A Component-Based Approach to Hand-Based Verification and Identification System

A dissertation submitted in partial fulfillment of the requirements for the degree of Doctor of Philosophy in Computer Science and Engineering

by

Gholamreza Amayeh

Prof. George Bebis/Dissertation Advisor

December, 2009
We recommend that the dissertation prepared under our supervision by

GHOLAMREZA AMAYEH

entitled

A COMPONENT-BASED APPROACH TO HAND-BASED VERIFICATION AND IDENTIFICATION SYSTEM

be accepted in partial fulfillment of the requirements for the degree of

DOCTOR OF PHILOSOPHY

Prof. George Bebis, Advisor

Prof. Mark Pinsky, Committee Member

Dr. Mircea Nicoles, Committee Member

Dr. Monica Nicoles, Committee Member

Dr. Zong Tian, Graduate School Representative

Marsha H. Read, Ph. D., Associate Dean, Graduate School

December, 2009
Abstract

Hand-based verification/identification represent a key biometric technology with a wide range of potential applications both in industry and government. Traditionally, hand-based verification and identification systems exploit information from the whole hand for authentication or recognition purposes. To account for hand and finger motion, guidance pegs are used to fix the position and orientation of the hand. In this dissertation, we have investigated a component-based approach to hand-based verification and identification which improves both accuracy and robustness as well as ease of use due to avoiding pegs. Our approach accounts for hand and finger motion by decomposing the hand silhouette in different regions corresponding to the back of the palm and the fingers. To improve accuracy and robustness, verification/recognition is performed by fusing information from different parts of the hand. The proposed approach operates on 2D images acquired by placing the hand on a flat lighting table and does not require using guidance pegs or extracting any landmark points on the hand. To decompose the silhouette of the hand in different regions, we have devised a robust methodology based on an iterative morphological filtering scheme. To capture the geometry of the back of the palm and the fingers, we employ region descriptors based on high-order Zernike moments which are computed using an efficient methodology. The proposed approach has been evaluated both for verification and recognition purposes on a database of 101 subjects with 10 images per subject, illustrating high accuracy and robustness. Comparisons with related ap-
proaches involving the use of the whole hand or different parts of the hand illustrate the superiority of the proposed approach. Qualitative and quantitative comparisons with state-of-the-art approaches indicate that the proposed approach has comparable or better accuracy. As an extension of our work, we investigate the problem of gender classification from hand shape. It has been motivated by studies in anthropometry and psychology suggesting that it is possible to distinguish between male and female hands by considering certain geometric features. For classification, we compute the distance of a given part from two different eigenspaces, one corresponding to the male class and the other corresponding to female class. We have experimented using each part of the hand separately as well as fusing information from different parts of the hand. Using a small database containing 20 males and 20 females, we report classification results close to 98% using score-level fusion and LDA. Also, we address the template aging issue. We introduce a technique by decomposing the hand silhouette into the different parts and analyzing the confidences of these parts in order to lead to global optimization of templates. In the proposed method, first the hand silhouette is divided in different parts corresponding to the fingers. Then the confidence of each finger, as well as its identity, is evaluated by a Support Vector Data Description (SVDD). The confidence of a query hand is determined by the maximum confidence of all fingers. If the maximum confidence is higher than a threshold, the boundaries of all fingers’ SVDDs are incrementally updated to learn the variations of the input data. The motivation behind this technique is that the temporal changes that may occur in the fingers are uncorrelated in such a way that the confidence of each finger can be significantly different from the others. As a result those fingers with difficult intra-class variations can be used in the update process by this technique. The experimental results show the effectiveness of the proposed technique in comparison to the state of the art self-update technique specially at low false acceptance rates.
Dedication

I would like to dedicate this to my wonderful parents, Maryam and Ahmad, who have raised me to be the person I am today. Also I would like to dedicate this to my sisters, Bita and Bahar, and my brother, Alireza, for their unconditional support. I would like to dedicate this to my love, Sara, for her unconditional love. Finally, I would specially like to dedicate this to my supervisor, Prof. George Bebis, who has been with me every step of the way, through good times and bad.
Acknowledgment

The work presented would not be possible without the years of encouragement and guidance of my advisor, Prof. George Bebis, on technical and personal matters. I would also like to thank my family and the numerous friends I have made during my academic career for all of their support.

I would like to thank my committee members Dr. Mircea Nicolescu, Dr. Monica Nicolescu, Prof. Mark Pinsky and Dr. Zong Tian for reviewing my dissertation and offering suggestions. I would also like to thank all of the members of Computer Vision Laboratory, especially Dr. Ali Erol and Dr. Konstantinos Veropoulos for their valuable advice and helpful comments.

This research was supported in part by the National Science Foundation (NSF) under EPSCoR RING-TRUE III grant No. 0447416 and in part by the National Aeronautics and Space Administration (NASA) under EPSCoR grant No. NCC5-583.
Table of Contents

Abstract i

Dedication iii

Acknowledgment iv

List of Tables ix

List of Figures xii

1 INTRODUCTION 1

1.1 Hand Biometrics ........................................... 2

1.2 Goals and Objectives ....................................... 2

1.3 Contributions of Dissertation: A Summary .................. 5

1.4 Outline of Dissertation ..................................... 7

2 LITERATURE REVIEW 8

2.1 Introduction .................................................. 8

2.2 Hand Geometry ............................................... 10

2.3 Hand Contour ................................................ 11

2.4 Hand Surface Shape ........................................... 12

3 SYSTEM OVERVIEW AND PREPROCESSING 16
3.1 System Overview .................................................. 16
3.2 Image Acquisition System ........................................ 17
3.3 Preprocessing ...................................................... 19
   3.3.1 Binarization .................................................. 19
   3.3.2 Hand-Arm Segmentation ..................................... 20
   3.3.3 Finger-Palm Segmentation ................................... 24
   3.3.4 Postprocessing of Fingers ................................... 24

4 HIGH ORDER ZERNIKE MOMENTS COMPUTATION 28
   4.1 Zernike Moments ................................................ 28
   4.2 Computation of High Order Zernike Moments ................... 30
   4.3 Comparisons with other methods .............................. 33

5 FEATURE EXTRACTION AND FUSION 37
   5.1 Extracting Zernike Features .................................. 37
   5.2 Fusion Techniques .............................................. 41

6 EXPERIMENTAL RESULTS 45
   6.1 Hand-Based Verification Results .............................. 45
      6.1.1 Verification using whole hand ............................ 46
      6.1.2 Verification using different parts of the hand ......... 48
      6.1.3 Verification by fusing information ..................... 51
   6.2 Hand-Based Identification Results .......................... 54
      6.2.1 Identification using whole hand ......................... 55
      6.2.2 Identification using different parts of the hand ...... 55
      6.2.3 Identification by fusing information ................... 58
   6.3 Comparison between global and component-based hand representations 59
   6.4 Comparisons with other approaches ......................... 62
6.4.1  Comparison between different segmentation methods ........ 66
6.5  System Performance Over Time ................................. 70

7  GENDER CLASSIFICATION 78
7.1  Introduction ....................................................... 78
7.2  Fourier Descriptors .............................................. 80
7.3  Gender Classification Using Different Parts of the Hand ....... 81
7.4  Experimental Results ............................................ 86
    7.4.1  Gender Classification Results using Different Parts of the Hand 86
    7.4.2  Gender Classification Results using Fusion .................. 89

8  TEMPLATE UPDATE 93
8.1  Introduction ....................................................... 93
8.2  Matching and Update through SVDD ............................. 96
8.3  Experimental Results ............................................ 98

9  CONCLUSIONS AND FUTURE WORK 105
9.1  Conclusions ....................................................... 105
9.2  Future work ...................................................... 107

Bibliography 108
List of Tables

3.1 The effect of the extra morphological closing operator . . . . . . . . . . 26
3.2 Mean and Variance of matching distances . . . . . . . . . . . . . . . . 27
3.3 Mean and Variance of non-matching distances . . . . . . . . . . . . . 27
4.1 The terms $\chi_{m,k}$ required to be computed up to order 10 . . . . . . 32
4.2 Differences between Zernike moments up to order 50 . . . . . . . . . . . 32
4.3 Reconstruction error . . . . . . . . . . . . . . . . . . . . . . . . . . . 35
4.4 Computational complexity of various methods . . . . . . . . . . . . . 36
6.1 Verification using whole hand . . . . . . . . . . . . . . . . . . . . . 48
6.2 Comparison using different parts of the hand for verification . . . . . 50
6.3 Detailed comparison of different fusion strategies . . . . . . . . . . . . 53
6.4 Fusion using majority voting for verification . . . . . . . . . . . . . . 53
6.5 Processing time for traditional and proposed methods . . . . . . . . . 65
6.6 Previous published research efforts . . . . . . . . . . . . . . . . . . . 76
6.7 Time lapse verification performance . . . . . . . . . . . . . . . . . . 76
6.8 Time lapse identification performance . . . . . . . . . . . . . . . . . 77
7.1 Measurements of hand breadth and length . . . . . . . . . . . . . . . . 79
7.2 Gender classification using feature-level fusion . . . . . . . . . . . . . 91
7.3 Gender classification using score-level fusion . . . . . . . . . . . . . . 91
7.4 Gender classification using decision-level fusion . . . . . . . . . . . . 92
8.1 True acceptance rates of different fingers .......................... 101
8.2 True acceptance rates and equal error rates .......................... 102
List of Figures

2.1 The input image in a typical hand-based biometric system . . . . . . . 11
2.2 (a) An intensity image sample . . . . . . . . . . . . . . . . . . . . . 12
2.3 (a) An intensity image sample . . . . . . . . . . . . . . . . . . . . . 13
2.4 (a) Segmented intensity hand image . . . . . . . . . . . . . . . . . . 13
3.1 Main stages of the proposed system . . . . . . . . . . . . . . . . . . . 17
3.2 Image acquisition system . . . . . . . . . . . . . . . . . . . . . . . . 18
3.3 Color image . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 20
3.4 Multi resolution scheme . . . . . . . . . . . . . . . . . . . . . . . . . 23
3.5 Binarized image . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 23
3.6 Hand silhouette . . . . . . . . . . . . . . . . . . . . . . . . . . . . . . 25
3.7 (Top) The junctions of finger with the palm . . . . . . . . . . . . . . 26
4.1 Common terms in the computation of Zernike moments . . . . . . . . 31
4.2 Dot product between basis function . . . . . . . . . . . . . . . . . . . 33
4.3 Original and reconstructed images . . . . . . . . . . . . . . . . . . . . 34
4.4 Reconstructed images using moment of order up to 60 . . . . . . . . . 34
5.1 Original and reconstructed images . . . . . . . . . . . . . . . . . . . . 38
5.2 (a) Original image (top left) and reconstructed images . . . . . . . . 40
5.3 (a) Original and (b) reconstructed images . . . . . . . . . . . . . . . 41
6.1 Average ROC curves using whole hand for verification . . . . . . . . . 47
6.2 Average ROC curves using different parts of the hand ................. 49
6.3 Verification results using different parts of the hand ................. 50
6.4 Average ROC curves for verification ............................... 52
6.5 Comparison of the four different fusion strategies .................. 53
6.6 Average CMC curves using whole hand ........................... 56
6.7 Average CMC curves using different parts of the hand ............. 57
6.8 Identification results using different parts .......................... 58
6.9 Average CMC curves for identification ............................. 60
6.10 Comparison of different fusion strategies ........................... 61
6.11 Comparison of the three most representative approaches ............ 62
6.12 (a), (b) Images of the same hand containing finger motion ........ 62
6.13 (a) A sample hand image ........................................... 64
6.14 Comparison between geometric features and Zernike descriptors ... 65
6.15 A sample hand contour and its landmarks ........................... 67
6.16 (a) A sample hand contour ........................................... 68
6.17 Average ROC curves .................................................. 71
6.18 Comparison of morphological-based and landmarks-based segmentation 72
6.19 Effect of time lapse .................................................... 73
6.20 Effect of time lapse .................................................... 75

7.1 Computing the masculine/feminine characteristic ......................... 83
7.2 Distribution of distances for the male/female eigenspaces ............... 84
7.3 Male/Female distributions using LDA .................................. 85
7.4 Gender classification results using ZMs and MD ......................... 87
7.5 Gender classification results using ZMs and k-NN ......................... 88
7.6 Gender classification results using ZMs and LDA ......................... 88
7.7 Gender classification results using FDs and MD ......................... 89
<table>
<thead>
<tr>
<th>Section</th>
<th>Title</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>7.8</td>
<td>Gender classification results using FDs and (k)-NN</td>
<td>90</td>
</tr>
<tr>
<td>7.9</td>
<td>Gender classification results using FDs and LDA</td>
<td>90</td>
</tr>
<tr>
<td>7.10</td>
<td>Best gender classification results</td>
<td>92</td>
</tr>
<tr>
<td>8.1</td>
<td>Temporal change of a hand shape over</td>
<td>95</td>
</tr>
<tr>
<td>8.2</td>
<td>(a) A color image sample</td>
<td>99</td>
</tr>
<tr>
<td>8.3</td>
<td>Verification results of each finger</td>
<td>103</td>
</tr>
<tr>
<td>8.4</td>
<td>Performance of the hand verification system</td>
<td>104</td>
</tr>
</tbody>
</table>
Chapter 1

INTRODUCTION

Recently, there has been increased interest in developing biometrics-based verification and identification systems which has led to intensive research in fingerprint, face, hand, ear, and iris authentication and recognition. Each biometric has its own strength and weakness depending on the specific application and its requirements. Hand-based biometrics is among the oldest live biometrics-based authentication modalities. The existence of several hand-based authentication commercial systems and patents indicate the effectiveness of this type of biometric. Although hand-based live authentication has a long history and a considerable market share [1], most studies addressing enhancements of this technology are rather recent [2]. Increases in computing power and advances in computer vision and pattern recognition are expected to enable the implementation of more accurate, robust, and easier to use systems. Removal of pegs, to improve convenience, and use of more powerful features to represent the shape of the hand represent promising research directions in this area.
1.1 Hand Biometrics

The geometry of the hand contains relatively invariant features of an individual, however, geometric features of the hand (e.g., finger length/width, area/size of the palm) are not as distinctive as fingerprint or iris features. Therefore, hand-based biometric systems have been employed mostly in small-scale person authentication applications. In this study, we demonstrate the application of hand-based biometrics for identification purposes as well. The main difference between verification and identification is that in the case of verification, an unknown subject is compared against a specific subject in the database to verify his/her identity (i.e., ”Am I who I claim”). In the case of identification, an unknown subject is compared against all the subjects in the database to establish his/her identity (i.e., ”Who am I?”). Therefore, identification can be thought as verifying an unknown subject against all subjects in the database. As a result, identification is more time consuming and prone to errors.

There are several reasons for developing hand-based verification/identification systems. First, the shape of the hand can be easily captured in a relatively user friendly manner by using conventional CCD cameras. Second, this technology is more acceptable by the public in daily life mainly because it lacks a close connection to forensic applications. Finally, there has been some interest lately in fusing different biometrics to increase system performance [57][3]. The ease of use and acceptability of hand-based biometrics make hand shape a good candidate in these heterogeneous systems.

1.2 Goals and Objectives

In this dissertation, we have developed a novel, peg-free approach to hand-based verification and identification which does not require extracting any landmark points on
the hand and it is not sensitive to hand and finger motion. The proposed approach operates on 2D hand images acquired by placing the hand on a planar lighting table without any guidance pegs. There are several important ideas behind the proposed approach. First, to deal with the issue of hand and finger motion, we decompose the silhouette of the hand in different regions corresponding to the back of the palm and fingers. This is performed using a robust methodology based on an iterative morphological filtering scheme. To avoid touching fingers and simplify segmentation, subjects are required to stretch their hand prior to placing it on the lighting table. No other restrictions are imposed on the subjects. Second, in contrast to traditional approaches that represent the shape of the explicitly using hand-crafted geometrical measurements, we represent the geometry of each part of the hand implicitly using high-order Zernike moments [83]. Finally, to improve verification/identification accuracy and robustness, we fuse information from different parts of the hand. It is worth mentioning that the use of high-order moments is not practical for many applications due to their noise sensitivity. However, this is not an issue in the context of our application since we use a robust image acquisition process which provides very high quality hand images as shown in Chapter 3.

Moments have been used before in a wide range of applications in image analysis, and object recognition [70]. In the area of biometrics, preliminary results have been reported using various types of moments (e.g., geometric, Zernike, pseudo-Zernike, and Legendre moments) for palmprint verification [47] and [21]. Zernike moments are quite attractive for representing the geometry of the hand due to having minimal redundancy (i.e., employ orthogonal basis functions [69]), providing invariance to translation, rotation, and scale, and demonstrating robustness to noise [70]. In most applications, the use of Zernike moments has been limited to low-orders only or small low-resolution images due to high computational requirements and lack of accuracy.
due to numerical errors. Capturing the shape of the hand in sufficient detail, however, would require computing moments of rather high-orders. Although there have been several efforts to reduce computational complexity by employing quantized polar coordinate systems, such transformations have an effect on accuracy. In this study, we employ an improved algorithm, proposed in one of our earlier studies [9], which can speed up the computation of high-order Zernike moments without sacrificing accuracy. To keep computational complexity low, we avoid redundant computations by detecting common terms and using look-up tables. To preserve accuracy, we avoid any coordinate transformations and employ arbitrary precision arithmetic.

Fusing information from different biometric modalities (i.e., face, fingerprint, hand) has received considerable attention lately, however, fusing information from different parts of the same biometric has been considered to a lesser extent. For example, Ross and Govindarajan [56] have reported a feature-level fusion scheme which combines hand and face features. Kumar and Zhang [37] have investigated feature selection of hand shape and palm print features. Cheung et al. [17] have proposed a two-level fusion strategy for multimodal biometric verification. Jiang and Su [32] have proposed fusing faces and fingerprints to improve verification accuracy. Our approach is mostly related to component-based approaches in object detection and recognition [4] [63], face detection/ recognition [25], and person detection [44]. The key idea behind them is representing objects in terms of their parts and geometrical relationships. Among them, the most relevant approach to ours is the face recognition approach reported in [25]. In that study, information from different parts of the face was fused at the feature level using Support Vector Machines (SVMs) [18]. Here, we report results using several different fusion strategies including feature-level, score-level and decision-level. Earlier versions of our work have appeared in [7], [8], and [6].
1.3 Contributions of Dissertation: A Summary

We considered the problem of complexity of high order Zernike moments computation. We discussed the lack of accuracy due to numerical errors and we have showed, through an experimental analysis, that traditional approaches fails to calculate high order Zernike moments correctly.

We proposed an efficient algorithm to accurately calculate Zernike moments at high orders. To preserve accuracy, we do not use any form of coordinate transformation and employ arbitrary precision arithmetic. The computational complexity is reduced by detecting the common terms in Zernike moments with different order and repetition.

We considered the improvement of efficiency, accuracy, and robustness of hand-based verification due to the fact that geometric features of the hand (e.g., finger length/width, area/size of the palm) are not as distinctive as fingerprint or iris features.

We proposed a system operates on 2D hand silhouette images acquired by placing the hand on a planar lighting table without any guidance pegs, increasing the ease of use compared to conventional systems. In particular, we proposed using high-order Zernike moments to represent hand geometry, avoiding the more difficult and prone to errors process of hand-landmark extraction (e.g., finding finger joints).

We considered hand and finger motion by decomposing the hand silhouette in different regions corresponding to the back of the palm and the fingers. First, the hand is segmented from the forearm using a robust, iterative methodology based on morphological operators. Then, the hand is segmented into six regions corresponding to the palm and the fingers using morphological operators again. To improve accuracy and robustness, verification/recognition is performed by fusing information from different parts of the hand.
We considered the problem of gender classification from hand shape. Our method has been motivated by studies in anthropometry and psychology suggesting that it is possible to distinguish between male and female hands by considering certain geometric features. For classification, we compute the distance of a given hand from two different eigenspaces, one corresponding to the male class and the other corresponding to female class.

We considered template aging issue due to substantial intra-class variations of biometric data which decreases the performance of the system. The majority of existing techniques in the literature, namely self update, update a template set by using a confidently verified input sample in order to avoid the introduction of impostors into the template set of a client. Therefore these techniques can only exploit the input sample very similar to the current template set leading to local optimization of a template set.

We proposed a technique by decomposing the hand silhouette into the different parts (i.e. fingers) and analyzing the confidences of these parts in order to lead to global optimization of templates. In the proposed method, first the hand silhouette is divided in different parts corresponding to the fingers. Then the confidence of each finger, as well as its identity, is evaluated by a Support Vector Data Description (SVDD). The confidence of a query hand is determined by the maximum confidence of all fingers. If the maximum confidence is higher than a threshold, the boundaries of all fingers SVDDs are incrementally updated to learn the variations of the input data. The motivation behind this technique is that the temporal changes that may occur in the fingers are uncorrelated in such a way that the confidence of each finger can be significantly different from the others.
1.4 Outline of Dissertation

The rest of this dissertation is organized as follows: Chapter 2 discusses the previous related work in hand based biometric systems. The necessary background material on Zernike moments and their computational complexity are discussed in Chapter 4. Chapter 3 presents our methodology for separating the hand from the arm and decomposing the hand silhouette in different parts corresponding to the back of the palm and the fingers. Representing the geometry of the shape of the palm and the fingers using Zernike moments and various fusion strategies investigated in this study are discussed in Chapter 5. The experimental results of verification, identification experiments are presented in Chapter 6. The details of the gender classification task from hand shape are discussed in Chapter 7. Chapter 8 considers template aging issue in the proposed system for verification purpose. Finally, Section 9 provides our conclusions and directions for future work.
Chapter 2

LITERATURE REVIEW

2.1 Introduction

The majority of hand-based biometric systems employ geometric measurements and are based on research limited to considerably old patents and commercial products [29]. In these systems, users are asked to place their hand on a flat surface and align it, with the help of some guidance pegs. The alignment operation simplifies feature extraction and allows for high processing speeds. A mirror is usually used to obtain a side view of the hand using a single camera. In most cases, a few hand-crafted geometric features (e.g. length, width and height of the fingers, thickness of the hand, aspect ratio of fingers and palm, etc.) are extracted, making it possible to construct a small template (i.e., 9 bytes in some commercial systems).

Removal of pegs, to improve convenience, and use of more powerful feature extraction techniques to capture the shape of the hand more accurately represent promising research directions in this area. Several studies have reported that peg-based alignment is not very satisfactory and represents in some cases a considerable source of failure [61][30]. Although peg removal provides a solution to reduce user inconvenience, it also raises more challenging research issues due to the increase in intra-
class variance. Nevertheless, most recent studies have concentrated on the design of peg-free systems.

Extracting extremities of the hand contour, such as finger valleys and finger tips, is usually the first processing step in these systems. In peg-free systems, fingers are not guaranteed to be at the same position and orientation at different acquisition times; therefore they need to be segmented and identified in the input images. Analysis of the silhouette contour to locate fingertips and palm-finger intersections, which basically corresponds to curvature local maxima, provides an effective solution to the segmentation problem [74][77]. Once the fingers have been segmented, geometric features such as finger length and width can be measured at predefined points along the finger axes [3][77][60][80].

Using geometric features helps to reduce storage requirements but can not represent hand shape in detail. Moreover, accurate localization of various landmark points on the fingers is not a straightforward task. Some studies have introduced new features capturing the full finger shape. Jain and Duta [31] have used the silhouette contour of the fingers and an iterative closest point (ICP) alignment algorithm to compute a shape distance which is used as a measure of similarity. Ma et al. [81] have followed a similar approach using B-Spline curves. Xioang et. al. [74] introduced a semi-geometric approach by extracting geometric features after aligning the fingers which are represented using ellipses. Kumar et. al [36] used palm-print and hand geometric features where the extremities of the hand contour were used to measure finger length and palm width. Recently, Yoruk et. al. [82] introduced a more accurate and detailed representation of the hand using the Hausdorff distance of the hand contour, and Independent Component Analysis (ICA)[18]. Their approach requires registering the silhouettes of the hand images using the locations of fingertips and valleys. This study is among a few studies where hand-based biometrics have been
demonstrated both for verification and recognition purposes.

A marginally different feature extraction approach, which involves reconstructing the 3D surface of the hand, was proposed in [79]. Using a range sensor to reconstruct the dorsal part of the hand, local shape index values of the fingers were used as features in matching. In a related study, Lay et. al. [38] projected a parallel grating onto the dorsal part of the hand to extract features that indirectly capture 3D shape information. Use of multiple enrollment templates is an effective method to improve the recognition accuracy of any biometric system. In the hand-based biometrics domain, using multiple enrollment templates is vital part of any system mainly due to the lower distinctiveness of hand-shape. User-specific statistical models, such Mixtures of Gaussians [53][62], have shown to improve system accuracy [77][80][53][62]. We categorize the previous researches based on the features used for recognition.

2.2 Hand Geometry

As it was mentioned, the majority of existing systems employ hand geometric features for recognition or authentication. It has been illustrated in the literature that these features work well and can be computed efficiency. Many of the current hand geometry based systems consist of a platen to place the hand for image capture and a camera to capture a digital image as shown in Fig. 2.1. Usually there are guide pegs on the platen to help ensure proper placement of the user’s hand. The guide pegs of the system serve as control points and aid in choosing the axes. As it can be seen in Fig. 2.1, hand and finger measurements are taken across 16 axes.

The techniques used for hand geometry based systems are simple, relatively easy to use, and inexpensive. In addition, hand geometry based systems tend to have the highest user acceptability compared to other biometric identifiers such as the fingerprint and iris pattern. Despite these advantages, there are limits to the use of
hand geometry as a biometric identifier. One limit is its low discriminating power. Most researchers believe that hand geometry alone cannot be used to identify an individual when using large data sets. The performance of hand geometry-based systems is very sensitive to hand placement. It has been shown that some hand geometry-based systems can be circumvented with the use of a fake hand. This makes hand geometry a poor choice for use as a biometric feature in unattended high security applications.

2.3 Hand Contour

As an alternative to taking hand and finger measurements along fixed axes, as in hand geometry-based systems, researchers have investigated using the contour of the hand’s silhouette as a biometric feature [31]. Platen pegs are used for proper placement of the hand during image capture. Before comparison, the pegs are removed from the hand images using a mask image that contains the known positions of the platen pegs.

Next, the hand contour is extracted from the hand image using a mean-shift unsupervised segmentation. The five pairs of fingers to be compared are extracted
Figure 2.2: (a) Hand contour, and (b) point correspondences [31].

from the contours and aligned separately. To accomplish this automatically, the point set pairs of each finger are aligned based on a quasi-exhaustive polynomial search of point pair matchings between them. This step addresses the issue of hand placement on performance by aligning the fingers before the features are extracted. During the finger extraction and alignment step, a set of point correspondences is produced.

Fig 2.2 illustrates the technique. Fig 2.2(a) shows hand contours extracted from two different subjects, and Fig 2.2(b) shows point correspondences represented as green segments. The mean alignment error between two sets of contour points is used to quantify the match quality. A match exists between two hands if the MAE falls below a predetermined threshold. The performance is comparable to that of commercially available hand geometry based systems [31].

2.4 Hand Surface Shape

Woodard and Flynn in [78] presented a novel approach for personal identification which utilizes finger surface features as a biometric identifier. This approach includes three main steps: hand segmentation, finger extraction and template generation. In order to work with only the range image pixels lying on the surface of the hand, the
task of hand segmentation is required. To simplify this task in [78], the intensity image of the hand is used instead of its range image (see Fig. 2.3), since there is a pixel to pixel correspondence between intensity and range images. Therefore a combination of edge and skin detection techniques are employed to the intensity image to reliably segment the hand from the image, as shown in Figure 2.4(a), thereby allowing for segmentation in the range image.

After obtaining hand silhouette from the intensity hand image, the convex hull of the contour of the hand silhouette is used to locate the valleys between the fingers represented as circles in Figure 2.4(b). The valley positions are used as segment boundaries allowing for the index, middle, and ring fingers to be extracted and processed individually [78]. The subjects are instructed to place their hand in relatively the same position so it is assumed that the relative positions of index, middle, and
ring fingers are consistent among the collected images. The shaded areas of Figure 2.4(c) represents the extracted finger pixels. To address finger pose variations, each finger mask along with its corresponding range pixels is rotated and centered in a 80 × 240 output finger range image in which the major axis of the finger mask is coincident with the horizontal axis.

For each valid pixel of the finger mask in the output image, a surface curvature estimate is computed with the corresponding range data. The principal curvature values, \( k_{\min} \) and \( k_{\max} \), are calculated for each finger surface point \( p \) by following formula [78]:

\[
\begin{align*}
k_{\min,\max} &= \frac{f_{xx} + f_{yy} + f_{xx}f_{yy} - 2f_{x}f_{y}f_{xy}}{2(1+f_{xx}^2+f_{yy}^2)^{3/2}} \pm \sqrt{\frac{(f_{xx} + f_{yy} + f_{xx}f_{yy} - 2f_{x}f_{y}f_{xy})^2 - f_{xx}f_{yy} - f_{xy}^2}{(1+f_{xx}^2+f_{yy}^2)^3}} \\
\end{align*}
\]  

(2.1)

where \( f_{x}, f_{y}, f_{xy}, f_{xx} \) and \( f_{yy} \) are the partial derivatives of point \( p \). In [78], it has been suggested that the range data be smoothed prior to curvature estimation in order to limit the effects of noise. The computed principal curvature values are then used to compute the Shape Index, \( SI \), value at each pixel, given by the following formula [78]:

\[
SI = \begin{cases} 
\frac{1}{2} - \frac{1}{\pi} \arctan \left( \frac{k_{\max} + k_{\min}}{k_{\max} - k_{\min}} \right) & k_{\max} \geq k_{\min} \\
\text{not valid} & k_{\max} < k_{\min}
\end{cases}
\]

(2.2)

\( SI \) is a scalar in \([0, 1]\) with values that allow shape classification. In the rare case in which the computed principal curvature values are equal, thereby forcing the shape index formula to be undefined at a particular pixel, the shape index value at that pixel is assigned the value of zero. The match score is the sample normalized
correlation coefficient given by the following formula [78]:

\[
C_{SI_q,SI_t} = \frac{\sum_{i,j} (SI_q(i,j) - \overline{SI}_q)(SI_t(i,j) - \overline{SI}_t)}{\sqrt{\sum_{i,j} (SI_q(i,j) - \overline{SI}_q)^2 \sum_{i,j} (SI_t(i,j) - \overline{SI}_t)^2}}
\]  

(2.3)

where \(SI_q(i,j)\), \(SI_t(i,j)\) are valid shape index values and \(\overline{SI}_q\), \(\overline{SI}_t\) are the sample mean shape index values in the query and template images, respectively. In [78], different fusion technique such as average, median and maximum fusion rules employed to combine information of index, middle and ring fingers. Using UND database the best results have been reported for authentication and verification using maximum and average rules at score level.
Chapter 3

SYSTEM OVERVIEW AND PREPROCESSING

3.1 System Overview

Fig. 3.1 shows the main stages of the proposed system. The image acquired by a VGA resolution CCD camera is binarized and goes through the segmentation module. During segmentation, the arm is separated from the hand and discarded from further processing. Then, the hand is further processed to segment the palm and the fingers. Feature extraction is performed by computing the Zernike moments of each part of the hand separately. The resulting representation is invariant to translation, rotation and scaling. Verification/Identification is performed by fusing information from different parts of the hand. We have experimented with different fusion strategies including feature-level, score-level, and decision-level fusion. We employ multiple enrollment templates per subject and compute similarity scores using the minimum distance between a query image and the templates of the subjects.
3.2 Image Acquisition System

Our image acquisition system consists of a VGA resolution CCD camera and a flat lighting table, which forms the surface for placing the hand. The direction of the camera is perpendicular to the lighting table as shown in Fig. 3.2(a). The camera has been calibrated to remove lens distortion. In practical settings, both the camera and the lighting table can be placed inside a box to completely eliminate light interferences from the surrounding environment. Also, the whole system can be made much smaller than the one shown in Fig. 3.2(a) which is very bulky and was built for experimentation reasons only. Nevertheless, the experimental set up in our laboratory, shown in Fig. 3.2(a), provides high quality images without requiring us to put
much effort to control the environment. It should be mentioned that capturing high quality hand images is critical for our application as it allows us to use high-order Zernike moments without worrying about noise sensitivity issues.

When users place their hand on the surface of the lighting table, an almost binary, shadow and noise free, silhouette of the hand is obtained as shown in Figs. 3.2(b) and (c). During acquisition, subjects are required to stretch their hand and place it inside a rectangular region marked on the surface of the table. This is to avoid touching fingers, ensure the visibility of the whole hand, and avoid perspective distortions. No restrictions were imposed on the orientation of the hand.

Figure 3.2: (a) Image acquisition system, (b, c) Images of the same hand acquired by the system.
3.3 Preprocessing

This stage includes the binarization of the hand image and its segmentation into different regions corresponding to the arm, hand, palm, and fingers. Our current set up yields very high quality images, which are almost free of shadows and noise. As a result, binarization can be performed using a fixed threshold. To separate the forearm from the hand, first we detect the palm by finding the largest circle that can be prescribed inside the hand-arm silhouette. Then, we take the intersection of the forearm with the circle’s boundary. To separate the fingers from the palm, first we filter out the fingers using morphological closing [23]. Then, the palm is subtracted from the hand silhouette. Specific details are provided below.

3.3.1 Binarization

The hand images can be captured using a gray scale camera; however, we used a color CCD camera as it was already available in our laboratory. To obtain a grayscale image, we used the luminance value \( Y_{i,j} \) of each pixel \((i, j)\):

\[
Y_{i,j} = 0.299R_{i,j} + 0.587G_{i,j} + 0.114B_{i,j}
\]  

(3.1)

where \( R_{i,j} , G_{i,j} , B_{i,j} \) denote the RGB values of the pixel. Figures 3.3(a) and (b) show the original color and grayscale images respectively. The binary value \( B_{i,j} \) of a pixel \((i, j)\) was calculated as follows:

\[
B_{i,j} = \begin{cases} 
1 & \text{if } Y_{i,j} < T \\
0 & \text{otherwise}
\end{cases}
\]  

(3.2)
where $T$ is a fixed threshold which was determined experimentally. In all of the experiments reported in this study, $T = 0.5$ was used. Figure 3.3(c) shows the output of the binarization process. The resulting silhouettes are very accurate and consistent due to the image acquisition set up. This is critical as high-order Zernike moments are sensitive to small changes in silhouette shape.

![Figure 3.3: (a) Color image, (b) grayscale image and (c) binarized image.](image)

### 3.3.2 Hand-Arm Segmentation

The binary silhouette obtained during image acquisition is the union of the hand with the arm. The arm does not have many distinctive features while its silhouette, at different acquisition sessions, is not expected to be the same due to clothing and freedom in hand placement (see Figs. 3.2(b) and (c)). To segment the arm, we assume that the user is not wearing very loose clothing on the arm. Under this assumption, the palm becomes the thicker region of the silhouette, which enables its detection by finding the largest circle that can be prescribed inside the silhouette. We use a robust, iterative morphological closing scheme based on a circular structuring element [23] to find the largest circle. Our algorithm can be summarized as follows:

1. Initialize the radius of the circular structuring element $D$ to a very large value (e.g. $R = 85$)
2. Apply morphological closing on the image using $D$

3. If the output is an empty image then set $R = R - 1$ and go to step 2; otherwise, the resulting image corresponds to the largest circle inside the silhouette.

This algorithm has shown to work well in our experiments, however, its main drawback is that it is time consuming due to its iterative nature and the use of morphological operators. This is especially true when the size of the hand is relatively small. For example, it requires 29 morphological closing operations on a $480 \times 640$ assuming that the radius of the largest circle inside the hand silhouette is about 57 pixels which is typical for smaller hands. Implementing the algorithm in MATLAB 7.4.0 on a 3.19 GHz 64-bits machine with 2GB of RAM, it would take more than 6 second to segment the hand and forearm in this case.

One way to speed up processing is by reducing the number of iterations [10]. This can be done by initializing the radius of the structuring element $D$ more conservatively. To address this issue, we have developed a multi-resolution scheme which operates on different resolution images of the hand. First, the largest circle inside the hand silhouette is found approximately but fast using a low resolution image of the hand. Next, to refine the position and size of the circle found, the same process is repeated at a higher resolution. To reduce the number of iterations, the radius of the structuring element at higher resolutions is initialized using the radius of the circle found at lower resolutions. This process is repeated until the highest resolution hand image (i.e., original hand image) is processed.

To represent a hand image at different resolution levels, we scale it down by simply down-sampling it. Figure 3.4 illustrates the hierarchy of hand images obtained by down-sampling the input image three times, each time by a factor of two. The multi-resolution algorithm can be summarized as follows:
1. Generate a hierarchy of different resolution hand images by down-sampling the input image.

2. Initialize the radius of the circular structuring element $D$ to a small value (e.g. $R_0 = 11$)

3. Consider the lowest resolution hand image.

4. Find the largest circle ($R_{\text{max}}$) inside the hand silhouette.

5. Set the radius of the circular structuring element $D$ to $2 \times R_{\text{max}} + 2$

6. If a higher resolution image is available, go to step 4; otherwise, stop;

In our experiments, the multi-resolution hierarchy contains four levels, that is, we scale down the original image three times. The algorithm starts by processing the lowest resolution hand image which is 8 times smaller than the original one. At this level, the largest circle prescribed inside the hand silhouette can be found very quickly (i.e., typically, within 4-5 iterations). When considering higher resolutions, the number of iterations stays low by initializing the radius of the structuring element conservatively based on the size of the circle found at lower resolutions. Considering the small hand example mentioned earlier, it takes 5 iterations at the lowest resolution image and only 2 iterations at the highest resolution (i.e., original) image. Overall, segmenting the hand and forearm reduces processing time from 6 seconds to 0.69 seconds for this example. The average processing time on 250 sample images was 0.58 seconds.

Fig. 3.5(b) shows the output of the algorithm on the sample image shown in Fig. 3.5(a). Once the largest circle has been found, the arm can be segmented by finding its intersection with the circle and the boundary of the hand-arm region. Figure 3.5(c) shows the resulting hand silhouette after discarding the arm region.
Figure 3.4: Multi resolution scheme to reduce computation cost of Hand-Arm Segmentation module. The binarized image is down sampled up to 3 scales with sampling rate equal to $\left(\frac{1}{2}\right)^{scale}$.

Figure 3.5: (a) Binarized image, (b) largest circle that can be prescribed inside the hand-arm silhouette (c) segmented hand silhouette.
3.3.3 Finger-Palm Segmentation

To simplify finger segmentation, subjects were instructed to stretch their hand during image acquisition in order to avoid touching fingers; however, finger motion was unavoidable. Several sample images collected from the same subject are shown in Figure 3.2. As it can be observed, the relative position of the fingers varies significantly from sample to sample. The processing steps of the finger segmentation module are shown in Fig. 3.6. First, a morphological closing operator based on a circular disk is applied on the hand image as shown in Fig. 3.6(a). The radius of the structuring element was experimentally set to 25 pixels, making it thicker than the widest finger in our database. The closing operation filters out the fingers from the silhouette as shown in Fig. 3.6(c). The remaining part of the silhouette corresponds to the palm, which is subtracted from the hand image to obtain the finger segments as shown in Fig. 3.6(d). It should be mentioned that an alternative way to segment the fingers from palm is by detecting certain landmark points on the hand, such as fingertips and valleys. This solution, however, would be more prone to errors due to inaccuracies in landmark detection.

To identify each of the fingers quickly, we assume that hand rotations are less than 45 degrees [10]. In our prototype system, larger rotations would correspond to purposeful, unnatural hand placement by the users. In general, it would be possible to deal with larger rotations by using additional information for each finger such as length, width, aspect ratio, and area. To extract each finger and the back of the palm, we use connected component analysis [24].

3.3.4 Postprocessing of Fingers

A closer examination of the results shown in Fig. 3.6(d) reveal that the segmented fingers have sharp tails at the locations where they meet the palm. The curvature
of the hand contour at these locations is smoother for the little, point, and thumb fingers as shown in Fig. 3.7(left). As a result, there might be significant differences in the length of the tails corresponding to these fingers as shown in Fig. 3.7(a), where different samples from the same subject are shown. In some cases, especially when the hand is small (i.e., mostly for female hands), there are significant differences in the length of the tails, which introduces significant errors in the computation of the Zernike moments. This can be illustrating by observing the differences in the size of the circles enclosing the fingers in Fig. 3.7(a) versus those in Fig. 3.7(b) where the tails have been removed using post-processing.

To keep these errors as low as possible, we post-process each finger by applying an extra morphological closing step as shown in Fig. 3.7(b). The structuring element was chosen experimentally to be a simple 4 by 4 square with values set to one. Table 3.1 illustrates the effect of this step by showing the normalized distances between the pairs of corresponding fingers shown in Fig. 3.7. Obviously, this step improves matching scores considerably.

Tables 3.2 and 3.3, present statistical results (i.e., mean and variance) to further support the benefits of this step in terms of matching and non-matching distances. For each finger, we have computed all possible matching and non-matching distances
Figure 3.7: (Top) The junctions of finger with the palm in the hand counter where their curvature is too smoother than the others. (Bottom) Pairs of segmented little, point, and thumb fingers. Each pair corresponds to two different samples of the same subject. (a) before applying the additional step, and (b) after applying the additional step.

Table 3.1: The effect of the extra morphological closing operator on the normalized distances between the Zernike moments (up to order 20) of the segmented finger pairs before (Figure 3.7(a)) and after (Figure 3.7(b)) the extra step.

<table>
<thead>
<tr>
<th>Pair of Fingers</th>
<th>$d_{before}$</th>
<th>$d_{after}$</th>
</tr>
</thead>
<tbody>
<tr>
<td>Little</td>
<td>0.5904</td>
<td>0.0901</td>
</tr>
<tr>
<td>Point</td>
<td>0.7881</td>
<td>0.1135</td>
</tr>
<tr>
<td>Thumb</td>
<td>0.7424</td>
<td>0.1253</td>
</tr>
</tbody>
</table>
in our database, before and after post-processing, using Zernike features up to order 20 (121 features). Since our database contains 101 people with 10 images per person, there were 4545 matching distances and 1010000 non-matching distances for each finger. Our results indicate that finger post-processing reduces the overlap between matching and non-matching distances significantly in the case of the little, point, and thumb fingers; however, it does not seem to have an important effect in the case of the ring and middle fingers. This was an expected result since there are more segmentation problems with these fingers due to their greater motion freedom.

Table 3.2: *Mean* and *Variance* of matching distances for each finger before and after post-processing.

<table>
<thead>
<tr>
<th>Finger</th>
<th>( \mu_{\text{before}} )</th>
<th>( \mu_{\text{after}} )</th>
<th>( \sigma_{\text{before}} )</th>
<th>( \sigma_{\text{after}} )</th>
</tr>
</thead>
<tbody>
<tr>
<td>Little</td>
<td>0.1724</td>
<td>0.0998</td>
<td>0.1039</td>
<td>0.0873</td>
</tr>
<tr>
<td>Ring</td>
<td>0.1085</td>
<td>0.0817</td>
<td>0.1004</td>
<td>0.0952</td>
</tr>
<tr>
<td>Middle</td>
<td>0.0823</td>
<td>0.0810</td>
<td>0.0983</td>
<td>0.0995</td>
</tr>
<tr>
<td>Point</td>
<td>0.1928</td>
<td>0.0716</td>
<td>0.1287</td>
<td>0.0886</td>
</tr>
<tr>
<td>Thumb</td>
<td>0.1843</td>
<td>0.1205</td>
<td>0.1262</td>
<td>0.0843</td>
</tr>
</tbody>
</table>

Table 3.3: *Mean* and *Variance* of non-matching distances of each finger before and after post-processing.

<table>
<thead>
<tr>
<th>Finger</th>
<th>( \mu_{\text{before}} )</th>
<th>( \mu_{\text{after}} )</th>
<th>( \sigma_{\text{before}} )</th>
<th>( \sigma_{\text{after}} )</th>
</tr>
</thead>
<tbody>
<tr>
<td>Little</td>
<td>0.3564</td>
<td>0.2869</td>
<td>0.1162</td>
<td>0.0341</td>
</tr>
<tr>
<td>Ring</td>
<td>0.3072</td>
<td>0.2861</td>
<td>0.0531</td>
<td>0.0289</td>
</tr>
<tr>
<td>Middle</td>
<td>0.2859</td>
<td>0.2870</td>
<td>0.0290</td>
<td>0.0268</td>
</tr>
<tr>
<td>Point</td>
<td>0.3672</td>
<td>0.2715</td>
<td>0.1454</td>
<td>0.0248</td>
</tr>
<tr>
<td>Thumb</td>
<td>0.4136</td>
<td>0.3120</td>
<td>0.1216</td>
<td>0.0616</td>
</tr>
</tbody>
</table>
Chapter 4

HIGH ORDER ZERNIKE MOMENTS COMPUTATION

4.1 Zernike Moments

Zernike moments are based on a set of complex polynomials that form a complete orthogonal set over the interior of the unit circle [83]. They are defined as the projection of the image on these orthogonal basis functions. Specifically, the basis functions $V_{n,m}(x,y)$ are given by

$$V_{n,m}(x,y) = V_{n,m}(\rho, \theta) = R_{n,m}(\rho)e^{j m \theta}$$

(4.1)

where $n$ is a non-negative integer, $m$ is a non-zero integer subject to the constraints $n - |m|$ is even and $|m| < n$, $\rho$ is the length of the vector from origin to $(x,y)$, $\theta$ is the angle between the vector $\rho$ and the $x$-axis in a counter clockwise direction, and $R_{n,m}(\rho)$ is the Zernike radial polynomial which is defined as follows:

$$R_{n,m}(\rho) = \sum_{k=|m|, n-k\text{even}}^{n} \frac{(-1)^{n-k}}{|n-k|! \frac{n+k}{2}} \frac{n+k}{2}! \rho^k$$
Note that $R_{n,m}(\rho) = R_{n,-m}(\rho)$. The basis functions in Eq. 4.1 are orthogonal, therefore, satisfying the constraint:

$$\frac{n + 1}{\pi} \int \int_{x^2 + y^2 \leq 1} V_{n,m}(x, y)V_{p,q}^*(x, y) = \delta_{n,p} \delta_{m,q} \quad (4.3)$$

where

$$\delta_{a,b} = \begin{cases} 1 & \text{if } a = b \\ 0 & \text{otherwise} \end{cases} \quad (4.4)$$

The Zernike moment of order $n$ with repetition $m$ for a digital image function $f(x, y)$ is given by [34]:

$$Z_{n,m} = \frac{n + 1}{\pi} \sum_{x^2 + y^2 \leq 1} f(x, y)V_{n,m}^*(x, y) \quad (4.5)$$

where $V_{n,m}^*(x, y)$ is the complex conjugate of $V_{n,m}(x, y)$. To compute the Zernike moments of a given image, the center of mass of the object is taken to be the origin. The magnitude of the Zernike moments is rotation invariant by its definition (See Eq. 4.5). Taking the center of mass of the object as the origin of the coordinate system makes them translation invariant as well. Additionally, to provide scale invariance, the object is scaled inside the unit circle [13].

The function $f(x, y)$ can then be reconstructed by the following expression [34]:

$$\tilde{f}(x, y) = \sum_{n=0}^{N} \frac{C_{n,0}}{2} R_{n,0}(\rho) +$$
\[
\sum_{n=1}^{N} \sum_{m>0} (C_{n,m} \cos m\theta + S_{n,m} \sin m\theta) R_{n,m}(\rho)
\]  

(4.6)

where \( N \) is the maximum order of Zernike moments used, while \( C_{n,m} \) and \( S_{n,m} \) denote the real and imaginary parts of \( Z_{n,m} \) respectively.

### 4.2 Computation of High Order Zernike Moments

A method to improve the speed of Zernike moments computation involves using a quantized polar coordinate system. In [46], Mukundan et al. proposed a recursive algorithm for the computation of Zernike and Legendre moments using polar coordinates. In [59], Belkasim et al. introduced a different recursive algorithm using radial and angular expansions of Zernike orthonormal polynomials. For an \( M \times M \) image, the angles were quantized to \( 4M \) and the radii were quantized to \( M \) levels. In a more recent study, Gu et al. [26] employed the "square to circle" transformation of Mukundan et al. [46] and more efficient recursive relationships to develop an even faster algorithm, however, its accuracy was still limited to that of [46] due to the quantization step in the coordinate transformation.

A side effect of quantization is that errors are introduced in the computation of high-order Zernike moments (see next Section). In this work, we have adopted a novel algorithm which avoids using any quantization, therefore, the computation of the moments is as accurate as in the traditional approach (i.e., no approximations). To save computation time, the improved algorithm finds the terms that occur repeatedly in various orders and avoids recomputing them. Additional computations can be saved using a look-up table. To ensure high accuracy, it uses arbitrary precision arithmetic.

Specifically, by substituting Eqs. 4.2 and 4.1 in Eq. 4.5 and re-organizing the
terms, the Zernike moments can be computed as follows:

\[ Z_{n,m} = \frac{n+1}{\pi} \sum_{x^2+y^2 \leq 1} ( \sum_{k=|m|}^{n} \beta_{n,m,k} \rho^k e^{-jm\theta} f(x, y)) \]

\[ = \frac{n+1}{\pi} \sum_{k=|m|}^{n} \beta_{n,m,k} \left( \sum_{x^2+y^2 \leq 1} e^{-jm\theta} \rho^k f(x, y) \right) \]

\[ = \frac{n+1}{\pi} \sum_{k=|m|}^{n} \beta_{n,m,k} \chi_{m,k} \quad (4.7) \]

The terms \( \chi_{m,k} \), defined in Eq. 4.7, are common in the computation of moments having the same repetition as shown in Fig. 4.1 in the case of repetition \( m=0 \). In general, to compute Zernike moments up to order \( N \), we would need to compute \( \chi_{m,k} \) for each repetition. However, computing \( \chi_{m,k} \) only once would be sufficient for computing Zernike moments of any order and any repetition by simply taking linear combinations as shown in Eq. 4.7. Moreover, the coefficients \( \beta_{n,m,k} \) (see Eq. 4.2) do not depend on the input image or the coordinates; therefore, they can be stored in a small look-up table to save additional computations.

\[ Z_{0,0} = \beta_{0,0,0} \chi_{0,0} \]
\[ Z_{2,0} = \beta_{0,0,0} \chi_{0,0} + \beta_{2,0,2} \chi_{0,2} \]
\[ Z_{4,0} = \beta_{0,0,0} \chi_{0,0} + \beta_{2,0,2} \chi_{0,2} + \beta_{4,0,4} \chi_{0,4} \]
\[ Z_{6,0} = \beta_{0,0,0} \chi_{0,0} + \beta_{2,0,2} \chi_{0,2} + \beta_{4,0,4} \chi_{0,4} + \beta_{6,0,6} \chi_{0,6} \]
\[ Z_{8,0} = \beta_{0,0,0} \chi_{0,0} + \beta_{2,0,2} \chi_{0,2} + \beta_{4,0,4} \chi_{0,4} + \beta_{6,0,6} \chi_{0,6} + \beta_{8,0,8} \chi_{0,8} \]
\[ Z_{10,0} = \beta_{0,0,0} \chi_{0,0} + \beta_{2,0,2} \chi_{0,2} + \beta_{4,0,4} \chi_{0,4} + \beta_{6,0,6} \chi_{0,6} + \beta_{8,0,8} \chi_{0,8} + \beta_{10,0,10} \chi_{0,10} \]

Figure 4.1: Common terms in the computation of Zernike moments up to order 10 with zero repetition.

An important issue in the computation of high-order Zernike is the issue of numerical precision. Depending on the image size and maximum order, double precision arithmetic does not provide enough precision and serious numerical errors can be
Table 4.1: The terms $\chi_{m,k}$ required to be computed up to order 10

<table>
<thead>
<tr>
<th>repetition $m$</th>
<th>$\chi_{m,k}$</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>$\chi_{0,0}, \chi_{0,2}, \chi_{0,4}, \chi_{0,6}, \chi_{0,8}, \chi_{0,10}$</td>
</tr>
<tr>
<td>1</td>
<td>$\chi_{1,1}, \chi_{1,3}, \chi_{1,5}, \chi_{1,7}, \chi_{1,9}$</td>
</tr>
<tr>
<td>2</td>
<td>$\chi_{2,2}, \chi_{2,4}, \chi_{2,6}, \chi_{2,8}, \chi_{2,10}$</td>
</tr>
<tr>
<td>3</td>
<td>$\chi_{3,3}, \chi_{3,5}, \chi_{3,7}, \chi_{3,9}$</td>
</tr>
<tr>
<td>4</td>
<td>$\chi_{4,4}, \chi_{4,6}, \chi_{4,8}, \chi_{4,10}$</td>
</tr>
<tr>
<td>5</td>
<td>$\chi_{5,5}, \chi_{5,7}, \chi_{5,9}$</td>
</tr>
<tr>
<td>6</td>
<td>$\chi_{6,6}, \chi_{6,8}, \chi_{6,10}$</td>
</tr>
<tr>
<td>7</td>
<td>$\chi_{7,7}, \chi_{7,9}$</td>
</tr>
<tr>
<td>8</td>
<td>$\chi_{8,8}, \chi_{8,10}$</td>
</tr>
<tr>
<td>9</td>
<td>$\chi_{9,9}$</td>
</tr>
<tr>
<td>10</td>
<td>$\chi_{10,10}$</td>
</tr>
</tbody>
</table>

Table 4.2: Differences between Zernike moments up to order 50, computed using double precision and arbitrary precision arithmetic for a $300 \times 300$ image.

<table>
<thead>
<tr>
<th>Order , repetition</th>
<th>0</th>
<th>2</th>
<th>4</th>
<th>...</th>
<th>40</th>
<th>42</th>
<th>44</th>
<th>46</th>
<th>48</th>
<th>50</th>
</tr>
</thead>
<tbody>
<tr>
<td>42</td>
<td>7.28e-4</td>
<td>6.60e-4</td>
<td>1.91e-4</td>
<td>...</td>
<td>1.17e-17</td>
<td>3.82e-17</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>44</td>
<td>3.50e-3</td>
<td>5.57e-3</td>
<td>1.11e-3</td>
<td>...</td>
<td>1.52e-15</td>
<td>1.30e-17</td>
<td>1.04e-17</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>46</td>
<td>3.97e-1</td>
<td>6.48e-3</td>
<td>5.25e-3</td>
<td>...</td>
<td>2.12e-14</td>
<td>1.48e-15</td>
<td>9.06e-17</td>
<td>2.60e-18</td>
<td></td>
<td></td>
</tr>
<tr>
<td>48</td>
<td>1.86e0</td>
<td>6.91e-2</td>
<td>4.39e-2</td>
<td>...</td>
<td>5.25e-14</td>
<td>5.92e-14</td>
<td>3.11e-16</td>
<td>1.20e-16</td>
<td>3.47e-18</td>
<td></td>
</tr>
<tr>
<td>50</td>
<td>1.38e1</td>
<td>1.81e0</td>
<td>1.06e-1</td>
<td>...</td>
<td>7.52e-12</td>
<td>2.67e-13</td>
<td>1.69e-14</td>
<td>8.60e-16</td>
<td>4.65e-17</td>
<td>2.17e-18</td>
</tr>
</tbody>
</table>

introduced in the computation of the moments. This is demonstrated in Table 4.2 which shows the differences between Zernike moments up to order 50, computed using double precision and arbitrary precision arithmetic for a $300 \times 300$ image. As it can be observed, the error becomes more and more significant with increasing order and decreasing repetition.

Fig. 4.2 shows the effect of numerical errors on the orthogonality of the basis functions. As it can be observed in Fig. 4.2(a), obtained using double precision arithmetic, the orthogonality of the basis functions is violated seriously. On the other hand, the orthogonality of the basis functions is preserved using arbitrary precision arithmetic as shown in Fig. 4.2(b) (i.e., only one delta peak is present).
Figure 4.2: Dot product between basis function of order $n = 43$ and repetition $m = 7$ with other basis functions up to order 50 using (a) double precision arithmetic and (b) arbitrary precision arithmetic.

4.3 Comparisons with other methods

We have compared the accuracy of our algorithm with several other algorithms [46][59][26] using the fidelity of reconstruction as a criterion. The test image that we used in our experiments is shown in Fig. 4.3 (top). This is a $64 \times 64$ image and Zernike moments up to order 40 were utilized for reconstruction. Figs. 4.3(a), 4.3(b) and 4.3(c) show the results using Mukundan’s method [46], Gu’s method [26] and our method respectively. As it can be observed, the former two algorithms give poor reconstructions mainly because of the square to circle transformation. The effect of the transformation is clearly visible in the reconstructed images.

The reconstruction results using Belkasim’s [59] method and Zernike moments up to order 60 is shown in Fig. 4.4(a) while the reconstruction results using our method is shown in Fig. 4.4(b). We used arbitrary precision arithmetic in the implementation of Belkasim’s method to make the comparison fair. It can be observed that Belkasim’s method introduces some distortions at the edges while our method produces smoother edges in general.
Figure 4.3: Original (top) and reconstructed images using moments of order up to 40: (a) Mukundan’s method, (b) Gu’s method, and (c) our method.

Figure 4.4: Reconstructed images using moment of order up to 60: (a) Belkasim’s method, and (b) our method.

To make the differences between the two methods more clear, we have computed reconstruction errors for each method, shown in Table 4.3, using different orders. The formula use to compute the error is shown below:

\[ \varepsilon_r = \frac{\sum_x \sum_y |\tilde{f}(x, y) - f(x, y)|^2}{\sum_x \sum_y f(x, y)^2} \]  

(4.8)

where \( f(x, y) \) is the original image and \( \tilde{f}(x, y) \) is the reconstructed image (up to order \( N \)).
In general, it would be reasonable to expect that the reconstruction error decreases as the order of moments increases. Our method exhibits this behavior, however, Belkasim’s method behaves quite differently which indicates that the quantization of the polar coordinates has a serious effect on the computation of higher-order moments.

Table 4.3: Reconstruction error using our method and Belkasim’s method.

<table>
<thead>
<tr>
<th>Order</th>
<th>Our method</th>
<th>Belkasim’s method</th>
</tr>
</thead>
<tbody>
<tr>
<td>35</td>
<td>0.0647</td>
<td>0.0648</td>
</tr>
<tr>
<td>40</td>
<td>0.0621</td>
<td>0.0628</td>
</tr>
<tr>
<td>45</td>
<td>0.0596</td>
<td>0.063</td>
</tr>
<tr>
<td>50</td>
<td>0.0370</td>
<td>0.0557</td>
</tr>
<tr>
<td>55</td>
<td>0.0203</td>
<td>0.0645</td>
</tr>
<tr>
<td>60</td>
<td>0.0133</td>
<td>0.0665</td>
</tr>
</tbody>
</table>

Table 4.4 shows the number of multiplications and additions required by each method. We have assumed an image of size $M \times M$ pixels and Zernike moments up to order $N$. In the case of our method, first we need $M^2N$ multiplications to compute $\rho_k^k f(x, y)$, $k = 0, 1, ..., N$. Then, we must compute $\chi_m,k = \sum_x \sum_y e^{-jm\theta} \rho_k^k f(x, y)$. The number of $\chi_{m,k}$ required to compute Zernike moments up to order $N (even)$ is $\frac{N}{2} (\frac{N}{2} + 1)$. Since there is no need for any multiplication when $m = 0$ and $\chi_{m,k}$ is a complex number, this step requires $M^2N(\frac{N}{2} + 1) + (\frac{N}{2} + 1)^2$ multiplications and $2(M^2 - 1)(\frac{N}{2} + 1)^2$ additions. For large $N$ and $M$, the number of multiplications and additions required to compute $Z_{n,m}$ is negligible according to Eq. 4.7. Asymptotically, our method has comparable computational complexity with Belaksim’s method (i.e., $O(N^2M^2)$ multiplications) although Belaksim’s method performs less additions (i.e., $O(NM^2)$ versus $O(N^2M^2)$).
<table>
<thead>
<tr>
<th>Method</th>
<th>Number of Addition</th>
<th>Number of Multiplication</th>
</tr>
</thead>
<tbody>
<tr>
<td>Mukundan’s method</td>
<td>$\frac{N^2M^2}{2} + \frac{1}{8}NM^3$</td>
<td>$2N^2 + N^2M^2 + \frac{1}{4}MN^3$</td>
</tr>
<tr>
<td>Belkasim’s method</td>
<td>$N(M + 2)(M - 1)$</td>
<td>$\frac{N^2M^2}{2} + 2MN$</td>
</tr>
<tr>
<td>Gu’s method</td>
<td>$\frac{3}{8}N^2M + 2NM^2 + \frac{1}{12}N^3M + \frac{1}{4}N^2M^2$</td>
<td>$\frac{N^2M}{2} + 2M^2N$</td>
</tr>
<tr>
<td>Our method</td>
<td>$2(\frac{N}{2} + 1)^2(M^2 - 1)$</td>
<td>$\frac{N^2M^2}{2} + 2M^2N$</td>
</tr>
</tbody>
</table>
Chapter 5

FEATURE EXTRACTION AND FUSION

5.1 Extracting Zernike Features

In this step, we represent the geometry of the back of the palm and the fingers implicitly using Zernike moments. A critical issue at this stage is choosing the order of Zernike moments appropriately in order to capture sufficient shape information for verification and identification purposes. Our experimental results indicate that capturing important shape details for verification/identification purposes requires using high-order moments.

In general, using very high-order moments would preserve more and more information. Fig. 5.1 demonstrates this idea using a 300 \times 300 binary image, shown at the top left corner, which contains information at various levels of detail. The reconstructed images using moments up to order 20 contain only a rough silhouette of the wolf. The reconstructions using moments up to order 50 show the head of wolf clearly, however, the letters in the logo are still blurred. Using orders up to 70 improves the reconstruction of the letters in the logo as well. Using very high-orders is
not practical, however, due to information redundancy and computational complexity issues. Moreover, Liao et. al. [39] have shown that there is an inherent limitation in the precision of arbitrary high-order Zernike moments due to the circular geometry of the domain.

![Figure 5.1: Original and reconstructed images using different orders of Zernike moments.](image)

Here, we used the average reconstruction error (i.e., Eq. 4.8) on a large number of palm and finger images to decide the appropriate order for our application. Our objective was to preserve important details while keeping the orders as low as possible. Specifically, by analyzing the reconstruction error, the maximum order chosen for the fingers was 20 while the maximum order chosen for the back of the palm was 30. Fig. 5.2 (a), shows several finger reconstructions using different orders. Fig. 5.2(b) shows
the reconstruction error using different orders in this case. As it can be observed, the error almost saturates for orders higher than 40. This is also visually evident from the finger reconstructions shown in Fig. 5.2(a). In general, using orders higher than 20 does not offer major improvements. Therefore, to keep computational cost low, the highest order chosen in the case of fingers was 20. Similar experiments and analysis in the case of the back of the palm revealed that the highest order useful for verification/identification purposes was 30. It should be mentioned that the reconstruction criterion used here to select the order of Zernike moments might not yield the most discriminant moments [52]. In the future, we plan to investigate feature selection schemes [66] to select a subset of discriminant Zernike moments for each part of the hand.

Using a similar analysis to represent the geometry of the whole hand, we found that orders as high as 70 were required. Fig. 5.3 (a) shows a hand image while Fig. 5.3 (b) shows several reconstructions using different orders. The reconstruction error is shown to the right of Fig. 5.3. Clearly, using a component-based representation of the hand offers major computational savings.

Computing very high-order Zernike moments is quite computationally expensive, especially when precision is a requirement. The algorithm proposed in Section 4.2 was initially implemented in C++ using arbitrary precision arithmetic (i.e., 200 digits) on a 2.66 GHzpentium IV with 256 MB memory. In this case, it takes about 6 minutes to compute Zernike moments up to order 70, while it only takes 35 seconds to compute Zernike moments up to order 30. We have verified experimentally that moments up to order 30 can be computed quite accurately without resorting to arbitrary precision arithmetic. In our application, using double precision instead of arbitrary precision to compute moments up to order 36 yields an error less than 0.5%. Using double precision in C++ on a 3.19 GHz 64-bits machine with 2GB of RAM,
Figure 5.2: (a) Original image (top left) and reconstructed images (left to right, top to bottom) up to order 2, 5, 10, 20, 30, 40, 50, 60 and 70, (b) reconstruction error.
it takes less than 0.01 seconds on the average to compute moments up to order 30. The time savings using double precision are significant and can be further improved by computing the Zernike moments of different parts of the hand in parallel. In practice, a hybrid implementation can be employed where the use of arbitrary precision arithmetic is restricted only to orders higher than 36, therefore, reducing computational complexity. It should be mentioned that using feature selection [66] to choose a subset of discriminant Zernike moments, as mentioned earlier, will further decrease time requirements. Hardware implementations could also be considered for real time applications [35][48].

5.2 Fusion Techniques

At this step, we fuse information from different parts of the hand to improve verification/identification accuracy and robustness. In general, fusion can be implemented
at different levels. In this study, we have experimented with three different fusion strategies: feature-level, score-level, and decision-level fusion.

In feature-level fusion, the features extracted from the fingers and the back of the palm can be fused to create a more compact and powerful feature set. Commonly, feature-level fusion is performed using dimensionality reduction or feature selection [52]. In score-level fusion, the matching scores of the fingers and the palm can be fused to obtain an overall score. The sum rule or the weighted sum rule are common score-level fusion techniques [27]. In decision-level fusion, verification/identification results based on different parts of the hand can be fused to obtain an overall authentication decision. Majority voting, and AND/OR rules are widely used decision-level fusion techniques [27]. We provide more details in the following subsections.

**Feature-Level Fusion using Principal Component Analysis**

Using Principal Components Analysis (PCA) [52] for feature-level fusion is a very common approach. According to this approach, the feature vectors of the back of the palm and the fingers are combined into a single feature vector. Then, PCA is applied to map them into a lower dimensional space. The resulting PCA features are linear combinations of the original finger and palm features.

**Score-Level Fusion Using Weighted Sum**

The weighted sum rule has been extensively investigated in the literature and it is probably the most straightforward fusion strategy at the score-level. First, we compute matching scores between corresponding parts of the hand (i.e., back of the palm and fingers) in the query and and the template. Then, the matching scores are combined into a single score using a weighted sum rule as shown below:
\[ S(Q, T) = \sum_{i=1}^{6} \alpha_i S(Q_i, T_i) \] (5.1)

where \( S \) is the similarity measure (e.g., Euclidean distance) between the query \( Q \) and the template \( T \). \( Q_i \) and \( T_i \) represent the \( i \)-th part of the hand, \( i = 1, 2, ..., 6 \). In our system, the first five parts correspond to the little, ring, middle, point and thumb fingers while the sixth part corresponds to the back of the palm. The parameters \( \alpha_i \) are the weights associated with the \( i \)-th part of the hand; they need to satisfy the following constraint:

\[ \sum_{i=1}^{6} \alpha_i = 1 \] (5.2)

The key issue with this method is determining a set of appropriate weight values.

**Score-Level Fusion using Support Vector Machines**

A Support Vector Machine (SVM) is a binary classifier that maps input patterns \( X \) to output labels \( y \in \{-1, 1\} \) [52]. In general, an SVM has the following form:

\[ f(X) = \sum_{i \in \Omega} \alpha_i y_i K(X, X_i) + b \] (5.3)

where \( \alpha_i \) are the Lagrange multipliers, \( \Omega \) corresponds to the indices of the support vectors for which \( \alpha_i \neq 0 \), \( b \) is a bias term, \( X \) is an input vector, and \( K(X, X_i) \) is a kernel function. Classification decisions are based on whether the value \( f(X) \) is above or below a threshold. We have employed SVM to implement an alternative score-level fusion strategy. Given a pair of hands to be verified, the input vector \( X \) is composed of the matching scores between corresponding parts of the hand. Assigning the input vector to the class ”1” implies that both hands come from the same subject while assigning it to the class ”-1” implies that they come from different subjects.
Decision-Level Fusion Using Majority Voting

Majority voting is among the most straightforward decision-level fusion strategies. In this case, the final decision is based on the output results of several matchers. In the context of our application, first we verify/identify each subject using different parts of the hand (i.e., fingers and palm). Then, if three or more parts of the hand yield a positive verification/identification, then verification/identification is considered successful; otherwise, the subject is rejected.
Chapter 6

EXPERIMENTAL RESULTS

In order to evaluate the proposed system, we have collected hand images from 101 people of different age, sex and ethnicity. For each subject, we collected 10 images of their right hand during the same session. To test the performance of our system on time passage, additional hand images were collected in a separate session from 20 of these subjects 9 months later. During each session, subjects were asked to stretch their hand and place it inside a square area drawn on the surface of the lighting table; no other restrictions were imposed on the subjects. To capture different samples within each session, subjects were asked to remove their hand from the lighting table, relax it for a few seconds, and then place it back again. We report results both on hand-based verification and recognition [10].

6.1 Hand-Based Verification Results

For person verification, one must differentiate a genuine hand from imposter hands as the user provides his/her hand image in support of his/her claimed identity. For this purpose, we calculate the Euclidean distance between the hand of the applicant and each of his/her templates in the database and take the minimum distance $D$: 
\[ D = \min_i \{ ||Q - T_i|| \}, i = 1, \ldots, k \] (6.1)

where \( Q \) corresponds to the query hand, \( T_i \) corresponds to the \( i \)-th template of a given subject in the database, and \( k \) corresponds to the number of templates of that subject. If \( D \) is below a threshold, verification is successful; otherwise, the subject is rejected.

In the following subsections [10], we present the results of several different experiments. First, we investigate the performance of a baseline system which uses the whole hand for verification. Then, we investigate the verification power of different parts of the hand by implementing several systems that perform verification using each part of the hand separately. Finally, we evaluate the proposed system which fuses information from different parts of the hand for verification.

### 6.1.1 Verification using whole hand

To provide a baseline for comparisons, first we experimented with a simpler system that uses the whole hand for verification. In this case, a global representation of the hand is used for verification. Preliminary results based on this approach have been reported in an earlier work [7], however, this section presents results based on more comprehensive experiments and a larger database.

The first step in this baseline system is to separate the arm from the hand using the methodology presented in Section 3.3.2. Then, the geometry of the silhouette of the whole hand is represented using Zernike moments. As mentioned in Section 5.1, capturing the shape details of the whole hand requires computing Zernike moments up to order 70; this yields feature vectors containing 1296 components.

To test the approach, we used different number of samples (i.e. 3, 4, and 5) for each subject as enrollment templates. To account for regularities in the choice of the
templates, we repeated the experiments 30 times, each time choosing the enrollment templates randomly. The remaining samples were used to construct matching and non-matching sets and estimate the False Acceptance Rate (FAR) and False Reject Rate (FRR) of the system. Fig. 6.1(a) shows the average ROC curves obtained using this procedure. As it can be observed, using more templates improves verification accuracy, however, it this would also increase verification time.

Since the size of the feature vectors was very high, we have also experimented with PCA to reduce their dimensionality. Using a similar procedure, we repeated the experiments 30 times, choosing 3, 4, and 5 templates randomly each time. In each experiment, the eigenvectors were computed from the covariance matrix of the enrolled templates by preserving 99.9% of the information. Fig. 6.1(b) shows the average ROC curves obtained using PCA features. Table 6.1 provides comparative results in terms of the Equal Error Rate (EER), as well as the mean, and standard deviation of the True Acceptance Rate (TAR) when $FAR = 0.1\%$. As it can be observed, PCA improves verification results by increasing TAR while at the same time reducing its standard deviation. Overall, the best verification performance using the
whole hand was obtained with PCA features and 5 enrollment templates per subject.

Table 6.1: Verification using whole hand: comparison using raw and PCA features.

<table>
<thead>
<tr>
<th>Enrollment Size</th>
<th>3</th>
<th>4</th>
<th>5</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Features</strong></td>
<td>Raw</td>
<td>PCA</td>
<td>Raw</td>
</tr>
<tr>
<td>Number of Features</td>
<td>1296</td>
<td>182-203</td>
<td>1296</td>
</tr>
<tr>
<td>EER (%)</td>
<td>3.55</td>
<td>2.69</td>
<td>2.95</td>
</tr>
<tr>
<td>TAR (%) (FAR=1%)</td>
<td>94.22</td>
<td>95.84</td>
<td>95.62</td>
</tr>
<tr>
<td>σTAR (%) (FAR=1%)</td>
<td>1.62</td>
<td>1.60</td>
<td>1.61</td>
</tr>
</tbody>
</table>

6.1.2 Verification using different parts of the hand

To investigate the verification power of different parts of the hand, we experimented with several systems, each performing verification using a different part of the hand. In this case, local representations of the hand were used for verification. Each system was tested using 5 enrollment templates (i.e., using less templates results in lower accuracy) and repeating the experiments 30 times as before by choosing the enrollment templates randomly each time. For each system, we report the average ROC curves obtained. To ensure that the comparison was fair, we used the same training and test data as in the case of the whole hand. To calculate the distance between corresponding parts of the hand (i.e., fingers or back of the palm) in the query and the template hands, we used Eq. 6.1 as before.

Fig. 6.2 (blue solid line) shows the average ROC curves obtained for each part of the hand. In addition, we performed experiments using PCA to reduce the dimensionality of the feature vectors. In each case, we preserved 99.9% of the information. Fig. 6.2 (red dashed line) shows the results obtained in this case. Also, Table 6.2 shows specific details for each case using raw and PCA features when FAR = 0.1%. As it can be observed, PCA features improve accuracy slightly only in the case of the back of the palm.
Figure 6.2: Average ROC curves using different parts of the hand for verification: (a) little, (b) ring, (c) middle, (d) index, (e) thumb, and (f) back of the palm. Each experiment was performed 30 times using 5 samples for each subject as enrollment templates.
Table 6.2: Comparison using different parts of the hand for verification: raw versus PCA features.

<table>
<thead>
<tr>
<th>Finger</th>
<th>Feature</th>
<th>Little</th>
<th>Ring</th>
<th>Middle</th>
<th>Index</th>
<th>Thumb</th>
<th>Palm</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td># of Features</td>
<td>121</td>
<td>23-24</td>
<td>121</td>
<td>18</td>
<td>121</td>
<td>16</td>
</tr>
<tr>
<td></td>
<td>EER (%)</td>
<td>1.77</td>
<td>1.78</td>
<td>1.62</td>
<td>1.66</td>
<td>1.28</td>
<td>1.44</td>
</tr>
<tr>
<td></td>
<td>TAR (%) (FAR=1%)</td>
<td>96.9</td>
<td>96.6</td>
<td>97.3</td>
<td>97.1</td>
<td>98.3</td>
<td>97.8</td>
</tr>
<tr>
<td></td>
<td>σTAR (%) (FAR=1%)</td>
<td>1.3</td>
<td>1.41</td>
<td>0.79</td>
<td>0.8</td>
<td>0.54</td>
<td>0.72</td>
</tr>
</tbody>
</table>

To illustrate performance differences between various parts of the hand more clearly, we have plotted all six ROC curves, corresponding to raw features, on the same graph shown in Fig. 6.3. As it can be observed, the best performance was obtained using the index, middle, and ring fingers. Among them, the index yielded the best results. On the other hand, the thumb yielded the lowest performance among all parts. This can be explained by the fact that the thumb has higher degree of freedom than any other part, making it difficult to fix its position.

![Figure 6.3: Verification results using different parts of the hand and raw features.](source_image)
6.1.3 Verification by fusing information from different parts of the hand

In this subsection, we report results by fusing information from different parts of the hand for verification. To ensure that our results are comparable to the previous experiments, we used the same evaluation methodology as well as the same training and test sets. Using feature-level fusion, we combined the feature vectors of each part of the hand into a single feature vector yielding 861 features. Using PCA and keeping 99.9% of the information, yields between 72 and 81 features. In the case of score-level fusion using the weighted-sum rule, we experimented with different sets of weight values, using the results from the previous section as a guide. The best results, reported below, were obtained using the following values: \( w_1 = \frac{0.5}{12} \) (little finger), \( w_2 = \frac{2.5}{12} \) (ring finger), \( w_3 = \frac{3.0}{12} \) (middle finger), \( w_4 = \frac{4.5}{12} \) (index finger), \( w_5 = \frac{0.5}{12} \) (thumb), and \( w_6 = \frac{1.0}{12} \) (back of the palm). The weights were fixed in all experiments. In the case of score-level fusion using SVMs, we experimented using different parameter values. The best results (i.e., on the average) were obtained using the Gaussian kernel with \( \sigma = 0.01 \) and \( C = 1 \) (i.e., cost term). These parameter values were kept fixed in all of our experiments.

Fig. 6.4 shows the average ROC curves obtained for each fusion strategy using 3, 4, and 5 templates per subject. In general, using more enrollment templates per subject improves verification performance although it would also increase verification time. Among the four fusion strategies considered, decision-based fusion performs best. Between the two different decision-based fusion schemes considered, majority voting performs best. Feature level fusion based on PCA had the lowest performance, however, it should be mentioned that PCA reduces the size of templates more than ten times.

Fig. 6.5 compares all fusion strategies on the same graph assuming 5 enrollment
Figure 6.4: Average ROC curves for verification: (a) feature-level fusion based on PCA, (b) score-level fusion based on the weighted sum rule, (c) score-level fusion based on SVMs, and (d) decision-level fusion based on majority voting. In each case, we performed the experiments 30 times using 3, 4, and 5 templates per subject.

Additional details are shown in Table 6.3 which compares the fusion strategies, using 5 enrollment templates, in terms of EER and the mean and the standard deviation of TAR when FAR = 0.1%. As it can be observed, all fusion have improved verification performance, for example, TAR is more than 99.4% when FAR is more than 0.1%. Table 6.4 shows specific details in the case of majority voting.
Figure 6.5: Comparison of the four different fusion strategies for verification using 5 enrollment templates per subject.

Table 6.3: Detailed comparison of different fusion strategies for verification using 5 enrollment templates per subject.

<table>
<thead>
<tr>
<th>Method</th>
<th>PCA</th>
<th>Weighted Sum</th>
<th>Majority Voting</th>
<th>SVM</th>
</tr>
</thead>
<tbody>
<tr>
<td>EER(%)</td>
<td>0.523</td>
<td>0.052</td>
<td>0.044</td>
<td>0.136</td>
</tr>
<tr>
<td>TAR(%) (FAR=0.1%)</td>
<td>99.47</td>
<td>99.98</td>
<td>99.98</td>
<td>99.86</td>
</tr>
<tr>
<td>σ_{TAR}(%) (FAR=0.1%)</td>
<td>0.231</td>
<td>0.052</td>
<td>0.059</td>
<td>0.12</td>
</tr>
</tbody>
</table>

Table 6.4: Fusion using majority voting for verification: mean and standard deviation of TAR when FAR = 0.1%.

<table>
<thead>
<tr>
<th>No. of Training vectors</th>
<th>3</th>
<th>4</th>
<th>5</th>
</tr>
</thead>
<tbody>
<tr>
<td>TAR (%)</td>
<td>99.92</td>
<td>99.96</td>
<td>99.98</td>
</tr>
<tr>
<td>σ_{TAR} (%)</td>
<td>0.1115</td>
<td>0.0697</td>
<td>0.0594</td>
</tr>
</tbody>
</table>
6.2 Hand-Based Identification Results

For person identification, the user does not provide any identity claim, but the system must find out the user’s identity by comparing him/her to a database of enrolled users. Assuming that there are \( N \) subjects in the database and that each subject \( i \) has \( k_i \) templates, then the total number of templates stored in the database is \( K = k_1 + k_2 + ... + k_N \). Given a query hand \( Q \), we compute the Euclidean distance between \( Q \) and all the templates \( T_j, j = 1, ..., K \) in the database. Then, the identity of the user is established by finding the minimum distance, that is, by finding the subject whose template(s) best match the query hand:

\[
i^* = \arg\min_{i} \{||Q - T^i_j||\}, j = 1, ..., k_i, i = 1, ..., N
\]  

(6.2)

where \( T^i_j \) corresponds to the \( j \)-th template of the \( i \)-th subject in the database. In our experiments, we have assumed a "closed-universe" [28], that is, we have assumed that the user is among the subjects stored in the database (i.e., has already enrolled). The "closed-universe" model allows to investigate how good is our recognition algorithm at identifying a query hand by not only asking the question "is the top match correct?" but also the question "is the correct match among the top n matches?". This allows to determine how many templates must be examined in order to get a desired level of performance. In this context, we report identification results by using Cumulative Match Characteristic (CMC) curves which are plots of true match rate versus rank [28]. It should be mentioned that in the case of an "open-universe" [28] (i.e., the user has not enrolled), the minimum distance must also be below a threshold in order to be able to reject imposters.

In the following subsections [10], we report identification results based on similar experiments as in the case of verification. First, we investigate the performance
of a baseline system which uses the whole hand for identification purposes. Then, we investigate the recognition power of different parts of the hand by implementing several systems that perform identification using each part of the hand separately. Finally, we evaluate the proposed system which fuses information from different parts of the hand for identification.

### 6.2.1 Identification using whole hand

To provide a baseline for comparisons, first we experimented with a simpler system that uses the whole hand for identification. In this case, a global representation of the hand was used for identification. Fig. 6.6(a) shows the average CMC curves obtained using this procedure while Fig. 6.6(c) shows the corresponding standard deviations. As it can be observed, using more templates improves recognition accuracy, however, it also increases recognition time. Since the size of the feature vectors was very high, we have also experimented with PCA to reduce their dimensionality. Using a similar procedure like in the case of verification (i.e., see Table 6.1), we obtained the average CMC curves shown in Fig. 6.6(b). The corresponding standard deviations are shown in Fig. 6.6(d). As it can be observed, PCA has almost identical performance to the approach using raw features.

### 6.2.2 Identification using different parts of the hand

To investigate the identification power of different parts of the hand, we experimented with several systems, each performing recognition using a different part of the hand. Each system was tested using 5 enrollment templates and repeating the experiments 30 times as before by choosing the enrollment templates randomly each time. For each system, we report the average CMC curve obtained. To ensure that the comparison was fair, we used the same training and test data as in the case of the whole hand. To
Figure 6.6: Average CMC curves using whole hand for identification: (a) raw features, (b) PCA features, (c) standard deviation of raw features, and (d) standard deviation of PCA features. Each experiment was repeated 30 times, using 3, 4, and 5 enrollment templates per subject.

calculate the distance between corresponding parts of the hand (i.e., fingers or back of the palm) in the query and the template hands, we used Eq. 6.2 as before. Fig. 6.7 (blue solid line) shows the average CMC curves obtained for each part of the hand. In addition, we performed experiments using PCA to reduce the dimensionality of the feature vectors. In each case, we preserved 99.9% of the information (i.e., see Table 6.2 for details). Fig. 6.7 (red dashed line) shows the results obtained in this case. As it can be observed, PCA has slightly worse recognition accuracy to the approach using raw features, especially for low ranks.
Figure 6.7: Average CMC curves using different parts of the hand for identification: (a) little, (b) ring, (c) middle, (d) index, (e) thumb, and (f) back of the palm. Each experiment was performed 30 times using 5 samples for each subject as enrollment templates.
To illustrate performance differences between various parts of the hand more clearly, we have plotted all six CMC curves, corresponding to raw features, on the same graph shown in Fig. 6.8(a). The corresponding standard deviations are shown in Fig. 6.6(b) Fig. 6.8(b). As it can be observed, the best performance was obtained using the index, middle, and ring fingers. Among them, the index yielded the best recognition results (i.e., both higher accuracy and lower standard deviation). On the other hand, the thumb yielded the lowest recognition performance among all parts (i.e., both worst accuracy and higher standard deviation). These results are consistent with those obtained for verification.

6.2.3 Identification by fusing information from different parts of the hand

In this subsection, we report results by fusing information from different parts of the hand for identification. In particular, we tested the same fusion strategies except score-level fusion using SVMs since the use of SVMs for identification would require extending SVMs to multiple-class classification. Such an extension would require a
large number of training samples (i.e., enrollment templates) per subject to guarantee good performance. As previously, we used the same evaluation methodology as well as the same training and test sets for consistency. Fig. 6.9 shows the average CMC curves obtained for each fusion strategy using 3, 4, and 5 templates per subject. In general, using more enrollment templates per subject improves identification performance although it would also increase identification time. Among the three fusion strategies considered, score-level fusion had slightly better performance (i.e., higher accuracy and lower standard deviation) for low ranks. Fig. 6.10(a) compares all three fusion strategies on the same graph assuming 5 enrollment templates. Fig. 6.10(b) shows the corresponding standard deviations.

6.3 Comparison between global-based and component-based hand representations

Representing the hand in terms of its components and performing verification or identification using different parts of the hand separately or fusing information from different parts of the hand offers important advantages both in terms of time and accuracy. In terms of time, a component-based representation of the hand allows for representing shape information using a smaller set of features and lower Zernike moment orders. In terms of accuracy, a component-based representation, using fusion or individual parts of the hand, improves verification and identification performance compared to using a global-based representation.

Fig. 6.11(a) shows the ROC curves corresponding to the three most representative verification approaches tested here: (i) whole hand, (ii) index finger only, and (iii) fusion based on majority voting. Obviously, fusion improves performance significantly (e.g., when FAR=1%, TAR increases from 96.06% in the case of whole hand to 100%
Figure 6.9: Average CMC curves for identification using (a) feature-level fusion based on PCA, (b) score-level fusion based on the weighted sum rule, and (c) decision-level fusion based on majority voting. In each case, we performed the experiments 30 times using 3, 4, and 5 templates per subject.
Figure 6.10: ((a) Comparison of different fusion strategies for identification using 5 enrollment templates per subject; (b) Standard deviation.

in the case of fusion). Similarly, Fig. 6.11(b) shows the CMC curves corresponding to the three most representative identification approaches tested here: (i) whole hand, (ii) index finger only, and (iii) fusion based on weighted sum. Obviously, fusion improves identification performance significantly (e.g., when rank=1, recognition accuracy increases from 96.75% in the case of whole hand to almost 100% in the case of fusion).

It should be mentioned that the main reason that the baseline approach (i.e., whole hand) did not perform very well is because it cannot tolerate well finger motion. As shown in Figs. 3.2(b) and (c), finger motion is unavoidable in different sample images of the same subject. Although Zernike moments can tolerate some degree of finger motion (e.g., 6 degrees rotation about the axis being perpendicular to the joint of the finger with the palm), they are sensitive to larger finger motions. Moreover, they cannot tolerate well situations where the hand is bent at the wrist. Fig. 6.12, illustrates that finger motion affects the Zernike moments of all orders. Segmenting the hand in different parts alleviates these problems.
Figure 6.11: Comparison of the three most representative approaches for (a) verification and (b) identification, using 5 enrollment templates per subject: (i) whole hand, (ii) index finger, and (iii) majority voting (for verification) and weighted sum (for identification).

Figure 6.12: (a), (b) Images of the same hand containing finger motion, (c) normalized Zernike moment differences.

6.4 Comparisons with other approaches

In this section, we report both qualitative and quantitative results between our method and methods reported in the literature. Table 6.6 shows a qualitative comparison of the performance of our system and methods reported in the literature. Since there is no standard acquisition method and no benchmark databases, quan-
tative comparisons of different systems should be considered only indicative and not conclusive. To make the comparison more fair, for each study considered, we report several other factors including the number of subjects, the number of images per person, the number of enrollment templates, the use/no-use of pegs, the type of features, and the distance measure. The results reported for our system in Table 6.6 correspond to using 5 enrollment templates. Our database size is comparable to most of the systems reported in the table while our error rates are better than or equal even to the ones reported on much smaller databases.

As it can be observed from Table 6.6, the majority of existing systems employ hand geometric features for verification or identification. It has been illustrated in the literature that these features work well and can be computed efficiently. To better assess the performance of our method, we have performed quantitative comparisons, using the same database, to investigate whether Zernike descriptors offer any potential advantages over geometric features in terms of robustness and accuracy. The geometric features used in our experiments is a subset of the features introduced by Sanchez-Reillo et al. in [53, 62]. Figure 6.13(a) shows a sample image taken by their image acquisition system. They used 31 features (see Figure 6.13(b)): width of four fingers and palm in different locations (18 features), height of middle and little fingers and palm (3 features), distances between the three inter-finger points (3 features) and angles between the inter-finger points and horizontal line (3 features), distances between a middle point of the finger and the middle point of the straight line between the inter-finger point and the last height where the finger width is measured (4 features). However, their image acquisition system uses a mirror to capture a side view of the hand in addition to a top view of the hand as shown in Figure 6.13(a). Since our system captures a top view of the hand only, we cannot extract the height of the little and middle fingers as well as the palm (3 features). Therefore, we have used
Figure 6.13: (a) A sample hand image taken using the image acquisition system in [53, 62], (b) Location of measurement points for feature extraction in [53, 62] and (c) the main measured distances according [53, 62] on a sample hand image in our database.

only 28 features in our experimental comparisons. Figure 6.13(c) shows the main distances measured on the binarized hand images from our database.

Systems employing pegs to fix the position of the hand, such as [53, 62], use pre-destined axes to facilitate feature extraction. In the case of peg-free systems, several landmarks on the hand, such as fingertips and valleys, must be extracted in order to define the same or similar axes [74, 81, 77, 19]. Here, we compute the curvature of the hand boundary to extract the fingertip and valley locations by detecting curvature minima and maxima. The same methodology has been employed in several other peg-free systems including [81, 74, 77]. We provide more details about the landmarks extraction algorithm in the next section.

Figure 6.14(a) and (b) show the performance of hand based authentication (i.e., using majority voting) and identification (i.e., using weighted sum) between our method based on Zernike descriptors and the method of [53, 62] based on geometric features. In these experiments, 5 templates per person were employed for enrollment while the rest of them were used for testing. We report the average performance as before by repeating each experiment 30 times. As it can be observed, system performance using Zernike descriptors is superior to using geometric features both in the case of verification and identification. Therefore, Zernike descriptors seem to be more powerful
compared to geometric features. In terms of time, as we discussed in Section 5.1, it takes less than 0.01 seconds on the average to compute moments up to order 30 on a 3.19 GHz 64-bits machine with 2GB of RAM, assuming double precision. Table 6.5 compares the time requirements of the proposed method and the method based on geometric features. Although the preprocessing step of the proposed method is more time consuming, it is most robust than detecting landmark points on the hand as illustrated in the next section.

Table 6.5: Processing time for traditional and proposed methods. All numbers are in milli-seconds.

<table>
<thead>
<tr>
<th>Method</th>
<th>Preprocessing</th>
<th>Feature Extraction</th>
</tr>
</thead>
<tbody>
<tr>
<td>Traditional</td>
<td>Constructing Axes / 30†</td>
<td>Geometric / 20†</td>
</tr>
<tr>
<td>Proposed</td>
<td>Hand-Arm Seg. / 580†</td>
<td>Palm-Finger Seg. / 140†</td>
</tr>
</tbody>
</table>

†: ms in MATLAB 7.4 on a 3.19 GHz, 64-bit machine with 2GB RAM
‡: ms in Visual Studio 2005 on a 3.19 GHz, 64-bit machine with 2GB RAM

Figure 6.14: Comparison between geometric features and Zernike descriptors: (a) authentication results, (b) identification results.
6.4.1 Comparison between morphological-based finger segmentation versus landmarks-based finger segmentation

It could be claimed that segmenting the fingers and the palm could be accomplished more efficiently using landmark points on the hand than the algorithm described in Section 3.3.3 based on morphological operators. The purpose of the experiment reported in this section is to investigate this claim by comparing finger segmentation using morphological operators versus landmark points. There are two main objectives behind our comparisons: (i) to investigate the computational efficiency of each method and (ii) to investigate the effect of segmentation errors on verification and identification performance.

Figure 6.15 illustrates how landmark points on the hand could be used for finger segmentation. In particular, Figure 6.15(a) shows the location of the fingertips (blue dots) and valleys (red dots) while Figure 6.15(b) illustrates how to segment the fingers from the hand using these landmarks. It should be noted that, to segment the thumb, little finger, and index finger, some auxiliary points, namely $A'$, $C'$ and $D'$, need to be considered where $ID'$ is equal to $ID$, $HC'$ is equal to $HC$, and $EA'$ is equal to $EA$ [74, 81, 77, 19].

Accurate extraction of landmark points on the hand is a crucial step for peg-free systems [74, 81, 77, 19]. A common approach to detect and extract the fingertips and valleys involves using curvature information on the boundary of the hand [81, 74, 77]. The main idea is detecting curvature minima (i.e., fingertips) and maxima (i.e., valleys); we have adopted this methodology here. The curvature $k$ of a planar curve, at a point on the curve, is defined as the instantaneous rate of change of the slope of the tangent at that point with respect to arc length, and it can be expressed as follows:

$$ k = \frac{d\theta}{ds} $$
Figure 6.15: (a) A sample hand contour and its landmarks. Points $E - I$ and $A - D$ show the location of the fingertips and valleys; (b) finger segmentation using the landmarks in (a) and some auxiliary points.

\[ k(t) = \frac{(\dot{x}(t)\ddot{y}(t) - \ddot{x}(t)\dot{y}(t))^2}{(\dot{x}(t)^2 + \dot{y}(t)^2)^{\frac{3}{2}}} \]  

(6.3)

where $\dot{x}(t)$, $\ddot{x}(t)$, $\dot{y}(t)$, and $\ddot{y}(t)$ are the first and second derivatives of $x(t)$ and $y(t)$ respectively, and $(x(t), y(t))$ is the parametric representation of the curve. To account for noise, $x(t)$ and $y(t)$ are typically smoothed using a Gaussian function $g(t, \sigma)$ [45, 15]. The smoothed curve curvature $k(t, \sigma)$ can be expressed as follows:

\[ k(t, \sigma) = \frac{(\dot{X}(t, \sigma)\ddot{Y}(t, \sigma) - \ddot{X}(t, \sigma)\dot{Y}(t, \sigma))^2}{(\dot{X}(t, \sigma)^2 + \dot{Y}(t, \sigma)^2)^{\frac{3}{2}}} \]  

(6.4)

where $\dot{X}(t, \sigma)$ and $\ddot{X}(t, \sigma)$ are defined at the convolution of $x(t)$ with the first and second derivatives of $g(t, \sigma)$ correspondingly. $\dot{Y}(t, \sigma)$ and $\ddot{Y}(t, \sigma)$ can be defined similarly. Before computing the curvature, the hand boundary is re-sampled at 1,024, equal-distant, points [45, 15]. Figure 6.16(b) shows the curvature of the hand contour shown in Figure 6.16(a). Choosing the value of $\sigma$ is critical to ensure both good...
Figure 6.16: (a) A sample hand contour, (b) the curvature of the hand contour. $R$ is a reference point which is used for identifying the landmarks. The hand boundary has been re-sampled at 1,024, equal-distant, points.

detection and localization. In general, smaller $\sigma$ values lead to better localization, however, noise could give rise to false positives. On the other hand, larger $\sigma$ values reduce false positives but good localization is difficult. To address this issue, multi-resolution schemes have been proposed (i.e., curvature scale-space [45]), however, time requirements are higher. Since our system produces images of high quality, we have found that a $\sigma$ value equal to 20 yields good detection and localization results.

It can be noted by observing Figure 6.16(b) that the curvature of the finger valley between the index finger and the thumb is much smaller compared to the other valleys. As a result, detecting and localizing point $D$ is more difficult than the other three valleys. Moreover, it is easy to confuse its location with other points on the hand boundary, having similar curvature values such as point $X$. To deal with this issue, we use a reference point (i.e., $R$) as shown in Figure 6.16(a). A disadvantage of using the landmarks shown in Figure 6.16(a) for partitioning the hand is that the palm can not be separated from the hand silhouette. To address this issues, additional
landmark points would be needed, for example, on the opposite sides of the wrist. However, detecting and localizing these points reliably would be difficult since the curvature in the wrist region is quite as Figure 6.16(b) illustrates.

The goal of our first experiment in this section is to investigate how errors in partitioning the hand could affect verification and identification performance. Therefore, we have performed experiments to compare landmarks-based segmentation versus morphological-based segmentation. Since the palm cannot be segmented using landmark points, our comparisons used information from the fingers only. First, we computed Zernike descriptors for each finger up to order 20, segmented using landmarks or morphological operators. Then, we performed verification experiments using each finger separately in the spirit of the experiments reported in Section 6.1.2.

Figure 6.17 shows the verification results obtained for each finger. Each experiment was performed 30 times, each time using 5 samples per person for enrollment. Our results indicate that morphological-based segmentation has better performance than landmarks-based segmentation in the case of the little, ring, middle, and index fingers. In the case of the thumb, morphological-based segmentation performs better for FAR rates smaller than 0.04. Obviously, segmenting the thumb from the hand is more challenging than the rest of the fingers due to its higher flexibility. These results suggest that combining morphological-based with landmarks-based segmentation for separating the thumb from the hand might yield better results than either approach alone although it would be more time consuming. Next, we performed both verification and identification experiments by fusing information from the fingers using both majority voting and weighted sum. Figure 6.18 shows the results obtained in this case for each segmentation method. As it can be observed, landmarks-based segmentation has similar performance to morphological-based segmentation in the case of verification using majority voting. However, morphological-based segmentation performs
better than landmarks-based segmentation in all other cases.

It is worth noting by comparing Figure 6.5 to Figures 6.18(a),(b) that using information from the palm in addition to information from the fingers for verification (i.e., Figure 6.5) does not lead to significantly better results than using information from the fingers only (i.e., Figures 6.18(a),(b)). Similar conclusions can be made in the case of identification by comparing Figure 6.10 with Figures 6.18(c),(d). On the other hand, segmenting the palm from the hand is much more expensive than segmenting the fingers from the hand. This is because segmenting the fingers from the hand requires applying a single morphological closing operation using a fixed radius disk structure element. However, segmenting the palm from the hand requires applying the iterative process described in Section 3.3.2. If the palm is disregarded, then there are no significant differences in terms of time between landmarks-based segmentation and morphological-based segmentation. In particular, implementing both methods in MATLAB 7.4.0 on a 3.19 GHz 64-bits machine with 2GB of RAM, landmarks-based segmentation takes 0.09 seconds on average while morphological-based segmentation takes 0.14 seconds on average.

6.5 System Performance Over Time

In this section, we report several results to illustrate the performance of the proposed method over large lapses of time. In this context, we recorded 10 new samples from 20 of the 101 subjects after a period of 9 months (i.e., 200 images). These samples were used to test the performance of our system when there is a substantial passage time between the acquisition of the template and test images. In a similar manner as before, we repeated each experiment 30 times using 3, 4, and 5 samples from our initial data collection as enrolment templates. To keep results consistent, we used exactly the same enrolment templates in each experiment as in our previous experiments.
Figure 6.17: Average ROC curves using morphological-based and landmarks-based segmentation for verification using each finger separately: (a) little, (b) ring, (c) middle, (d) index, and (e) thumb. Each experiment was repeated 30 times using 5 enrollment templates per subject and the average is reported.
Figure 6.18: Comparison of morphological-based and landmarks-based segmentation using different fusion strategies: (a) verification using weighted sum, (b) verification using majority voting, (c) identification using weighted sum and (d) identification using majority voting. Each experiment was repeated 30 times using 5 enrollment templates per subject and the average is reported.
Figure 6.19: Effect of time lapse: (a) average ROC curves for verification based on decision-level fusion (majority voting) using 3, 4, and 5 templates per subject; (b) average CMC curves for identification based on score-level fusion (weighted sum) using 3, 4, and 5 templates per subject. The experiment was performed 30 times using the same enrollment templates as in the previous experiments. For testing, we used 200 images from 20 of the 101 subjects, taken 9 months later.

Figs. 6.19(a) and (b) show the average ROC and CMC curves respectively obtained in this case. As it can be observed by comparing Fig. 6.19(a) with Fig. 6.4(d), and Fig. 6.19(b) with Fig. 6.9(b) there is a small deterioration in system performance over time, however, this is quite reasonable and acceptable.

To further test the performance of our method on time lapse, we performed more experiments using a publicly available hand database provided by University of Notre Dame [78]. This database was created by collecting data on three different sessions. In the first session, two images from 132 subjects were collected. In the second session, which was conducted a week later, three images were collected from the same 132 subjects. The third session, which was conducted 15 weeks later from the second session, three images were collected from 177 subjects of which 86 had participated in the first two data collections [78]. The database contains both range and color images, each being $640 \times 480$ in size.

In our experiments, we used the color images of the same 86 subjects who partici-
pated in all three sessions. To extract the hand silhouette, we used the same algorithm described in [78]. However, since lighting was not uniform in all images, some areas of the palm have low contrast. As a result, the hand silhouette was defected many times and we were not able to segment the palm satisfactorily. Therefore, we decided to use only information from the fingers in our experiments. Similarly to other experiments, we computed Zernike descriptors for each finger up to order 20. Verification was performed using the majority voting rule while identification was performed using the weighted sum rule. In both cases, we used the samples from one of the three sessions as enrollment templates and the samples from the other two sessions for testing.

For consistency reasons, we adopted the same setup as in [78] to form the gallery (i.e., enrollment) and probe (i.e., test) image sets. That is, gallery images were chosen to be images collected prior to those chosen as probe images [78]. Following this rule, only images collected during the second week could serve both as probe and gallery images. For each time lapse, we performed two experiments by switching the enrollment samples with the test samples. Since the number of samples was not equal in all sessions (e.g., 2 samples per person in the first session and 3 samples per person in the second and third sessions), we report average performance for each time lapse.

Figure 6.20 shows the results obtained for each time lapse. Fig. 6.20(a) shows that the performance of our method is close to 98% when FAR is equal to 0.01; these results do not change much for different time lapses. Figure 6.20(b) illustrates that recognition is robust over time. There is a slight inconsistency for ranks 1 and 2 (i.e., the recognition rate based on 16 weeks time lapse is higher than 1 and 15 weeks time lapse), however, this is probably due to the unequal size of the data sets.

Table 6.7 reports the Equal Error Rate (EER) and TAR for each method when FAR is equal to 5% assuming a 16 week time lapse as reported in [78]. As it can be observed, our approach shows better performance. Table 6.8 reports the identification
Figure 6.20: Effect of time lapse: (a) average ROC curves for verification using score-level fusion (weighted sum) of fingers; (b) average CMC curves for identification using score-level fusion (weighted sum) of fingers. 86 subjects participated in 3 separate sessions in which 2, 3 and 3 images were taken from each subject (total 688 images). Results for each method assuming 1 week and 16 weeks time lapse as reported in [78]. Again, our methods show better performance. Moreover, our method seems to be more robustness over time since the identification rate does not change significantly.
Table 6.6: Previous published research efforts in hand based personal verification/identification.

<table>
<thead>
<tr>
<th>System(s)</th>
<th># of people per person</th>
<th>Pegs</th>
<th>Feature(s)</th>
<th>Verification Performance</th>
<th>Identification Performance</th>
</tr>
</thead>
<tbody>
<tr>
<td>Jain [2]</td>
<td>50</td>
<td>Yes</td>
<td>Geometric features [11] (16 features)</td>
<td>FAR=0.01</td>
<td>-</td>
</tr>
<tr>
<td>Wong [77]</td>
<td>22</td>
<td>12-15</td>
<td>No 13 geometric features [11] and 3 fingertip regions</td>
<td>FAR=0.022</td>
<td>-</td>
</tr>
<tr>
<td>Jain [31]</td>
<td>53</td>
<td>2-15</td>
<td>Yes Contour of five fingers</td>
<td>FAR=0.01</td>
<td>-</td>
</tr>
<tr>
<td>Reillo [53][62]</td>
<td>20</td>
<td>10</td>
<td>Yes Geometric features [11] deviation and angles between inter-finger points (31 features)</td>
<td>FAR=0.01</td>
<td>-</td>
</tr>
<tr>
<td>Ma [81]</td>
<td>20</td>
<td>6</td>
<td>No 4 B-Spline curves, length of thumb and width of palm</td>
<td>FAR=0.01</td>
<td>-</td>
</tr>
<tr>
<td>Kumar [3]</td>
<td>100</td>
<td>10</td>
<td>No Geometric features [11] and hand area (16 features)</td>
<td>FAR=0.01</td>
<td>-</td>
</tr>
<tr>
<td>Bulatov [80]</td>
<td>70</td>
<td>10</td>
<td>No 30 geometric features [11] and FRR [80] ≈ 0.32</td>
<td>FAR=0.01</td>
<td>-</td>
</tr>
<tr>
<td>Ribaric [60]</td>
<td>130</td>
<td>5</td>
<td>No 20 geometric features [11] and FRR [60] ≈ 0.33</td>
<td>FAR=0.153</td>
<td>-</td>
</tr>
<tr>
<td>Xiong [74]</td>
<td>108</td>
<td>5</td>
<td>No Width of 4 fingers at 45 different location</td>
<td>FRR=1.15×10⁻²</td>
<td>98.81×10⁻²</td>
</tr>
<tr>
<td>Oden [49]</td>
<td>35</td>
<td>not clear</td>
<td>No Combination of implicit polynomials and geometric features (16 features)</td>
<td>FAR=0.01</td>
<td>95.0×10⁻²</td>
</tr>
</tbody>
</table>

| Our method  | 100                    | 10   | No Zernike moments (861 features for fingers and palm.) | FAR=0.01                 | 99.98×10⁻²                |

1. Out of 500 images, only 360 images were used and 140 images were discarded.
2. Estimated from ROC curve in [2].
3. A total of 288 images were used.
4. A total of 353 images were used.
5. Not all possible non-matching pairs were used.
6. The minimum error rate, which is the sum of FAR and FRR.
7. A total of 714 images were used.
8. This is the best EER using 5 training vectors and GMM for verification [62].
9. Hand geometry was used to improve the performance of palmprint-based verification. We have estimated FRR using only hand geometry information from the ROC curve in [3] when FAR=0.01.
10. A multi-modal biometric system was designed in [60] using fingerprint, palmprint, and hand geometry. The FAR and FRR reported here relates to hand geometry only, see [60].
11. Geometric features such as length and width of the fingers, width of palm, thickness of hand and middle finger, etc.
12. The database includes 458 people with 3 samples per person. EER has been reported for different populations (i.e. 20, 35, 50, 100, and 458).

Table 6.7: Time lapse verification performance comparison between our method and Woodard’s method [78] using the Notre Dame University database.

<table>
<thead>
<tr>
<th>Time lapse</th>
<th>Method</th>
<th>Woodard [78]</th>
<th>Proposed</th>
</tr>
</thead>
<tbody>
<tr>
<td>16 Week</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

| EER        | 5.5%   | 1.72%         |
| TAR (FAR=5%)| 94%    | 99.3%         |
Table 6.8: Time lapse identification performance comparison between our method and Woodard’s method [78] using the Notre Dame University database.

<table>
<thead>
<tr>
<th>Time lapse</th>
<th>1 Week</th>
<th>16 Week</th>
</tr>
</thead>
<tbody>
<tr>
<td>Method</td>
<td>Woodard [78]</td>
<td>Proposed</td>
</tr>
<tr>
<td>Recognition Rate</td>
<td>91%</td>
<td>97.7%</td>
</tr>
<tr>
<td></td>
<td>Woodard [78]</td>
<td>Proposed</td>
</tr>
<tr>
<td>Recognition Rate</td>
<td>94%</td>
<td>98.4%</td>
</tr>
</tbody>
</table>
Chapter 7

GENDER CLASSIFICATION

7.1 Introduction

Gender classification is an important problem with a variety of practical applications. For example, a robust gender classification system could provide a basis for performing passive surveillance using demographic information or collecting valuable consumer statistics in a shopping center. It could also be used to improve the performance of biometric systems such as face authentication and recognition [73]. In computer vision, the majority of studies on gender classification are based on face since visual information from human faces provides important cues for gender classification. A recent study comparing different gender classification approaches using face information can be found in [41]. A very small number of studies have also investigated the use of modalities other than face including gait, [64], iris [71] and fingerprint [14].

In this study, we investigate the problem of gender classification from human hands. To the best of our knowledge, this is the first study attempting to address the issue of gender classification from hand images in computer vision. However, extracting gender information from human hands has been studied for a long time fields such as anthropology and psychology. For example, several studies have found
Table 7.1: Measurements of hand breadth and length (based on centimeters) [5].

<table>
<thead>
<tr>
<th>Gender</th>
<th>$N$</th>
<th>min</th>
<th>max</th>
<th>$\mu$</th>
<th>$\sigma$</th>
<th>min</th>
<th>max</th>
<th>$\mu$</th>
<th>$\sigma$</th>
<th>$\mu$</th>
</tr>
</thead>
<tbody>
<tr>
<td>Male</td>
<td>125</td>
<td>7.3</td>
<td>9.4</td>
<td>8.45</td>
<td>0.40</td>
<td>15.3</td>
<td>21.0</td>
<td>18.89</td>
<td>0.88</td>
<td>44.73</td>
</tr>
<tr>
<td>Female</td>
<td>125</td>
<td>6.7</td>
<td>8.8</td>
<td>7.48</td>
<td>0.38</td>
<td>14.8</td>
<td>20.4</td>
<td>17.22</td>
<td>0.92</td>
<td>43.46</td>
</tr>
</tbody>
</table>

that there is significant difference between the hand dimensions of males and females. In 1875, Ecker [20] noted that the following three relations of relative finger length may be observed in a human hand: i) the index finger is shorter than the ring finger; ii) the index finger is equal in length to the ring finger; and iii) the index finger is longer than ring finger. In 1877, Mantegazza [42] found that all three relations occur in both sexes, however, a relatively long index finger is found more frequently in females than in males. In 1930, George [22] studied a Canadian population and found that relation (i) is more frequent in males while relation (ii) is predominant in females. Similar results have been reported in more recent studies [16].

In [43], McFadden and Shubel studied all six possible ratios between the index, middle, ring and little fingers in both genders. Their results indicate that the ratio exhibiting the largest gender difference was the relative lengths of the index and ring fingers. In another study, Agnihotri et al. [5] found that the average hand breadth and hand length are about 1 cm and 1.5 cm correspondingly greater in male subjects than in female subjects. By defining hand index as the ratio of hand breadth over hand length, they found that the average hand index in males was more than 44% while the average index in females was less than 44%. Table 7.1 provides more details. Based on these results, they suggested using this value as a threshold for determining gender by hand dimensions. However, to date there is no conclusive evidence as to which features determines gender robustly. It appears that gender can not be determined using a single feature, but rather involves the combination of multiple features.

Motivated by these studies, we investigate the problem of gender classification.
from hand images by extracting more powerful hand features. Our goal is building a system that can distinguish between male and female subjects using hand shape information. For classification, we compute the distance of a given part from two different eigenspaces, one corresponding to the male class and the other corresponding to female class. We have experimented using each part of the hand separately as well as fusing information from different parts of the hand.

Determining gender information from hand images has several important advantages. First of all, capturing an image of the hand can be done more robustly than capturing an image of the face. There are several biometric systems today that can capture high quality hand images by controlling the position and orientation of the hand as well as illumination [65]. Second, assuming that the hand is placed on a flat surface for image acquisition purposes, which is typical for hand-based authentication applications, hand appearance shows less variability compared to face appearance which is affected by many factors such as facial expression changes. Finally, gender information from hand images could be very valuable in improving the accuracy and robustness of hand-based authentication and identification systems [7][6].

7.2 Fourier Descriptors

MPG-7 divides shape descriptors in two categories: contour-based and region-based. Contour-based shape descriptors use the shapes boundary to extract shape information, while region-based shape descriptors exploit the shapes region to represent shape information. In this study, we experimented Fourier Descriptor as a contour descriptor besides Zernike Moment as a region descriptor.

Fourier Descriptors have been used in a wide range of applications to describe the boundary of an object. Let us consider a closed contour \( C \) in the complex plane. In this case, the \( x-y \) coordinates of each point in the boundary become a complex
number $x + jy$. By tracing the boundary in a counterclockwise direction with uniform velocity, a complex function $z(t) = x(t) + jy(t)$ is obtained with parameter $t$. The velocity is chosen such that the time required to traverse the contour is $2\pi$. If $z(k)$ is a uniformly re-sampled version of $z(t)$ of dimension $N$, its Discrete Fourier Transform (DFT) is given by following equation:

$$z(k) = \sum_{n=0}^{N} a_n e^{\frac{j2\pi nk}{N}}$$ (7.1)

where $a_n$ is Fourier coefficient of $z(k)$. The Fourier Descriptors of the closed contour $C$ are defined by taking the inverse transform:

$$a_n = \frac{1}{N} \sum_{k=0}^{N} z(k) e^{-\frac{j2\pi nk}{N}}, \quad n \in \{1, 2, ..., N\}$$ (7.2)

To normalize the FDs with respect to translation, rotation, scale, and starting point, we used the methodology proposed in [75]. In [75], the dimensionality of the contour ($N$) must be power of 2. Since the average number of points in the boundary of different parts of the hand was in the range of $[2^7, 2^8]$ for the fingers and $[2^8, 2^9]$ for the palm, we re-sample the finger and palm contours in 256 and 512 points respectively.

### 7.3 Gender Classification Using Different Parts of the Hand

To investigate the discrimination power of different parts of the hand for gender classification, we considered each part of the hand separately. To represent each part compactly, we applied Principal Component Analysis (PCA) [52]. Specifically, we built two different eigenspaces for each part of the hand, one for the male class and
the other for the female class. Given that the hand is decomposed in six parts (i.e., five fingers and the palm), we built a total of twelve eigenspaces.

To represent an instance of given part, we compute its distance from the male and female eigenspaces. This was performed by projecting the part in each eigenspace and reconstructing it from its projections. To compute the error in each eigenspace, we computed the difference between the original representation of the part and its reconstruction. Specifically, let us assume that $\Omega_{m/f}$ corresponds to the representation of an instance $\Phi$ of some part $p$ in the male/female eigenspaces; then $\Omega_{m/f}$ is given by:

$$
\Omega_{m/f} = \sum_{k=0}^{M} \omega_{m/f}^{k} u_{m/f}^{k} + \bar{\Phi}_{m/f}
$$

(7.3)

where the projection $\omega_{m/f}^{k}$ of $\Phi$ in the male/female eigenspaces can be computed as follows:

$$
\omega_{m/f}^{k} = \hat{u}_{m/f}^{k}(\Phi - \bar{\Phi}_{m/f})
$$

(7.4)

$M$ represents the dimensionality of the eigenspaces, $u_{m/f}^{k}$ is the $k$th eigenvector in the male/female space and $\bar{\Phi}_{m/f}$ is the average male/female vector (i.e., computed from the training set). Also, $\hat{u}_{m/f}^{k}$ is the transpose of $u_{m/f}^{k}$. To measure the masculine/feminine characteristic of $\Phi$, the Euclidean distance $\varepsilon_{m/f}$ between $\Phi$ and its projection onto the male/female eigenspaces is computed:

$$
\varepsilon_{m/f} = \|\Phi - \Omega_{m/f}\|
$$

(7.5)

Therefore, each part $p$ is represented as a distance vector $E = [\varepsilon_{m}, \varepsilon_{f}]^{T}$. Figure 7.1 illustrates this process. In our experiments, we preserved the same amount of information for the male/female eigenspaces, however, this could be varied depending
on the shape descriptor.

Figure 7.1: Computing the masculine/feminine characteristic for an instance Φ of some part of the hand.

Figure 7.2 shows the distribution of distances $\varepsilon_{m/f}$ in the case of the little finger using ZMs and FDs. Due to lack of space, we do not show the distributions of the other parts of the hand. This male/female eigenspaces in this figure were generated using 12 males and 12 females from our database and preserving 99% information.

To classify a query distance vector $E$, we experimented with three different classifiers: *Minimum Distance* (MD), *k-Nearest Neighbors* (kNN), and *Linear Discriminant Analysis* (LDA) [52]. In the case of the MD classifier, the minimum distance from the male/female eigenspace was used to determine the gender of the input part. In the case of a tie, we arbitrarily classified the input as male. In the case of the kNN classifier, we determined the gender by finding the most dominant gender classification among the top $k$ closest distances between the query and the training data. Again, in the case of a tie, we arbitrarily classified the input as male. In the case of LDA,
Figure 7.2: Distribution of distances for the male/female eigenspaces in the case of little finger: (a) ZMs, (b) FDs.
each distance vector was represented by a single value since our problem is a two-class classification problem. A threshold needs to be used in this case in order to separate the two classes. Figure 7.3 shows the male/female distributions for the little finger using LDA. These graphs should be related to the classification results presented in Section 7.4.

Figure 7.3: Male/Female distributions using LDA: (a) using ZMs distances shown in Figure 7.2(a), (b) using FD distances shown in Figure 7.2(b).
7.4 Experimental Results

To evaluate the proposed approach, we built a small database containing 40 people. Although we had a larger database, 101 people, unfortunately, we had not recorded gender information for each subject in that database. The population of male and female subjects was equal (i.e., 20 males and 20 females). For each subject, we collected 10 images of his/her right hand in different directions and positions. Time lapse between captured samples of same subject was a few minutes. Besides asking the subjects to stretch their hand and place it inside a square area drawn on the surface of the lighting table, no other restrictions were imposed.

To evaluate the proposed system, we used cross-validation based on leave-one-out approach. Although leave-one-out is computationally expensive, it is well suited for small datasets. Moreover, it has been shown to be an almost unbiased estimator of the true error rate of a classifier [76]. Using a leave-one-out approach, we repeated each experiment 40 times, each time using all samples of one person (10 out of 400 samples) for test and the rest of samples (390 out of 400 samples) for training set. Therefore, for each experiment, we report the average over 40 trials. Zernike moments were computed up to order 20 and 30 for the fingers and palm respectively. The number of features was 121 for each finger and 256 for the palm. The number of Fourier coefficients was 256 for the fingers and 512 for the palm.

7.4.1 Gender Classification Results using Different Parts of the Hand

In this subsection, the performance of each part of the hand is presented using different shape descriptors and classifiers as described in Section 7.3. Different portion of information has been preserved in building the male/female eigenspaces (i.e. 90%,
95%, 97% and 99%) [11].

Figures 7.4 - 7.9 show gender classification results using different parts of the hand and different combinations of features (i.e., ZMs or FDs) and classifiers (i.e., MD, kNN, and LDA). In particular, figures 7.4, 7.5 and 7.6 show gender classification results using Zernike moment. The results are consistent across all classifiers, showing that the best accuracy is achieved by the thumb while the lowest accuracy is achieved by the ring finger. When considering FDs (i.e., Figures 7.7, 7.8 and 7.9), the ring finger shows comparable performance with the rest of parts.

![Zernike Moments](image)

Figure 7.4: Gender classification results using ZMs and MD classifier.

Comparing ZMs with FDs, the results are rather mixed. For certain parts (e.g., ring finger), FDs seem to be performing better, while for others (thumb and palm), ZMs seem to be performing better. An interesting observation, however, is that the amount of information preserved when building the male/female eigenspaces seems to affect the FDs more than the ZMs. This is illustrated more clearly in Figures 7.5.
Figure 7.5: Gender classification results using ZMs and $k$-NN classifier where $k \in \{1, 5, 10, 15, 20\}$.

Figure 7.6: Gender classification results using ZMs and LDA.
and 7.8. Overall, the highest accuracy in the case of ZMs was obtained using the thumb and the LDA classifier (i.e., close to 89%). In the case of FDs, the highest accuracy was obtained using the little finger and either the kNN or LDA classifiers (i.e., close to 87%).

Figure 7.7: Gender classification results using FDs and MD classifier.

7.4.2 Gender Classification Results using Fusion

In this section, we report classification results by fusing information from different parts of the hand. Since LDA performed the best in our previous experiments, we only considered LDA for classifying different parts of the hand for fusion purposes. Table 7.2 shows the classification results obtained using feature-level fusion. As discussed in Section 5.2, we experimented with fusing two types of features: (i) the original ZMs or FDs and (ii) the distance vectors $E$. As it can be observed, feature-level fusion improved classification results in certain cases, both for ZMs and FDs.
Figure 7.8: Gender classification results using FDs and $k$-NN classifier where $k \in \{1, 5, 10, 15, 20\}$.

Figure 7.9: Gender classification results using FDs and LDA.
Table 7.2: Gender classification using feature-level fusion.

<table>
<thead>
<tr>
<th>Preserved info</th>
<th>Zernike Moments</th>
<th>Fourier Descriptor</th>
</tr>
</thead>
<tbody>
<tr>
<td>90% (PCA)</td>
<td>91.75</td>
<td>93.5</td>
</tr>
<tr>
<td>95% (PCA)</td>
<td>90.75</td>
<td>92.25</td>
</tr>
<tr>
<td>97% (PCA)</td>
<td>92.25</td>
<td>91</td>
</tr>
<tr>
<td>99% (PCA)</td>
<td>93</td>
<td>89</td>
</tr>
</tbody>
</table>

Table 7.3: Gender classification using score-level fusion.

<table>
<thead>
<tr>
<th>Preserved info</th>
<th>Zernike Moments</th>
<th>Fourier Descriptor</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Sum Rule</td>
<td>Weighted Sum</td>
</tr>
<tr>
<td>90% (PCA)</td>
<td>94.5</td>
<td>97.75</td>
</tr>
<tr>
<td>95% (PCA)</td>
<td>91.25</td>
<td>96</td>
</tr>
<tr>
<td>97% (PCA)</td>
<td>91.75</td>
<td>96.25</td>
</tr>
<tr>
<td>99% (PCA)</td>
<td>92.25</td>
<td>96.5</td>
</tr>
</tbody>
</table>

Table 7.3 shows the classification results obtained using score-level fusion (i.e., weighted sum rule). The main issue with this method is determining a set of appropriate weight values. Here, we chose the weight values by considering the discrimination power of each part of the hand as described in Section 7.3. For comparison purposes, we also experimented with the simple Sum rule where all the weights are equal (i.e., average over scores). As it can be observed, score-level fusion improves classification results significantly, both for ZMs and FDs. As expected, the weighted sum rule outperforms the simple sum rule.

Table 7.4 shows the classification results obtained using decision-level fusion (i.e., majority voting). As it can be observed, decision-level fusion improves classification results both for ZMs and FDs. Figure 7.10 shows the best classification rates obtained using different fusion strategies and features. Comparing all three fusion strategies, it is clear that score-level fusion outperforms the other two. Since different parts of the hand have different discrimination power, it is not surprising that score-level fusion using weighted-sum outperforms decision-level fusion. Obviously, weighting each part of the hand appropriately has an important impact on system error. It is
Table 7.4: Gender classification using decision-level fusion.

<table>
<thead>
<tr>
<th>Majority Voting</th>
<th>Preserved information</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>90% (PCA)</td>
</tr>
<tr>
<td>Zernike Moments</td>
<td>94.5</td>
</tr>
<tr>
<td>Fourier Descriptors</td>
<td>95.25</td>
</tr>
</tbody>
</table>

worth noticing that when each part of the hand is assigned the same weight (i.e., sum rule), score-level fusion performs almost the same as decision-level fusion. Overall, the best performance (i.e., 98%) was obtained using score-level fusion and FDS. It should be observed, however, that the performance of ZMs using score-level fusion is very close to that of FDs.

Figure 7.10: Best gender classification results using different fusion strategies and shape descriptors.
Chapter 8

TEMPLATE UPDATE

8.1 Introduction

A typical biometric verification system operates by acquiring biometric data (i.e. hand) from a subject and comparing it against the template set of that subject, stored in a database, in order to verify a claimed identity. Most systems store multiple templates of a person in order to account for variations observed in biometric data. In fact the biometric measurements tend to have a large intra-class variability. Thus, it is possible for the stored template data to be significantly different from those obtained during system’s operation, resulting in an inferior performance (higher false rejection rate) of the biometric system. Figure 8.1 shows an example of this variation. As you can see, shape of little finger in Figure 8.1(b) is curvier compare to one in Figure 8.1(a). Also thumb is bent more in Figure 8.1(b). More over cutting fingernails in Figure 8.1(b) causes some changes in the shapes of point, ring and little fingers (i.e. length of the finger).

Substantial intra-class variations are exhibited in the input data, non representative of the enrolled templates, which decreases the performance of the system. This issue has been recently faced by template update techniques. The earlier approaches,
known as supervised learning methods, were based on enrolling multiple templates per person representing temporary variations, and by repeating the process of enrollment over period of the time to capture variations in the biometric data. Uludag et al. [72] proposed two simple methods to perform template update using the newly acquired data. In the first method, namely *Batch Update*, all current templates are replaced with templates selected from the newly acquired data set, thereby capturing temporal changes (i.e. in fingerprints) [72]. In the second method, namely *Augment Update*, both the current template set and the newly obtained data set are considered when performing template update [72]. These methods need a supervisor, who has to assign identity labels to the input data to be used for update, and it makes the update process very expensive, time consuming, and inefficient.

To overcome the drawback of aforementioned methods, self-update methods based on semi-supervised learning have been developed. They are self-update systems as they update themselves by iteratively classifying the unlabeled samples and modifying the enrolled templates with highly confidently classified data, using their own knowledge gained from previously augmented template set. These techniques can be categorized as *Online* and *Off-line* methods. In online update methods [33][58], templates are updated as soon as an input data arrives, however in off-line update methods [40][55][54] templates are updated after the batch of unlabeled data is collected.

Jiang and Ser [33] proposed an online fingerprint template updating algorithm by merging the input data into the template database during the system’s operation. Ryu et al. [58] proposed an online minutiae-based fingerprint template adaptation algorithm. The algorithm updates a template by using a query fingerprint, which is successfully verified by the fingerprint matcher as a high quality genuine input. Liu et al. [40] introduced an off-line update technique based more on the recent
samples and less on the older samples with application in face recognition. Roli et al. [55] proposed an off-line semi-supervised version of the classical PCA-based face recognition algorithm to update the eigenfaces and the templates. Recently Rattani et al. [54] proposed an off-line graph-based approach to template update by its application to face verification, as a case study.

Self update techniques operate at high acceptance threshold in order to avoid the introduction of impostors into the template set of a client. The impostors’ introduction leads to the so called effect of creep in of errors which strongly decrease the effectiveness of update. Moreover, due to operation at high acceptance threshold, these approaches can only exploit the input data near to the current templates resulting in local optimization of the template set and non exploitation of many difficult and informative intra-class variations.

This chapter introduces a global optimization approach to hand template update based on fusion. This method analyzes overall confidence of a hand by decomposing the hand silhouette in different parts (i.e. fingers) and fusing confidences of all fingers in order to better exploit difficult samples and use them in template update process. The maximum rule is employed to fuse confidences of all fingers. So, to update a hand template set, having a high confident finger in the verified hand sample is
sufficient. The motivation behind this technique is that the temporal changes that may occur in the fingers are uncorrelated in such a way that the confidence of each finger can be significantly different from the others. As an example, consider the hand image in Figure 8.1(a) as a enrolled template and the other one in Figure 8.1(b) as a query sample. A typical self update system can not exploit the little, ring, index and thumb fingers due to having large variation from current enroled template, just middle finger very similar to the current template can be exploited. In our proposed method [12], since the middle finger is classified as a high confidence sample, therefore all fingers can be exploited by the system. However the proposed method has the potential to identify difficult intra-class variations compare to a typical self update system, the effect of imposter introduction in the proposed method can be worse than the typical self update system. The reason is that in the typical self method each introduced imposter affects only a specific finger of a client, while in our proposed method an introduced imposter affects all fingers of a client. To avoid that a tighter acceptance threshold, compare to the typical self update method, is chosen in the proposed system. The matching for each finger is performed by a Support Vector Data Description (SVDD). Support Vector Data Description (SVDD) is a technique which uses support vectors in order to model a data set [68]. The SVDD represents one class of known data samples (i.e. a subject templates) in such a way that for a given test sample it can be recognized as known (i.e. genuine attempt), or rejected as unknown (i.e. imposter attempt).

8.2 Matching and Update through SVDD

As mentioned earlier, the matching score $s_i$ of each finger is evaluated by a Support Vector Data Description (SVDD). A normal data description gives a closed boundary around the data (i.e. a finger template set) which can be represented by a hyper-
sphere $F(R, a)$. The volume of this hyper-sphere with center $a$ and radius $R$ should be minimized while containing all the data (i.e. finger templates). As proposed in [68] the extension to more complex distributions is straightforward using kernels. For generalization purpose, slack variables $\epsilon_i \geq 0$ are introduced. The error function to be minimized is defined as:

$$F(R, a) = R^2 + C \sum_i \epsilon_i \quad (8.1)$$

subject to:

$$\|z_i - a\|^2 \leq R^2 + \epsilon_i \quad \forall i. \quad (8.2)$$

Using Lagrange optimization the above results in:

$$L = \sum_i \alpha_i K(z_i, z_i) - \sum_{i,j} \alpha_i \alpha_j K(z_i, z_j) \quad \forall \alpha_i : 0 \leq \alpha_i \leq C \quad (8.3)$$

where $\alpha_i$ is a Lagrange multiplier and $K(z_i, z_j)$ is a kernel function. In this study we employed radial basis function as kernel function. When a sample falls in the hyper-sphere then its corresponding Lagrange multiplier is $\alpha_i \geq 0$, otherwise it is zero. After optimizing the function in (8.3) the following equality constraint must hold:

$$\sum_i \alpha_i = 1 \quad (8.4)$$

When a query sample is applied to a trained SVDD, the output is its distance to the center of the hyper-sphere. In the context of our application, this distance (multiply by $-1$) is considered as matching score $s_i$. As a result matching score $s_i$ can be in the range of $(-\infty, 0]$.

In the enrollment stage, using templates of a client a SVDD classifier is trained for each finger. Therefore a subject is represented by a set of 5 SVDDs corresponding to his/her fingers. The support vectors and their corresponding Lagrange multipliers
are stored as the classifier information for each finger. This information is used later in the verification process.

During system’s operation, hand template update is performed iteratively by adding verified hand sample with high confidence. Using high confident hand sample, we update the Lagrange coefficients $\alpha_i$ in trained SVDDs classifiers using an incremental learning algorithm [51]. This method is based on the theorem proposed by Osuna et al. in [50].

After verifying an identity, maximum rule is used in score level to determine the overall confidence level of the query hand. Therefore the maximum matching score $s_{\text{max}}$ of the fingers represent the confidence of the query hand. Higher matching score indicates higher confidence. If $s_{\text{max}}$ is greater than a threshold $T_c$, then the query hand is used in update process. Usually this threshold $T_c$ is much higher than the threshold $T_s$ in verification process to avoid the introduction of impostors (false positive) into the system.

### 8.3 Experimental Results

We performed our experiments using the database provided by University of Notre Dame [79]. As it was mentioned before, this database was created by collecting data on three different sessions. In the first session, two images from 132 subjects were collected. In the second session, which was conducted a week later, three images were collected from the same 132 subjects. The third session, which was conducted 15 weeks later from the second session, three images were collected from 177 subjects of which 86 people had participated in the first two data collections [79]. The database contains both range and color images, each being $640 \times 480$ in size. In our experiments, we used the color images of the same 86 subjects who participated in all three sessions. To extract the hand silhouette from a color image, we used the same
algorithm described in [79]. As proposed in [79], we employed a combination of edge and skin detection techniques on the color image to extract hand silhouette from the intensity images. Figure 8.2 shows an intensity image sample and its hand silhouette extracted by aforementioned method.

Here, in our experiments, all hand images in the first session were used as initial templates. Since only 2 templates represent the initial template set for each client, it can not exhibit a large intra-class variations of the hand. The rest of hand images in the second and third sessions, with 6 images per client, were used as queries. As a result the query images provided 516 genuine attempts and 43860 imposter attempts. It should be noted that a template image never becomes a query image even as an impostor for other clients.

Due to training of the SVDD incrementally at each iteration, the order of training data may affect the learning process. To account for the variation in the order of the genuine queries, we repeated each experiment 10 times, each time choosing a random order of the genuine input for each client and reported the average performance. It should be noted that the relative order of genuine queries in second and third sessions does not change. In other words, a genuine sample from the third session never appears before any genuine query in the second session.
The threshold for self update techniques is always evaluated on initial template set, since it is the only set available in real environments. Threshold is evaluated on this template set by comparing each template to the templates of all the other clients thus estimating the impostor distribution and selecting a threshold value at a specific false acceptance rate (i.e. $FAR = 1\%$). Followed by the same methodology the threshold was chosen $T_c = -0.003$ in our experiments.

To make a base line for our experiments, we employed a conventional verification system which does not utilize any update scheme. In this system, SVDDs are trained once using initial templates in the first session. Therefore the order of query samples is not important in this case.

Figure 8.3 shows the performance of different fingers before and after utilizing a self update system at different confidence thresholds. Figure 8.3(e) indicates that the substantial intra-class variation in the thumb is so large. As a result its performance is the lowest one among other fingers and self update technique is not effective in this case. As you can see in Figure 8.3(a-f), adopting very high threshold (i.e. $T_c = -0.001$) limits the capability of the system to capture the substantial intra-class variations in the subject’s input data. Also adopting low threshold (i.e. $T_c = -0.01$) introduces impostors into the template set of a client resulting in poor performance. Table 8.1 shows the true acceptance rate (TAR) of the conventional verification system before and after utilizing self update when false acceptance rate (FAR) is equal to 1\%. As you can see in table 8.1, self update technique improves the performance slightly as it can exploit only the patterns similar to the enrolled templates which leads to local optimization and non-exploitation of many difficult and informative intra-class variations.

For comparison purpose, we employed a typical self update system in which each part of the hand (i.e. finger) is updated independently using a high confident verified
Table 8.1: True acceptance rates of different fingers before and after utilizing self update technique when false acceptance rate is equal to 1%.

<table>
<thead>
<tr>
<th>Method</th>
<th>Little</th>
<th>Ring</th>
<th>Middle</th>
<th>Point</th>
<th>Thumb</th>
</tr>
</thead>
<tbody>
<tr>
<td>No Update</td>
<td>74.4%</td>
<td>76.2%</td>
<td>80.6%</td>
<td>74.5%</td>
<td>57.2%</td>
</tr>
<tr>
<td>Self Update</td>
<td>77.3%</td>
<td>77.0%</td>
<td>81.2%</td>
<td>75.7%</td>
<td>58.3%</td>
</tr>
</tbody>
</table>

query finger (with respect to its matching score above a threshold $T_c$). As a result, each time the SVDD of one or more fingers might be updated using a query hand sample.

Figure 8.4 shows the performance of the hand verification system before and after utilization of the update techniques: typical self update and proposed fusion-based self update. Also table 8.2 shows the equal error rate and true acceptance rate ($TAR$) of the hand verification system, when $FAR$ is equal to 0.1% and 1.0%, before and after utilization of the update techniques$^1$.

As it can be seen, the typical self update method improve the overall performance of the system slightly, while the proposed method improved the performance of system significantly. In fact the proposed approach can exploit difficult and informative intra-class variations. Our investigation indicated that there is no correlation between the confidence of different fingers in the hand, because the temporal changes that may occur in the fingers are uncorrelated. Therefore in a genuine hand, however some fingers may have high matching scores, some others may have very low confidence (i.e. thumb). In a typical self update system, these fingers with low confidence, which have informative intra-class variations, never can participate in the update procedure. In our proposed system, these fingers can contribute in the update process if and only if one of the fingers has a high confidence.

In the proposed method, the effect of imposter introduction is greater than the

---

$^1$All these methods are implemented in MATLAB 7.6.0 using data description toolbox[67].
typical self update method. The reason is that in the typical self method each introduced imposter affects only a specific finger’s SVDD, while in our proposed method an introduced imposter affects all fingers’ SVDDs. To reduce the effect of imposters’ introduction, a tighter acceptance threshold compare to the typical self update method is chosen in the proposed system. In figure 8.4 the threshold for the typical self update was -0.003 and for our proposed method was -0.0025.

Table 8.2: True acceptance rates and equal error rates of the hand verification system before and after utilizing a typical self update and proposed fusion-based template update.

<table>
<thead>
<tr>
<th>Method</th>
<th>TAR_{(FAR=0.1%)}</th>
<th>TAR_{(FAR=1%)}</th>
<th>EER</th>
</tr>
</thead>
<tbody>
<tr>
<td>No Template Update</td>
<td>87.4%</td>
<td>95.3%</td>
<td>2.50%</td>
</tr>
<tr>
<td>Typical Self Update</td>
<td>88.0%</td>
<td>95.8%</td>
<td>2.49%</td>
</tr>
<tr>
<td>Fusion-Based Update</td>
<td>93.6%</td>
<td>98.0%</td>
<td>1.50%</td>
</tr>
</tbody>
</table>
Figure 8.3: Verification results of each finger using different confidence thresholds in a typical self update system. SVDD of each finger in a client profile is updated using a genuine finger input with matching score higher than $T_c$. 
Figure 8.4: Performance of the hand verification system before and after utilizing a typical self update and proposed fusion-based template update. In all methods, matching scores of the fingers were fused by majority voting technique in verification process.
Chapter 9

CONCLUSIONS AND FUTURE WORK

9.1 Conclusions

We have presented a new approach to hand-based verification and identification using a component-based representation of the hand and fusion. The proposed method has several advantages including that it is peg-free, it does require the extraction of any landmark points on the hand, it is independent of the position and orientation of the hand, and tolerates finger motion very well. The only restriction imposed by our system is that users must stretch their hand during image acquisition to avoid touching figures.

Our system represents the geometry of the fingers and the back of the palm using translation, rotation and scale invariant Zernike moments. To improve the computational efficiency and accuracy of high-order Zernike moments, we have adopted an improved algorithm that avoids redundant computations and uses arbitrary precision arithmetic and look-up tables. Using a database of 1010 images from 101 subjects and 5 enrollment templates per subject, we obtained a TAR=99.98% when FAR=0.1%
and EER=0.044 for verification, and 99.98% accuracy for identification. Comparisons with alternative approaches using the whole hand or individual parts of the hand, illustrate the superiority of the proposed approach both in terms of speed and accuracy. Also, qualitative comparisons with systems reported in the literature indicate that our system performs comparable or better.

Implementing the proposed system on a 3.19 GHz 64-bits machine with 2GB of RAM, the computation of Zernike moments up to order 20/30 for the fingers and the palm using double precision architecture in Visual C++ Studio 2005 is less than 0.01 second. The total preprocessing time for hand-arm segmentation and finger-palm segmentation using MATLAB 7.4.0 is less than 0.73 second on average. Time can be further improved without sacrificing accuracy significantly by disregarding the palm as discussed in Section 6.4. Therefore the proposed system can be employed for on-line applications.

Also we investigated the problem of gender classification from hand images using the component-based framework presented in this study. For classification, we compute the distance of a given hand from two different eigenspaces, one corresponding to the male class and the other corresponding to female class. Although the dataset used in our experiments is rather small and all the data was obtained in the same session, we believe that the results presented in this study are quite encouraging.

Finally we introduced a technique by decomposing the hand silhouette and fusing the confidence of the fingers in order to lead to global optimization of templates. This update technique has the potential to identify difficult intra-class variations. The motivation behind this technique is that the temporal changes that may occur in the fingers are uncorrelated in such a way that the confidence of each finger can be significantly different from the others.
9.2 Future work

For future work, first we plan to perform larger scale verification and identification experiments by increasing the size of our database. This would allow us to obtain more accurate error estimates. Moreover, we plan to perform additional tests to evaluate the robustness of our method when there is substantial passage time between the template and test images. Second, we plan to investigate the idea of combining multiple templates into a single, "super-template", in order to build more accurate models for each subject and reduce storage requirements. Third, we plan to investigate feature selection schemes in order to reduce the dimensionality of the feature vectors without sacrificing discrimination power. This would also reduce time requirements since we would need to compute a small number of Zernike moments only. Fourth, we plan to investigate the benefits of integrating hand-based gender classification with hand-based authentication/identification. Fifth, we plan to evaluate the proposed gender classification technique on populations of different ages. For example, an interesting problem would be to investigate whether hand shape can be used to distinguish the gender of children. Sixth, we plan to investigate the possibility of ethnicity and age classification from hand shape. Seventh, we plan to evaluate the proposed template update framework in multimodal biometric systems (i.e. fingerprint, palm and hand). Finally, we plan to perform additional comparisons with other methods in the literature using the same database.
Bibliography


