the indexing based approach provides a potential alternative to the traditional classification based fingerprint identification approach.

Chapter 7

FUNDAMENTAL PERFORMANCE ANALYSIS – PREDICTION AND VALIDATION

7.1 INTRODUCTION

The foundations of modern fingerprint recognition were established by the research of Sir Francis Galton and Sir Edward Henry at the end of 19th century. Two fundamentally important conclusions were reached through extensive experiments on different age groups within different races (Halici et al. [35]):

- Fingerprints of a person are *permanent*, i.e., once the fingerprint is fully formed by the seventh month of fetal life, it won't change naturally;
- Fingerprints of individuals are *unique*, i.e., no two fingerprints are ever found to be exactly the same.

Both conclusions were the building blocks of research in this field over the last century.

After Galton and Henry, work on fingerprint recognition was extended and refined. In the early 20th century, fingerprints were formally used as valid signs of identity by law-enforcement agencies. Fingerprints have been accepted in the courts of several countries from the 19th century, e.g. Argentina (1892), India (1897), France (1902), and subsequently in England and the United States [124]. Now, in most countries' courts, fingerprint identification is taken positively for a defendant's guilt. People accepted fingerprint identification as science without any doubt. However, recently in *Daubert Hearing* in the U.S. court [124] on September 13th, 1999, the *Daubert Opinion* states that:

- Trial judge must still screen scientific evidence to ensure it is relevant and reliable;
- The focus, of course, must be solely on principles and methodology, not on the conclusions they generate;
- Factors the court should consider include: 1) testing and validation; 2) peer review; 3) rate of error; 4) general acceptance.

The opinion governs the admissibility of scientific evidence in U.S. federal court and many state and local jurisdictions have adopted it.

The second conclusion regarding the uniqueness of fingerprints is challenged because of the *Daubert Opinion*. Despite the fact that all fingerprint examiners believe in it and in reality it has not been found that two fingerprints are exactly the same, it is very difficult to prove it empirically. The millions of sets of fingerprints are never compared against one another for possible duplication. Judge Joyner presented the oral decision on *Daubert Hearing* on September 13th, 1999 [124]. The defense motion to exclude fingerprint evidence and testimony was denied. However, from *Daubert Hearing* to May 22, 2002, there are 37 legal challenges to fingerprints in the United States of America.

Because of their stability, minutiae, including endpoint and bifurcation, have been adopted in most existing fingerprint recognition systems. Generally, the similarity of two fingerprints depends on the number of corresponding minutiae. The larger the number of the cor-

responding minutiae, the more similar the two fingerprints are. Thus, we can define the *error rate* of fingerprint matching as:

The probability of two randomly chosen fingerprints, which are impressions of different fingers, has sufficient similarity.

Obviously, the lower the probability, the more unique the different fingerprints are.

Fingerprint recognition performance is typically determined experimentally. The major limitations of experimental performance evaluation are (Boshra and Bhanu [10]):

- It does not tell us why this is the expected performance. Such an insight is important for designing better recognition systems, as it can provide fundamental answers to questions such as:
 - 1) what are the performance limits?
 - 2) How much distortion can be tolerated without degrading performance?
 - 3) Is a given matching criterion sufficient to achieve desired levels of performance?
 - 4) How many model objects can be accommodated without substantially degrading performance?

Without a fundamental understanding of these questions, design of object recognition systems will remain an art rather than a science;

- The performance obtained experimentally depends on the algorithm and *actual* implementation of the recognition system, which may result in significant difference of performance.

Fingerprints have long been used for person authentication. However, there is not enough scientific research to explain the probability that two fingerprints, which are impressions of different fingers, may be taken as the same one. In this Chapter, we propose a formal frame-

work to estimate the fundamental algorithm independent error rate of fingerprint matching [107]. Unlike a previous work [81], which only measures minutiae's positions and orientations, in our model, we account for ridge counts between different minutiae as well as minutiae's positions and orientations. The error rates of fingerprint matching obtained by our approach are significantly lower than that of previously published research. Results are shown using NIST-4 fingerprint database. These results contribute towards making fingerprint matching a science and settling the legal challenges to fingerprints.

7.2 RELATED RESEARCH AND CONTRIBUTIONS

7.2.1 Related Research

There have been several research efforts for analyzing the performance of feature-based object recognition. Most of these efforts address the problem of discriminating objects from clutter. A representative sample of these approaches can be outlined as follows [10]:

- Grimson and Huttenlochar [34] used a statistical occupancy model for estimating the distribution of the fraction of consistent data/model feature pairs for an erroneous hypothesis. This distribution was obtained under the assumptions of bounded uncertainty and uniform clutter models. It was used to determine the minimum fraction of consistent feature pairs required to achieve a specific probability of false alarm;
- Sarachik [88] estimated probability distributions of weighted votes conditioned on hypothesis validity and invalidity. These distributions were obtained assuming a Gaussian uncertainty model, and uniform clutter and occlusion models. The likelihood-ratio test was used to decide on whether to accept the given hypothesis;
- Alter and Grimson [2] calculated the likelihood-ratio test as a decision criterion. This work considered both bounded and

- Gaussian uncertainty models, in addition to uniform clutter and occlusion models;
- Lindenbaum [65] extended the modeling of clutter to include objects of known shapes as well as random features. This hybrid model, which considered the similarity between clutter and model objects, was used to determine bounds on the number of features required to reliably distinguish objects, at any poses, from clutter;
 - Lindenbaum [64] presented a probabilistic method for deriving bounds on the number of features required to achieve recognition with a certain degree of confidence. This method considered similarity between objects, bounded uncertainty and occlusion. Extreme-case analysis was used to model the interaction between similarity and occlusion, thus leading to the generation of relatively loose bounds;
 - Boshra and Bhanu [10] extended the above efforts by addressing the problem of object/object discrimination. The method they presented considered data distortion factors such as uncertainty, occlusion and clutter, in addition to object similarity. A target recognition task, involving real synthetic aperture radar images and point features, was used for validating the proposed method. For all the test sets, the results showed that the proposed method consistently predicted reasonable tight bounds on actual performance.
 - Bhanu and Han [13] proposed a kinematics-based approach to recognize human by gait. A Bayesian based statistical analysis is performed to evaluate the discriminating power of extracted features. Through probabilistic simulation, they not only predict the probability of correct recognition (PCR) with regard to different within-class feature variance, but also obtain the upper bound on PCR with regard to different human silhouette resolution. In addition, the maximum number of people in a database is obtained given the allowable error rate.
 - Some biometrics experts believe that iris is one of the biometric signs, whose theoretical error rate is lower than that of finger-print. Daugman [24] presented a iris recognition approach,

which is based on using quadrature 2D Gabor wavelets. Based on the experimental results, Daugman also showed the theoretical error rates of iris recognition as a function of the decision criterion employed.

The research on the performance of fingerprint recognition belongs to one of the two categories:

- 1) Find the probability of the amount of details presented in a single fingerprint;
- 2) Find the probability of the amount of details presented as correspondences in two different fingerprints.

Galton [32] assumed that 24 independent square regions could cover a fingerprint and he could correctly reconstruct any of the regions with a probability of $1/2$ by looking at the surrounding ridges. Accordingly, Galton's formulation of the uniqueness of a fingerprint is given by: $(1/16) \times (1/256) \times (1/2)^{24}$, where $1/16$ is the probability of the occurrence of a fingerprint type, and $1/256$ is the probability of occurrence of the correct number of ridges entering and exiting each of the 24 regions. Most research on fingerprint performance is similar to Galton's work, which measures the amount of details in a single fingerprint. A more detailed introduction to this kind of work can be found in Pankanti et al. [81].

In Bhanu and Tan [15], we use a model-based approach to find the local correspondences between two different fingerprints. An analysis of the probability of finding false correspondences is also introduced. First, we compute S , the size of the search space, and then compute S_1 , the size of finding one correspondence within some uncertainty area. Based on the assumption that the models are uniformly distributed in the search space, the probability of finding one correspondence is S_1/S . Using Binomial distribution and Poisson approximation, it can be extended to compute the probability of finding more than one correspondences between two different fingerprints. Problems with this approach are: 1) Assumption that models are uniformly distributed in the

search space is not realistic; 2) Estimation of error rate in this approach depends on the recognition algorithm used.

An important work was presented in Pankanti et al. [81], which measures the amount of detail needed to establish correspondence between two fingerprints. To estimate the probability of correspondence, they make the following assumptions:

- Only endpoint and bifurcation are considered and they do not distinguish them;
- Minutiae are distributed in a fingerprint uniformly and no two minutiae are very close to each other, so there is no overlap between their uncertainty areas;
- Correspondence of a minutiae pair is independent and each correspondence is equally important;
- Fingerprint quality is not explicitly taken into account.

The probability that there are exactly ρ corresponding minutiae between n query minutiae and m template minutiae is given by
$$\binom{m}{\rho} \binom{W-m}{n-\rho} / \binom{W}{n}$$
, where $W=A/C$ and is assumed to be an integer ($A>>C$), A is the total area of overlap between the template and query fingerprints after reasonable alignment, and C is the area of tolerance for uncertainty. Accordingly, the probability of matching q minutiae in both position and direction is given by

$$\sum_{\rho=q}^{\min(m,n)} \left(\binom{m}{\rho} \binom{W-m}{n-\rho} / \binom{W}{n} \right) \times \binom{\rho}{q} \times l^q (1-l)^{\rho-q}$$

where l is the probability of two position-matched minutiae having similar direction.

The key problem with this model is that it does not exploit the non-geometric relations (e.g., ridge count) between different minutiae. Ridge counts are important features for fingerprint recognition [52].

7.2.2 Contributions

Following are the major contributions of our work:

- a) We develop a formal framework for estimating the error rate of fingerprint recognition, so that we may have a fundamental insight into the performance of fingerprint recognition when the triplet based algorithm is used. In our model, we measure the relations between different minutiae. We address the error rate estimation problem in the following context:
 - Both template and query fingerprint are represented by locations of 2D point features that are discretized at some resolution;
 - An instance of a query data is assumed to be obtained by applying some 2D transformations to the template data, considering positional uncertainty, occlusion and clutter. It has been a challenge to model these factors in a single approach for performance prediction. For an approach in the context of object recognition in synthetic aperture radar imagery, see [11];
- b) Our estimation of error rate is fundamental with respect to the pairs or triplets of minutiae features as a basic representation unit. We validate the performance using entire NIST-4 fingerprint database. The performance characterizations of our approach is basic and will have an impact on various point matching problems and applications in computer vision and pattern recognition.

7.3 TECHNICAL APPROACH

In our model, we not only measure minutiae's positions and orientations, but also the relations between different minutiae to find the probability of correspondence between fingerprints. To estimate the probability, we make the assumptions similar to those in Pankanti et al. [81]. We also assume that the template and query fingerprints are

well aligned. Note that this assumption is not explicitly stated in [81], although it is one of the assumption there.

Suppose

- 1) The overlap area of the template and query fingerprints is A ;
- 2) The number of minutiae in the template and query fingerprints are m and n , respectively;
- 3) The uncertainty area of a minutia is C , $C=\pi r^2$, where r is the radius of the uncertain area;
- 4) The probability of matching one minutia in the query fingerprint with one minutia in the template fingerprint is p , $p=C/A$;
- 5) The number of corresponding minutiae in the query fingerprint is k .

Figure 7.1 shows the relations among these variables.

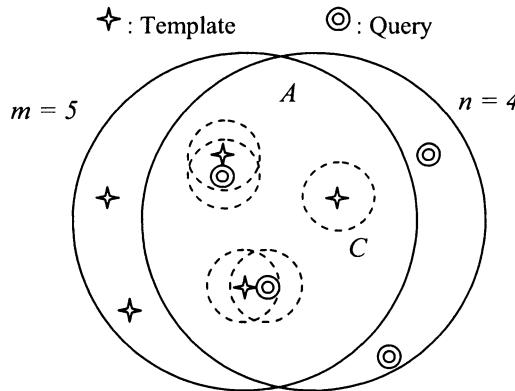


Figure 7.1. Illustration of the variables. The matching takes place from query to template.

Obviously, these k corresponding minutiae in the query fingerprint may correspond to i minutiae in the template fingerprint, where $1 \leq i \leq k$ (Figure 7.1 only shows the case when $i=k$). We can find $P_{M,N}\{M=i\}$,

$N=k\}$, the probability of k minutiae in the query fingerprint correspond to only i minutia in the template fingerprint, by the following procedure:

1) The probability that one minutia in the query corresponds to a given minutiae in the template is p and the probability that one minutia does not correspond to any given minutiae in the template is $(1-mp)$. Thus, the probability that k minutiae in the query fingerprint correspond to a given minutia in the template fingerprint is p^k , and the probability of $(n-k)$ minutiae in the query fingerprint that do not correspond to any given minutia in the template fingerprint is $(1-mp)^{n-k}$. So, when $i = 1$, if the given minutia can be any minutia in the template fingerprint, then we have

$$i = 1, P_{M,N} \{M = 1, N = k\} = \binom{n}{k} \binom{m}{1} (1 - mp)^{n-k} p^k$$

2) The probability that any one of k minutiae in the query corresponds to any one of two given minutiae in the template is $2p$, and they can not both correspond to only one minutia. Thus, the probability of k minutiae in the query fingerprint correspond to two given minutiae (not only one minutia) in the template fingerprint is $(2p)^k - 2p^k$, and the probability of $(n-k)$ minutiae in the query fingerprint that do not correspond to any one of the two given minutiae in the template fingerprint is $(1-mp)^{n-k}$. So, when $i = 2$, if two given minutiae can be any two minutiae in the template fingerprint, then we have

$$i = 2, P_{M,N} \{M = 2, N = k\} = \binom{n}{k} \binom{m}{2} (1 - mp)^{n-k} \left[(2p)^k - \binom{2}{1} p^k \right]$$

3) Similarly, we have

$$i = 3, P_{M,N} \{M = 3, N = k\} = \binom{n}{k} \binom{m}{3} (1 - mp)^{n-k} \left[(3p)^k - \binom{3}{2} \left[(2p)^k - \binom{2}{1} p^k \right] - \binom{3}{1} p^k \right]$$

4) In general, we can find the probability of k minutiae in the query fingerprint corresponding to only i minutia in the template fingerprint as:

$$\begin{aligned} P_{M,N} \{M = i, N = k\} &= \binom{n}{k} \binom{m}{i} (1 - mp)^{n-k} H_i \\ H_1 &= p^k, \text{ and } H_i = (ip)^k - \sum_{j=1}^{i-1} \left\{ \binom{i}{j} H_j \right\} \text{ for } i \geq 2 \end{aligned} \quad (7.1)$$

where N is the number of corresponding minutiae between the template and query fingerprints.

In equation (7.1), we only consider the position of a minutia in the matching. If we consider both position and direction of a minutia in the matching at the same time, then we have,

$$P_{M,N} \{M = i, N = k\} = \binom{n}{k} \binom{m}{i} (1 - mp_d)^{n-k} H_i \quad (7.2)$$

where $H_1 = p_d^k$, $H_i = (ip_d)^k - \sum_{j=1}^{i-1} \left\{ \binom{i}{j} H_j \right\}$ for $i \geq 2$ and $k \geq i$,

$$p_d = \frac{p}{360} \times 2r_l, \text{ and } r_l \text{ is the uncertainty in the minutiae directions.}$$

Figure 7.2 explains p_d .

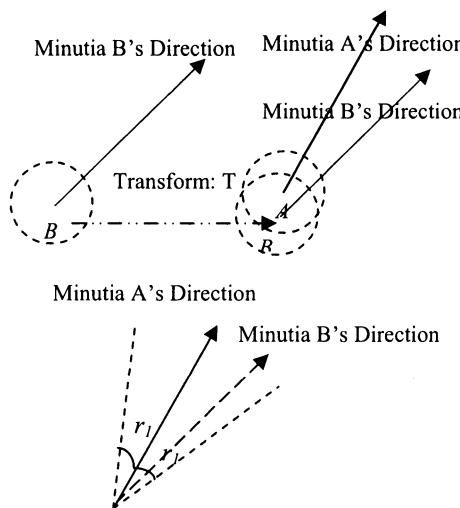


Figure 7.2. After transform T , minutiae B is in minutiae A's uncertainty area. If minutiae B's ridge direction is in minutiae A's directional uncertainty area, then we think their directions to be the same and minutiae A and B are the corresponding minutiae.

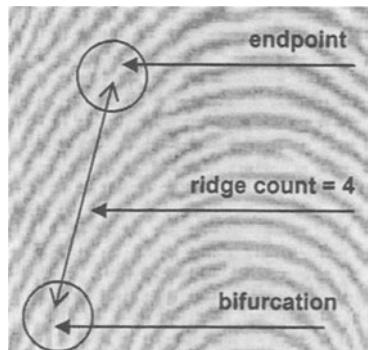


Figure 7.3. Examples of minutiae and ridge count.

In the following, we explicitly consider ridge counts in our model. Figure 7.3 shows an example of ridge count between two minutiae, which defines the relations between two different minutiae. If we measure ridge counts, we expect the matching to be more reliable and the performance of the recognition system will be better, since there are more constraints between a pair of minutiae. Ridge count changes with distance between two minutiae, average distance between ridges, orientation of ridges, etc. It is difficult to build a model of ridge counts. However, it is not necessary for us to know the model of ridge counts explicitly. What we need to know is p_r , the probability that two ridge counts, which are between a pair of corresponding minutiae in the template and query fingerprints respectively, can be considered to have the same ridge count. Because of bad quality fingerprints, it is not easy to develop an effective algorithm to detect ridge counts. Thus, we have to consider the uncertainty of ridge counts in matching. For simplicity, we assume:

$$p_r = \frac{2T_r + I}{r_{\max} - r_{\min}} \quad (7.3)$$

where r_{\max} and r_{\min} are the maximum and minimum ridge counts between two minutiae in a fingerprint, respectively, and T_r is threshold to tolerate the uncertainty of ridge counts in ridge counts matching.

7.3.1 Two-Point Model

First, we consider a simple case, which only uses a pair of minutiae to construct the model. Each pair of minutiae can form a side, and i minutiae can form $g_2 = i C_2$ sides. Suppose M_2 is the number of models, which are constructed by pairs of minutiae, that are supposed to satisfy all the criteria in the matching and $P_{2,M}\{\bullet\}$ is the probability of 2-point model, then we have

$$P_{2,M} \{M_2 = s \mid M = i\} = \begin{cases} \binom{g_2}{s} p_r^s (1 - p_r)^{g_2 - s}, & 0 \leq s \leq g_2 \\ 0, & \text{otherwise} \end{cases} \quad (7.4)$$

In this case, the expectation of the number of minutiae pairs that satisfy all the criteria in the matching when $N=k$ is:

$$\begin{aligned} E\{M_2 = s, N = k\} &= \sum_{i=2}^{\min(m,n,k)} P_{2,M} \{M_2 = s, M = i, N = k\} \\ &= \sum_{i=2}^{\min(m,n,k)} [P_{M,N} \{M = i, N = k\} \times \\ &= \sum_{i=2}^{\min(m,n,k)} [P_{M,N} \{M = i, N = k\} \times \\ &\quad P_{2,M} \{M_2 = s \mid M = i, N = k\}] \end{aligned} \quad (7.5)$$

Thus, the Probability Distribution Function (*PDF*) of 2-point model is

$$E\{M_2 = s\} = \sum_{k=2}^n \{E\{M_2 = s, N = k\}\} \quad (7.6)$$

7.3.2 Three-Point Model

A more complex case uses 3 points (triplets of noncolinear minutiae) to build the model and measures each side of every 3-point model to predict the performance. Since for each 3-point model, there are three sides and those sides are not independent, they may have effect on other models. It is difficult to get a closed-form equation for $P_{3,M}\{M_3=s \mid M=i\}$, where M_3 is the number of models that are constructed by triplets of minutiae that satisfy ridge count constraints in the matching and $P_{3,M}\{\bullet\}$ is the probability of 3-point model. In this work, we use a statistical method to estimate $P_{3,M}\{M_3=s \mid M=i\}$ and its 95% confidence interval $P_{3,M,CI=95\%}\{M_3=s \mid M=i\}$.

$$\begin{aligned}
 E_{CI=95\%} \{M_3 = s, N = k\} &= \sum_{i=3}^{\min(m,n,k)} P_{3,M,CI=95\%} \{M_3 = s, M = i, N = k\} \\
 &= \sum_{i=3}^{\min(m,n,k)} [P_{M,N} \{M = i, N = k\} \times \\
 &\quad P_{3,M,CI=95\%} \{M_3 = s \mid M = i, N = k\}] \\
 &= \sum_{i=3}^{\min(m,n,k)} [P_{M,N} \{M = i, N = k\} \times \\
 &\quad P_{3,M,CI=95\%} \{M_3 = s \mid M = i\}]
 \end{aligned} \tag{7.7}$$

where $CI=95\%$ means the 95% confidence interval.

Based on equation (7.7), we can compute the expectation of the probability distribution function (*PDF*) of the number of 3-point models as,

$$E_{CI=95\%} \{M_3 = s\} = \sum_{k=3}^n \{E_{CI=95\%} \{M_3 = s, N = k\}\} \tag{7.8}$$

This is an important result. Although we may find the *PDF* of the number of n -point model for $n > 3$, we believe that 3-point model is simple and may provide an effective constraint for evaluating the performance of fingerprint matching.

- **Simulation:** Suppose

- 1) There are s triangles;
- 2) In order to construct these s triangles, the minimum number of sides and points that we need are h_s and h_p ;
- 3) h_t is the number of possibilities that can construct s triangles using h_s sides and h_p points;
- 4) h_n , the maximum possible number of feature points which are needed to construct the s triangles.

Figure 7.4 shows the procedure that we use to find h_s , h_p and h_t for different s . We used $h_n = 8$ in our simulations. Table 7.1 shows the

results of simulation procedure shown in Figure 7.4 for $1 \leq s \leq 15$. Figure 7.5 shows an example of constructing 2 triangles using the minimum number of 5 sides and 4 points.

- 1) $h_s=10000, h_t=0, h_p=0;$
- 2) Simulate $\binom{h_n}{2}$ loops, the variables associated with those loops are $L_i, i = 1, 2, 3, \dots, \binom{h_n}{2}$, and L_i 's values are from 0 to 1. Within all these loops, do step 3) to 10);
- 3) Suppose L_i represent the status of the sides, which connect any two of the h_n feature points, 1 represents the ridge count of the side is within the threshold, otherwise, L_i is 0. Find t_s , the number of triangles L_i forms, t_s , the associated sides, which is the number of 1s in L_i , and t_p , the number of feature points associated with those t_s sides;
- 4) **If** ($t_s == s$) **Then**
- 5) **If** ($t_s < h_s$) **Then** $h_s=t_s, h_t=0, h_p=t_p$; **End If**;
- 6) **If** ($t_s == h_s$) **Then**
- 7) **If** ($t_p == h_p$) **Then** $h_t=h_t+1$; **End If**;
- 8) **If** ($t_p < h_p$) **Then** $h_t=0, h_p=t_p$; **End If**;
- 9) **End If**;
- 10) **End If**;

Code for simulating loops:

- 1) $c = \binom{h_n}{2};$
- 2) **For** ($t = 0; t < c; t++$) {
- 3) *i[t] = 0; I[t] = 1;* }
- 4) **While** (1) {
- 5) **For** ($t = c - 1; t \geq 0; t--$) {
- 6) **If** ($i[t]++ \neq I[t]$) **Then** break; **End If**;
- 7) *i[t] = 0;* }
- 8) **If** ($t < 0$) **Then** break; **End If**; //end of loop
- 9) //do anything with $i[]$, $i[]$ is the value for all the c variables
- 10) } // While

Figure 7.4. Basic procedure to estimate the lower bound of $P_{3,M}\{M_3=s \mid M=i\}$.

Table 7.1. Simulation results for different s triangles.

s	1	2	3	4	5	6	7	8
h_s	3	5	7	6	8	10	9	11
h_p	3	4	5	4	5	6	5	6
h_t	1	6	70	1	30	900	10	585
s	9	10	11	12	13	14	15	
h_s	12	10	12	13	13	15	16	
h_p	6	5	6	6	6	7	7	
h_t	180	1	60	45	60	5670	6552	

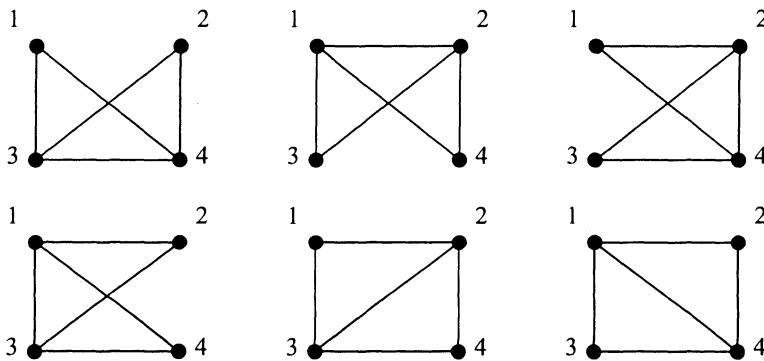


Figure 7.5. To construct 2 triangles, the minimum number of sides and points that are needed are 5 and 4, respectively. There are 6 variations of 4 points and 5 associated sides to form 2 triangles.

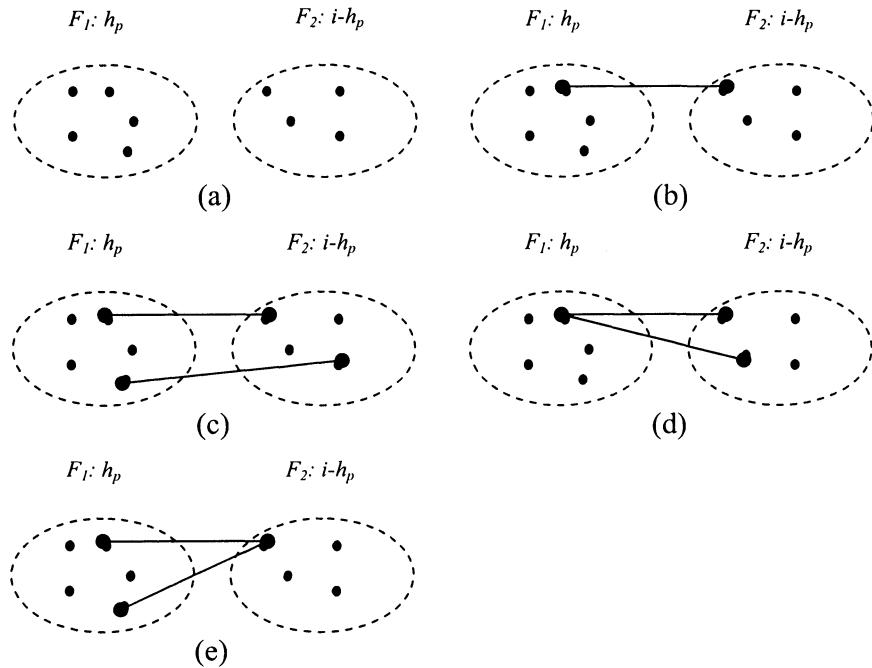


Figure 7.6. Relations between two sets of feature points F_1 and F_2 .

- **Analysis for Lower Bound:** Besides using simulation to find $P_{3,M}\{M_3=s \mid M=i\}$, we can also do an analysis and estimate a lower bound for it. For a given s , suppose there are i feature points, $i > h_p$, then we can divide these i feature points into two feature sets, F_1 and F_2 , which includes h_p and $(i-h_p)$ feature points, respectively. All those triangles are constructed in F_1 . Figure 7.6 shows some possible relations between F_1 and F_2 . Note that, for each pair of feature points, there is a probability p_s to connect them to form a side, for simplicity, we draw only a few sides in Figure 7.6.

For Figure 7.6(a), all the lines between F_1 and F_2 can not satisfy the ridge count constraints and there is no triangle in F_2 , we use P_a , given

below, as the lower bound of $P_{3,M}\{M_3=s \mid M=i\}$ for the case shown in Figure 7.6(a),

$$P_a = h_t \cdot \binom{i}{h_p} \cdot p_s^{h_p} \cdot (1-p_s)^{\binom{h_p}{2} - h_p + h_p(i-h_p)} \cdot f_a$$

$$\text{where } f_a > \sum_{i=0}^4 f_i, \quad f_0 = (1-p_s)^{s_2}, \quad f_1 = s_2 \cdot p_s \cdot (1-p_s)^{s_2-1},$$

$$f_2 = s_3 \cdot p_s^2 \cdot (1-p_s)^{s_2-2}, \quad f_3 = s_4 \cdot p_s^3 \cdot (1-p_s)^{s_2-3},$$

$$f_4 = s_5 \cdot p_s^4 \cdot (1-p_s)^{s_2-4}, \quad s_2 = \binom{i-h_p}{2}, \quad s_3 = \binom{s_2}{2},$$

$$s_4 = \binom{s_2}{3} - \binom{i-h_p}{3}, \text{ and } s_5 = \binom{s_2}{4} - \binom{i-h_p}{3} \cdot (s_2 - 3).$$

f_i is the probability that there are i lines that satisfy the constraints and there is no triangle in F_2 . Since $(1-p_s)$ is the probability that no side in F_2 satisfies the constraints and totally there are s_2 sides in F_2 , thus, we have $f_0 = (1-p_s)^{s_2}$. If there is one side in F_2 that satisfies the constraints, for a certain side, the probability is $p_s \cdot (1-p_s)^{s_2-1}$. Since the side can be any one in F_2 , we have $f_1 = s_2 \cdot p_s \cdot (1-p_s)^{s_2-1}$. Similarly, we can find the expressions for f_2, f_3 , and f_4 .

For Figure 7.6(b), only one line between F_1 and F_2 can satisfy the constraints and there is no triangle in F_2 , we use P_b as the low bound of $P_{3,M}\{M_3=s \mid M=i\}$ for the case shown in Figure 7.6(b):

$$P_b = h_t \cdot \binom{i}{h_p} \cdot p_s^{h_p+1} \cdot (1-p_s)^{\binom{h_p}{2} - h_p + h_p(i-h_p)-1} \cdot h_p \cdot (i-h_p) \cdot f_b$$

Similarly, we can find P_c, P_d and P_e corresponding to Figures 7.6(c) to 7.6(e) as:

$$P_c = h_t \cdot \binom{i}{h_p} \cdot \sum_{q=2}^{\min(h_p, i-h_p)} \left(p_s^{h_s+q} \cdot (1-p_s)^{\binom{h_p}{2} - h_s + h_p(i-h_p)-q} \cdot \binom{h_p}{q} \cdot \binom{i-h_p}{q} \cdot q! \right) \cdot f_c,$$

$$P_d = h_t \cdot \binom{i}{h_p} \cdot p_s^{h_s+2} \cdot (1-p_s)^{\binom{h_p}{2} - h_s + h_p(i-h_p)-2} \cdot h_p \cdot \binom{i-h_p}{2} \cdot f_d,$$

$$P_e = h_t \cdot \binom{i}{h_p} \cdot p_s^{h_s+2} \cdot (1-p_s)^{\binom{h_p}{2} - h_s + h_p(i-h_p)-2} \cdot (i-h_p) \cdot \binom{h_p}{2} \cdot f_e$$

where $f_d > (1-p_s)^{s_2} + (s_2-1) \cdot p_s \cdot (1-p_s)^{s_2-1} + (s_2-1) \cdot (s_2-2) \cdot p_s^2 \cdot (1-p_s)^{s_2-2}$
and $f_b = f_c = f_e = f_a$.

f_d is different from f_a . It is because for the case in Figure 7.6(d), a line in F_2 may have effect on the number of triangles. So, we have

$$P_{3,M} \{M_3 = s \mid M = i\} > \sum_{x \in \{a,b,c,d,e\}} P_x \quad (7.9)$$

Using equations (7.7), (7.8) and (7.9), we have a lower bound

$$E\{M_3 = s\} > \sum_{k=3}^n \left\{ \sum_{i=3}^{\min(m,n,k)} \{P_{M,N} \{M = i, N = k\} \times \sum_{x \in \{a,b,c,d,e\}} P_x\} \right\} \quad (7.10)$$

Equation (7.10) only provides us a loose lower bound. A much more detailed analysis is needed for obtaining a tight bound.

7.4 EXPERIMENTS

7.4.1 Database

We use NIST special fingerprint database 4 (*NIST-4*) [115] in our experiments, which is a publicly available fingerprint database. Since the fingerprints in *NIST-4* are collected by an ink-based method, a large portion of the fingerprints is of poor quality and contain certain other objects, such as characters and handwritten lines. The size of the fingerprint images is 480×512 pixels with a resolution of 500 DPI. *NIST-4* contains 2000 pairs of fingerprints. Each pair is a different impression of the same finger. The fingerprint is coded as a *f* or *s* followed by 6 numbers, which means the fingerprint image is the first or second impression of certain finger. Some sample fingerprints in *NIST-4* are shown in previous Chapters, e.g. Figures 2.4, 3.8, 4.5 and 5.10.

Table 7.2. Parameters used in estimation.

Parameters	Meaning	Value
r	Radius of uncertain area of a minutia	15 pixels
C	Uncertainty area of a minutia	225π pixels ²
A	Overlap area between template and query	512×480 pixels ²
p	Probability of having one pair of corresponding minutiae between template and query (position)	$\approx 2.9 \times 10^{-3}$
r_l	Uncertainty in minutiae directions	22.5°
p_l	Probability of having one pair of corresponding minutiae between template and query (position and orientation)	$\approx 3.6 \times 10^{-4}$
m	Number of minutiae in template	60
n	Number of minutiae in query	60
p_r	Probability that two ridge counts can be considered to be the same	0.15
T_r	Threshold to account for the uncertainty of ridge counts in matching	1

7.4.2 Parameters

The parameters we used in our estimation are shown in Table 7.2. They are based on our experience and published results by others [81] using NIST-4 database. In order to estimate p_r , we use the Genetic Algorithm (GA) based approach introduced in [102][103] to find the optimal transformation between each genuine matching in NIST-4 fingerprint database. Based on the optimal transformation, the pairs of corresponding minutiae in two fingerprints of each genuine matching are found. Then, we find P_{rc} , the distribution of ridge counts, and P_{rd} , the difference of ridge counts between two fingerprints of each genuine matching. Figure 7.7 and Figure 7.8 show these two distributions. From these figures, we have:

$$\begin{aligned}
 P_{rc}\{0 \leq \text{ridge counts} \leq 20\} &= 92.5\%, \\
 P_{rc}\{0 \leq \text{ridge counts} \leq 30\} &= 99.3\%, \\
 P_{rd}\{0 \leq \text{ridge counts difference} \leq 1\} &= 78.0\%, \\
 P_{rd}\{0 \leq \text{ridge counts difference} \leq 2\} &= 91.5\%.
 \end{aligned}$$

Obviously, different choices of the spread of ridge counts and ridge counts difference may result different estimation of p_r . To be more conservative, we take the spread of ridge counts and ridge counts difference as 20 and 1, respectively. That is, $r_{max} - r_{min} = 20$ and $T_r = 1$, so $p_r = 0.15$.

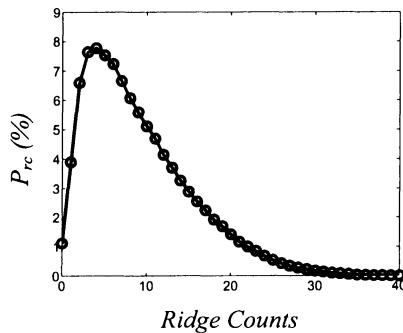


Figure 7.7. Distribution of ridge counts.

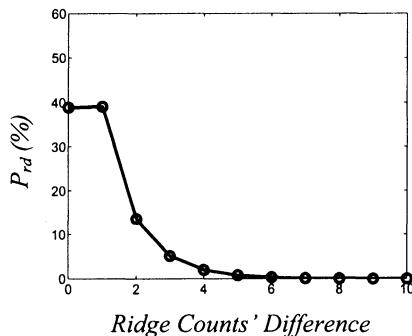


Figure 7.8. Distribution of ridge counts' difference.

7.4.3 Estimation of $P_{3,M}\{M_3=s \mid M=i\}$

It is computationally expensive to search all possible situations to find $P_{3,M}\{M_3=s \mid M=i\}$. As i increases, the size of the search space will explode to 2^q and $q=iC_2$. Therefore, we use simulations to estimate its mean and 95% confidence interval. The basic simulation procedure is shown in Figure 7.9. We performed the tests 100 times. For each test, we repeat the basic simulation procedure 10^8 times. Figure 7.10 shows the 3D surface of estimated $P_{3,M}\{M_3=s \mid M=i\}$. Figure 7.11 shows the detail of $P_{3,M}\{M_3=s \mid M=i\}$ for different value of M with their 95% confidence interval. From Figure 7.11, we observe that the 95% confidence interval of the estimated distribution is very small, i.e. the maximum length of 95% confidence interval for each M in Figure 7.11 are 7.9×10^{-6} , 2.3×10^{-5} , 4.1×10^{-5} , and 2.0×10^{-5} , respectively. Note that the results of the 100 tests are very consistent.

<p>INPUT: the number of corresponding feature points n_c; OUTPUT: s_t, the number of corresponding triangles between the template and query fingerprints;</p>
--

- 1) **For** $i = 1$ **to** n_c-1 ; **For** $j = i+1$ **to** n_c
- 2) Generate the ridge counts for template and query: $tr_{i,j}$ and $qr_{i,j}$;
- 3) **Next j; Next i**
- 4) $s_t=0$;
- 5) **For** $i = 1$ **to** n_c-2 ; **For** $j = i+1$ **to** n_c-1 ; **For** $k = j+1$ **to** n_c
- 6) **If** ($| tr_{i,j} - qr_{i,j} | \leq T_r$ and $| tr_{i,k} - qr_{i,k} | \leq T_r$ and $| tr_{j,k} - qr_{j,k} | \leq T_r$)
- 7) $s_t=s_t+1$;
- 8) **End if**;
- 9) **Next k; Next j; Next i**

Figure 7.9. Basic simulation procedure.

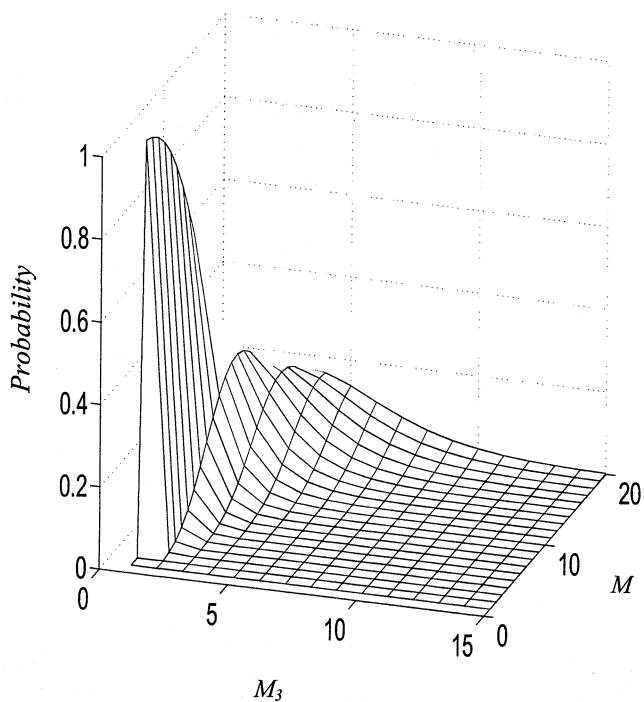


Figure 7.10. Estimated $P_{3,M}(M_3=s | M=i)$.

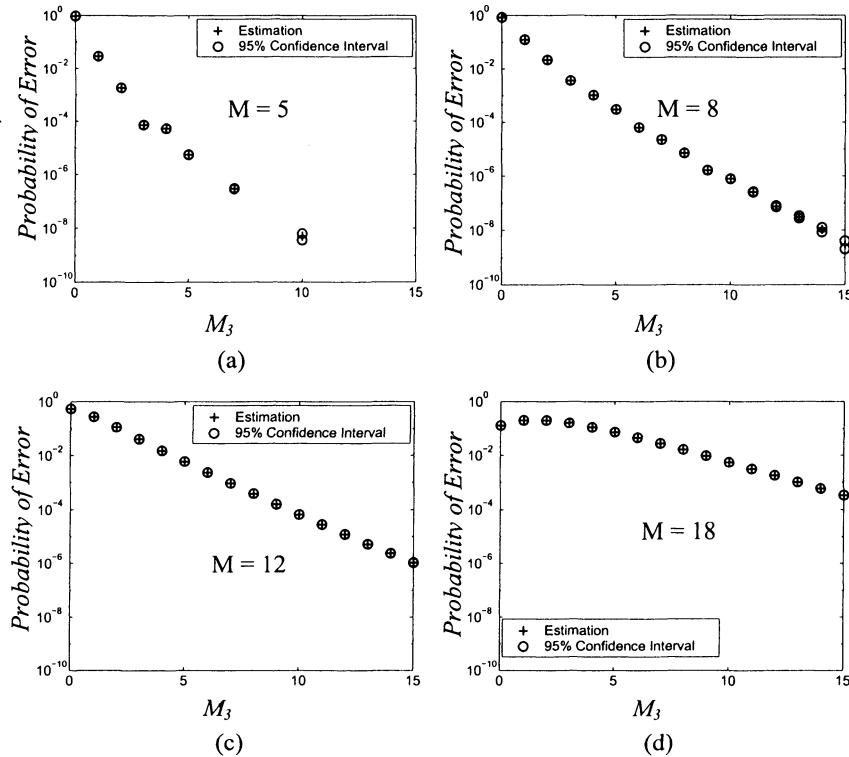


Figure 7.11. Simulation results of $P_{3,M}\{M_3=s \mid M=i\}$ and its 95% confidence intervals for various M : (a) $M=5$; (b) $M=8$; (c) $M=12$; (d) $M=18$.

7.4.4 Results

Figure 7.12 and Figure 7.13 show the distributions of the 2-point model and 3-point model, respectively. From these figures, we observe that, the 3-point model has more discriminating power than the 2-point model. For example, when $M_2=M_3=10$, $E_{CI=95\%}\{M_3=10\}=1.4\times10^{-10}$ while $E\{M_2=10\}=2.6\times10^7$. Note that the minimum number of minu-

tiae, which can have $M_2=M_3=10$, is 5. It also means that if we use the 3-point model for matching, the probability, with which we can find 10 similar 3-point models between two randomly chosen fingerprints is about 1 in 7.1 billion. Of course, the probability varies according to the parameters we use in our experiments. Table 7.3 shows the error rates for different m , n and s . We observe that if all the other parameters are fixed, the smaller the number of minutiae in the template and query fingerprints, the lower the probability that they are similar for the same s . Note that the value for the probability $p_r=0.15$ is a kind of simplification, since in general the ridge count between two minutiae depends on the distance between them, average distance between ridges, ridge's direction, resolution of fingerprint images, etc.

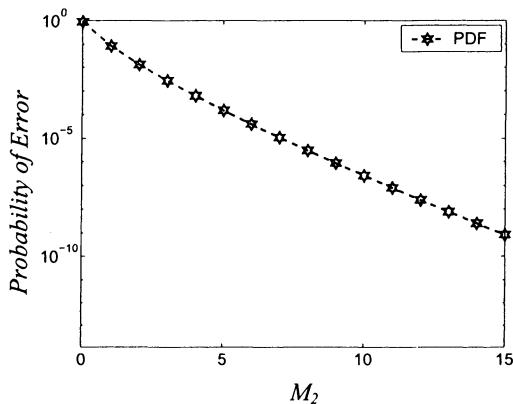


Figure 7.12. PDF of 2-point model's error rate.

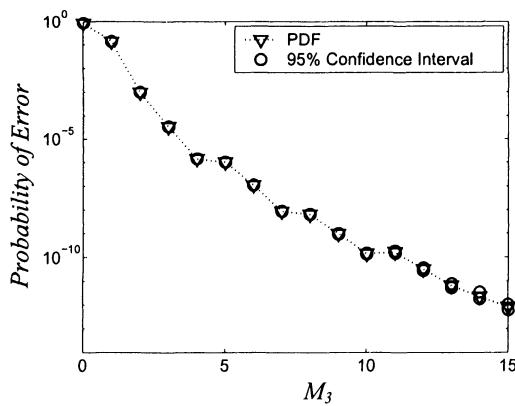


Figure 7.13. PDF of 3-point model's error rate.

Table 7.3. Estimated error rate for different m , n and s .

$m = n$	$E_{CI=95\%}\{M_3=s\}$			
	8	10	12	15
60	6.5×10^{-9}	1.4×10^{-10}	3.2×10^{-11}	8.5×10^{-13}
45	3.2×10^{-10}	2.8×10^{-12}	8.2×10^{-13}	1.2×10^{-14}
36	3.0×10^{-11}	1.3×10^{-13}	4.7×10^{-14}	4.4×10^{-16}
20	5.1×10^{-14}	4.1×10^{-17}	2.0×10^{-17}	4.1×10^{-20}

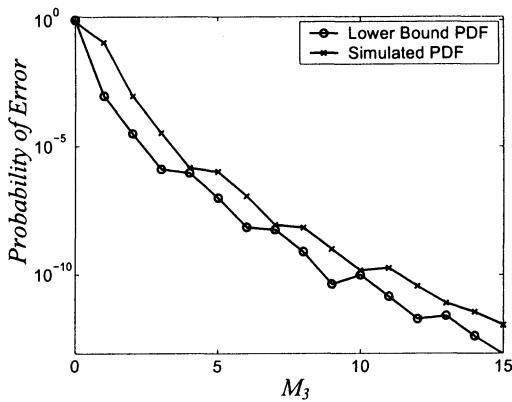


Figure 7.14. Lower bound of the performance.

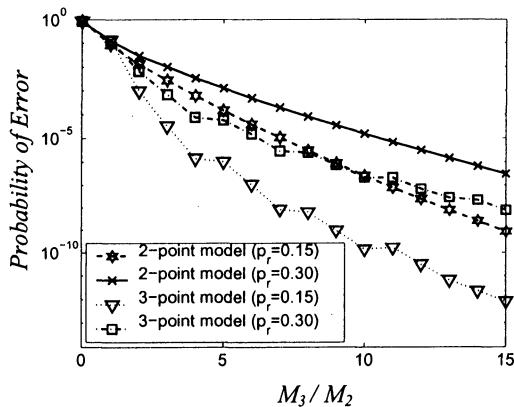


Figure 7.15. Comparison of different models' estimation.

Figure 7.14 shows the lower bound of the performance estimated by analysis. At certain points, analysis results get a good estimation, while at other points there is a deviation. The reason for this is that the analysis is an approximation, it takes into account the simplified cases while the real cases are quite complex. Figure 7.15 shows the comparison of the *PDFs* of 2-point model and 3-point models with differ-

ent p_r . Obviously, p_r is very important to determine the *PDF*. The more similar the relations among minutiae, the higher the probability that two fingerprints are from the same finger.

Figure 7.16 shows the *PDF* of the number of matching minutiae computed by the model presented in Pankanti et al.'s work ($m = 60$ and $n = 60$). Based on their model, if an expert strictly adheres to the *12-point* guideline [81], there is overwhelming identifying evidence to his/her testimony. The probability of an imposter matching which has 12 minutiae between two different fingerprints is 7.3×10^{-6} . Note that 7.3×10^{-6} is estimated using the parameters based on NIST-4. From Table 7.3, we observe that $P_{3,M\{M_3=12\}}=3.2 \times 10^{-11}$ ($m = 60$ and $n = 60$). We define:

$$r_3 = \frac{p_3}{p_1} \quad \text{and} \quad r_2 = \frac{p_2}{p_1} \quad (7.11)$$

where p_3 and p_2 are the probability of the number of 3-point model and 2-point model estimated in our approach respectively and p_1 is the probability of the number of minutiae shown in Pankanti et al.'s work. Figure 7.17 shows r_3 and r_2 for $p_r=0.15$ and $p_r=0.30$ respectively. We observe that the mean of r_3 and r_2 are 3.68×10^5 and 4.56×10^2 for $p_r=0.15$ and 9.34×10^2 and 2.37×10^1 for $p_r=0.30$. It shows that the error rate estimated by our approach is much less than that in Pankanti et al.'s work. The most important reason is that the relations among different minutiae are explicitly modeled in our approach, while in [81] minutiae are considered as individuals. It is a well known fact that relations between features are crucial for recognition.

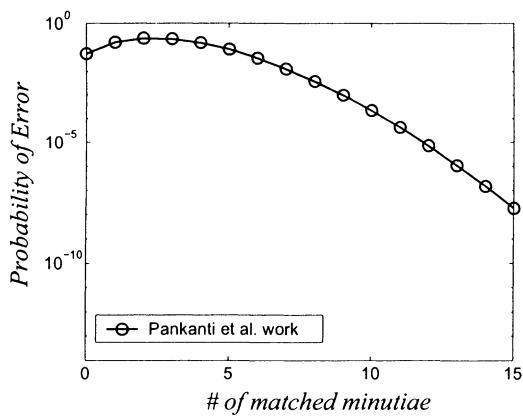


Figure 7.16. PDF of the number of matching minutiae in Pankanti et al.'s work [81].

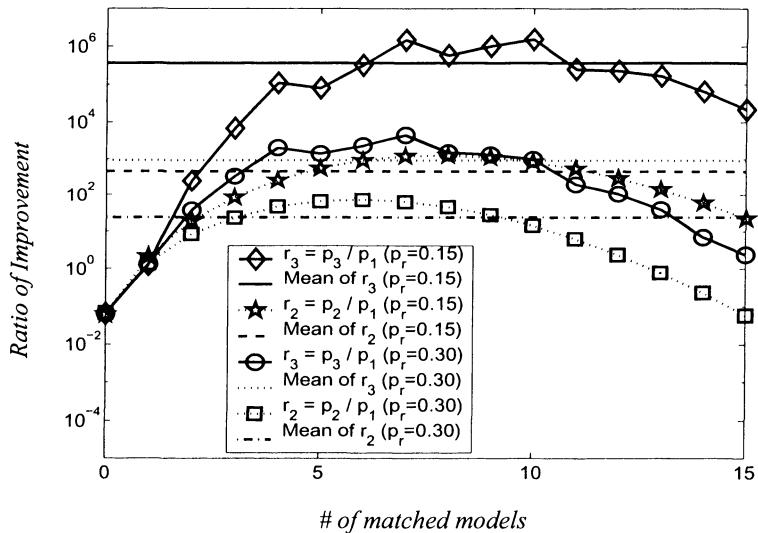


Figure 7.17. Ratio of estimation between our approach and Pankanti et al. [81].

7.4.5 Real Data

In [103], we present a Genetic Algorithm (GA) based approach to verify fingerprints. It finds the optimal transformation between two fingerprints. This approach takes into account not only the positions of minutiae, but also the ridges counts between minutiae. The fitness function of the approach is defined by a two-step process. First, it finds the number of corresponding minutiae between two fingerprints for a transformation proposed by the GA. If the number of corresponding minutiae is greater than a threshold, then it finds the number of corresponding triangles. Note that the number of corresponding triangles is the same as the number of 3-point models. We did $200,000 \times 5$ experiments to estimate the distribution of the number triangles of imposter matching. Figure 7.18 shows the comparison of the estimated distributions of error rate on real data and the result of our theoretical analysis. This shows that there is a significant gap between the theoretical error rates and those actually achieved.

The reasons for the difference between these two distributions are:

- The theoretical model assumes the template and query fingerprints are well aligned, while the alignment in real data matching has some errors;
- Because of the poor quality images, the minutiae orientation and ridge counts between different minutiae can not be detected precisely. Although we use PDF for ridge count, we do not consider the situation when ridge count can not be computed at all because of the poor quality of an image. There could be a function to account for it;
- The overlap area between the template and query fingerprints is less than the entire image area, so that the results on real data are worse than that of the theoretical results.

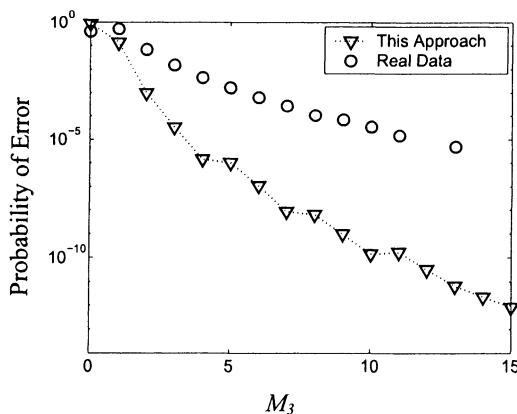


Figure 7.18. Comparison of theoretical results and real-data results.

7.4.6 Error Rates between Fingerprint and Iris

Biometric signs, which people usually use, include: voice, visible or infrared facial image, fingerprint, face profile, iris, ear, gait, key-stroke, signature, retina, etc. Among them, some experts believe that iris is one of biometric signs that has very low theoretical error rate. Daugman [24] presented a method to compute the theoretical error rate of iris recognition based on Hamming Distance acceptance thresholds. The error rates of iris recognition vary from 1 in 170 to 1 in 10^{13} as the Hamming Distance criterion varies from 0.40 to 0.26. From Table 7.3 with $m = 60$ and $n = 60$, we can see that the error rates for 10 and 15 3-point models in fingerprint are similar to that of using iris biometric whose Hamming Distance is 0.30 and 0.27, respectively. According to the analysis in [24], 0.30 is a low value of Hamming Distance, and it provides high confidence for iris recognition. So we claim that if we use 3-point models in fingerprint recognition, the theoretical error rate of fingerprint recognition will be similar to that of iris recognition. Thus, from theory, fingerprint is also one of the biometric signs that have a very low theoretical error rate.

7.5 CONCLUSIONS

People have long believed that everyone's fingerprints are unique. However, there is very little scientific research on estimating the error rate of fingerprint matching, which is very important for the insight into the performance of fingerprint recognition and its practical applications. Most previous works either measure the amount of details in a single fingerprint or only measure the details of minutiae in corresponding fingerprints individually. To the best of our knowledge, this work is the first, which not only measures the amount of detail needed to find correspondence between two fingerprints, but also measures the relations between different minutiae. Compared with a previously published research [81], on the average, our model reduces the error rate by a factor of 4.56×10^2 for 2-point model and 3.68×10^5 for 3-point model, respectively. The comparison between the theoretical result and the real-data result, which uses entire NIST-4 fingerprint database, shows future research work on fingerprint matching in real-world applications is needed, so that the performance of the fingerprint matching algorithm can be close to the theoretical result. The theoretical result of our work provides a framework to find the limits of error rate for fingerprint matching. It helps people to understand that fingerprint biometric is a emerging scientific field and fundamental performance bounds help towards settling the legal challenges to fingerprints. Since our approach is independent of the implementation of the triplet based recognition algorithm, it can be used to estimate the error rate of other point based recognition systems in computer vision and pattern recognition.

Chapter 8

SUMMARY AND FUTURE WORK

8.1 SUMMARY

The purpose of our research is to develop computational algorithms for an automatic fingerprint recognition system, which is able to achieve high performance with high confidence. We have developed several algorithms for this purpose. These algorithms include:

- **Templates based minutiae extraction algorithm:** We have developed a new technique, based on the use of learned templates, which statistically characterizes the minutiae features. Templates are learned from examples by optimizing a criterion function using Lagrange's method. To detect the presence of minutiae, templates are applied with appropriate orientations to the binarized fingerprints only at selected potential minutia locations. Performance measures, which evaluate the quality and quantity of extracted minutiae and their impact on an identification system, are used to evaluate the significance of learned templates. The experimental results on NIST-4 fingerprint da-

tabase show that learned templates can improve both the features and the performance of the identification system.

- **Triplets of minutiae based fingerprint indexing algorithm:** We have presented a model-based approach, which efficiently retrieves correct hypotheses using novel features of triangles formed by the triplets of minutiae as the basic representation unit. Triangle features that we use are its angles, handedness, type, direction and maximum side. Geometric constraints based on other characteristics of minutiae are used to eliminate false correspondences. Experimental results on NIST-4 fingerprint database show that our indexing approach efficiently narrows down the number of candidate hypotheses in the presence of translation, rotation, scale, shear, occlusion and clutter. We also perform scientific experiments to compare the performance of our approach with another prominent indexing approach and show that the performance of our approach is much better.
- **Genetic Algorithm based fingerprint matching algorithm:** We have developed a fingerprint matching approach based on Genetic Algorithms (GA), which finds the optimized global transformation between two different fingerprints. In order to deal with low quality fingerprint images, which introduce significant occlusion and clutter of minutiae, we design the fitness function based on the local properties of each triplet of minutiae. The experimental results on NIST-4 fingerprint database not only show that the proposed approach can achieve good performance even when a large portion of fingerprints in the database are of poor quality, but also show that the proposed approach is better than another approach, which is based on mean-squared error estimation.
- **Genetic Programming based feature learning algorithm for classification:** We have developed a fingerprint classification approach based on a novel feature-learning algorithm. Unlike current research for fingerprint classification that generally uses

visually meaningful features, our approach is based on Genetic Programming (GP), which learns to discover composite operators and features that are evolved from combinations of primitive image processing operations. The experimental results show that our approach can find good composite operators to effectively extract useful features. Using a Bayesian classifier, without rejecting any fingerprints from NIST-4 database, the correct rates for 4 and 5-class classification are 93.3% and 91.2% respectively, which compare favorably and have advantages over the best results published to date.

- **Comparison of classification and indexing for identification:** We have developed a two-step fingerprint identification approach based on the triplets of minutiae. Features that we use to find the potential corresponding triangles include angles, triangle orientation, triangle direction, maximum side, minutiae density and ridge counts. In the first step, based on the number of corresponding triangles between the query fingerprint and the model database constructed offline, hypotheses are generated. In the second step, false corresponding triangles are eliminated by applying constraints to the transformation between two potential corresponding triangles. We have also integrated the classification algorithm into the fingerprint identification system. The extensive comparisons between indexing and classification techniques are performed. On NIST-4 database, the comparison shows that, although correct classification rate can be as high as 92.8% for 5-class problems, the indexing approach performs better based on size of search space and identification results.
- **Fundamental performance analysis of fingerprint matching:** We have developed a formal framework to estimate the fundamental error rate of fingerprint matching using pairs and triplets of minutiae features. Unlike a previous work which only measures minutiae's positions and orientations, in our model, we account for ridge counts between different minutiae as well

as minutiae's positions and orientations. The error rates of fingerprint matching obtained by our approach are significantly lower than that of previously published research. Results are shown using NIST special fingerprint database 4. These results contribute towards making fingerprint matching a science and help settling the legal challenges to fingerprints.

8.2 FUTURE WORK

Although the algorithms we have developed can achieve a good performance in fingerprint recognition, we believe that there are still many problems that need to be addressed to make automatic fingerprint recognition system more effective and efficient in real-world applications [118]. In the following, we list the possible directions, which can improve the performance of our algorithms:

- Fingerprint image quality greatly affects the performance of fingerprint recognition. If the recognition system can reject bad quality fingerprints before processing them, the recognition performance may be improved greatly. Generally, this is not possible in real-world scenarios and the algorithm has to make the most out of what is given to it. This is because until today, there is no good method to evaluate the quality of a fingerprint. It will be interesting to develop systematic methods with firm mathematical foundations to evaluate the quality of a fingerprint and classify them from very good to poor categories.
- All features in fingerprints, including minutiae and gray scale features, are equally weighted in recognition. Although some researchers attempt to combine different recognition approaches, to the best of our knowledge, none of them use different weights for different minutiae or different gray scale features in different areas. It is possible to improve the performance by assigning different weights to different features.

- In almost all fingerprint recognition systems, feature extraction and recognition are designed and implemented separately. However, the performance of recognition has always been used to evaluate the performance of feature extraction. It is possible to develop a systematic method to combine these two procedures and achieve a better recognition performance.
- In almost all fingerprint recognition systems, the distortion of human skin is simplified as a linear distortion. One important reason for this simplification is that it is not easy to describe the actual distortion by simple mathematical equations. A better understanding of the distortion may help us to design a better system. It is possible to use other sensing modalities, e.g., the- rmal profile of a finger, to develop multimodal fingerprint matching techniques.
- Some human experts may perform better than computers (in accuracy) for fingerprint recognition. In most cases, human experts can recognize different fingerprints at the first glance. Even for two similar fingerprints, if enough overlapped areas are presented, human experts can recognize them without much effort. What method human experts' eyes and brains use to do this work? To answer this question, we believe that it needs coopera- tion of researchers from different fields, such as Computer Sci- ence, Electrical Engineering, Cognitive Science, Statistics, Social Science, etc. Then we can hope to achieve a better fingerprint recognition performance, which is close to the theoretical bound.

References

- [1] A.S. Abutaleb and M. Kamel, A genetic algorithm for the estimation of ridges in fingerprints, *IEEE Trans. Image Processing*, 8(8), pp.1134-1139, 1999.
- [2] T.D. Alter and W.E.L. Grimson, Verifying model-based alignments in the presence of uncertainty, *Proc. IEEE Conference on Computer Vision and Pattern Recognition*, pp. 344-349, 1997.
- [3] A. Almansa and T. Lindeberg, Fingerprint enhancement by shape adaptation of scale-space operators with automatic scale selection, *IEEE Trans. Image Processing*, 9(12), pp. 2027-2042, 2000.
- [4] J.C. Amengual, A. Juan, J.C. Perez, F. Prat, S. Saez, and J.M. Vilar, Real-time minutiae extraction in fingerprint images, *Proc. Int. Conference on Image Processing and its Applications*, vol.2, pp. 871-875, 1997.
- [5] T. Back, U. Hammel and H.P. Schwefel, Evolutionary computation: comments on the history and current state, *IEEE Trans. Evolutionary Computation*, 1(1), pp. 3-17, 1997.
- [6] M. Ballan, Directional fingerprint processing, *Proc. of Intel. Conference on Signal Processing*, pp. 1064-1067, 1998.

- [7] A.M. Bazen and S.H. Gerez, Systematic methods for the computation of the directional fields and singular points of fingerprints, *IEEE Trans. on Pattern Analysis and Machine Intelligence*, 24(7), pp. 905-919, July 2002.
- [8] G. Bebis, S. Louis, Y. Varol, and A. Yfantis, Genetic object recognition using combinations of views, *IEEE Trans. Evolutionary Computation*, 6(2), pp. 132-146, 2002.
- [9] S. Bernard, N. Boujemaa, D. Vitale and C. Bricot, Fingerprint classification using kohonen topologic map, *Proc. International Conference on Image Processing*, vol. 3, pp. 230 –233, 2001.
- [10] M. Boshra and B. Bhanu, Predicting performance of object recognition, *IEEE Trans. Pattern Analysis and Machine Intelligence*, 22(9), pp. 956-969, 2000.
- [11] M. Boshra and B. Bhanu, Predicting an upper bound on SAR ATR performance, *IEEE Transactions on Aerospace and Electronic Systems*, vol. 37, no. 3, pp. 876-888, 2001.
- [12] B. Bhanu, M. Boshra and X. Tan, Logical templates for feature extraction in fingerprint images, *Proc. Int. Conference on Pattern Recognition*, vol. 2, pp. 850-854, 2000.
- [13] B. Bhanu and J. Han, Bayesian-based performance prediction for gait recognition, *Proc. IEEE Workshop on Human Motion and Video Computing*, pp. 145-150, 2002.
- [14] B. Bhanu and X. Tan, Learned templates for feature extraction in fingerprint images, *Proc. IEEE Conference on Computer Vision and Patter Recognition*, vol. 2, pp. 591-596, 2001.
- [15] B. Bhanu and X. Tan, A triplet based approach for indexing of fingerprint database for identification, *Proc. Int. Conference on Audio- and Video- Based Biometric person Authentication*, pp. 205-210, 2001.

- [16] B. Bhanu and X. Tan, Fingerprint indexing based on novel features of minutiae triplets, *IEEE Trans. on Pattern Analysis and Machine Intelligence*, 25(5), pp. 616-622, 2003.
- [17] G.T. Candela, P.J. Grother, C.I. Watson, R.A. Wilkinson and C.L. Wilson, PCASYS --- a pattern-level classification automation system for fingerprints, Technical Report NISTIR 5647, NIST, Apr. 1995.
- [18] R. Cappelli, A. Lumini, D. Maio and D. Maltoni, Fingerprint classification by directional image partitioning, *IEEE Trans. on Pattern Analysis and Machine Intelligence*, 21(5), pp. 402-421, 1999.
- [19] R. Cappelli, D. Maio, and D. Maltoni, Combining fingerprint classifiers, *Proc. Int. Workshop on Multiple Classifier Systems*, pp. 351-361, 2000.
- [20] A.V. Ceguerra and I. Koprinska, Integrating local and global features in automatic fingerprint verification, *Proc. Int. Conference on Pattern Recognition*, vol. 3, pp. 347-350, 2002.
- [21] J.H. Chang and K.C. Fan, Fingerprint ridge allocation in direct gray-scale domain, *Pattern Recognition*, vol. 34, pp.1907-1925, 2001.
- [22] S.B. Cho, Pattern recognition with neural networks combined by genetic algorithm, *Fuzzy Sets Systems*, 103(2), pp. 339-347, 1999.
- [23] M.M.S. Chong, H.N. Tan, J. Liu, and R.K.L. Gay, Geometric framework for fingerprint image classification, *Pattern Recognition*, 30(9), pp. 1475-1488, 1997.
- [24] J. Daugman, How iris recognition works, *Proc. IEEE International Conference on Image Processing*, vol. 1, pp. 33-36, 2002.
- [25] C. Domeniconi, S. Tari and P. Liang, Direct gray scale ridge reconstruction in fingerprint images, *Proc. IEEE Int. Conference*

- on Acoustics, Speech and Signal Processing*, vol.5, pp. 2941-2944, 1998.
- [26] D.C. Douglas Hung, Enhancement and feature purification of fingerprint images, *Pattern Recognition*, 26(1), pp. 1661-1671, 1993.
 - [27] G. Drets and H. Liljenstrom, Fingerprint sub-classification and singular point detection, *Intl. Journal of Pattern Recognition and Artificial Intelligence*, 12(4), pp. 407-422, 1998.
 - [28] A. Farina, Z.M. Kovacs-Vajna, and A. Leone, Fingerprint minutiae extraction from skeletonized binary images, *Pattern Recognition*, 32(5), pp. 877-889, 1999.
 - [29] FBI, Advanced latent fingerprint school, U.S. Department of Justice, Federal Bureau of Investigation, 1983.
 - [30] A.P. Fitz and R.J. Green, Fingerprint pre-processing on a hexagonal grid, *European Convention on Security and Detection*, pp. 256-260, 1995.
 - [31] A.P. Fitz and R.J. Green, Fingerprint classification using hexagonal fast fourier transform, *Pattern Recognition*, 29(10), pp. 1587-1597, 1996.
 - [32] F. Galton, *Fingerprints*, London, McMillan, 1892.
 - [33] R.S. Germain, A. Califano and S. Colville, Fingerprint matching using transformation parameter clustering, *IEEE Computational Science and Engineering Magazine*, 4(4), 42-49, 1997.
 - [34] W.E.L. Grimson and D.P. Huttenlocher, On the verification of hypothesized matches in model-based recognition, *IEEE Trans. on Pattern Analysis and Machine Intelligence*, 13(12), pp. 1201-1213, 1991.
 - [35] U. Halici, L.C. Jain and A. Erol, Introduction to fingerprint recognition, *Intelligent Biometric Techniques In Fingerprint and*

- Face Recognition*, (Eds.) L.C. Jain, U. Halici, I. Hayashi, S.B. Lee and S. Tsutsui, CRC Press, 1999.
- [36] U. Halici and G. Ongun, Fingerprint classification through self-organizing feature maps modified to treat uncertainties, *Proc. IEEE*, 84(10), pp. 1497-1512, Oct. 1996.
 - [37] J. H. Holland, *Adaptation in Natural and Artificial Systems: An Introductory Analysis with Applications to Biology, Control, and Artificial Intelligence*, MIT Press, 1992.
 - [38] D. Howard, S.C. Roberts, and R. Brankin, Target detection in SAR imagery by genetic programming, *Advances in Eng. Software*, 30(5), pp. 303-311, May 1999.
 - [39] C.T. Hsieh, Z.Y. Lu, T.C. Li and K.C. Mei, An effective method to extract fingerprint singular point, *Proc. Int. Conference on High Performance Computing in the Asia-Pacific Region*, vol. 2, pp. 696-699, 2000.
 - [40] A.K. Jain, R. Bolle and S. Pankanti, *Biometrics: Personal Identification in Networked Society*, Kluwer Academic Publishers, 1999.
 - [41] A.K. Jain and H. Lin, On-line fingerprint verification, *Proc. Intel. Conference on Pattern Recognition*, vol. 2, pp.596-600, 1996.
 - [42] A.K. Jain, H. Lin and R. Bolle, On-line fingerprint verification, *IEEE Trans. on Pattern Analysis and Machine Intelligence*, 19(4), pp.302-314, 1997.
 - [43] A.K. Jain, H. Lin, S. Pankanti and R. Bolle, An identity-authentication system using fingerprints, *Proc. of IEEE*, 85(9), pp. 1364-1388, 1997.
 - [44] A.K. Jain and S. Minut, Hierarchical kernel fitting for finger-print classification and alignment, *Proc. Int. Conference on Pattern Recognition*, vol. 2, pp. 469–473, 2002.

- [45] A.K. Jain, S. Prabhakar and L. Hong, A multichannel approach to fingerprint classification, *IEEE Trans. on Pattern Analysis and Machine Intelligence*, 21(4), pp. 348-359, Apr. 1999.
- [46] A.K. Jain, S. Prabhakar and S. Pankanti, Twin test: on discriminability of Fingerprints, *Proc. Int. Conference on Audio-and Video-Based Person Authentication*, pp. 211-216, 2001.
- [47] A.K. Jain, A. Ross and S. Prabhakar, Fingerprint matching using minutiae and texture features, *Proc. Int. Conference on Image Processing*, vol. 3, pp. 282-285, 2001.
- [48] A.K. Jain, S. Prabhakar, H. Lin and S. Pankanti, Filterbank-based fingerprint matching, *IEEE Trans. on Image Processing*, 9(5), pp. 846-859, 2000.
- [49] J.M. Johnson and Y. Rahmat-Samii, Genetic algorithm optimization for aerospace electromagnetic design and analysis, *Proc. IEEE Aerospace Applications Conference*, vol. 1, pp. 87-102, 1996.
- [50] J.M. Johnson, Genetic algorithm design of a switchable shaped beam linear array with phase-only control, *Proc. IEEE Aerospace Conference*, vol. 3, pp. 297–303, 1999.
- [51] G. Jones and B. Bhanu, Recognition of articulated and occluded objects, *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 21, no. 7, pp. 603-613, 1999.
- [52] X. Jiang and W.Y. Yau, Fingerprint minutiae matching based on the local and global structures, *Proc. Int. Conference on Pattern Recognition*, vol. 3, pp. 1038-1041, 2000.
- [53] X. Jiang, W.Y. Yau, and W. Ser, Detecting the fingerprint minutiae by adaptive tracing the gray-level ridge, *Pattern Recognition*, 34(5), pp. 999-1013, 2001.

- [54] T. Kamei and M. Mizoguchi, Image filter design for fingerprint enhancement, *Proc. Int. Symp. on Computer Vision*, pp. 109-114, 1995.
- [55] T. Kamei and M. Mizoguchi, Fingerprint preselection using eigenfeatures, *Proc. IEEE Conference on Computer Vision and Pattern Recognition*, pp. 918-923, 1998.
- [56] M. Kamijo, Classifying fingerprint images using neural network: deriving the classification state, *Proc. IEEE Int. Conference on Neural Networks*, vol. 3, pp. 1932-1937, 1993.
- [57] K. Karu and A.K. Jain, Fingerprint classification, *Pattern Recognition*, vol. 29, no. 3, pp. 389-404, 1996.
- [58] S. Kasaei, M. Deriche and B. Boashash, Fingerprint feature extraction using block-direction on reconstructed images, *Proc. IEEE TENCON*, vol.1, pp.303-306, 1997.
- [59] W.M. Koo and A. Kot, Curvature-based singular points detection, *Proc. Int. Conference on Audio- and Video-Based Biometric Person Authentication*, pp. 229-234, 2001.
- [60] Z.M. Kovacs-Vajna, A fingerprint verification system based on triangular matching and dynamic time warping, *IEEE Trans. Pattern Analysis and Machine Intelligence*, 22(11), pp. 1266-1276, 2000.
- [61] Z.M. Kovacs-Vajna, R. Rovatti and M. Fazzoni, Fingerprint ridge distance computation methodologies, *Pattern Recognition*, 33(1), pp. 69-80, 2000.
- [62] J.R. Koza, *Genetic Programming II: Automatic Discovery of Reusable Programs*, MIT Press, 1994.
- [63] T. Kawaguchi and M. Nagao, Recognition of occluded objects by a genetic algorithm, *Trans. of IEICE D-II*, J82(3), pp. 350-360, 1999.

- [64] M. Lindenbaum, Bounds on shape recognition performance, *IEEE Trans. on Pattern Analysis and Machine Intelligence*, 17(7), pp. 666–680, 1995.
- [65] M. Lindenbaum, An integrated model for evaluating the amount of data required for reliable recognition, *IEEE Trans. on Pattern Analysis and Machine Intelligence*, 19(11), pp. 1251–1264, 1997.
- [66] H. Lin, A.K. Jain, S. Pankanti and R. Bolle, Fingerprint enhancement, *Proc. IEEE Workshop on Applications of Computer Vision*, pp. 202-207, 1996.
- [67] H. Lin, Y.F. Wan and A.K. Jain, Fingerprint image enhancement: algorithm and performance evaluation, *IEEE Trans. on Pattern Analysis and Machine Intelligence*, 20(8), pp. 777-789, 1998.
- [68] H.K. Lam, F.H. Leung, P.K.S. Tam, Design and Stability Analysis of Fuzzy Model-Based Nonlinear Controller for Nonlinear Systems Using Genetic Algorithm, *IEEE Trans. on Systems, Man and Cybernetics - Part B: Cybernetics*, 33(2), pp. 250-257, 2003.
- [69] A. Lumini, D. Maio, and D. Maltoni, Inexact graph matching for fingerprint classification, *Machine Graphics & Vision*, 8(2), pp. 231-248, 1999.
- [70] Y. Lamdan and H.J. Wolfson, On the error analysis of ‘Geometric Hashing’, *Proc. IEEE Conference on Computer Vision and Pattern Recognition*, pp. 22-27, 1991.
- [71] D. Maio and D. Maltoni, Direct gray-scale minutiae detection in fingerprints, *IEEE Trans. on Pattern Analysis and Machine Intelligence*, 19(1), pp. 27-39, 1997.
- [72] D. Maltoni, D. Maio, A.K. Jain and S. Prabhakar, Handbook of fingerprint recognition, Springer, 2003.

- [73] G.L. Marcialis, F. Roli and P. Frasconi, Fingerprint classification by combination of flat and structure approaches, *Proc. Conference Audio- and Video- Based Person Authentication*, pp. 241-246, 2001.
- [74] C. Mandal, P.P. Chakrabarti and S. Ghose, GABIND: a GA approach to allocation and binding for the high-level synthesis of data paths, *IEEE Trans. on Very Large Scale Integration (VLSI) Systems*, 8(6), pp. 747-750, 2000.
- [75] B.M. Mehtre, Fingerprint image analysis for automatic identification, *Machine Vision and Applications*, 6(2), pp. 124-139, 1993.
- [76] S.M. Mohamed and H.O. Nyongesa, Automatic fingerprint classification system using fuzzy neural techniques, *Proc. IEEE Int. Conference on Fuzzy Systems*, vol. 1, pp. 358–362, 2002.
- [77] L. O’Gorman and J.V. Nickerson, An approach to fingerprint filter design, *Pattern Recognition*, 22(1), pp. 29-38, 1989.
- [78] A.C.M. de Oliveira and L.A.N. Lorena, A constructive genetic algorithm for gate matrix layout problems, *IEEE Trans. on Computer-Aided Design of Integrated Circuits and Systems*, 21(8), pp. 969-974, 2002.
- [79] E. Ozcan and C.K. Mohan, Partial shape matching using genetic algorithms, *Pattern Recognition Letters*, 18, pp. 987-992, 1997.
- [80] S. Pankanti, R.M, Bolle, and A. Jain, Biometrics: The future of identification – Guest Editors’ introduction, *IEEE Computer*, pp. 46-49, February 2000.
- [81] S. Pankanti, S. Prabhakar and A. K. Jain, On the individuality of fingerprints, *IEEE Trans. Pattern Analysis and Machine Intelligence*, 24(8), pp. 1010-1025, 2002.
- [82] S. Prabhakar, A.K. Jain, J.G. Wang, S. Pankanti and R. Bolle, Minutia verification and classification for fingerprint matching,

- Proc. Int. Conference on Pattern Recognition*, vol. 1, pp. 25-29, 2000.
- [83] M.S. Pattichis, G. Panayi, A.C. Bovik and S.P Hsu, Fingerprint classification using an AM-FM model, *IEEE Trans. on Image Processing*, vol. 10, no. 6, pp. 951–954, 2001.
 - [84] R. Poli, Genetic programming for feature detection and image segmentation, *Evolutionary Computation*, Ed. by T.C. Forgarty, pp. 110-125, 1996.
 - [85] Y. Qi, J. Tian, R.W. Dai, Fingerprint classification system with feedback mechanism based on genetic algorithm, *Proc. Int. Conference on Pattern Recognition*, vol. 1, pp. 163–165, 1998.
 - [86] N.K. Ratha, S.Y. Chen and A.K. Jain, Adaptive flow orientation-based feature extraction in fingerprint images, *Pattern Recognition*, 28(11), pp. 1657-1672, 1995.
 - [87] S.C. Roberts and D. Howard, Evolution of vehicle detectors for infrared line scan imagery, *Proc. Evolutionary Image Analysis, Signal Processing and Telecommunications*, pp. 110-125, 1999.
 - [88] V.K. Sagar and K.J.B. Alex, Hybrid fuzzy logic and neural network model for fingerprint minutiae extraction, *Proc. Int. Conference on Neural Networks*, pp. 3255-3259, 1999.
 - [89] V.K. Sagar, D.B.L. Ngo and K.C.K. Foo, Fuzzy feature selection for fingerprint identification, *Proc. Int. Carnahan Conference on Security Technology*, pp. 85-90, 1995.
 - [90] H. Saito and M. Mori, Application of genetic algorithms to stereo matching of images, *Pattern Recognition Letters*, 16, pp. 815-821, 1995.
 - [91] A.A. Saleh, R.R. Adhami, Curvature-based matching approach for automatic fingerprint identification, *Proc. Southeastern Symposium on System Theory*, pp. 171–175, 2001.

- [92] K.B. Sarachik, The effect of Gaussian error in object recognition, *IEEE Trans. on Pattern Analysis and Machine Intelligence*, 19(4), pp. 289–301, 1997.
- [93] B. Sareni, L. Krahenbuhl and A. Nicolas, Efficient genetic algorithms for solving hard constrained optimization problems, *IEEE Trans. on Magnetics*, 36(4), pp. 1027-1030, 2000.
- [94] A. Senior, A combination fingerprint classifier, *IEEE Trans. on Pattern Analysis and Machine Intelligence*, 23(10), pp. 1165-1174, Oct. 2001.
- [95] W.M. Shalash and F.A. Chadi, A fingerprint classification technique using multiplayer SOM, *Proc. National Radio Science Conf.*, Egypt, pp. 22-24, 2000.
- [96] B.G. Sherlock, D.M. Monro and K. Millard, Fingerprint enhancement by directional Fourier filtering, *IEE Proc. Vision Image Signal Process*, 141(2), 1994.
- [97] V.S. Srinivasan and N.N. Murthy, Detection of singular points in fingerprint images, *Pattern Recognition*, 25(2), pp. 139-153, 1992.
- [98] S.A. Stanhope and J.M. Daida, Genetic programming for automatic target classification and recognition in synthetic aperture radar imagery, *Proc. Evolutionary Programming VII*, pp. 735-744, 1998.
- [99] F. Su, J.A. Sun and A. Cai, Fingerprint classification based on fractal analysis, *Proc. International Conference on Signal Processing*, vol. 3, pp. 1471-1474, 2000.
- [100] X. Sun and Z.M. Ai, Automatic feature extraction and recognition of fingerprint images, *Proc. Int. Conference Signal Processing*, pp. 1086-1089, 1996.
- [101] P.W.M. Tsang, A genetic algorithm for aligning object shapes, *Image and Vision Comp.*, 15(11), pp. 819-831, 1997.

- [102] X. Tan and B. Bhanu, Fingerprint matching by genetic algorithm, Late-Breaking Papers for Genetic and Evolutionary Computation Conference, pp. 435-442, 2002.
- [103] X. Tan and B. Bhanu, Fingerprint verification by genetic algorithm, *Proc. IEEE Workshop on Applications of Computer Vision*, pp. 79-83, 2002.
- [104] X. Tan and B. Bhanu, Robust fingerprint identification, *Proc. IEEE Int. Conference on Image Processing*, vol. 1, pp. 270-280, 2002.
- [105] X. Tan and B. Bhanu, A robust two step approach for fingerprint identification, *Pattern Recognition Letters*, vol. 24, pp. 2127-2134, 2003.
- [106] X. Tan, B. Bhanu and Y. Lin, Learning composite operators for fingerprint classification, *Proc. Int. Conference on Audio- and Video-Based Biometric Person Authentication*, pp. 318-326, 2003.
- [107] X. Tan and B. Bhanu, On the fundamental performance for fingerprint matching, *Proc. IEEE Conference on Computer Vision and Pattern Recognition*, vol. 2, pp. 499-504, 2003.
- [108] X. Tan, B. Bhanu and Y. Lin, Fingerprint identification: classification vs. indexing, *Proc. IEEE International Conference on Advanced Video and Signal Based Surveillance*, pp. 151-156, 2003.
- [109] M. Tico and P. Kuosmanen, A multiresolution method for singular points detection in fingerprint images, *Proc. IEEE Int. Symp. on Circuits and Systems*, vol.4, pp.183-186, 1999.
- [110] M. Tico and P. Kuosmanen, An algorithm for fingerprint image postprocessing, *Proc. Asilomar Conference on Signal, Systems and Computers*, vol. 2, pp. 1735-1739, 2000.

- [111] A. Toet and W.P. Hajema, Genetic contour matching, *Pattern Recognition Letters*, 16, pp. 849-856, 1995.
- [112] K. Uchida, T. Kamei, M. Mizoguchi, and T. Temma, Fingerprint card classification with statistical feature integration, *Proc. Int. Conference on Pattern Recognition*, pp. 1833-1839, 1998.
- [113] J.L. Wayman, Error rate equations for the general biometric system, *IEEE Robotics & Automation Magazine*, 6(1), pp. 35-48, 1999.
- [114] A. Wahab, S.H. Chin, and E.C. Tan, Novel approach to automated fingerprint recognition, *IEE Proc. Visual Image Signal Process*, 145(3), pp. 160-166, 1998.
- [115] C.I. Watson and C.L. Wilson, NIST special database 4, fingerprint database, U.S. National Institute of Standards and Technology, 1992.
- [116] A.J. Willis and L. Myers, A cost-effective fingerprint recognition system for use with low-quality prints and damaged fingerprints, *Pattern Recognition*, 34(2), pp. 255-270, 2001.
- [117] C.L. Wilson, G.T. Candela, and C.I. Watson, Neural network fingerprint classification, *Journal of Artificial Neural Networks*, 1(2), pp. 203-228, 1993.
- [118] E.P. Wood and A.K. Jain, (Eds.), Biometric Research Agenda, *Report of the NSF Workshop*, July, 2003.
- [119] Q.H. Xiao and H. Raafat, Fingerprint image postprocessing: a combined statistical and structural approach, *Pattern Recognition*, 24(10), pp. 985-992, 1991.
- [120] Y. Yao, G.L. Marcialis, M. Pontil, P. Frasconi and F. Roli, Combining flat and structured representations for fingerprint classification with recursive neural networks and support vector machines, *Pattern Recognition*, 36(2), pp. 397-406, Feb. 2003.

- [121] D.S. Yeung, Y.T. Cheng, H.S. Fong and F.L. Chung, Neocognitron based handwriting recognition system performance tuning using genetic algorithm, *Proc. IEEE Int. Conference on Systems, Man, and Cyber.*, 5(5), pp. 4228-4233, 1998.
- [122] M. Zaki, A. El-Ramssi and R. Omran, A soft computing approach for recognition of occluded shapes, *The Journal of Systems and Software*, 51, pp. 73-83, 2000.
- [123] D. Zhang, Biometric Solutions: For Authentication in An E-World, Kluwer Academic Publishers, 2002.
- [124] <http://www.forensic-evidence.com>
- [125] <http://www.identix.com>
- [126] <http://www.authentec.com>
- [127] <http://www.iosoftware.com>
- [128] <http://www.digitalpersona.com>
- [129] <http://www.sony.com>
- [130] <http://www.veridicom.com>
- [131] <http://www.policensw.com/fingerprints/finger01.html>
- [132] <http://www.biometrics.org/html/examples/fingerprint.html>

Index

A

- adaptive smooth 24
angle 21, 41, 68, 124
angle change 40
Arch 5, 85, 119
assumption 126, 141

B

- Bayesian 83, 97, 138, 176
bifurcation 19, 26, 41, 140
Binomial distribution 139
biometrics 1, 138, 188
block 11, 66, 103, 181
block diagram 13, 90, 121

C

- classification 5, 83, 115, 180
clutter 10, 80, 138, 170
comparison 31, 82, 123, 171
computation operator 90, 107
confusion matrix 111, 128
constraints 43, 151, 171

- core 44, 84, 127
Correct Index Power 48, 130
corresponding points 61, 127
corresponding triangles 46, 165

D

- Daubert Opinion 135
delta 13, 84, 86
directional 84, 145, 185

E

- endpoint 26, 135, 140
error rate 138, 139, 171

F

- False Acceptance Rate 78, 133
false corresponding triangles 121, 171
feature extraction 29, 172, 185
feature generation operator 95
feature vector 60, 97, 116
fitness value 67, 104
five classes 83, 117, 171

four classes	104, 115	O
G		
Gabor wavelets	138	occlusion 56, 141, 170
gait	176	optimization 62, 185
generation	96, 138	orientation 26, 165, 184
generational GP	100	
Genetic Algorithm	65, 170	P
Genetic Programming	96, 181	PDF 96, 165
Genuine Acceptance Rate	78, 131, 133	performance 81, 188
GV	32	postprocessing 13, 186, 187
H		primitive feature 94, 103, 107
Handedness	41, 124	probability 70, 142, 163
Henry System	5, 83, 85	
hypothesis	55, 137, 171	R
I		resolution 89, 154, 160
identification	120, 183	ridge 124, 181
indexing	38, 120, 186	Right Loop 83, 85, 119
indexing score	38	ROC 79, 131
iris	177	rotation 56, 170
L		S
Lagrange's method	169	scale 62, 177, 182
learning	87, 170	shear 56, 170
Left Loop	85, 119	simplification 160, 173
lower bound	162	simulation 138, 157
M		singular points 84, 185, 186
matching	75, 144, 183	steady-state GP 100
Maximum Side	42, 68, 124	
minutiae	32, 135, 178	T
model	136, 184	template 92, 169, 176
N		Tented Arch 85, 119
NIST-4	48, 116, 171	terminal 92
		termination condition 71
		three-point model 159, 166
		training 97, 111, 128
		translation 60, 170
		triangle 67, 151, 171
		Triangle Direction 42, 68, 124
		Triangle Type 41
		triplets 82, 170, 177

<i>Index</i>		191
two-point model	163, 167	
<i>U</i>		<i>V</i>
uncertainty	119, 146, 175	verification
		60, 131, 186
		<i>W</i>
		Whorl
		119