Evaluation of Candidate Fingerprint Features for Employment within Template-Free Biometric Cryptosystems

A Thesis Submitted to the University

Of Kent

For the Degree of Doctor of Philosophy

In Electronic Engineering

By Karl Harmer

August 2009

"It's been emotional"

Big Chris – Lock, Stock and Two Smoking Barrels

Dedication

In loving memory of my grandpa, Leslie Hunnisett, and close family friend, Joan Mitchell, who both sadly passed away during the course of my studies.

Acknowledgements

First and foremost I would like to thank my family for all the support they have given me during my Ph.D. I am eternally indebted to them and without them this thesis would not have been possible. I would also like to express my gratitude to Elizabeth Lyimo, for I would not have returned to academia to pursue my Ph.D without her.

Secondly, I would like to extend my gratitude to my supervisor Dr Gareth Howells for his assistance, guidance and encouragement throughout my time at the University of Kent. I would also like to wish Gareth all the best with the commercialisation of this project and with all his research endeavours. Likewise, I would like to express my thanks and best wishes to Professor Michael Fairhurst, Dr Farzin Deravi and Dr Weiguo Sheng for their input and support.

Finally, I would like to thank the Engineering and Physical Sciences Research Council for providing me with this opportunity and monetary support.

To all my friends at the Department of Electronics at the University of Kent, I wish you all the best with your studies and careers, and I especially want to thank all of you that have made my time in Kent highly enjoyable. I will miss the tea-breaks!

Abstract

With the increasing employment of biometric authentication worldwide, there have been significant concerns over the security and privacy of the biometric templates, which are stored and used for matching. This thesis investigates a relatively new area of biometric authentication, known as biometric encryption. Biometric encryption is an alternative form of authentication where an encryption key is either generated, or retrieved, instead of a binary result indicating verification. The key can subsequently be used to encrypt and decrypt data using existing encryption algorithms, such as triple-DES, AES and RSA. The purpose of this thesis is to investigate the feasibility of generating stable and strong encryption keys directly from biometric features, without the use of templates. To accomplish this, a proof-of-concept was created using the fingerprint biometric modality. Fingerprints currently possess a limited range of features, although they are highly variable and result in many user distributions overlapping. Therefore, if consistent, highly entropic keys can be generated using fingerprint features, other modalities, whose feature distributions are better behaved, should improve the performance of the generated key. By performing statistical tests, on the countless variants of existing and novel feature vectors, the best performing feature vectors were then selected for deployment in a template-free biometric encryption system. Although the results were not exceptional, an encryption key, with 21 effective bits, was reproduced correctly ~80% of the time with 55% unique keys in the resultant key-space. These results are encouraging, although significant improvements would be necessary in order to obtain the performance required of a practical system.

List of Publications

1. K. Harmer, G. Howells, W. Sheng, M. C. Fairhurst and F. Deravi, "Fingerprint Core Detection using Orthogonal Tracing", submitted to IEEE Transactions in Pattern Analysis and Machine Intelligence.
2. W. Sheng, G. Howells, M. C. Fairhurst, F. Deravi and K. Harmer, "Consensus Fingerprint Matching with Genetically Optimised Approach", Pattern Recognition, Vol. 42, No. 7, pp 1399 – 1407, July 2009.
3. K. Harmer, G. Howells, W. Sheng, M. C. Fairhurst and F. Deravi, "A Peak – Trough Detection Algorithm based on Momentum", 2008 International Congress on Image and Signal Processing (CISP 2008).
4. W. Sheng, K. Harmer, G. Howells, M. C. Fairhurst and F. Deravi, "A Genetic Algorithm for Fingerprint Matching based on an Integrated Measure", IEEE Congress on Evolutionary Computing (IEEE CEC 2008).
5. K. Harmer, G. Howells, W. Sheng, M. C. Fairhurst and F. Deravi, "Fuzzy Vault Fingerprint Smartcard Implementation using an Orientation Based Feature Vector", 2008 ECSIS Symposium on Bio-Inspired, Learning and Intelligent Systems for Security (BLISS 2008).
6. K. Harmer, G. Howells, W. Sheng, M. C. Fairhurst and F. Deravi, "A Novel Level-Independent Fingerprint Segmentation Algorithm based on Fourier Energy", 2007 International Conference on Image Processing, Computer Vision and Pattern Recognition (IPCV'07)
7. K. Harmer, G. Howells, W. Sheng, M. C. Fairhurst and F. Deravi, "A Computationally Efficient Fingerprint Segmentation Algorithm using Digital Closed Curves", 2007 International Conference on Image Processing, Computer Vision and Pattern Recognition (IPCV'07)
8. W. Sheng, K. Harmer, G. Howells, M. C. Fairhurst and F. Deravi, "Fingerprint Matching with an Evolutionary Approach", Second International Conference on Biometrics (ICB 2007).

Table of Contents

List of Figures

List of Tables

Chapter 1. Introduction

Facilitated by the widespread use of the internet, an ever-increasing number of governmental, commercial and personal users alike are benefiting from "having the world on their doorsteps". Access to an increasing cornucopia of knowledge and information, and massive interconnectivity, providing global communication on an unprecedented scale, are the most significant advantages of internet adoption. However, connecting to an immense mesh of anonymous systems and devices is also extremely hazardous to both the data residing within and the data transmitted to and from a connected device. It is foolish to assume that with the anonymity that the internet affords, that every entity connected to it is credible.

Connected devices and their associated data can be compromised, which facilitates data corruption (either by deletion or alteration) and control (e.g. botnets provide external control to the device). This, by extension, allows disreputable entities to masquerade as the compromised device or to extract private or confidential data, which potentially facilitates identity fraud or theft. Most internet users are aware of the risks attached and are able to defend their devices with virus guards, firewalls and hardened login names and passwords. However, lay-users may believe this is where the threat to the device ends, which is a common misconception. The security measures outlined attempt to protect the device from being compromised, although they offer minimal protection against the compromising of data when the device is compromised. Furthermore, these measures do not offer any protection to the communication channel, whereby the data can be compromised by sniffing (passive interception and reading of network traffic).

1.1. Research Motivation

Current research estimates that a computer connected to the internet is probed for vulnerabilities every 39 seconds [4] and an un-patched computer is compromised by malware (malicious software) within 20 minutes of being connected to the internet [5]. Coupled with the ingenuity of modern social engineering practices and unsuspecting users, an internet-connected device is constantly at risk. Only one successful attempt is enough to compromise personal and confidential data. Therefore, in order to minimize the fraudulent use of the compromised data and maintain its authenticity and integrity, it is highly recommended that confidential, if not all, data is encrypted.

Data encryption is a fundamental tool for the protection of electronically mediated data, which translates plain text data into seemingly random data (cipher text). Plain text is encrypted using an encryption key and, depending on the encryption methodology employed (asymmetric or symmetric), the same or another key is used to decrypt the data. However, the integrity of encryption is questionable when the security of the encryption keys cannot be absolutely guaranteed. In order to strengthen encryption, it is recommended that the encryption key be very

long to increase resistance to a brute force attack. As the key can be very long, it is very difficult for a human to recite such a long stream of bits, so the encryption key is often determined from a simpler, memorable passphrase. This passphrase weakens the security of the encryption key as it is generally short, uses alphanumeric characters and can be related to the user. As the user knows this passphrase it can be tricked or coerced from them and be simple enough to be guessed or cracked by brute force. Therefore, in order to maintain key-strength without the user being aware of the encryption key, or passphrase, key generation / release from biometric data has been proposed in recent years.

Biometric encryption is the process of extracting physiological and / or behavioural characteristics from an individual in order to generate encryption keys for the encryption and decryption of data. This is a logical extension to traditional biometric identification / authentication, which has been around for many decades. However, traditional security systems, which exploit biometric data, require the employment of templates that encapsulate the data extracted from the given biometric of each user. This is still true in many biometric encryption systems, whereby an encryption key is released / generated upon a successful match of a stored template and a query template. This presents a number of disadvantages. The main disadvantage is that an unauthorized user or system administrator can gain access to the template and obtain data required for circumventing the system. Another drawback to this method is the reluctance of users to submit biometric samples, knowing that they will be used to generate templates and then stored within a system. Biometrics represent a description of who a person is and many people have a certain degree of attachment to their biometric details and are therefore reluctant to relinquish these details to a government, or company, that they have little faith in and an unproven system. Therefore, the research presented in this thesis investigates and develops techniques for the direct generation of encryption keys from biometric features.

There are significant benefits for such a system. Without the need for templates, this approach addresses the aforementioned disadvantages of template-based systems, while ensuring the security system is as strong as the biometric and encryption algorithms employed. Therefore, in theory, the only mechanisms for decrypting the data is by submitting another biometric sample (abstract representation of the encryption key) or by breaking the cipher as a plain text representation of the key is never outputted or stored by the system. Anyway, in the unlikely event that the system was compromised there are not any templates or biometric data stored, so no other system protected by that biometric can be compromised by replaying data. The concept of direct key generation from biometric features has been around for many years, but it is a difficult problem due to the high intra-user variability and, as a result, little work in this area has been published. The existing literature employs user-specific helper data (data that has no connection to actual biometric information) to model the intra-user variability. Therefore, an investigation into and the development of a template-free, biometric encryption system is extremely topical and novel.

Although the purpose of this investigation is to develop a template-free biometric cryptosystem, the work presented predominantly comprises the investigation of fingerprint features and their suitability in such an application. This extensive investigation in to fingerprint features was the main work-package of this project and this is reflected in the layout of this thesis. As the analysis of fingerprint features was of great importance, the bulk of this thesis is dedicated to the extraction and analysis of these features. Due to the extent of the features investigation, an unconventional thesis layout was adopted to present the findings. The layout of this thesis follows the progression of the entire application, starting with fingerprint pre-processing, followed by feature extraction and ending in fingerprint-based biometric cryptosystems. Not surprisingly, this bottom-up approach to the layout presents the work in chronological order. An alternative layout could be based on the top-down approach as favoured by designers, where biometric cryptosystems would be presented first. This method would present the most novel contribution first, however the main emphasis of this project was the extraction and analysis of the features. Therefore, it was felt that fingerprint features should be presented before biometric cryptosystems. Besides, by structuring the thesis similar to that of the processes of an actual system, allows the findings to be presented in a logical order with a natural progression building up to the main contribution.

In addition to the bottom-up layout, it was also decided that each chapter presents its own literature review and "*mini-reviews*". This is because this thesis covers many diverse fields and, by each chapter having its own literature review and "*mini-reviews*", this thesis can be viewed as being compiled of multiple "*mini*" theses. By moving away from the single large literature review, decouples the chapters from the thesis so they can be extracted and read independently without needing the entire thesis. The structure of this thesis is outlined in greater detail later in this chapter.

1.2. Biometric Authentication Background

With the ever-increasing popularity of online e-commerce services and internet-based communications (such as data-sharing and teleconferencing), it becomes vital to protect the confidential data residing in the data packets. In order to ensure the integrity and the authenticity of the potentially confidential data, it is necessary to adopt some mechanism of authentication. Automated authentication systems have been researched for many decades and typically employ one or a combination of three authentication methodologies. These are token based (i.e. something an individual possesses, such as a key or fob), specific-knowledge based (i.e. something an individual knows, such as a password or personal identification number (PIN) and characteristic based (i.e. something that defines an individual, such as biometrics). However only within the last decade or so have biometric systems provided performance suitable for everyday use in secure applications.

Authentication of an individual is the act of determining whether an individual is who they claim to be. This is often referred to as one-to-one matching because the search-space is only one. In other words, query data is bound with a unique identifier (i.e. name) and if the unique identifier has been registered within a database, a direct comparison between the stored data and the query data is executed in order to determine whether they match. This is different from identification as identification is termed as one-to-many matching, whereby the identity is unknown and the query data is compared with every entry in a database to determine the most likely individual. Obviously, tokens and passwords/passphrases are not suitable for identification purposes. This is because tokens can be easily lost, stolen, surreptitiously copied or knowingly transferred to an unauthorised individual. Passphrases, PIN, etc can also be passed to an unauthorised individual, as an act of collusion, but they can also be forgotten or cracked / guessed depending on their strength. Even when these are implemented as a means of authentication they are considered weak approaches, which rely on trustworthy individuals for the systems to be effective.

Combinations of token and knowledge based authentication can also be implemented in an attempt to strengthen the authentication mechanism (e.g. the chip and PIN system in the United Kingdom, UK). However, through acts of collusion, negligence or ingenuity, these systems can still be compromised. Biometrics though cannot be easily passed to other individuals and they cannot be lost as they belong to the individual, ensuring that the individual is present at the time of authentication. Furthermore, the user does not know what data is extracted from the biometric and the representation of that data, which is used in the authenticating process and, therefore, cannot be obtained via collusion. However, biometrics are within the public domain and certain modalities of biometrics (e.g. fingerprints and face) can be surreptitiously acquired and used for future spoofing (i.e. posing as an authorized user to gain access) attempts. Additionally, biometric data can be stolen from computers and networks, although spoof prevention and biometric template security are currently hot research topics.

1.2.1. Choosing a Biometric

Biometric based authentication systems provide true authentication in the sense that the systems authenticate based on characteristics which describe an individual. Biometrics is the measurements of these characteristics and they can be used to identify an individual uniquely. There are many different biometric modalities (i.e. class of biometric, such as voice, fingerprint, gait, face, iris, etc) which can be broadly divided in to two categories, physiological and behavioural. The physiological category extracts features which relate to the physiological characteristics of a person. Example physiological biometrics includes fingerprint patterns, palm print patterns, iris patterns and retinal blood vessel patterns. A behavioural biometric arises from actions carried out

by an individual, such as signatures and keystrokes. It is quite common to combine biometric modalities into a multimodal (multi-biometric) system to strengthen security.

When choosing a biometric for use in a biometric-based authentication system, there are four highly desired properties that the biometric should possess. These are universality, uniqueness, permanence and collectability. In other words, the biometric should be obtainable from everybody, provide a high degree of discriminatory information, be invariant to aging and be practical to extract. However, different biometrics provide varying degrees of the desired properties. In addition to the properties desired of the biometric itself, the method of how the biometric is extracted also requires certain properties. Biometric extraction methods should achieve a high performance rate, be universally accepted by the users and be robust to circumvention. Table 1-1 presents a number of popular biometric modalities and the strengths of their respective properties as presented in [6]. It should be noted that the column for circumvention is inverted with respect to other columns (i.e. high circumvention is bad, whereas high in any other column is good). Furthermore, some of the ratings presented within Table 1-1 are contentious (e.g. low universality of signatures) and are dependent on the target application, its users and the subjectivity of the examiner.

Table 1-1: Biometric Modalities & Respective Property Strengths

	Universality	Uniqueness	Permanence	Collectable	Performance	Acceptability	Circumvention
Face	HIGH	LOW	MED	HIGH	LOW	HIGH	HIGH
Fingerprint	MED	HIGH	HIGH	MED	HIGH	MED	MED
Hand Geometry	MED	MED	MED	HIGH	MED	MED	MED
Iris	HIGH	HIGH	HIGH	MED	HIGH	LOW	LOW
Hand / Finger Vein	MED	MED	MED	MED	MED	MED	LOW
Signature	LOW	LOW	LOW	HIGH	LOW	HIGH	HIGH
Voice	MED	LOW	LOW	MED	LOW	HIGH	HIGH

The first four columns of Table 1-1 are inherent to the biometric modality and so are relatively stable. However the last three columns represent the level at which the current generation of extraction methods have achieved. Therefore, theoretically they should improve with time. Compiling a table akin to Table 1-1 provides a useful tool in deciding which modality would be best-suited to an automated authentication system. Depending on the requirements of the system, a single or combination of modalities can be employed in order to fulfil the system's specification. Example requirements of an authentication / identification system include speed, accuracy, discriminative power and intrusiveness.

The research presented in this thesis investigates direct key generation from biometric samples for implementation in a cryptosystem. Therefore, it is important to select a specific modality to examine. For the purpose of this research, the fingerprint modality was chosen for the following reasons. With reference to Table 1-1, it is clear that the fingerprint modality rates highly in all of the properties desired of a biometric modality. It is also the most mature biometric modality and has been highly researched [6]. Furthermore, the fingerprint modality also performs relatively well in large-scale applications due to the high degree of individuality and universality inherent to fingerprints. Therefore, logically, the extracted key-space should be very large without any key repetitions.

1.2.2. The Fingerprint Modality

Among all the automated biometric systems available, fingerprint based identification is the oldest and most familiar. A fingerprint is the resultant friction ridge pattern of an impression from the skin of a finger. Evidence of rudimentary fingerprint identification can be traced back to ancient Babylon and China, where impressions of a fingerprint were used as a signature for business transactions. Even evidence supporting awareness of fingerprints can be seen in prehistoric picture writing in Nova Scotia, where a hand is depicted with ridge patterns. However, it wasn't until much later (arguably the 14^{th} or 17^{th} century) when early fingerprint pioneers investigated the individuality of fingerprints. As a result of this discovery, fingerprints, as a means for personal identification, have been employed since the late 19^{th} century. Fingerprints superseded the inferior, but standard form of identification at the time, the Bertillon system at the beginning of the 20^{th} century when the Bertillon system's individuality was numerously questioned. This was a catalyst for England's Scotland Yard and the United States of America's (USA) adoption of fingerprints for criminal investigations.

As the number of fingerprint records increased, it became difficult to administer these records across states and law enforcement departments in the USA. Therefore the fingerprint records were consolidated in to one database. This heralded the first large-scale repository of fingerprints, which was commissioned in 1924 by the USA congress to assist the newly-formed identification department of the Federal Bureau of Investigation (FBI). Over time, the number of records processed by the FBI increased dramatically into hundreds of millions. With the advancement of technology, the difficult and arduous tasks of updating and maintaining a manual database gradually began to be replaced by an automated system, the Automated Fingerprint Identification System (AFIS), during the 1970s. Since the 1970s, the research and development of automated fingerprint methods have continued unabated, steadily improving recognition rates while reducing error rates.

Due to the individuality of fingerprints, a database could conceivably contain thousands, or even millions, of fingerprints. In fact, the FBI database already contains 55 million sets of fingerprints [7]. As identification is a one-to-many paradigm, to search the entire database is a very computationally expensive and takes time. In order to speed up the process, the search space is reduced by indexing the database. This can be achieved by ruling out fingerprints that do not match certain criteria. The most common of which is the pattern class. A fingerprint can be classified into one of three basic patterns, where each pattern can be broken down further into subgroups of the patterns. This hierarchical structure of fingerprint patterns is probably the most popular of fingerprint pattern classification and is known as the Henry Classification System (named after its creator Sir Edward Richard Henry) [6].

In Table 1-2, each fingerprint pattern class (approximate percentage of population is given in brackets) is broken down into its component subclasses and Figure 1-1 presents fingerprints that demonstrate the most popular fingerprint types. It should be noted that the subtypes of the loop pattern are defined as either radial or ulnar depending on the side of the hand the tail points. However in practice it is not always known which hand the fingerprint belongs to so the loop type is generally referred to as either a right-loop or a left-loop depending on the side of the image the tail resides. It is clear from the approximate breakdown of fingerprint types in Table 1-2 that fingerprints which belong to the loop category represent a considerable percentage of the population. Therefore, different methods exist, which attempt to reduce the search space further, based on alternative criteria. It is common for large databases to possess certain fields which, when interrogated, reduce the search space. This is commonly called indexing.

Table 1-2: Fingerprint Pattern Types

Arch Type (5%)	Loop Type (65%)	Whorl Type (30%)
Plain Tented	Ulnar Radial	Plain Accidental Double Loop Peacocks Eye Composite Central Pocket Loop

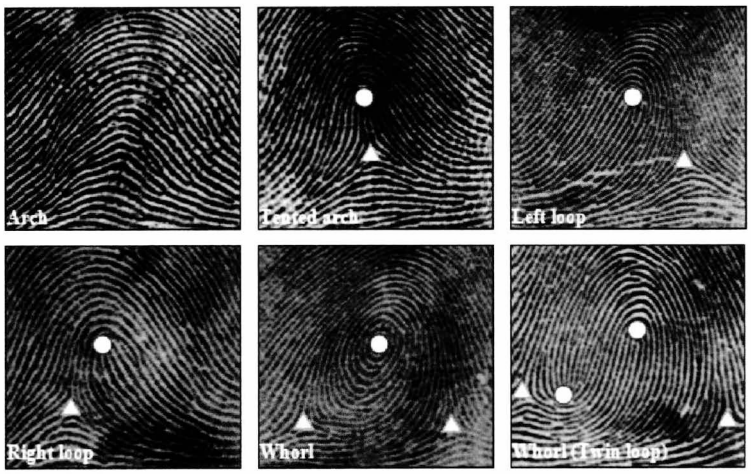


Figure 1-1: Example of Fingerprint Classes

As fingerprint technology advanced, fingerprint authentication became viable. Unlike identification where a subset of possible candidates is produced, authentication requires a high degree of certainty that the individual is who they claim to be. For fingerprint identification, a query template is compared with every entry within a database and each entry is ranked based on a matching score. The database administrator can request that the search return the best *n* matches based on the score, although empirical thresholds are often employed to ascertain reject matches that do not exhibit significant similarities. Therefore, identification is not an exact science and the performance of an identification system can be assessed by the regularity of the correct match being returned within the best *n* matches rather than the correct match always being the best match. Only when matching algorithms improved significantly so that they provided a high degree of certainty that a query and a registered fingerprint are identical could fingerprints conceivably be implemented for use in a high security authentication system.

For most parts, fingerprint identification and authentication systems are identical. The only differences are that for an authentication system the identity of the individual is known and that it results in a yes/no answer regarding whether the individual is who they claim to be. Both systems employ an acquisition stage, a pre-processing / enhancement stage, a feature extraction and representation stage and a comparison stage, similar to any other pattern recognition system. As with any pattern recognition system, the most important aspect is the identification, extraction and representation of features. The performance of a pattern recognition system is directly related to the reproducibility of features as they reduce the dimensionality of the pattern into a workable representation.

Fingerprint features can be categorised in a number of ways. One approach to feature categorisation is to view each feature as belonging to one of three levels. Level one contains the features pertaining to the ridge structure and pattern, while level two features comprise solely of the Galton characteristics (i.e. minutiae) and level three features encompasses all dimensional

attributes of the ridges (i.e. pores, ridge distances, shape descriptors, scars / creases, etc) [8]. As feature extractors in automated fingerprint systems developed, the definition of level three features has altered to accommodate new types of features. The general definition of level three features is now "any feature that does not belong in levels one and two" [9]. An alternative approach to categorising fingerprint features is at the scale the features are extracted from, which is either local or global. Minutiae features are an example of a local feature whereas any image-based feature is a global feature [10]. Unfortunately, features of the fingerprint modality are highly variable, which presents severe complications that will need to be addressed when applied to a template-free implementation. Fingerprint features will be explored in more detail later in the thesis.

1.2.3. Typical Biometric System Framework

Biometric systems have been integrated worldwide providing security solutions for law enforcements, internet banking, access control (digitally and physically) and ATMs. The most popular automated systems employed are the fingerprint and iris modality. However, all biometric systems are essentially a signal (an image is viewed as a 2D signal) processing and pattern recognition task. Therefore, the basic structure of the majority of systems employed is very similar. The fundamental processes of a biometric system are highlighted in Figure 1-2.

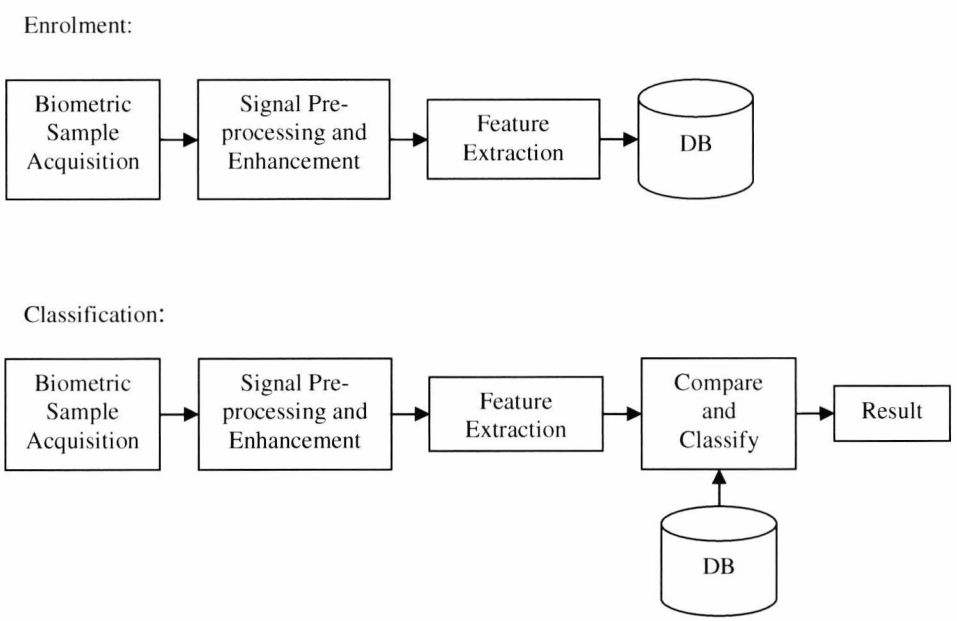


Figure 1-2: Processes Involved in User Enrolment and Subsequent Classification

The flowcharts presented in Figure 1-2 represent a rudimentary biometric system. The features extracted are commonly represented in the form of a template and the template is then stored in the database for future comparisons and classification. It is clear from the Figure 1-2 that certain

aspects of the framework are common with many automated recognition systems. Signal pre-processing, enhancement, feature extraction and classification are the fundamental parts of most implementations. Therefore, a biometric system will have to overcome the same challenges, which are inherent in all pattern recognition systems.

1.2.4. Challenges

Automated recognition systems require the acquisition of a sample from an analogue (real) world. In order to acquire a sample, the sample must be presented to a sensor, which will subsequently convert the analogue version of the sample to a discrete, digitized representation for further computer processing. When an analogue signal is converted to a digital signal, there is an inevitable loss of information through quantisation. Furthermore, the environment in which the sample was acquired can seriously affect the quality of the signal acquired. For example, an image-based system may be susceptible to illumination differences, orientation variation and partial obscurity (e.g. dirt on lens). Therefore, it is highly beneficial that the signal is enhanced and pre-processed.

Both signal enhancement and pre-processing are essential to minimize variation and normalize the signal to a common base on which subsequent processing may build upon. Enhancement is mainly employed to improve the quality of the image and primarily consists of noise filtering and the accentuating of particular features. Whereas pre-processing methods attempt to provide a normalized signal, in the sense that certain features of every signal passed to the subsequent feature extractor stage are consistent. Pre-processing is also implemented to highlight regions of the signal which either do not belong to the actual biometric (segmentation) or that the signal is significantly corrupted that features extracted from these regions are of poor quality (quality estimation).

For all pattern recognition tasks, it is important that intra-sample variations are minimized throughout the process in order to attain the best results. If the intra-sample variance is not minimized at every level, then the variance will accumulate at every stage. Therefore it is important to reduce the variance right from the outset because the digital representations of biometric signals are inherently highly variable (i.e. no two representations of the same biometric are identical). By reducing the variance at the initial stage, the variation of the extracted feature vectors should also be reduced. This should pave the way for simpler matching / classification algorithms because the necessity for the variance to modelled when matching should also be reduced.

Feature selection is another important factor which should be taken into consideration when implementing an automated pattern recognition system. Extracting a set of features that possess certain characteristics can also improve performance. In order to obtain the best results, a feature, or combined feature set, must be highly discriminative (i.e. low variation between signals obtained

from same origin, while high variation between signals from different origin), consistently reproducible and invariant to affine transformations and scale. Unfortunately, identifying features, or combined feature sets, which provide these characteristics is not a straightforward task. In some cases, a feature that is highly discriminatory is easily identifiable, but due to the high variability of biometric data, it may not be possible to extract the feature consistently. Fingerprint minutiae are an example of such a feature. Minutiae have long been identified as being highly discriminatory, but the representation of this feature is liable to change between subsequent acquisitions of the same fingerprint. Therefore, identifying a suitable representation of the feature is just as important as identifying the feature itself. Optimum feature selection is often preferred as it simplifies classification considerably, whereas sub-optimum feature selection can increase classifier complexity because it will need to model the feature variances.

1.2.5. Security Issues

Many pattern recognition systems are often implemented in benign applications, which replace repetitive, mundane or difficult tasks. However, biometric-based systems are often implemented in security applications. Even in non-security based applications, the use of biometrics instigates privacy concerns, which need to be addressed if biometrics is to be universally accepted. There is a high degree of scepticism and worry associated with the use of biometrics, especially with regards to storage and the eventual use of biometric data. Biometric data are measurements which define an individual and the thought of another person having unauthorised access to and potentially stealing the information heightens the fear of identity theft and misuse. To alleviate fears and boost public confidence, biometric-based systems should provide a certain level of transparency and provide the highest degree of privacy and security. This is a very difficult problem, which needs to be addressed.

Unfortunately, ensuring that an authentication system is completely secure is not a trivial task because biometric-based systems are fraught with security weaknesses [11]. Every stage of a biometric authentication system presents opportunities for circumvention. This is not aided by the fact that biometrics and the algorithms of the systems which employ them are not always secret. Biometric data exists in the public domain whereby an individual's biometric information can be surreptitiously obtained (i.e. taking photos of a face, recording voices or lifting latent fingerprints). Therefore, it is possible to replicate them with the intent of spoofing a biometric system and claiming someone else's identity. Furthermore, many commercial products that implement a form of biometric authentication / identification patent their methods in order to protect their intellectual property. Also, the public requests a certain level of transparency so they can allay their fears regarding the use and storage of their biometric data. However, the more transparent the system, the more information is available in the public domain. This unfortunately presents another weakness because patents and biometric data usage can be publicly accessible, which allows

unscrupulous characters to increase their understanding of a product in order to mount an attack on the system. Figure 1-3 outlines possible attack points within a traditional biometric-based authentication system [11].

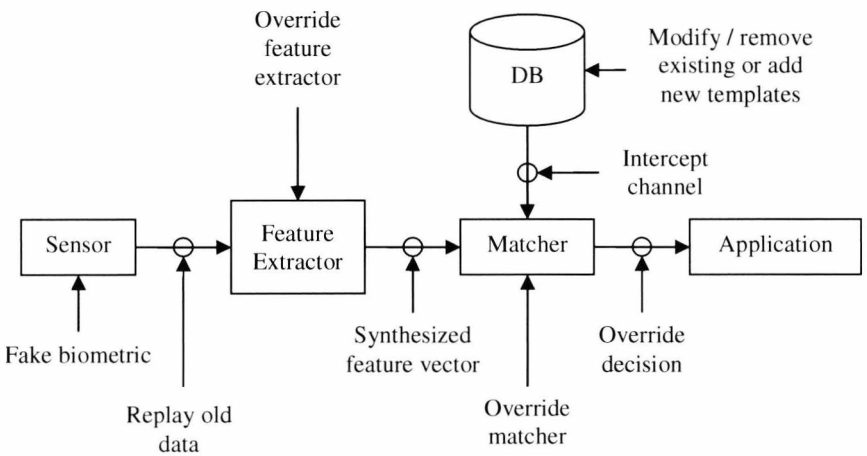


Figure 1-3: Attack Points in a Biometric-Based Security System

Probably the most common attack on a biometric system is at the sensor level (i.e. presenting fake biometrics or replaying old data) because other people's biometric data can easily be obtained. The sensor level is also the most accessible part of the system as it provides the interface at which the public presents their biometric samples. If the accessibility is high, then it is logical to assume that more people will attempt to spoof the system at that level. Hence the urgency in the research for detecting / preventing spoof attempts. Furthermore, the sensor of a biometric system is commonly provided by a third-party. This means that documentation on how to implement the sensor is probably in the public domain. By educating themselves in the communication protocol, it is possible for someone to extract the data while in transit between the sensor and the biometric system's innards. The data can then be replayed at a later date for unauthorized access. However, circumventing the system using this method would be very difficult to achieve if the system is supervised. This is because it would require access to the communication channel. If the sensor is hard-wired (physically connected to the biometric system), then it would be very difficult to surreptitiously sniff / modify the connection without being detected.

Although spoofing at the sensor level is probably the easiest and most common attack, its malevolence is minimal compared to the havoc, which could potentially occur when attacked at other points located within the biometric system's framework. An attack at the sensor level will affect the privacy and security of a single, individual user, which is bad enough. Furthermore, overriding the decisions, feature extractors and matchers will also allow unauthorized access under the guise of an authorized user, but an attack at these points could also facilitate a denial of service (DoS) attack. A DoS attack renders the system unavailable to its users. This would be infuriating

for both users and system administrators, but the privacy and security of the entire user-base will not be compromised. However, circumvention at the database level could be catastrophic as it can compromise the privacy and security of all users.

It is therefore prudent to protect the database. The obvious method to protect the database is to encrypt its contents. By encrypting the templates, an attacker will only uncover seemingly random patterns preventing access to the data and to whom it belongs. However, the data can be obtained if the attacker has administrator rights (i.e. an attack from within) or an administrator is conspiring with an external individual (collusion) either by force or willingly. To prevent actual biometric data from being compromised, the templates are often either unrelated directly to the biometric data or a one-way transformation of the feature data. As the transformation is one-way, it prevents the biometric data being obtained from the template if the template is ever compromised.

While template manipulation and encryption protects the identity of an individual and their biometric data, it does not prevent an attacker from gaining unauthorized access or causing harm to the system. By deleting or modifying the database or certain records, the attacker could easily launch a DoS attack Furthermore, if the attacker has sufficient knowledge of the system, then it is plausible that they could overwrite an existing template with their template, allowing subsequent unauthorized access. The communication channel also presents a significant weakness, especially if the database is remote.

There are four locations in a biometric system where template storage can occur. A template can either be stored remotely in a centralized database or locally on individual workstations, on individual biometric sensors or on individual tokens [12]. When the templates are stored locally on sensors or on tokens, the user will need to present these items whenever they need to be authenticated. Therefore they can be lost or stolen easily, where subsequent tampering can take place. Conversely, storing the templates locally on individual workstations reduces the possibility of loss, but the user cannot authenticate at multiple locations. However a centralized database allows authentication at multiple locations without the need of possessing an additional item. Unfortunately, centralized storage creates additional security concerns.

Centralized storage requires a link to a remote database and it is therefore possible to snoop on the communication channel and intercept outgoing and incoming transmissions. Even though the data transferred between the system and the database may be encrypted, the transmission could be altered or the attacker could hijack the channel in order to copy and analyze the packets for the facilitation of a replay attack. Protecting the database at all levels increases the level of security, but the possibility of attacking the database still exists. One school of thought for the improvement of security is to remove the possibility of weaknesses. If the database is removed from the equation, then the weaknesses associated with it will also be eliminated. This is a difficult task because in

order to authenticate or identify an individual a comparison stage is required. However, this can be ascertainable by extending the concept of biometric encryption.

1.2.6. Biometric Encryption

Biometric encryption is the process of encrypting / decrypting data using one or more biometrics of an individual. This is commonly achieved by obtaining a key from the biometric data. In essence, this is similar to biometric authentication. In biometric authentication an individual states who they are and provides biometric samples as evidence of this person. The biometric samples are compared to that of an enrolled template and a yes / no decision is determined. This essentially converts the biometric data into a 1-bit representation. Biometric encryption in many cases follows the same steps but produces an n-bit representation. This n-bit representation of the biometric data is, in theory, a unique value, which can be viewed as a password / PIN. Therefore, it can be used similarly to that of knowledge-specific authentication systems. However, in this application the user is taken out of the equation and does not know the unique value that is defined by their biometric features. By employing encryption methodologies, biometric encryption also removes the need for the biometric system to know the password / PIN. Authentication is implied by the successful encryption / decryption of data.

A common approach to biometric encryption is to bind the key with the template [13], where upon authentication a key is released. Another popular approach is to securely store an unknown key within a metaphorical vault [14-20], which the biometric data unlocks. These approaches will be explored in further detail later. However in these approaches, storage is still required, which introduces the same weaknesses outlined earlier. Therefore, the work presented in this thesis investigates the direct generation of individual keys from biometric samples. By generating keys directly, this eliminates the need for template creation, storage and matching, thus reducing the number of attack points. A further investigation into biometric encryption is provided in "Biometric Encryption", chapter 7.

1.3. The Scope and Research Objectives

The research undertaken aims to investigate the feasibility of employing fingerprint features within a template-free biometric cryptosystem. The methodology developed seeks to fulfill two key objectives through fingerprint feature extraction, feature analysis and direct key generation. The main research objectives are:

1. To assess the suitability of individual fingerprint features based on their respective intra- and inter-sample variances.
2. To assess the stability of encryption keys generated from the optimal feature set determined from the feature analysis.

This research encompasses two key fields, which are fingerprint feature extraction and biometric encryption. Due to the variable nature of fingerprints, feature extraction was expanded to include fingerprint pre-processing as this can reduce variance attributed to affine transformations. In order to amalgamate these areas to produce a working system, a restricted scope of interest was applied. Certain aspects, which apply to biometric encryption that do not lie within scope of this research, include system attacking and cancellable keys / biometrics. Furthermore, this research does not investigate outliers and their effects. These out-of-scope topics are large and diverse themselves and would require extensive investigations if a direct key generation method is to be commercialized.

1.4. Research Challenges

There are many challenges that must be addressed. However, the majority of these difficulties can be categorized under one umbrella term, variance. It is important for every pattern recognition system to minimize the intra-sample variation of each feature to obtain the best performance. This is especially true with fingerprints as this biometric modality is notorious for its variability. Unfortunately, intra-sample variation can accumulate down the chain (i.e. a knock-on effect) of processes within the system. Any variance exhibited at the acquisition stage of a fingerprint system would be a fraction of the total variance exhibited in the resultant feature vector. If the feature vector is used to derive an encryption key, this in turn will increase the variability of the encryption key for that user. Therefore, it is important to minimize intra-sample variation at each stage of the automated system.

1.4.1. Intra-user Variability

The difference between two images of the same fingerprint can be attributed to variance in any combination of greyscale intensity (through inconsistent pressure, dirt, sweat or dryness), orientation, translation, captured region (i.e. partial overlap) or scale (i.e. if two different fingerprint capture devices are used). In most systems, some of these variances are taken into account by the matching algorithm. However, in a direct key generation application, the matching stage is omitted. As a result of this, the intra-sample and inter-sample variations have to be modelled at the feature extraction stage. This places a great deal of emphasis on feature selection. The ideal feature vector will have a very low intra-sample variation and a very high inter-sample variation. This characteristic is an indication of the reproducibility of the feature vector. Therefore, it is necessary to investigate the available fingerprint features, determine how reproducible each feature is and, if need be, adapt the features so they are more resilient to certain variations.

Some of these variations can be suppressed by implementing a pre-processing stage, which can crop, rotate, translate and enhance the fingerprints. It is common practice in image-based fingerprint systems to obtain a reference point and define a region of interest (ROI) centred about it [21]. However, consistent placement of the ROI is entirely dependent on consistent detection of the reference point. If the reference point is detected in different relative locations, the ROI will not capture the same region. Unfortunately, in certain fingerprint types (i.e. arches) reference points are not easily identifiable. Even when a consistent reference point can be located, a reliable orientation estimate in other fingerprint types (i.e. whorl) cannot be extracted preventing consistent alignment. In general, most image-based systems employ a matching stage, which handles alignment so whorls do not present any problems at the feature extractor stage. This is not the case with arch-type fingerprints because the variation in the relative location of the reference point tends to result in lower performance rates for this fingerprint type.

1.4.2. Direct Key Generation

The other considerable challenge that will need to be addressed is the direct generation of the encryption keys, without storing templates or helper data, which are consistently reproducible (key stability) with a large number of effective bits (key entropy). Even when assuming that consistent feature extraction is possible (i.e. the same feature vectors results for each image of the same fingerprint), it is inevitable that many feature vectors of different fingerprints will overlap in the feature space, especially when the feature space is small. Therefore, the feature vectors that overlap could potentially generate the same binary patterns. Furthermore, if multiple feature vectors of a fingerprint overlap with a consistent group of fingerprints, then the number of effective bits could reduce dramatically as the group will have similar binary patterns. This will effectively reduce the uniqueness of the key. This could be a possible scenario for fingerprints which are of the same type.

These problems can arise even when the feature vectors are consistently extracted for each fingerprint. Now, coupled with the fact that these feature vectors in practice will almost certainly have differing degrees of variation, gives rise to further problems because the key is no longer stable. Therefore, direct key generation becomes a problem of stability and uniqueness.

1.5. Thesis Outline

The research undertaken encompasses aspects from a variety of fields. In order to minimize cross-referencing between chapters, the content of this thesis is organized thusly.

Chapter 2 presents an in-depth literature review and discussion of current fingerprint pre-processing methods and proposes two novel fingerprint segmentation algorithms. Furthermore, a

novel approach to peak-trough detection is proposed as peak-trough detection occurs frequently in fingerprint pre-processing and feature extraction methods.

Chapter 3 outlines the testing framework to which the feature catalogue experiments adhere to. While chapters 4, 5 and 6 present collections of fingerprint features belonging to different levels of features (chapter 4: level-1 features, chapter 5: level-2 features and chapter 6: level-3 features). These three chapters start with an in-depth literature review of current fingerprint features pertaining to the respective level. Following which, an outline of the features to be investigated and the extraction methods employed are presented. The results of the experiments, outlined by the testing framework, are then presented and conclusions are drawn for each respective feature level. Although many of the features presented within these chapters are pre-existing, the contribution of this work lay in the testing itself where a multitude of scenarios are investigated by employing statistical tests directly on the features rather than appraising overall system performance..

Chapter 7 presents an in-depth literature review and discussion of current biometric encryption techniques and other relevant works related to direct key generation. In this chapter a novel fingerprint-based fuzzy vault is proposed and an investigation into direct key generation is carried out on the resultant feature vectors of the feature catalogue. Conclusions are then drawn to conclude this chapter.

Chapter 8 is the final chapter, which draws general conclusions concerning the entire research carried out and provides concluding remarks and recommendations for future work.

Chapter 2. Fingerprint Pre-processing

Automated biometric systems, as with other image / signal processing application, require the acquisition of data from a variety of sensors sited within variable environments. Therefore, images from the same sample but acquired from different sensors, or at different times from the same sensor, will almost certainly differ. In fact, there are many factors that contribute to this intra-sample variation. The majority of these factors are outlined in [22]. It is the objective of the pre-processing stage to minimize the variation between signals from the same sample.

The variations exhibited between signals of the same sample are often attributed to non-linear distortions, partial capture, noise and transformations. Distortion, partial capture and transformations (in either time or spatial domain) can also be considered as noise due to their random nature. However, in this sense, noise will be defined as a random addition to the signal (i.e. residual ridges, dirt, sweat, etc), which can cause signal deterioration. Transformations and region omissions, due to partial capture, are independent of the quality of the signal. Non-linear distortions, however, are a random addition. Unfortunately, handling distortions at the pre-processing stage is extremely difficult. This is because the system does not have any previous knowledge of the query signal at the pre-processing stage, so it would not detect any distortion. Only when comparing images, can an estimation of distortion be made. Many automated fingerprint systems handle distortion at the matching stage [6]. Although, Ratha et al [23], measured the forces and torques applied to the scanner and reject the capture if excessive force is applied. Therefore, as non-linear distortion handling is shifted to the matching stage, and it is not possible to enhance something that is not there (as in partial capture), pre-processing can be viewed as a two-stage process. In order to minimize intra-sample variations, the signal must be located and transformed to a common space and then cleaned in order to obtain the best possible, consistent representation of the biometric sample.

In order to clean a 2D signal (image) and suppress the influence of noise, image enhancement algorithms can be employed. Furthermore, the signal can be located by applying segmentation, which is an effort to handle translational variance. However, rotational variation is best handled at the matching stage, where comparisons can be executed. This is not an ideal scenario because a matching stage is not always implemented, such as in direct key generation for biometric encryption. Hence, either physically restricting the possible rotation of the finger at the point of capture or estimating rotation at the pre-processing or feature extraction stage would be preferable. In this chapter, previous work for both fingerprint enhancement and segmentation is outlined and assessed. Based on these assessments, two new fingerprint segmentation methods are presented. A brief discussion regarding a suitable method of estimating fingerprint orientation will also be presented.

2.1. Previous Work

2.1.1. Fingerprint Enhancement Methods

Fingerprint enhancement methods are employed to reduce the influence of noise and accentuate the ridges. The degree of image enhancement varies depending on the features the system extracts. Minutiae-based systems employ the highest level of enhancement because minutiae are features pertaining to individual ridges. Therefore the enhancement method must result in high quality ridge structure. Image-based methods usually do not require the same degree of enhancement because the emphasis does not lie on extracting individual ridges. Normalisation [24] or histogram equalization [25] of the region of interest (ROI) is suitable for image-based enhancement as these methods preserve all the information residing in the image for use by the feature extractor, whereas filtering methods, used excessively with non-optimal parameters, can smooth out important information and drastically effect the image. An enhancement phase of an automated fingerprint system can be employed at two stages in the system. An enhancement algorithm can be applied either directly to the raw image or to a ridge-segmented (binarised) image.

2.1.1.1. Raw Image Enhancement

In many systems, an enhancement method is applied to the entire, raw image. At this point, the entire image can either be enhanced using an algorithm which treats all regions as the same or an algorithm that is adaptive to different regions of the fingerprint. A non-adaptive algorithm will not take into consideration regions of different structure or of poor quality. As the properties of regions differ, the quality of enhancement will also vary because "one size does not fit all". Image normalization and histogram equalization are examples of non-adaptive enhancement, although these can also be applied to local neighbourhoods. Unfortunately, these methods only improve the distinctiveness of ridges. However, in smudged or wet regions, just improving ridge clarity is not enough as ridges will be broken or merged together. Therefore, an adaptive algorithm allows the regions' characteristics to slightly alter the method to improve enhancement.

Essentially any adaptive method can be seen as using the underlying quality estimation, except that the features extracted are not amalgamated to create an overall estimation of quality. An adaptive method employs exploits the contextual information of the local neighbourhood and extracts features which are then used to tweak and tune the enhancement technique. Many of these features are similar to that which is used to estimate the quality of the region, but the regions are not classified into differing levels of quality to produce a quality map. Therefore, adaptive algorithms encompass methods which utilise filters, whose parameters are chosen based on the properties of the local neighbourhood, and hybrid methods. Hybrid methods are basically enhancement technique selection algorithms, whereby the appropriate enhancement scheme is chosen based on the characteristics of the local neighbourhood.

2.1.1.1.1. Hybrid Methods

An example of an enhancement scheme selection method is that presented by Yun et al [26]. From a prior, segmented image, the quality of the foreground is deduced by implementing Ward's clustering algorithm to classify the quality of the region based on a feature vector containing the mean, variance, block directional difference, orientation change and ridge / valley thickness. Based upon the heuristically derived quality estimation, dry and smudged regions were determined and appropriate combinations of morphological and logical operators were employed to enhance the areas accordingly. However Wang et al [27] also adopted a selection method but replaced the morphological and logical operators with quantitative filters. The parameters of these filters were automatically adjusted in accordance of the quality factors of that region.

2.1.1.1.2. Filtering Methods

Employing filters that adapt based on local characteristics is a very popular approach to fingerprint enhancement. Probably the most popular filter employed in nearly all aspects of a fingerprint system is the Gabor filter. The Gabor filter, when projected in the spatial domain, is a Gaussian envelope modulated by a complex sinusoidal carrier [24] and, due to the shape of the waveform, is implemented frequently. A fingerprint is seen as an oriented, periodic texture pattern [28] and the 2D Gabor filter has an oriented, periodic shape. Therefore, the parameters of the Gabor filter can be tuned to the underlying sinusoidal structure of the fingerprint region. When the Gabor filter and the fingerprint region are aligned correctly the ridges are accentuated, the noise is suppressed and the ridge smoothed. Horton et al [29] compiled a report focussing on the costs and benefits of using Gabor filters in fingerprint systems.

Many enhancement techniques available employ Gabor filters in some way or form. Originally, Gabor filter parameters were constants derived empirically and a bank of Gabor filters tuned at different orientations was compiled [2]. The entire bank of filters would be used to create multiple filtered images. The enhanced image was then determined by partitioning the filtered images and determining which partition of each filtered image produced the best result. However, it is now common to determine the local orientation and estimate the frequency (constant average or local estimated frequency) to tune the filter for each region separately. Many proposed algorithms are conceptually similar, although different methods are used to estimate the filters parameters [2, 30-34].

The main drawback of using Gabor filters is execution time. Convolution is a slow process when the filter radius is quite large. Therefore, other variants of the Gabor filter can be implemented. The easiest method is to set any Gabor coefficients below a threshold (epsilon) to zero. Other methods

for reducing computation time is to separate the Gabor filter [31], implement a circular [34] or half Gabor filter [32].

There are many alternatives to employing the Gabor filter. Other filters, which are appropriate for highly oriented images have also been proposed. Amongst these filters is the directional filter [35-37]. The directional filter decomposes the image into wedge-shaped directional pass-band regions and reconstructs them as an enhanced image. Another method of directional-based filtering is presented by Wu et al [38], whereby a combination of eight directional median filters and an anisotropic filter were implemented. The anisotropic filter is designed to remove Gaussian noise and the directional median filters attempt to remove the impulse noise and repair broken ridges.

Anisotropic filters have also been numerously proposed. Greenberg et al [39] presented a structure adaptive anisotropic filter, which uses the local orientation and an anisotropic measure to control the shape of the filter kernel. The method proposed by Greenberg et al [40] is very similar to the Gabor filter methods except the Gabor filter is replaced with an anisotropic filter. Furthermore, Cheng et al [41] proposed a novel anisotropic non-linear diffusion filter which steered itself by local characteristics rather than just the local orientation.

The majority of the filtering methods outlined are implemented by convolving the filter kernel with the image. As mentioned previously, convolution can be a slow process. This process can be sped up in some cases by transforming the image and the filter kernel into the Fourier domain, multiplying them together and transforming back into the spatial domain. The Gabor filter can be applied in this way. It is important to note that this is only more efficient when the kernel size is large. If the kernel is very small, convolution is probably more efficient.

Some filters however are designed especially for implementation in the Fourier domain. Sherlock et al [42] presented a directional Fourier filter and Kamei & Mizoguchi [43] expanded on this and designed two distinct Fourier filters corresponding to ridge frequency and direction. Features from the resultant filtered images were then extracted to determine an energy function. The original image was enhanced by minimizing the energy function. Sharat et al [1] implemented [42], although the filter parameters were determined by the features simultaneously extracted from the Short Time Fourier Domain (STFT). Zhu and Zhang [44] further adapted the technique proposed in [43] by reducing the dependency on the ridge frequency and orientation estimation by employing a top-down iterative filtering method. It is important to note though, that Ko [45] analysed many of the discussed filtering methods (both spectral and spatial) and determined that by applying selective filters to the Fourier spectra, concluded that by employing different filters enhanced regions of the image, which would not have been possible if only one type of filter was applied throughout.

2.1.1.1.3. Wavelet Methods

Just as the Fourier transform provides an alternative representation of the signal, which overcomes the shortcomings of spatial domain filtering, wavelets provide a solution to some of the Fourier domain drawbacks. Fourier analysis decomposes a signal into its component sine waves of varying amplitudes and frequencies. Hence the Fourier spectrum represents the frequencies which are evident in the image and their respective strengths. However, it does not indicate where in the signal these frequencies occur because the component sine waves have an infinite time window so the components are not localized in time. Wavelet theory provides a mechanism that allows localized decomposition because a wavelet is a fast deteriorating signal and therefore short in duration. Wavelet analysis permits localization of the signal in both time and frequency unlike Fourier analysis which is localized in frequency only. This is obtained by decomposing the signal into shifted and scaled versions of the mother wavelet by means of a filter-bank [46]. It is therefore feasible to apply filtering methods to the resultant decomposition [47, 48], similar to that of applying a filter in the Fourier domain, or obtain features from the wavelet domain for filter / algorithm parameter estimation [49].

2.1.1.1.4. Scale-Space Methods

One of the consequences of wavelet analysis is a representation of the original signal at different scales. It has been postulated that additional, different information can be obtained when viewing a signal at different scales [50]. However there is an alternative method of analysing an image at different scales. Scale-space theory is a relatively new approach to fingerprint enhancement and is essentially a formal framework for processing images at different scales. This is conducted in such a way that fine-scale features can be successively suppressed and associated with a scaling parameter at each level in the scale-space. There are many different approaches to scale-space theory, which are examined by Larsson [50]. In Larsson's thesis [50], an in-depth introduction to scale-space theory is provided followed by a non-linear anisotropic diffusion scale-space approach, which enhances the fingerprint without relying on extracted features. However, Almansa and Lindeberg [51] also proposed an anisotropic scale-space, shape-adapted smoothing technique based upon second moment descriptors to analyse and control the diffusion process. Furthermore, Cheng et al [52] proposed a dyadic scale-space approach where the image is decomposed based on the dyadic scale and the details are used to smooth and threshold the image.

2.1.1.2. Binary Image Enhancement

The aforementioned enhancement methods are applied to the raw greyscale image. In many minutiae-based systems, an enhanced, binarized image is required for the minutiae extraction stage. Therefore a subsequent ridge segmentation, which is commonly an adaptive threshold algorithm, is often applied to extract the ridges. This results in a binary image where black pixels lie on ridges

and the background is depicted by white pixels. Following binarization, a thinning algorithm [53], which is a morphological operation that reduces the width of the lines (ridges) to one pixel, is applied. Depending on the quality of the image prior to ridge segmentation, the result of performing a ridge segmentation and thinning algorithm often corrupts ridge structure (i.e. false connection of ridges, introduction of short ridges, etc), in areas of poor quality as a result of pixel misclassification. Therefore a further enhancement phase can be applied to the resultant thinned, binarized image in order to improve the ridge structure. The majority of methods, which are applied to the binarized image, employ either heuristics or morphological operators.

Greenberg et al [54] simply rejected ridges under a certain length and Luo & Tian [55] proposed a novel rule-based algorithm derived from distances and orientations of ridges. However, Ratha et al [56] implemented a morphological opening, where the structuring element (SE) was an oriented, small rectangle parallel to the local ridge orientation. Ikeda et al [57] also employed morphological operators, which are based on an isotropic constructing element and Liang & Asano [58] employed generalised morphological operators (GMO) derived from the distance transform and integral image. However, the performance of a morphological operator-based algorithm relies heavily on the selection of a suitable SE [58]. An alternative approach to enhancing the binarized image is to detect the minutiae without enhancement and use heuristics to reject minutiae (i.e. post-processing). Employing post-processing is a common method, although it is possible that many true minutiae maybe rejected because they lie within a cluster and the post-processing method does not know which of the minutiae within the cluster are true or false. Enhancing, or de-noising, the binary image should reduce the amount of rejected, true minutiae by removing the false minutiae first.

2.1.1.3. Summary

Minutiae-based fingerprint systems are one of the most popular approaches for both identification and authentication. This is because minutiae are intuitive, simple to extract and are one of the most discriminatory fingerprint feature identified at present. As minutiae are highly discriminatory, it is recommended that automatic fingerprint identification / verification system employ minutiae in some way or form to take advantage of this characteristic. Although minutiae are simple to extract, the reliability of each point is directly related to the quality of enhancement and ridge segmentation. Minutiae are often missed, switched and added depending on the outcome of the enhancement stage. Therefore it is important to employ a high quality, reliable enhancement algorithm.

Many of the enhancement methods discussed claim great performance, but in publications, it is not uncommon for difficult images to be discarded or deployment of the method on only a handful of databases, where the performance of some databases are significantly poorer than others. It can also

be difficult implementing previously-published algorithms due to poorly articulated instructions, omitted details and misunderstandings. Furthermore, if a system is to be deployed in the public domain, the enhancement algorithm should perform well in all scenarios with a variety of different sensors. This is especially the case when templates are not used as it is even more important to suppress intra-sample variance to facilitate extraction of highly reproducible features. The best possible performance is required. Therefore, for this project, it was decided that, in order to extract minutiae fairly and with a high degree of reliability, proprietary software [59] was to be used to enhance the fingerprint and extract the minutiae.

2.1.2. Fingerprint Segmentation Methods

Segmentation, in the sense employed here, is used as an umbrella term. It encapsulates the techniques employed to identify specific regions of the signal (an image is a 2D signal), which belong to the true biometric data. Fingerprint segmentation methods can be applied on many different scales. Most commonly segmentation occurs on the neighbourhood scale (block-wise), whereby the image is partitioned (overlapping or non-overlapping) and the local neighbourhoods are assessed to determine whether foreground elements of the image are evident in that region. This category of segmentation primarily segments the region of interest (ROI). The finest scale at which segmentation is performed is at the pixel-level. A ROI segmentation method can also be performed at the pixel-level, or pixel-wise, but more commonly they are used for ridge segmentation. This is because pixel-wise ROI segmentation is slower than the block-wise approaches and is only marginally better at defining the boundary.

2.1.2.1. Ridge Segmentation

Ridge segmentation involves the individual labelling of pixels, indicating whether they belong to a ridge (foreground) or belong to the background. It is essentially a threshold scheme and, as the pixels, which belong to the foreground elements, are generally darker than those belonging to the background (due to contact with the sensor), a greyscale binarization method can be employed. Otsu's [3] threshold algorithm is a standard threshold method employed consistently throughout the realm of image processing. Otsu's method assumes the image is bi-level, obtains the histogram of the image and separates foreground from background by minimizing their respective intra-class variances. However, when the threshold method is applied to an entire image, many pixels may be misclassified due to the inconsistencies in the background. This will provide an inaccurate segmentation and therefore, local adaptive threshold methods [60, 61] are generally more suited to fingerprint images.

Employing a local adaptive threshold algorithm can also provide many misclassifications, especially in low contrast regions. "Ghosting" is the term used to describe the pixel

misclassifications within a region of low contrast, whereby the entire region should have been declared either background or foreground if the threshold method was applied globally. However, as the threshold was determined on a local scale, the method does not realise that the region is essentially mono-level and will therefore segment the region assuming it is bi-level. Huang et al [60] tries to solve this problem by employing an adaptive window size which increases when the Lorenz Information Measure (LIM) is below a threshold, which is estimated by using the Otsu's method on the LIM. The windows whose LIM exceeds the threshold is then binarized using Otsu's method. However this method still assumes that the entire image is bi-level on the global scale so the method segments incorrectly in mono-level images.

Greenberg et al [54] also proposed an adaptive binarization method, which enhanced the image as it binarized. The algorithm applied local histogram equalization and Wiener filtering to reduce the noise before employing a threshold algorithm derived from the local mean. Furthermore, Emiroglu [62] presented a fingerprint-specific threshold method, which employed a novel averaging scheme to derive thresholds. There are many different threshold schemes available, some of which are outlined in [63], and determining which methods are best suitable for fingerprints would be an arduous task. However, some comparisons of many popular threshold techniques can be found in [64, 65].

Segmenting individual ridges directly using a threshold algorithm can induce many errors in the ridge structure. Often the binarized image is further enhanced or, following minutiae extraction, post-processed to remove irregularities. Ridge segmentation is generally only performed in minutiae-based systems because minutiae are very easy to detect when ridges are completely separated from the background. However, image-based systems do not require such a low level segmentation. Image-based systems require the location of where the fingerprint lies in the image. This is commonly achieved by implementing ROI segmentation.

2.1.2.2. Region of Interest (ROI) Segmentation

ROI segmentation, as outlined previously, locates regions which contain foreground elements. This is performed by partitioning the image into overlapping or non-overlapping blocks and then determining whether the block belongs to the foreground or background. This is commonly referred to as block-wise, ROI segmentation (BWRS) and is generally more computationally efficient than ridge segmentation as the calculations are performed on individual neighbourhoods, rather than every pixel. BWRS generally results in a mask than can be used to physically crop the image or used as a map, although some pixel-wise methods extract the ROI instead of ridge segmentation [66]. However, as BWRS only highlights the regions of the image that are of interest, no alterations to the original image is carried out, therefore preserving all the information in the image. Furthermore, the noisy regions of the image are also preserved. These areas can be marked

in the mask so the subsequent feature extraction stage can determine how reliable the features will be. It also allows the possibility of localised enhancement. A mask that contains such information is commonly called a quality map and will be further expanded later in the chapter.

2.1.2.2.1. Spatial-Based Methods

The forerunner of BWRS was Mehtre [67] and it is still used occasionally as a benchmark for comparing new techniques. The algorithm presented partitions the image into non-overlapping blocks, obtains the direction of local grey level uniformity and transforms the image into a directional image. Segmentation is determined by the statistics derived from the directional image. Mehtre further developed this technique in [68] after noticing that, although the original method out-performed a grey-level statistics (variance) approach, it broke down in uniform regions (i.e. when variance is small). Therefore, the original method was refined and converted into a hybrid approach where a threshold determined whether the original directional image-based method or the variance technique was to be used.

Ratha et al [56] however, built upon these foundations by recognising that the greyscale variance along the direction orthogonal to the local dominant ridge orientation in a foreground region should be distinctly greater than the variance in the direction parallel to the ridge orientation. However in the background region, the variance in every direction should be low. Using this information, a variance field is created, from which a quality map is derived. The variance field and the quality map are used to segment the image. Garris & Wilson [69], who developed the NIST (National Institute of Standards and Technology) Fingerprint Image Software (NFIS), proposed a different BWRS approach entirely. An iteratively increasing threshold is used to create a succession of binary maps. The number of transitions (vertically and horizontally) between 0 and 1 determines the optimal map and the other maps are discarded.

All the BWRS methods outlined previously are simple, fast and perform well on images with clear ridge structure and a relatively uniform background. However, as they all assume that the original image is bi-level, any additional elements will also be classified as foreground, and regardless what they are. This is because many non-fingerprint elements can have similar variance measures or directionality. Therefore, these measurements are not highly discriminatory features of fingerprint foreground regions. This misclassification of additional artefacts poses few problems when the regions are disjoint because a simple connected components algorithm [25] can remove them. Although, if the artefacts overlap with the fingerprint region, then removing these objects based on variance or directionality alone becomes a little more complex.

Many BWRS approaches extract a combination of features derived from greyscale statistics and directional properties in order to achieve optimum separation. Several approaches train artificial

neural networks (ANN) or classifiers with these features. Bazen & Gerez [66] presented a pixel-wise method to extract the ROI, which extracted the mean, variance and coherence to compile a feature vector. A linear classifier was trained with this feature vector to classify each pixel. This method was further improved by the implementation of a Hidden Markov Model (HMM) [70]. However, Chen et al [71] and Yin et al [72] determined that a linear classifier was not suitable for pixel classification as pixels are not linearly separable. So Chen et al [71] adapted [66] into a BWRS and substituted the coherence feature with the block cluster degree to train the linear classifier. Whereas Yin et al [72] developed a quadric surface formula whose coefficients were determined from a back propagation (BP) ANN to obtain optimum classification. Zhu et al [73] also implemented a BP ANN for the initial stage of a two-part algorithm. However, instead of just using the mean / variance / coherence triple as a feature vector, 11 greyscale statistical features were extracted from an oriented window centred at the centre of a block. The feature vectors were inputted into a BP ANN to classify background and foreground blocks, and then a linear classifier was used to check the block classification as the second part of the algorithm.

2.1.2.2.2. Frequency-Based Methods

Many segmentation methods exploit the underlying sinusoidal structure of a fingerprint. The fingerprint can be seen as an oriented, periodic texture and the projection of ridges from a local neighbourhood resemble a sine wave when the quality is exceptional, but the shape of the waveform deteriorates with ridge quality. Therefore, the amplitude and frequency of the projected sine wave can be used to ascertain whether the block contains foreground elements and, perhaps, the quality of the ridge stricture. Hong et al [2] extracted the frequency and amplitude in the spatial domain by projecting the ridges of oriented windows centred at the centre of each block (dubbed the X-Signature) and counting the number of peaks to estimate the frequency. Furthermore, the greyscale range of the projected ridges was used to estimate the amplitude, which is a measure of the orthogonal variance. Empirical thresholds were employed for both amplitude and frequency to classify each block.

An alternative approach to obtaining frequency and amplitude is to transform the image into the Fourier domain, where the impulses represent the frequencies which are present in the image. The locations of the impulses also present the orientation on which these frequencies occur and the brightness of the impulse is related to the amplitude of the component sine wave. At present, it is not possible to locate where in the image the strong impulses occur if the entire image was transformed into the Fourier domain. This is because the Fourier transform is localized in frequency and not space (time in a 1D signal). Therefore, local neighbourhoods are transformed into the Fourier domain independently. This is termed Short Time Fourier Transform (STFT) and the features extracted from the STFT of each block can be used for classification.

Sharat et al [1] partitioned the image into overlapping blocks and obtained the STFT of each block. The block was classified using the Fourier energy, whereby a simple energy threshold obtained using Otsu's threshold algorithm was implemented. Ong et al [74] also employed features obtained form the STFT and an adaptive threshold for the second stage of a two-part algorithm. The first part implemented an orientation certainty threshold to coarsely define the segmented region. A threshold is the simplest method of classification but many approached have opted to employ ANNs for classification. Marques & Thome [75] trained a Multi-Layer Perceptron (MLP) ANN using the features extracted from the STFT for segmentation. Whereas Wang et al [76] extracted greyscale statistics and the main energy ratio of each STFT (i.e. the ratio of the sum of the two main impulses, ignoring the D.C component, and the sum of all the FFT components at the same frequency as the two main impulses), and used the resultant feature vector to train a Radial Basis Function (RBF) ANN for classification. Furthermore, Sato & Umezaki [77] also extracted features from the spatial and Fourier domain. The group delay spectrum was combined with a histogram of the horizontal pixel line and was fed into a Recurrent Neural Network (RNN) for classification.

2.1.2.2.3. Gabor-Based Methods

It is evident from the methods outlined that frequency and orientation are two features which are used frequently to segment fingerprint images. Therefore, employing Gabor filters for segmentation is a logical step because the Gabor kernel is an oriented, periodic waveform and will accentuate images which are in synchronisation with the kernel. Conversely, images which are out of synchronisation will blur significantly. Alonso-Fernandez et al [30] proposed that a bank of 8 differently oriented Gabor filters was applied to the image, resulting in 8 Gabor-filtered images. Each filtered image was partitioned in to overlapping blocks. In each image, the four overlapping blocks were combined to create one block feature and the standard deviation of the block feature at each orientation was used as the feature vector. A standard deviation threshold level was implemented to segment the image.

Bernard et al [78] also proposed employing a bank of Gabor filters. However, the bank contained 24 filters using 8 different orientations and 3 different frequencies, producing 24 filtered images. Each filtered image was partitioned into blocks and the coefficient of projection for each block in each filtered image was determined. Comparing respective blocks across all 24 images, the one with the best coefficient of projection was established. From this, 3 images were obtained; one of frequency, one of orientation and one for projection coefficients. A threshold applied to the resultant projection coefficient image was used for segmentation and a sinusoidal model derived from the features was used to segment the ridges. This method is very expensive in terms of computational resources and time, although concurrent processes can significantly reduce the time.

There are alternative approaches to segmentation which attempt to identify certain points within the fingerprint and define a boundary which encompasses all the detected points. These methods are often extracted pixel-wise to locate these feature points. Qun et al [79] developed a method, whereby feature dots were extracted depending on pixel intensities in an 8 pixel neighbourhood centred on a focus pixel. A convex hull is created which encapsulates all the feature dots that remain after a pruning stage based on heuristics. Wu et al [80] and Baig et al [81] both detect the strength of Harris corner points. However, the method proposed by Wu et al [80] detects the points and encapsulates those, whose strengths are above a threshold, in a convex hull. Whereas, the technique proposed by Baig et al [81] partitions the image into blocks and the blocks are classified based upon how many Harris points, above a strength threshold, are within the blocks. Morphological operators are then used to fill holes in the mask.

2.1.2.3. Quality Estimation

An extension of BWRS is the block-wise quality estimation (BWQE) of the fingerprint. BWQE algorithms are essentially segmentation methods but classify the regions into a greater range of classes. Whereas the BWRS methods classify the image into background and foreground, the BWQE methods attempt to classify the regions into their quality level. The BWQE methods can be viewed as multi-level segmentation or that BWRS methods are bi-level quality estimation algorithms. In fact, both BWRS and BWQE methods use similar feature sets for classification as the objective of both methods is to locate regions of strong ridge structure and clarity. It is also worth noting that a BWRS can be obtained from the result of a BWQE by simply employing a threshold.

As mentioned previously, there are many factors which influence the quality of the captured fingerprint image and many of them are discussed in [22, 82, 83]. These factors can adversely affect the performance of the system, so quality estimation methods have been devised for three main purposes, which are to screen acquired images, to localise poor quality and to provide helper data. Many systems employ some degree of screening, either at the pre-processing stage or at the matching stage. Image screening determines whether the acquired image is of sufficient quality that, with or without enhancement, features can be confidently extracted. This prevents unrecoverable, poor quality images from completely progressing through the system, which improves system performance. Furthermore, the majority of systems, in some way or form, implement quality estimation at the local level, either indirectly at the enhancement stage or directly by flagging poor quality regions for use at the feature extraction stage.

Previously, the notion that adaptive enhancement algorithms indirectly ascertain the underlying quality of individual regions has been postulated. These adaptive approaches change depending upon characteristics of a local neighbourhood and these features are similar to those used in quality

estimation methods. Furthermore, localized quality estimation is commonly employed as helper data at the feature extraction level. This helper data could possibly allow the feature extractor to implement "don't care" clauses during extraction or for ranking minutiae to aide post-processing.

Quality estimation can be determined by analysing features that are extracted on the global or local scale. Local features are used more frequently than the global features due to the global features' inability to identify individual regions of poor quality. There are numerous local features that can be extracted for quality estimation, although there are not that many global features available. Generally, local features are obtained by partitioning the fingerprint into overlapping or non-overlapping blocks and extracting the feature sets from each independently. Based on the extracted features, a quality index is determined to quantify the quality of that region.

2.1.2.3.1. Local Features

2.1.2.3.1.1. Local Orientation Based Features

Metrics derived from the local orientation are popular for quality estimation. Lim et al [84] determines numerous features for each block, which are consequently categorized into 4 classes (good, bad, blank or undetermined). The features extracted include the orientation certainty level (OCL), ridge frequency, ridge thickness and the ridge-to-valley thickness ratio. The OCL is a useful feature as it measures the energy concentration along the dominant direction. Smudged or dry regions have lower energy concentration than that of well-defined regions because the magnitude and direction of the local intensity gradients, along the dominant direction, fluctuate vigorously.

The OCL is also implemented in [84] to detect high curvature regions, which can be used to detect regions of invalid curvature. However, regions containing singularities are high-curvature regions, so the fact that high curvature is exhibited within that region does not necessarily mean that it is of poor quality. The method presented in [85] also exploits the dynamic nature of intensity gradients within low quality regions. Likewise, Chen et al [86] extract the local coherence of the intensity gradients, which reflect the clarity of the local ridge structure. Lee et al [87] however adopted a different gradient–based approach. By obtaining 2 probability density functions (PDF) from the projection of the 2D gradient vectors in both the ridge direction and its orthogonal, a similarity measure between the locally extracted PDFs and 2 ideal PDFs is used to determine the local quality index.

Another orientation field derived metric was presented in [88], where the average absolute difference of orientation within surrounding neighbourhoods, which was coined local orientation quality (LOQ) measure, was determined. The metric is similar to a consistency measure of the orientation field [89], which highlights regions that undergo significant direction change within

neighbourhoods of the orientation field. Again, the LOQ measure is a supporting feature that does not indicate whether the block is truly of poor quality because singularity regions also exhibit significant direction change. Zhu et al [90] also proposed using the orientation field, although as an input to an ANN to determine the correctness of the ridge orientation. However, Fronthaler et al [91] extracted symmetry features from the orientation tensor, from which two symmetry representations were obtained. One was for linear symmetry and the other was for parabolic symmetry. A high degree of linear symmetry is characteristic of coherent ridge structure, whereas a high degree of parabolic symmetry is distinctive of high curvature points, such as minutiae and singularities.

2.1.2.3.1.2. Fourier Domain Features

As with BWRS, many local features try to capitalize on the underlying sinusoidal structure of a fingerprint. A sine wave has two main features, which are frequency and amplitude, although a sine wave in the spatial domain will also have an orientation component. These features can be extracted from the Fourier domain, which decomposes the image (2D signal) into its component sinusoidal waves. Lim et al [84] investigated the strength of the coefficients at all frequencies and deduced the quality of local neighbourhoods based on how dominant the maximum frequency impulse is. The image quality measure (IQM) is a ratio derived from the Fourier coefficients and classifiers are trained on a combination of features to classify individual blocks into 1 of 4 classes (very good, good, bad and very bad).

2.1.2.3.1.3. Gabor Filter Based Features

As mentioned previously, Gabor filters are often used in all aspects in a fingerprint system due to the oriented, periodic shape, which accentuates ridges and suppresses noise. Alonso-Fernandez et al [30] developed a BWRS method based on features extracted from resultant Gabor-filtered images. However, a precursor to this method was proposed in [78]. The image was initially segmented akin to the BWRS method in [78] and followed by classifying the foreground regions in to either good or poor quality by using empirical thresholds. The employment of further empirical thresholds determined whether the poor quality blocks contain smudged or dry areas.

2.1.2.3.1.4. Pixel Intensity Based Features

Although all features are related in someway to the intensities of the pixels of an image, only features that are directly determined from the pixel intensities are considered in this section. The aforementioned features required the transformation of the pixel intensities in to another form (i.e. directional image, Fourier domain or filtered image) in order to obtain meaningful features. Greyscale statistics are evidently important when ascertaining the quality of a region because low variance is a characteristic of poor quality. Yi-Sheng et al [82] employed the mean and contrast of

local neighbourhoods to classify whether the region is smudgy or dry. Whereas in [92] proposed two methods to quantify local quality. The first approach obtained the local neighbourhood variance, while the second extracted the local contrast among ridges and valleys along the local direction. Shi et al [93] expanded on the variance approach by incorporating the local mean, variance, contrast and eccentric moment.

Other methods attempt to quantify the clarity of the ridges and valleys. Chen et al [86] obtained a measurement for clarity by extracting the sinusoidal-shaped wave (X-Signature) that local ridge structure exhibits and then computing the grey-level distribution of the segmented ridges and valleys by means of a threshold. Lim et al [94] also extracted a measure of clarity by determining a clustering factor. In this approach, the ridges and valleys are segmented by means of applying local thresholds and the clustering factor of each block is defined as the degree to which grey-values of ridge / valley pixels are clustered.

Local ridge projection (X-Signature) is a useful tool for obtaining features that are common to sine waves in the spatial domain. It is evident from the methods previously reviewed that frequency, amplitude and orientation are useful features for ascertaining local quality estimations. Hong et al [2], employs the X-Signature method to obtain the frequency, amplitude and variance in order to classify local blocks as being either recoverable or non-recoverable. While Kyungtae [95] obtains the wavelength (inter-ridge distance) from the locally extracted sinusoidal waveforms to determine whether a region is either wet or dry. The widths of ridges and valleys are useful features because in smudged regions the width of a ridge is wider than usual, whereas dry regions have the opposite effect.

Although the features described are categorized, many of the previously-proposed algorithms in fact use a combination of local features derived from the same or different domains / spaces. This is evident in the algorithms proposed in [2, 84, 88]. In addition to these combinational methods, the minutiae detection package (MINDTCT) provided as part of the NIST Fingerprint Image Software (NFIS) [96] combines the direction, low contrast, low flow and high curvature maps into one map quantifying local quality into 5 different categories. As with many classification tasks, multiple feature based classifiers often perform better than single feature-based methods because single feature-based methods are inflexible. For example, single feature-based classification methods often employ thresholds. Unfortunately, many query images may not adhere to the standard thresholds. Therefore, adding extra features allows classifiers to take the extra dimensionalities into account.

2.1.2.3.2. Global Features

Global features attempt to determine an estimation of the quality of the entire fingerprint itself. It is possible to determine a global quality estimate from the local quality indices. For example, the global estimate can be ascertained by averaging the local quality indices [86] or by defining a ratio based on the quality estimates of local neighbourhoods [97]. However, there are not many global features identified in the literature and the ones that are require the transformation of the image into an alternative domain / space. Lim et al [94] calculated the continuity of the orientation field and the uniformity of the frequency field and combined them to obtain a global estimate. Whereas Chen et al [88] and Fierrez-Aguilar et al [98] both transformed the ROI of the fingerprint into the Fourier domain and extracted the energy concentration at a range of frequencies. Good quality images should have high Fourier coefficients in no more than a few frequency bands.

2.1.2.4. Summary

Investigations into quality estimation [99, 100], have determined that the performance of a system degrades severely when poor quality regions / images are allowed to progress through the system. Therefore, it is important segment the fingerprint image. Mehtre [68] proposed a set of characteristics that a good segmentation method should exhibit. A good segmentation algorithm should (1) not be sensitive to the contrast of the image, (2) detect smudged and noisy regions, (3) be independent of whether the input image is an enhanced or raw image and (4) provide consistent results for images with various quality expected by the application. However, the majority of segmentation algorithms do not fulfil all of these requirements. For example, smudged and noisy regions are often classified as background when a segmentation algorithm is applied.

On the other hand, quality estimation methods preserve these poor quality regions, by producing a map of locally derived indices. It defers the decision process to subsequent stages of the system. Fierrez-Aguilar et al [98], Chen et al [88], Yun & Cho [26] and Wang et al [27] investigated the use of incorporating quality measurements at different stages of the fingerprint system. In all cases, the quality estimation is used to suppress intra-sample variation, which is the ultimate objective of including quality indices. However, not all automated fingerprint systems will be able to incorporate quality indices at different stages. For instance, in a multi-feature, matcher-less implementation, as in this project, not all features support the incorporation of quality indices and a matcher does not exist to benefit from the additional quality information to improve the performance. Therefore, the only conceivable stage at which quality estimation can be performed, in order to benefit all feature extractors, is at the pre-processing stage in a screening capacity. Pre-processing is essential in matcher-less implementations because the matching stage facilitates flexibility between samples of the same group. When this stage is omitted, intra-sample variance most be reduced elsewhere. The pre-processing stage is ideal to reduce rotation, translation and scale variations, because it will improve the reproducibility of variant features.

2.2. A Novel Level-Independent Fingerprint Segmentation Algorithm Based on Fourier Energy

As discussed above, fingerprint segmentation is a crucial stage in the majority of automated fingerprint systems. To a certain degree, segmentation governs the operational success of the overall system as it eliminates regions of noise and background. By retaining the ROI, subsequent processing is restricted to this region. This can improve fingerprint enhancement as unrecoverable regions are rejected, increase execution efficiency by reducing the area on which subsequent processing will perform, and ensure the extracted features possess factors indicative of good quality.

The fingerprint segmentation review section outlines many existing methodologies to obtain the ROI that employ a multitude of features. However, many of the techniques described focus heavily on separating background from foreground that it assumed only background and foreground elements will always be present (i.e. bi-level image). Unfortunately, this is not always the case because some images may have artefacts within the image or the image may contain only background or foreground regions (mono-level). Greyscale statistical, or directional, segmentation often misclassifies artefacts as foreground when they are not. It is possible that an object or noise that is apparent in a fingerprint image can have similar greyscale statistics and directionality that is similar to true foreground elements, which causes misclassification. This does not pose difficulties when the true fingerprint foreground and the false foreground (i.e. artefacts or noise) is disjoint because a simple connected components algorithm can be employed to remove the false foreground regions. However, if the true foreground is occluded by the false foreground, then the false foreground will remain. Therefore, features which are highly discriminatory are required in order to minimise the probability of this eventuality.

Texture features are commonly employed in fingerprint segmentation because the fingerprint image can be perceived as an oriented, periodic texture pattern. The pattern exhibited by the ridges within a local neighbourhood of a fingerprint is analogous to an oriented sinusoid, which gives rise to frequency and orientation components. The greyscale oscillations are a highly discriminatory feature for locating fingerprint foreground regions. If the variation between the texture features between the true foreground regions and other regions is significant, then an accurate segmentation can result. However, many of the existing techniques separate background and foreground elements by employing a single threshold. If external artefacts, which may be classified as foreground, are present then these regions can affect the threshold calculation, which will adversely affect the threshold estimation. This will result in poor background / foreground separation. Therefore, an approach to ascertaining multiple thresholds, based upon the Fourier energy, is proposed in order to improve segmentation.

2.2.1. Existing Short-Time Fourier Transform (STFT) Approach

Frequency-based features have long been exploited in automated fingerprint systems. Texture features in general are more resilient to variable contrast and noise than greyscale and directional statistics, and this is true of frequency-based features. There are two commonly used methods for obtaining frequency features. In the spatial domain, an oriented window is used to project the ridges orthogonal to the dominant ridge direction (commonly dubbed the X-Signature). The subsequent ridge projection is similar to a sine wave, which amplitude and frequency can be obtained from. An alternative method is to transform local regions of the fingerprint into the Fourier domain by means of the Fast Fourier Transform (FFT). This approach of partitioning an image and transforming each individual partition into the Fourier domain is often termed Short Time Fourier Transform (STFT).

Chikkerur et al [1] proposed the use of the STFT to segment a fingerprint image based on the Fourier energy map. The method involved the transformation of local regions to the Fourier domain and an energy map was subsequently generated, which was then subjected to Otsu's threshold algorithm. The algorithm is conducted as follows:

1. Partition the fingerprint image into overlapping cells. Cell dimensions for FFT must be a power of 2.
2. Perform 2D FFT [1] on each individual cell.
3. In each cell compute the Fourier energy using Equation 2-1.

$$E(x, y) = \log_{10}\left(\sum_{r}\sum_{\theta}|F(r,\theta)|^2\right)$$
$$|F(r,\theta)| = \sqrt{F_r(r,\theta)^2 + F_i(r,\theta)^2} \qquad \text{2-1}$$

   Where:

   $F_r(r, \theta)$ and $F_i(r, \theta)$ are the real and imaginary Fourier coefficients located at (r, θ).

   (x, y) is the cell coordinates.
4. Threshold energy map, E, using Otsu's threshold algorithm [3].
5. Retain largest connected component.
6. Fill holes using binary morphological operators.

It is important to note that the existing algorithm is performed on overlapping cells, whereby four component cells are combined to create an energy estimate for a single cell. To accomplish this, an average of the four component cells is used as the energy estimate. Furthermore, the algorithm can be optimized by exploiting the symmetrical nature of the Fourier transform. Therefore, only half of the spectrum needs to be evaluated. This changes the energy calculation in Equation 2-1 to that of Equation 2-2. In Equation 2-2, the polar coordinates have been replaced with Cartesian coordinates for simplicity.

$$E(x,y)=\log_{10}\left(\left(2\sum_{i=1}^{i<d}\sum_{j=1}^{j<d/2}|F(i,j)|^2\right)+\sum_{i=1}^{i<d}\left|F\left(i,\frac{d}{2}\right)\right|^2\right) \qquad \text{2-2}$$

$$|F(i,j)|=\sqrt{F_r(i,j)^2+F_i(i,j)^2}$$

Where:

$F_r(i, j)$ and $F_i(i, j)$ are the real and imaginary Fourier coefficients located at (i, j).

(x, y) is the cell coordinates.

d is the dimension of the cell, which must be a power of 2 for FFT.

The existing STFT algorithm segments the fingerprint image satisfactorily in images that possess significant variation between background and foreground elements. Unfortunately, many fingerprint images do not exhibit an obvious transition and this can lead to poor segmentation. Executing the algorithm as detail above on the Verifinger test database produces many misclassifications. Some examples are provided in Figure 2-1.

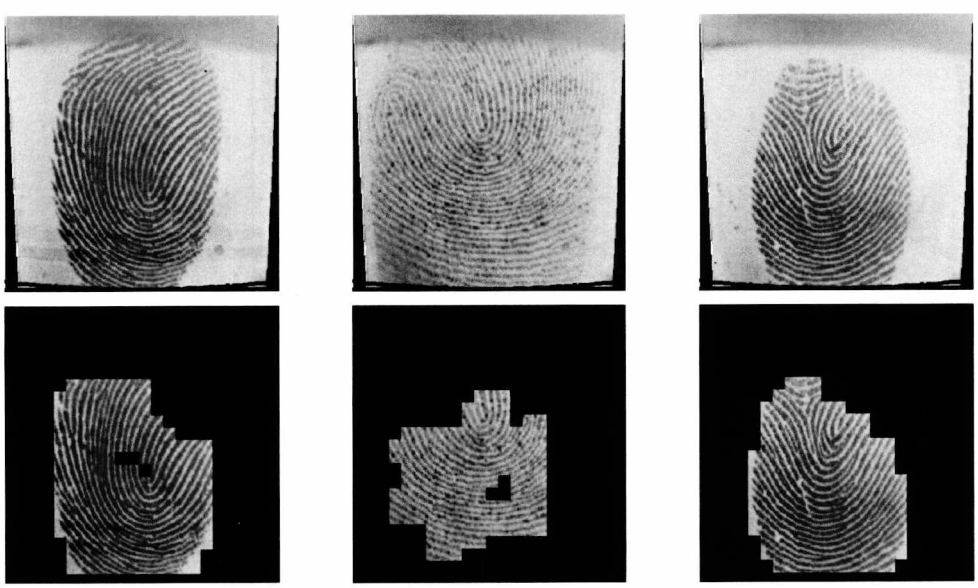


Figure 2-1: Example of Poor Classification from the Verifinger Test Database.

Figure 2-1 presents some poor segmentation results from employing the existing STFT algorithm. It should be noted that two tests were carried out whereby the quantization of the energy map was altered. The first test quantized the energy map into 100 equidistant bins and the second quantized the map into 1000 equidistant bins. The results showed that both tests resulted in almost identical segmentation. The top row of Figure 2-1 shows the original images before segmentation and the bottom row presents the results. It is clear from these results that the small, black border adversely affects the threshold decision. The transition between the black border and the grey background gives rise to strong frequency components which induces a new level in the energy map. This level

is not taken into account when applying Otsu's threshold algorithm as it assumes that there are always two levels present. Therefore, a novel, iterative threshold approach is presented in order to accommodate multi-level images.

2.2.2. Novel Multi-Level STFT Approach

The original STFT approach, as mentioned previously, performs well on bi-level energy maps because of the high variation between background and foreground components. However, as with many segmentation methods, the reliance on Otsu's threshold method can cause many misclassifications when the levels are not well defined or when multi-levels are present. One approach to multi-level threshold problems is the implementation of empirically-derived static thresholds. This method however is not feasible to threshold the energy map because the Fourier energy of the fingerprint is dependent on the contrast between ridges and valleys. The greater the contrast, the greater the effective amplitude of the component sine wave at that frequency, which in turn increases the energy. Hence the static threshold approach becomes contrast-specific. Alternative multi-level threshold approaches [101] do exist, however these often require the number of levels to be known *a priori*. The number of levels and the thresholds of these levels are not known so an alternative approach has been devised.

By investigating the behaviour of the energy map, it was noticed that the energy level exhibits jumps at transitions between levels within a sorted energy map. Locating these jumps provide threshold levels which can be used to segment the image. Figure 2-2 shows a bi-level image, its energy image and its sorted energy map.

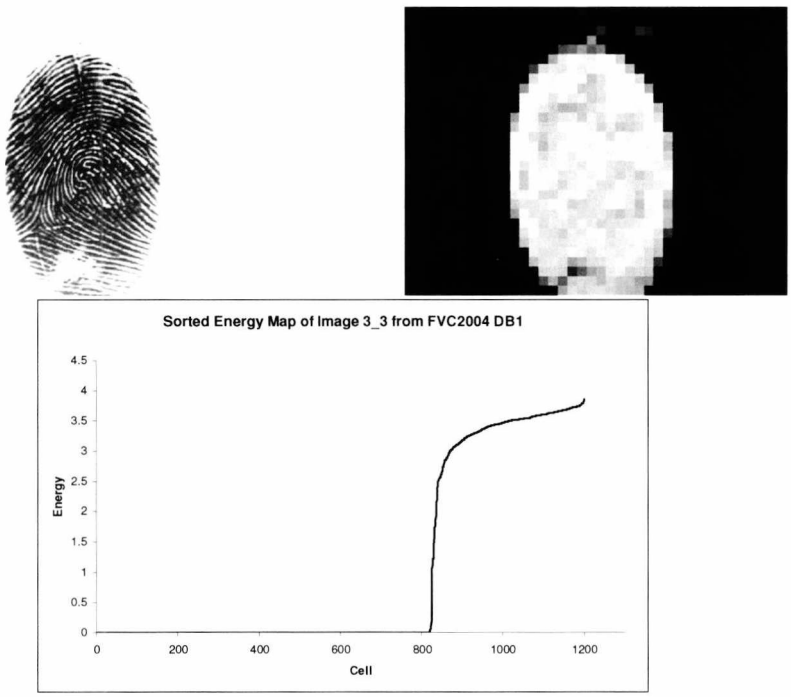


Figure 2-2: Example of Bi-Level Energy Map

The example in Figure 2-2 shows the bi-level fingerprint image of 3_3 from the FVC2004 DB1 database (top left), the energy image (top right) and the sorted energy map (bottom). It is clear that a significant incline is exhibited at about the 830th cell and that there seems to be only one significant jump. This characteristic is typical of a bi-level image and the level can subsequently be confidently segmented by using the energy level from cell 830 as a threshold. However, in other images there maybe numerous inclines and none as prominent as the jump in Figure 2-2. Figure 2-3 demonstrates the typical energy map of an image from the FVC2004 DB2 database.

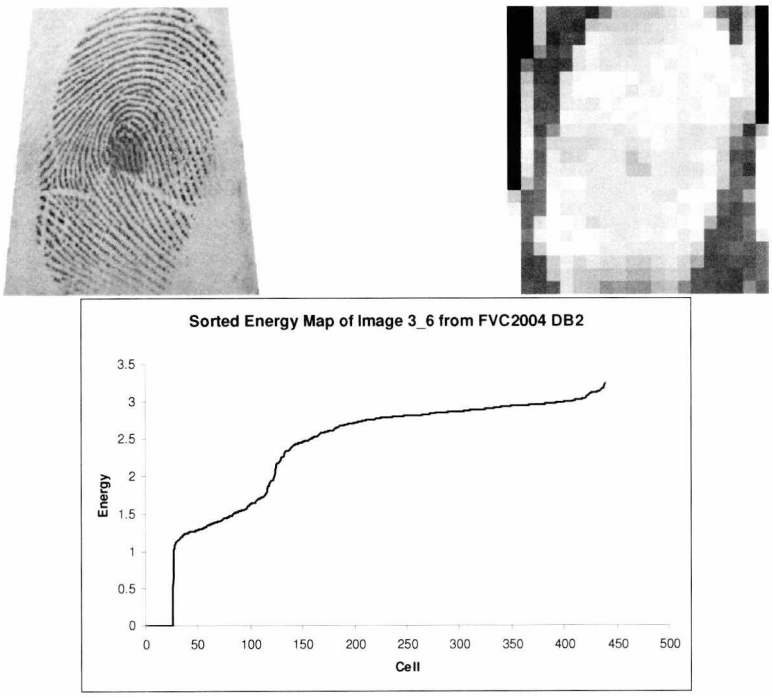


Figure 2-3: Example of Tri-Level Energy Map

The example shown in Figure 2-3, demonstrates the tri-level sorted energy map of image 3_6 from the FVC2004 DB2 database. It is clear that the curve does not exhibit as prominent jumps as that of the example presented in Figure 2-2 and that there are two inclines. One incline is located at the start of the curve at approximately the 7^{th} cell and the second is located around the 400^{th} cell. Therefore, the thresholds should be set to the energy values to which these cells refer to.

By locating these inclines and extracting the energy magnitude at where these jumps occur, the energy map can be successfully segmented. As with any signal that requires the detection of abrupt changes, this detection is best performed by taking the first derivative of the signal. The first derivative measures the rate at which the slope of the curve changes and therefore, when a large change occurs, the first derivative is also large. By analysing the first derivative of the sorted energy map, the sharp inclines can be detected and the subsequent energy thresholds can be obtained by using the point at which the inclines occur as indexes into the sorted energy map. The multi-level segmentation procedure is as follows.

1. Determine energy map as directed by the existing STFT approach. However, ignore the D.C. value in all cells when calculating the Fourier energy.
2. Sort the energy map, E_s.
3. Obtain first derivative by using Equation 2-3.

$$G(x) = E_S(x) - E_S(x-1)$$

Where:

$E_S(x)$ is the sorted energy map. **2-3**

X is the index in the map.

$G(x)$ is the array of first derivatives.

4. Normalize the array of first derivatives by implementing Equation 2-4. By normalizing the array, it allows a constant decrementing step to be applied across all images.

$$G_n(x) = \frac{G(x)}{\max(G(x))}$$

Where: **2-4**

$G(x)$ is the array of first derivatives.

$G_n(x)$ is the normalized array of first derivatives.

5. Smooth the array of first derivatives by using a Gaussian filter. An example of the intended output of this stage is shown in Figure 2-4.

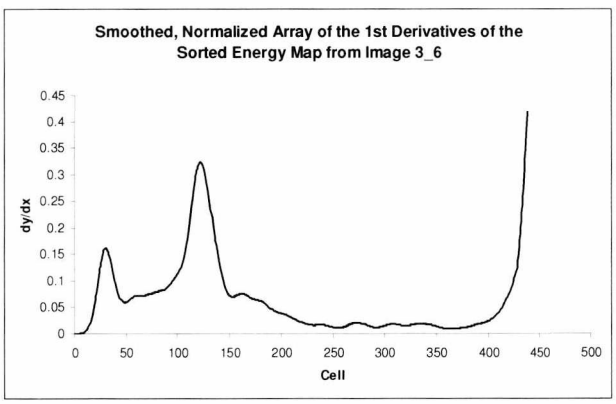


Figure 2-4: Example of a Smoothed, Normalized Array of 1st Derivatives of a Sorted Energy Map

6. Zero an accumulator array (A).
7. Set the initial threshold (t) to just under the maximum of the array of first derivatives.
8. If any inter-peak distances (i.e. local maximums) in accumulator array (A) is less than constant (c = 10% of total number of cells):
   a. Then go to step 9.
   b. Else increment accumulator according to Equation 2-5, decrement threshold (t) by a constant value (e.g. 0.01) and go back to step 8.

Where:

$A(x)$ is the accumulated value of cell x in the accumulator array.

$G_n(x)$ is the smoothed, normalized 1st derivative of cell x. **2-5**

t is the threshold which indicates the minimum number of cells required to be considered a level outright

9. By treating the first cell and the last cell as peaks, determine the inter-peak distances, retain the two greatest inter-peak distances and set all cells that do not lie within these regions to 0 in the segmentation map. An example of this process is demonstrated in Figure 2-5, whereby the double ended arrows depict inter-peak distance. These inter-peak distances reflect the area of each region in the energy map.

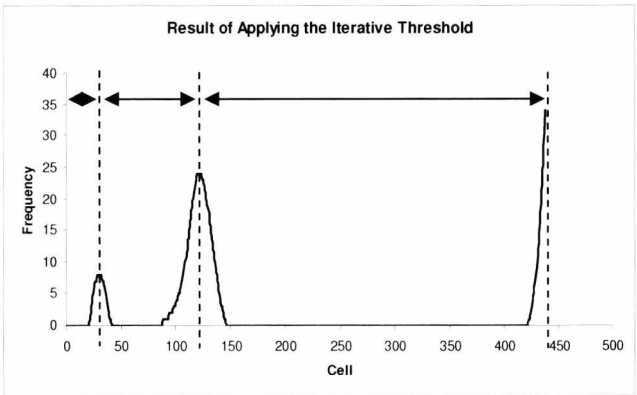


Figure 2-5: Result of Iterative Threshold

It is clear from Figure 2-5 that 3 regions have been detected. The first region ends at approximately the 30th cell, the second region ends at approximately the 120th cell and the third region ends at the very last cell. Regions 2 and 3 have the two greatest inter-peak distances, therefore any cells with energy lower than the energy in the 30th cell (i.e. 30th cell in the sorted energy map - $E_s(30)$) belongs in the background.

10. Out of the two regions remaining, the foreground is taken as the region which lies furthest along the x-axis. This is because foreground regions typically have higher energy than background regions. Therefore the example, shown in Figure 2-5, would classify the cells with energy less than the energy of the 120th cell (i.e. $E_s(120)$) as the background and the cells whose energy is greater than the energy of the 120th cell, $E_s(120)$, as the foreground.
11. Retain the largest connected component and fill in holes with either a mode filter or a binary morphological operator (e.g. close).

The algorithm outlined is slightly different from the originally proposed method in [102] at a few points. Firstly, the cells that extend past the input image's boundaries are rejected in this implementation, which essentially crops the image to a multiple of the cell size (i.e. size of block to be transformed into the Fourier domain). This prevents the noisy transition between padded regions and the image exhibited in [102]. Secondly, it was later discovered the transitions between levels was further pronounced when the D.C component in the Fourier domain was omitted from the energy calculation. This enhancement was added after realising that the energy of a completely white cell was different from a cell, which was completely black. By ignoring the D.C component compensated for this inequality. The final amendment was to iteratively lower the threshold level

rather than increasing it. This change was administered because the logic is simpler, although both methods resulted in the same threshold decision.

The iterative threshold method proposed can also be employed to count the number of transitions in any implementation. This is especially true in this scenario whereby the number of levels exhibited within the energy map can be input into a multi-threshold algorithm [101]. However in this case, the threshold estimates are simply obtained by using the peak locations as indexes into the sorted energy map, so there is no need to increase computational complexity by further implementing a multi-threshold algorithm.

2.2.3. Novel Level-Independent STFT Approach

The multi-level STFT approach provides very good segmentation on multi-level energy maps. However, as with the original STFT approach, no provisions are made for mono-level energy maps. Mono-level energy maps indicate that only background or foreground is apparent in the image. This does not pose too many problems when presented with live-scan images because the image acquired directly from a scanner generally has elements from both background and foreground. Although, if the system was to allow the use of static images, these images can often be cropped so only foreground remains. Furthermore, it may be beneficial for the system to recognise when an acquisition does not contain a fingerprint image (i.e. possessing only background (non-fingerprint) elements).

The simplest approach to solving the problem when only foreground elements are apparent in an image is to either pad the image on all sides with zeroes (black border), 255s (white border) or the mean pixel value, or extend the image's border pixels. This will essentially produce a border for all images and create an additional background level. This will enable the original STFT method to segment the image based on the premise that background and foreground elements are contained within the image. However, the original STFT method does not know if the image is mono- , bi- or multi-level, and therefore possibly adversely affect the threshold decision. In contrast, the Multi-Level STFT approach will determine that the energy map is mono-level, although it doesn't know whether it is full of background or foreground regions. Therefore, the Multi-Level STFT method assumes that it contains only foreground regions. However, this does not solve the problem when only background elements are presented. This eventuality is rare, although very poor images can be acquired that are not discernable from the background, presenting, in essence, a mono-level energy map.

In the original proposal [102], the image was initially screened to determine whether at least two levels were contained within the image. This was ascertained by performing a rudimentary segmentation based on local variance and if the image was deemed to be mono-level, the image is

segmented by implementing the X-Signature approach outlined in [2]. This is because the X-Signature method does not rely on background and foreground information to segment the image. However, the occurrence of mono-level images should be rare and therefore performing a screening stage, prior to all image segmentations, is an unnecessary computational overhead. In order to provide a STFT approach, that is level-independent and reduces computational overheads, a simplified approach to the originally proposed method [102] is presented.

1. Perform the Multi-Level STFT algorithm as outlined in the previous section. However determine how many levels are present at step 9.
2. If the number of levels is equal to 1:
    a. Then perform the X-Signature method [2] to determine whether the image contains only background or foreground regions.
    b. Else continue to segment the image with the Multi-Level STFT algorithm.

2.2.4. Experimental Results

The original STFT and the proposed level-independent method were applied to the Verifinger test database and to databases 1 and 2 (DB1 and DB2 respectively) of the 2004 Fingerprint Verification Competition (FVC2004) set of databases. The Verifinger test database and FVC2004 DB2 were selected because the images within these databases are the closest approximation to tri-level images than any of the other standard benchmarking databases. The algorithms were performed on FVC2004 DB1 in order to determine whether any difference between the segmentation results of both algorithms was exhibited on a simple bi-level data set. It is expected that the resulting segmentation differences between the two algorithms should be minimal because of the significant difference between the energy of the background and foreground regions.

Ascertaining the quality of a segmentation algorithm is highly subjective. A typical measure used to quantify how well a fingerprint is segmented is simply the ratio of correctly classified foreground regions. This requires the manual classification of foreground and background elements in every image. As this is necessarily a time consuming procedure, especially as FVC2004 DB1 and DB2 both contain 800 images, it would not be feasible to quantify successful segmentation in this way. In any case, manually classifying fingerprint regions is also highly subjective because poor quality maybe classified as background by some people and foreground by others. Therefore, in order to determine the quality of the proposed algorithm, the results are visually compared to that of the original STFT approach and a tally was kept of which algorithm obtained the best segmentation. Some typical segmentation results are provided in Figure 2-6 for the Verifinger test database and Figure 2-7 for FVC2004 DB2.

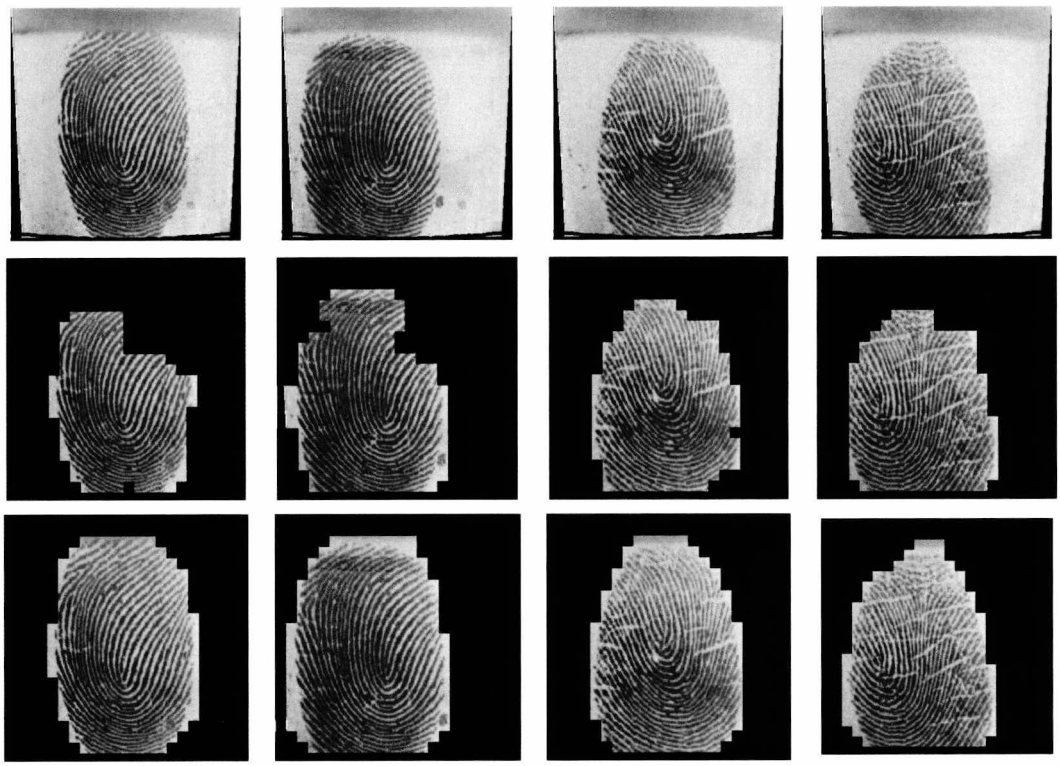


Figure 2-6: Example Results from the Verifinger Test Database. Top Row: Original Images. Middle Row: Existing STFT Approach. Bottom Row: Proposed Approach.

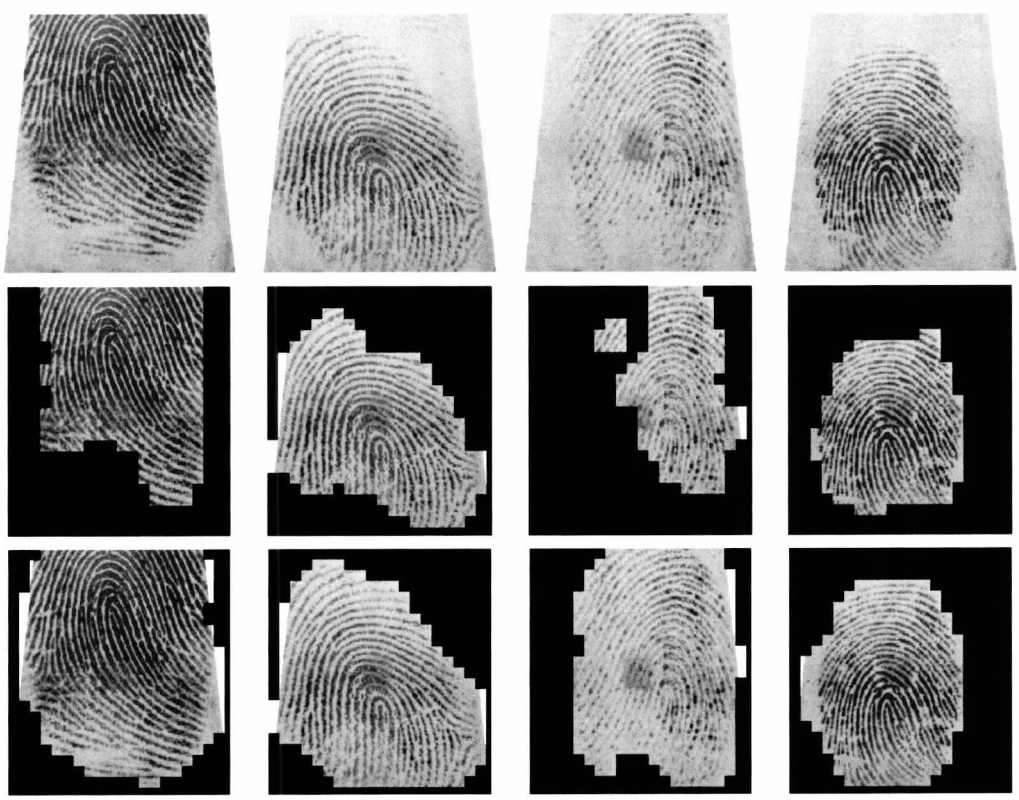


Figure 2-7: Example Results from FVC2004 DB2. Top Row: Original Images. Middle Row: Existing STFT Approach. Bottom Row: Proposed Approach.

It is evident from the example results provided that the proposed method produces better results than that of the original STFT approach. Overall, a significant improvement was exhibited by the proposed approach. The segmentation of the fingerprint images was found to be better in approximately 75% of the images in the Verifinger test database and approximately 92% of the images in FVC2004 DB2. As expected, minimal difference was exhibited between the results of the two algorithms when performed on FVC2004 DB1. Furthermore, the additions made in the proposed method do not increase the computational time significantly. A summary of the execution times of both algorithms applied to the 3 aforementioned databases is presented in Table 2-1.

Table 2-1: Execution Times

	Verifinger DB (s)	FVC2004 DB1 (s)	FVC2004 DB2 (s)
Existing – Min	0.031	0.109	0.031
Existing – Max	0.078	0.188	0.0782
Existing - Average	0.051	0.125	0.072
Proposed – Min	0.031	0.109	0.031
Proposed – Max	0.125	0.812	0.063
Proposed - Average	0.055	0.135	0.049

Unfortunately, problems are encountered when the smoothed, normalized array of 1st derivatives exhibits level transitions similar to those presented in Figure 2-8. The dashed, horizontal line depicts the determined threshold level.

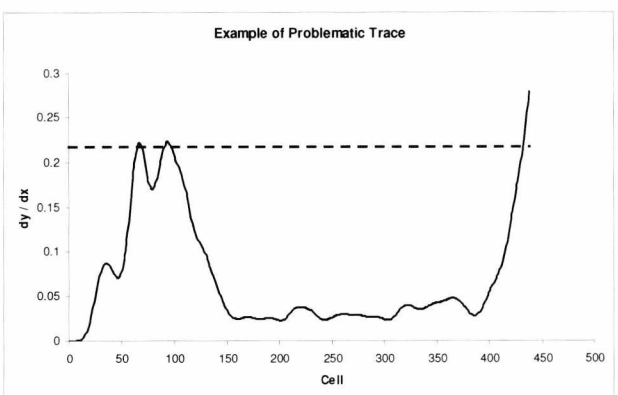


Figure 2-8: Example of Problematic Trace of the Smoothed, Normalized Array of 1st Derivatives

It is apparent from the trace presented in Figure 2-8 that the determined threshold line dissects the trace where two local maximums (at the 67th and 94th cells respectively) are close together and have similar magnitudes. As a result of this threshold decision, an entire level is overlooked as the termination rule is satisfied on account that the two peaks are too close together. This problem arises because the two local maximum's magnitude falls within the same quantization level, which

is derived from the maximum value in the array and the threshold decrementing step (0.01 in this implementation).

A simple remedy would be to reduce the threshold decrementing step further. However, if the values are equal (or difference is negligible) then this problem will still arise. Furthermore, reducing the step will also increase computational complexity by increasing the number of iterations. Alternatively, further heuristics maybe employed to determine this special case, although, as with most rule-based methods, there will always be one exception to the rules. Seeing as the rate of occurrence of this special case was less than 1% across all images, it was deemed unnecessary to invoke further heuristic rules.

2.2.5. Evaluation Remarks

A novel fingerprint segmentation algorithm has been introduced which addresses the potentially significant problems exhibited when multi-level images are exposed to an automated fingerprint system. The method employs an iterative threshold approach in order to segment the image based on the variations of the Fourier energy. The technique was applied to a number of standard databases and significant advances on the existing method were obtained. The performance of the algorithm is such that it can be employed as a practical segmentation tool for automated fingerprint systems.

However, it has come to light with further testing that the existing STFT method can be improved significantly with the addition of one minor detail. As with the proposed method, by simply omitting the D.C component, the variation between the foreground and background images is more pronounced. This little detail was not mentioned in [1] and incorporating the change improved the original STFT method so that its performance is similar to that of the proposed method for DB1 and DB2. The first incarnation of the proposed multi-level Fourier energy method [102] adapted the logarithm calculation (Equation 2-1) in order to segment the fingerprint without removing the D.C component. This was obtained by summing the logarithm of each individual Fourier coefficient rather than summating the coefficients and taking the logarithm of the sum. However, even with the removal of the D.C component, the proposed method slightly outperformed the improved, original method for the Verifinger test database. Although, this discovery dampens the significance of the proposed method, the iterative threshold decision-making and the level-counting method can be implemented in many different applications where multi-thresholds are required.

2.3. A Novel Computationally Efficient Fingerprint Segmentation Algorithm using Digital Closed Curves

Fingerprint segmentation forms a critical early stage of pre-processing for automated fingerprint systems where high efficiency can significantly improve overall recognition performance. However existing methods can be quite inefficient in terms of memory and time usage. It can be observed that a trade-off exists between quality and efficiency, whereby simpler, less computationally complex methods execute with high-speed but the quality of segmentation can be poor. Conversely, computationally complex methods employ classifiers, multiple features from various domains / spaces and modelling in order to improve segmentation results. However, all these enhancements incur more computational costs.

Many implementations do not require a fast segmentation method because the system may only employ a single feature for matching and, therefore, a result can be determined within an acceptable time-frame. However, in multi-feature implementations, more time is required to fuse the features and / or classifiers. This may impose restrictions on the allotted time available for pre-processing methods in order to ascertain a result within the same time-frame. Therefore, a novel, simple and intuitive method is proposed, which offers consistent segmentation at a fraction of the computational costs.

2.3.1. Digital Closed Curves (DCC) Approach

The proposed technique was primarily devised to be computationally efficient in order to address a major drawback of existing techniques. The method is loosely derived from the X-Signature technique described in [2]. However the proposed method eliminates the prerequisite of extracting directional components (i.e. orientation field), consequently making the DCC approach directionally independent. By eliminating the computation of the orientation field, time and resources are saved, decreasing the computational complexity.

The structure of a fingerprint is a highly discriminatory feature, which is often used to segment a fingerprint. Many methods try to exploit the structure by extracting the frequency of the sinusoidal structure exhibited by a fingerprint within local neighbourhoods [1, 2]. The X-Signature algorithm extracts the underlying sinusoidal waveform by projecting the ridges within an oriented window orthogonally to the dominant ridge direction. The resultant waveform is investigated with regard to its wavelength, amplitude and greyscale range. However, other waveform signatures can be obtained by traversing the borders of a digital closed curve. An example of how these signatures are extracted is presented in Figure 2-9.

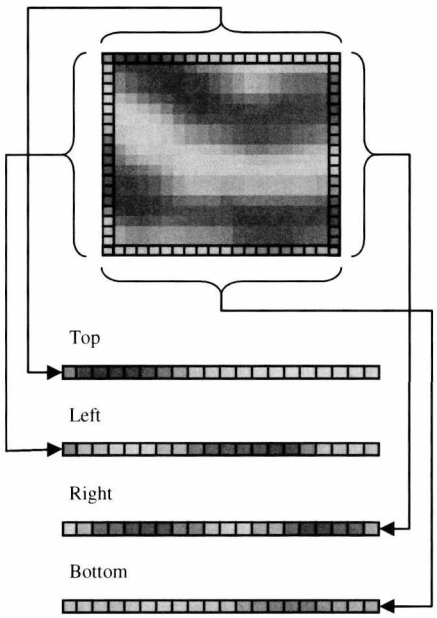


Figure 2-9: Extraction of Signatures

The signatures are simply obtained by traversing from left to right for the top and bottom boundaries, and from top to bottom for the left and right boundaries. By plotting the pixel intensities, four signature waveforms can be obtained. It was determined that the intersection of ridges with the boundaries of the digital closed curve, always produces a periodic waveform, independent of orientation, along at least two sides of the closed curve. The waveforms extracted from the example shown in Figure 2-9 are presented in Figure 2-10.

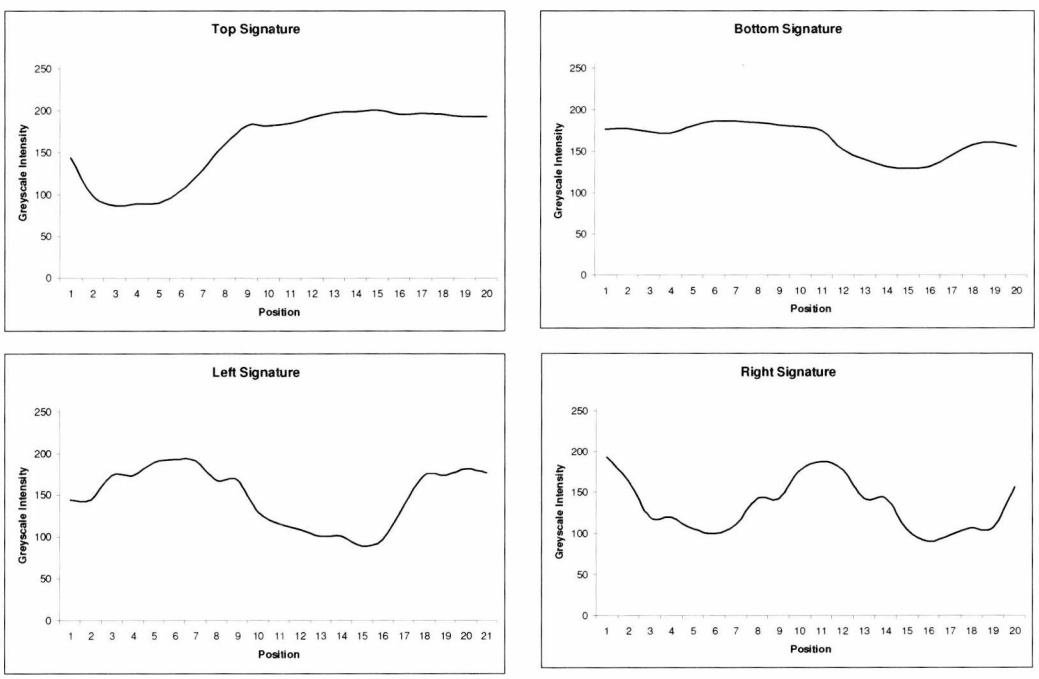


Figure 2-10: Signatures from a Foreground DCC

The left and right signatures presented in Figure 2-10 exhibit a relatively periodic structure with significant variation between the peaks and the troughs. This is indicative of foreground region. However, if the signatures were obtained from a background region, the signatures would exhibit a less structured shape with little variation. Example signatures extracted from a background region are presented in Figure 2-11. It is clear that these signatures exhibit little peak-trough amplitude and periodic behaviour, which is strongly indicative of a background region.

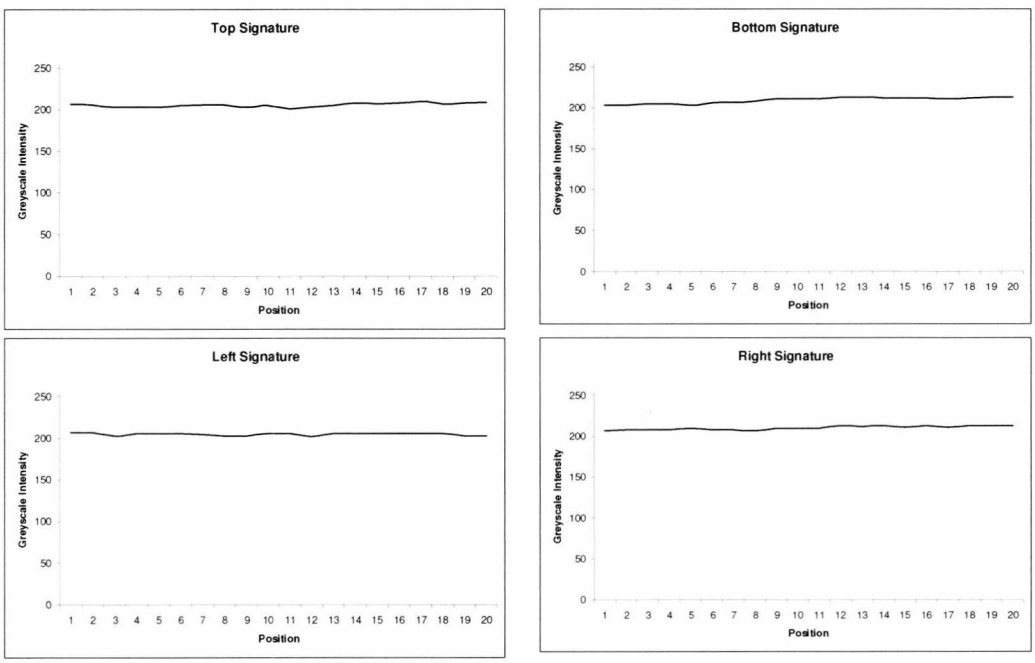


Figure 2-11: Signatures from a Background DCC

It is evident from the example signatures presented that there is significant differences between the signatures of a foreground region and that of a background region. Therefore, a fingerprint can be segmented by investigating the signatures of a digital closed curve. The DCC segmentation algorithm is as follows.

1. Partition the image into non-overlapping cells. In this implementation a cell of dimensions 16 x 16 was employed, although any size can be used.
2. Centre a DCC (square) about the midpoint of each cell. For an overlapping implementation the DCCs dimensions must be greater than that of the cell, and for non-overlapping, the DCC dimensions must be equal to the cells. For this implementation, the DCC dimensions were 20 x 20 so each dimension is double the average inter-ridge distance of a standard 500 dpi image, which is approximately 10 pixels [24]. Therefore the DCC should encapsulate a region containing a ridge-valley pair.
3. Extract the 4 signatures by projecting the top, bottom, left and right sides of the DCC.

4. Perform peak-trough detection on each of the 4 signatures. Example pseudo-code (Figure 2-12) of a simple peak-trough detection algorithm is presented. In this method, plateaux are taken as the midpoint between the locations where the sign change of gradient occurs.

```
start = -1, dec = 0, inc = 0;
for i = 0 to i < length -1
    diff = array[i+1] – array[i]
    if diff > 0
        if dec = 1 and start > -1)
            Trough located at (start + i) ÷2
        else if dec = 1
            Trough located at i
        dec = 0, inc = 1, start = -1
    else if diff < 0
        if inc = 1 and start > -1)
            Peak located at (start + i) ÷ 2
        else if inc = 1
            Peak located at i
        inc = 0, dec = 1, start = -1
    else if start = -1
        start = i
```

Figure 2-12: Pseudo code of peak/trough detector

5. Determine which signatures are of *suitable structure* by employing heuristic rules similar to that of the X-Signature method [2]. These heuristics are:
    a. The signature must contain at least one peak and one trough.
    b. The average peak-trough amplitude must be greater than a threshold. For this implementation the threshold was set at 40 as this ensures that there is significant variation between background and foreground.
6. The decision on whether the DCC contains background or foreground elements depends on which sides have *suitable structure*. The decision is based on the following conditions.
    a. If the number of sides which exhibit *suitable structure* is greater than 2, then the region is deemed to belong to the foreground.
    b. If there are 2 sides which exhibit *suitable structure* but they are on opposites sides of the DCC, then the region is deemed to belong to the foreground. If not, then the region is classified as the background.
    c. Otherwise the region is deemed to belong to the background.

Similarly to the X-Signature method, the DCC approach can be further improved by implementing filtering / interpolation, but with additional computational costs. The 4 signatures can be filtered with a low pass (mean or Gaussian) or median filter in order to aide peak-trough detection.

Furthermore, a median filter or binary morphological operators can be applied to the largest connected component of the resultant segmentation map to smooth the shape and fill in holes.

2.3.2. Experimental Results

Assessing the performance of a segmentation algorithm is highly subjective. A typical approach to quantifying the performance of a segmentation algorithm is to ascertain the ratio of the correctly classified regions. This would require manual classification of each region and would be very time consuming on very large databases. Therefore, this method of performance appraising is not feasible. In any case, the proposed DCC segmentation algorithm was not devised in order to obtain the best segmentation results, but was created to demonstrate that a simple segmentation method can provide comparable segmentation results at a fraction of the computational costs.

In order to assess the performance of the proposed DCC segmentation algorithm, the execution times and segmentation results were visually compared with 3 established alternative segmentation techniques. The algorithms were evaluated on the Fingerprint Verification Competition (FVC) 2004 set of databases. During the experimental phase, in order to maintain fairness, all preset values were kept constant for each database. The X-Signature's [2] oriented window's dimensions were maintained at 8 x 32 pixels, with a wavelength range of 3 and 25 pixels and a minimum variation of 40. Whereas the NFIS approach's [96] minimum and maximum foreground areas were set to 20% and 90% of the total image area respectively. Furthermore, the NFIS threshold increment was set to a third of its intensity range. The STFT method [1] does not require any parameters as the threshold is obtained automatically by employing Otsu's threshold algorithm, although the cell dimensions of the overlapping partitions was set to 16 x 16 pixels. Typical results of each of these segmentation methods are presented in Figure 2-13.

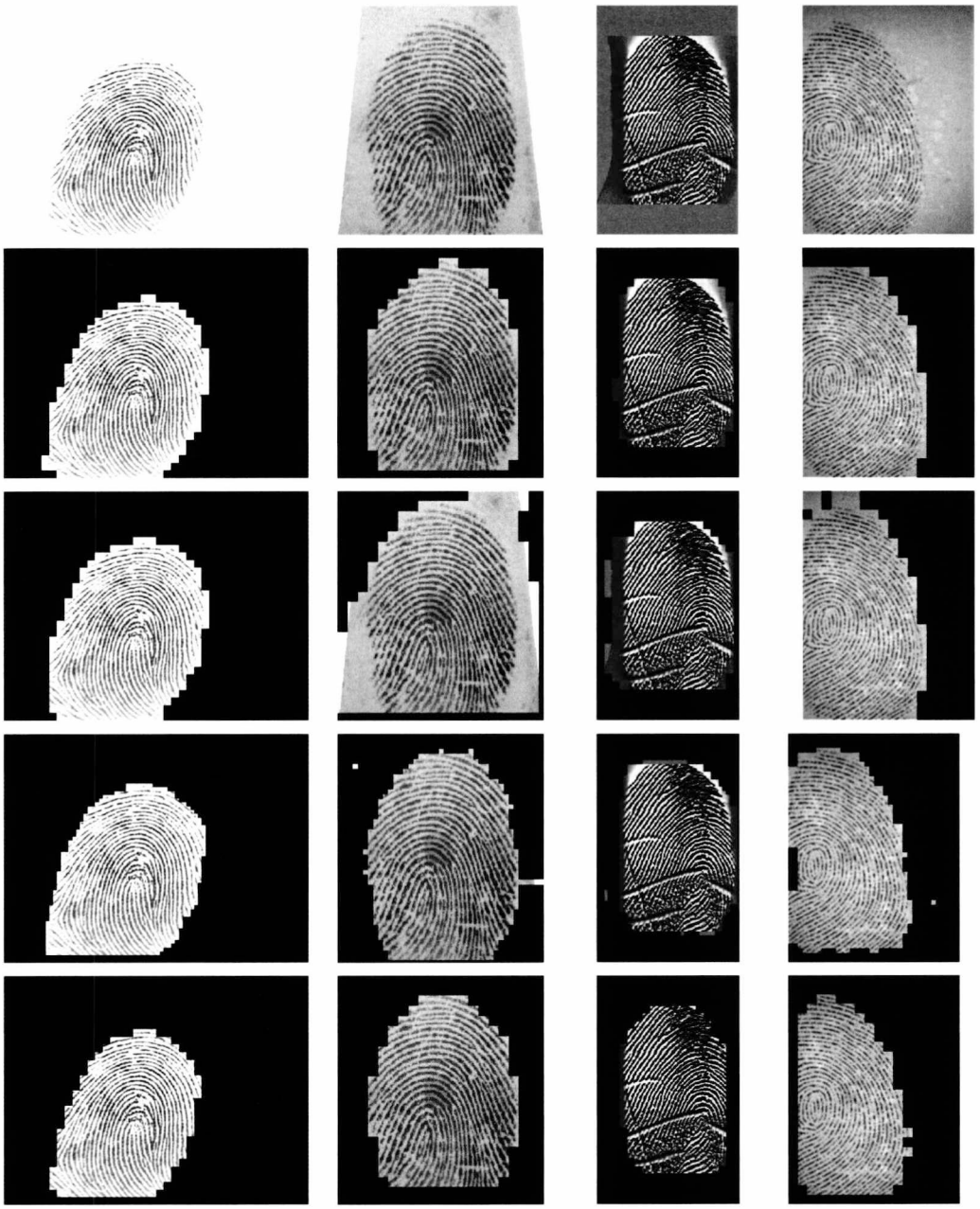


Figure 2-13: Example Segmentation Results

The test images were randomly chosen from each of the 4 FVC2004 databases and the organisation of Figure 2-13 is as follows.

- Row 1 contains the original images prior to segmentation.
- Row 2 contains the segmentation results obtained by employing the STFT method [1].
- Row 3 contains the results of the NFIS segmentation routine [96].
- Row 4 contains the segmentation results of the X-Signature technique [2].
- Row 5 contains the results of the proposed DCC algorithm with signature (Gaussian filter of 5 coefficients) and segmentation map (3x3 median filter) filtering.

In terms of segmentation quality, it was found that the STFT algorithm proposed in [1] was consistently the best across all four databases of the FVC2004 database set, proving that frequency is a highly discriminatory feature for fingerprint segmentation. Whereas the NFIS method [96] performed well on database 1 but struggled on databases 2, 3 and 4 due to the variation within the background of the images. However, the X-Signature [2] and the proposed DCC approaches produced similar segmentation results, although the X-Signature method performed slightly better. This was to be expected because the X-Signature indirectly incorporates directional information, which is a useful feature for fingerprint segmentation.

As mentioned previously, the objective of the proposed algorithm was to segment images with comparable quality to existing methods with minimal computational costs. It was determined that the proposed method segmented the entire FVC2004 set of databases with comparable quality as the X-Signature. However the proposed DCC method offers similar results with a significant reduction in execution times. These execution times are presented as box plots in Figure 2-14.

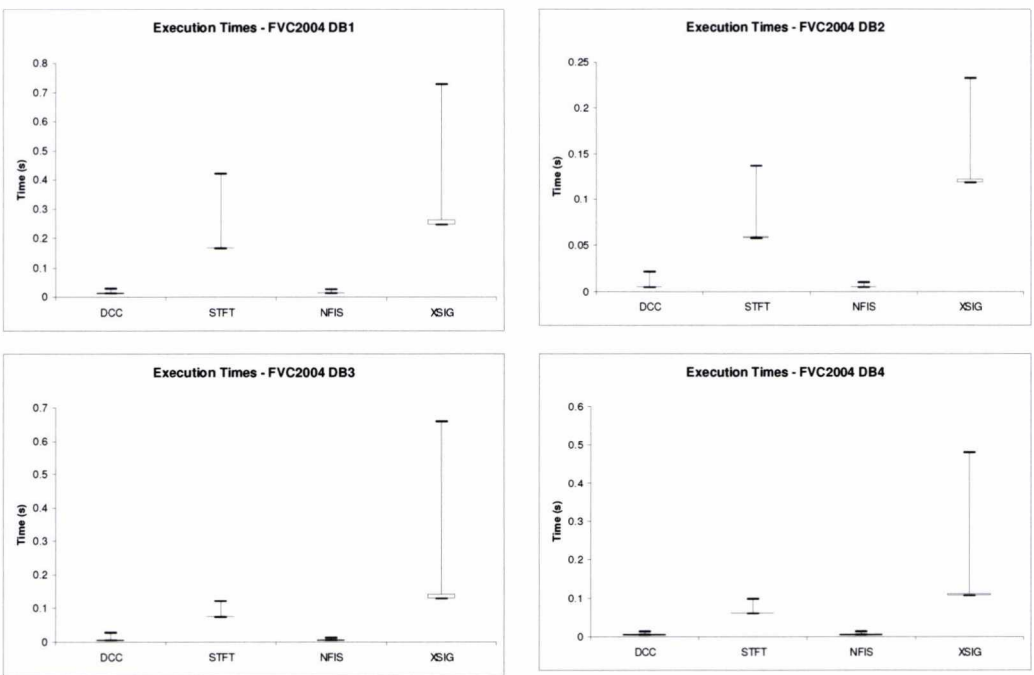


Figure 2-14: Execution Times for the FVC2004 Set of Databases

It is noticeable from the box plots that both the proposed DCC and the NFIS approaches are significantly faster than the other methods. However, the NFIS method in this experiment has been slightly biased to obtain results quickly with a cost that the segmentation quality is sub-optimal. The NFIS method in this experiment determines the best of four intermediate segmentation maps, which means that the threshold increases with a larger increment. This reduces the resolution of the threshold value, which prevents optimal segmentation. However, the number of intermediate maps was chosen so the algorithm could be executed at a comparable speed as the proposed DCC

method. This allows the segmentation quality of the proposed DCC method to be compared with another algorithm that can execute at similar speeds. In this case, the DCC method consistently provides better segmentation than the NFIS algorithm at a slightly faster average execution time. The average execution speed of each algorithm is presented in Table 2-2.

Table 2-2: Average Execution Times (sec)

	DB1	DB2	DB3	DB4
DCC	0.012466	0.005876	0.006074	0.004555
STFT	0.167135	0.062675	0.07647	0.06227
NFIS	0.013387	0.004952	0.006312	0.004986
XSIG	0.269678	0.129311	0.149378	0.115453

2.3.3. Evaluation Remarks

The proposed algorithm is a novel, fast and easy approach to fingerprint segmentation, which provides excellent results for average to high quality images with significantly reduced computational overhead compared to conventional well-known methods. However the proposed DCC technique produced inferior segmentation, with respect to the other methods, where regions of very low quality are present. Although, to some extent, the misclassifications resulting from low quality regions was also observed with the alternative techniques. Furthermore, the majority of very low quality images would be rejected in a fully implemented automated fingerprint system.

The results obtained from the proposed DCC method shows that the technique is suitable for a practical automated fingerprint system, which requires fast identification. Furthermore, it is also suitable for combinational / multi-stage segmentation algorithms as the implementation is quick and simple. In fact, the algorithm was originally devised as an alternative to the implementation of the X-Signature method proposed in A Novel Level-Independent Fingerprint Segmentation Algorithm Based on Fourier Energy. This is because the method does not rely on the relationship between background and foreground. In addition, as the algorithm is only implemented when the image is mono-level, the additional time imposed by implementing the proposed method is negligible.

2.4. A Novel Peak-Trough Detection Algorithm Based on Momentum

Peak-trough detection is an integral component of many signal and image processing algorithms. In fact, the subject of peak-trough detection has already been cursorily approached within the preceding chapters. Both of the proposed segmentation methods implement peak-trough detection in some guise. The proposed level-independent, Fourier-energy based segmentation algorithm

employed peak-trough detection to locate local maximums within the accumulator that results from the iterative threshold [102], whereas peak-trough detection was applied to the signatures obtained in the digital closed curve based segmentation method [103]. This is a necessary step in order to ascertain whether any sides of the digital closed curve exhibit *suitable structure*. In addition, it was applied to the X-Signature [2] in order to obtain a frequency and amplitude estimation of the projected ridges.

Peak-trough detection is essentially a utility function, which is used frequently in further algorithms. The method of peak-trough detection is often overlooked because it is only a supporting function within a larger algorithm. It is frequently taken for granted that the simple, traditional approach [104] performs well and is rarely substituted for a different approach, which may in fact improve performance. The traditional approach essentially detects sign changes in the first derivative of the signal (i.e. zero crossings). The basis of the traditional peak-trough detector is presented in Figure 2-12, although the method presented in Equation 2-6 breaks down where plateau regions are apparent. Therefore, the method can be extended to incorporate plateaux. An example of such an algorithm is presented in step 4 of the digital closed curve approach to fingerprint segmentation.

$$PT = ((X_i > X_{i+1}) AND (X_i > X_{i-1})) \otimes ((X_i < X_{i+1}) AND (X_i < X_{i-1})) \quad \textbf{2-6}$$

Where:

X_i is the value of the i^{th} sample of signal X.

PT is a boolean, which is true when a peak or trough is located.

The main drawback of the traditional method can also be viewed as a positive feature. The traditional approach detects all peaks and troughs, no matter how insignificant. Therefore, any slight fluctuation of the signal can result in a peak or trough being detected. The obvious benefit of this method is that no peaks or troughs will be neglected. However, it does not utilise any contextual information in order to neglect insignificant peaks or troughs. The pruning of the insignificant peaks or troughs can be employed as a post-processing stage. Heuristics and thresholds, determined a priori, can be applied to remove these insignificant detections. However, considerations must be taken in order to maintain the alternating pattern of peaks and troughs.

Alternatively, the signal can be filtered prior to peak-trough detection. This may require the signal to be buffered in order to retain samples before and after the focus sample. This will cause the peak-trough calculation to lag behind the signal by at least one sample. Furthermore, what is the best kernel size of the filter or how many smoothing iterations are required to sufficiently smooth the signal? Some signals may require more smoothing iterations than others to result in a suitable signal for peak-trough detection. Not all signals will undergo sufficient enhancement with constant filter parameters. In addition, too much smoothing can have a detrimental affect as useful features

can be filtered out too. By increasing either the filter kernel size or the number of smoothing iterations, the lag may increase and a larger buffer may be required. Even after signal enhancement, post-processing could still be required. All these things need to be considered in order to ensure that non-prominent peaks and troughs are discarded.

Alternative peak-trough detection methods do exist, although they are often application specific. For example Fisher & Naidu [105] examined five different techniques applied to finding the peak positions of a line or light strip. Many of these methods investigated required some sort of relationship or shape in order to produce the best results. Other approaches to peak-trough detection include employing thresholds and heuristics [106, 107]. However, all of these methods require a certain degree of prior knowledge about the application and therefore they would need to be adapted if they are to be implemented within a different application.

In conclusion, it is evident that most peak-trough detection algorithms perform best when their parameters are application specific, whether these include general shape of the peaks and troughs or thresholds (e.g. wavelength, amplitudes, deadbands, etc). Furthermore, post-processing maybe required in which heuristics or thresholds can be applied to remove insignificant peaks and troughs. To address these issues, a novel approach to peak-trough detection based on momentum is proposed. The main objective of the proposed method is to provide an accurate estimation of when a significant peak or trough occurs within a given signal, therefore reducing the dependency of pre- (signal smoothing) and post-processing (removal of insignificant peaks and troughs through thresholds and heuristics)

2.4.1. Novel Momentum-Based Peak-Trough Detection Algorithm

The original driver for devising an alternative peak-trough detection method was the need to terminate the algorithm after n significant peaks or troughs. This was because the algorithm was to be implemented into a larger algorithm. Evidently, if the traditional method was to be used, the entire signal needs to be analysed first to obtain all the peaks and troughs, then a post-processing stage would be needed to reject the insignificant ones. Although the algorithm was designed based on this primary purpose, the algorithm is sufficiently flexible to be implemented indefinitely.

The algorithm is based on the line-searching concept of a neural network learning algorithm [108]. In a neural network, a line-searching algorithm is often employed to find the optimal solution of a learning algorithm to adjust the weights of neurons in order to enable the network to converge on a solution. One approach is to find the global minima of the error rate of a learning algorithm (optimal solution) by utilising the momentum generated by the error rates of subsequent weight changes. This is because momentum can be viewed as contextual information as it is a measure representing the direction and rate of previous samples.

The basic premise of this approach is analogous to rolling a ball down a hill, whereby a basic implementation of Newtonian mechanics is used to determine the rate of descent and when the ball comes to a rest. As the rate of descent increases, the ball will gain momentum and, as the ball ascends, momentum is lost. Once the ball has lost all its momentum, the ball comes to a rest. The ball will come to rest either when a significant obstacle prevents the ball from continuing (i.e. global minima candidate) or when the ball loses momentum naturally through surface friction.

The line-searching algorithm would normally terminate at this point as a suspected global minimum has been located. However, this premature termination is not ideal for a peak-trough detection algorithm because the rest of the signal still remains and other peaks and troughs maybe located later in the signal. So how can the ball continue? Well, a signal must abide to an alternating pattern between peaks and troughs. Therefore, if a trough has been discovered, a peak must occur before another trough can be located. As troughs and peaks are defined as local minimum and local maximums respectively, if the samples following a trough are negated, the trough is now a peak and the samples will decrease until another local minimum is found, which will now be a peak. By continuously reflecting the signal with respect to the x-axis every time the ball comes to a rest, the entire signal can be traversed. A pictorial example of this reflection is present in Figure 2-15.

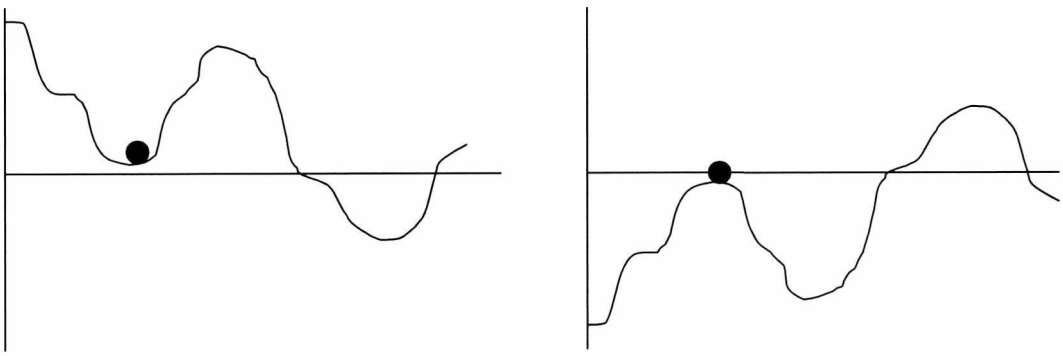


Figure 2-15: Signal Reflection

In Figure 2-15, the image on the left shows a snippet of a signal where the ball (i.e. the black circle) has come to a rest at a trough and the image on the right shows the same signal reflected in the x-axis. So the trough has now become a peak and the ball can continue to gather momentum until it comes to rest at the next trough, which is in fact a peak. The signal is then reflected in the x-axis again and the cycle repeats itself. By just simply reflecting the signal every time the ball comes to a rest, it is possible to locate peaks and troughs relatively well. However, few potential flaws are apparent.

The first potential flaw is that the basic model assumes that the implementation is frictionless. This causes a slight problem because the ball will not always lose all of its momentum in order to come to rest at a trough. In fact, if the following peak is not equal or greater than the peak at which the

ball currently resides, the ball will not come to a rest at the trough located somewhere between the points. Using Figure 2-15 as an example, the ball will not rest at the position indicated by the black circle because the following peak is not as high as the preceding peak. In order to overcome this problem, a friction parameter is incorporated into the algorithm to model the conservation of energy over flat surfaces and ascents. The addition of a friction parameter is based on the premise that, as a body descends, friction works against gravity, whereas a body ascends, friction assists gravity. An example of how friction and gravity act upon a body is presented in Figure 2-16, where F represents the direction of frictional force, G depicts the direction of gravity and M is the generated moment in that direction. For simplicity, the model proposed defines the mass of the ball as 1kg and gravity as $1ms^{-2}$.

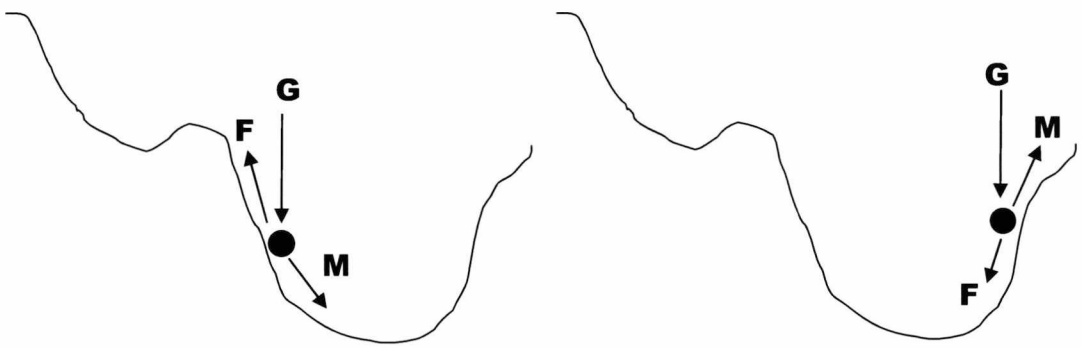


Figure 2-16: Forces Acting Upon the Ball

The second potential flaw is what occurs when the ball is at rest. This is essentially a question of overcoming static friction. This does not pose any problems if the ball is at a true peak because momentum will build up immediately. However, if the ball comes to rest on a plateaux or another flat, or flattish, region, then how does the ball overcome static friction?

There are two possible solutions. In reality, the ball will only move from a stationary position on a flat surface if an external force is acting upon it. This is because a body must overcome static friction in order to move. Therefore, in this situation, an initial momentum setting can be implemented in order to provide impetus to the ball. This is essentially modelling the occurrence of an external force. Alternatively, a threshold could be implemented, which prevents the algorithm from executing until the momentum has exceeded the threshold. Therefore, when the ball has come to a stop on a plateau, the algorithm will not execute until the ball is descending again. This is a preferred solution to the initial momentum setting because it prevents small peaks and troughs being detected when the ball comes to rest on a flattish surface, which would later need to be removed. The final algorithm is as follows.

1. Set momentum (m) and the reflect status (r) to 0. Set the friction parameter (μ) to a value between 0 and 1, where 0 indicates frictionless and 1 will prevent any motion. The

minimum momentum threshold (*mmt*) can be set to any value and is analogous to a filter. Set the interval (τ) to the between sample period, which is in this example 1.

2. Initiate either an infinite loop or a loop to interrogate a discrete number of samples.
   a. Read in a sample (X_t).
   b. if abs(m) is less than *mmt* and not the first sample
      i. $m = m + \left(\left| \frac{(X_t - X_{t-1})}{\tau} \right| \times (1 - \mu) \right)$
      ii. if $r = 0$

$$r = \begin{cases} 1, & if(X_{t-1} > X_t) \\ 2, & otherwise \end{cases}$$

   c. if abs(m) is greater than *mmt* and not the first sample
      i. $X_{PT} = \begin{cases} -X_t, & if(r = 2 \text{ and } -X_t <= X_{PT}) \\ X_t, & elseif(X_t <= X_{PT}) \end{cases}$
      ii. $t_{PT} = \begin{cases} t, & if(r = 2 \text{ and } -X_t <= X_{PT}) \\ t, & elseif(X_t <= X_{PT}) \end{cases}$
      iii. $v = \begin{cases} \frac{-(X_{t-1} - X_t)}{\tau}, & if(r = 2) \\ \frac{-(X_t - X_{t-1})}{\tau}, & otherwise \end{cases}$
      iv. $\mu' = \begin{cases} \mu, & if(v >= 0) \\ -\mu, & otherwise \end{cases}$
      v. $m = m + v(1 - \mu')$
      vi. if m is less than or equal to 0
         1. if $r = 2$
            a. $r = 1$
            b. t_{PT} is a peak
            c. $X_{PT} = X_t, \ if(X_t < X_{PT})$
         2. else
            a. $r = 2$
            b. t_{PT} is a trough
            c. $X_{PT} = \begin{cases} -X_t, & if(-X_t < -X_{PT}) \\ -X_{PT}, & otherwise \end{cases}$
         3. $m = \left| \frac{X_t - X_{PT}}{\tau} \right| \times (1 - \mu)$
   d. $X_{t-1} = X_t$

2.4.2. Experimental Results

In order to assess the performance of the proposed momentum based peak-trough detection algorithm, identical waveforms were fed into the proposed and traditional method and the resultant peak and trough locations were compared. Ideally, a better, quantitative method should be employed instead of the highly subjective qualitative method, which was implemented. However, a suitable measurement could not be determined because correct identification of peaks and troughs depends upon the application. Evidently, the traditional approach to peak-trough detection will always have a drop rate of zero, as it will locate every peak and trough. On the other hand, the traditional method may also have a lot of peak and troughs, which will need to be rejected during post-processing. It was decided that no post / pre-processing was to be implemented in order to illustrate the significant difference between the results. This ensures that both algorithms will only perform one pass and without buffering more than one sample. This is because the traditional method needs to buffer at least one sample in order to interrogate both the preceding and succeeding sample and the proposed approach needs to continually update the candidate trough point. In this application, the ideal scenario is that only prominent peaks and troughs are kept and the insignificant ones rejected.

Both algorithms were executed on a finite window even though both methods could be employed continuously. This was decided because a finite window would be more illustrative and is representative of the majority of situations a peak-trough detector may be employed within an automated fingerprint system. As many automated fingerprint systems employ the X-Signature algorithm, the signals presented to each algorithm were contained within a discrete window of 32 samples. This was due to the fact that the X-Signature produces a waveform within an oriented window containing 32 samples. The proposed method has been thoroughly tested on X-Signature-like waveforms and some example X-Signatures and their respective resultant peak-trough locations are provided in Figure 2-17. In Figure 2-17 the peaks and troughs are depicted as 1s and 2s respectively on the secondary scale.

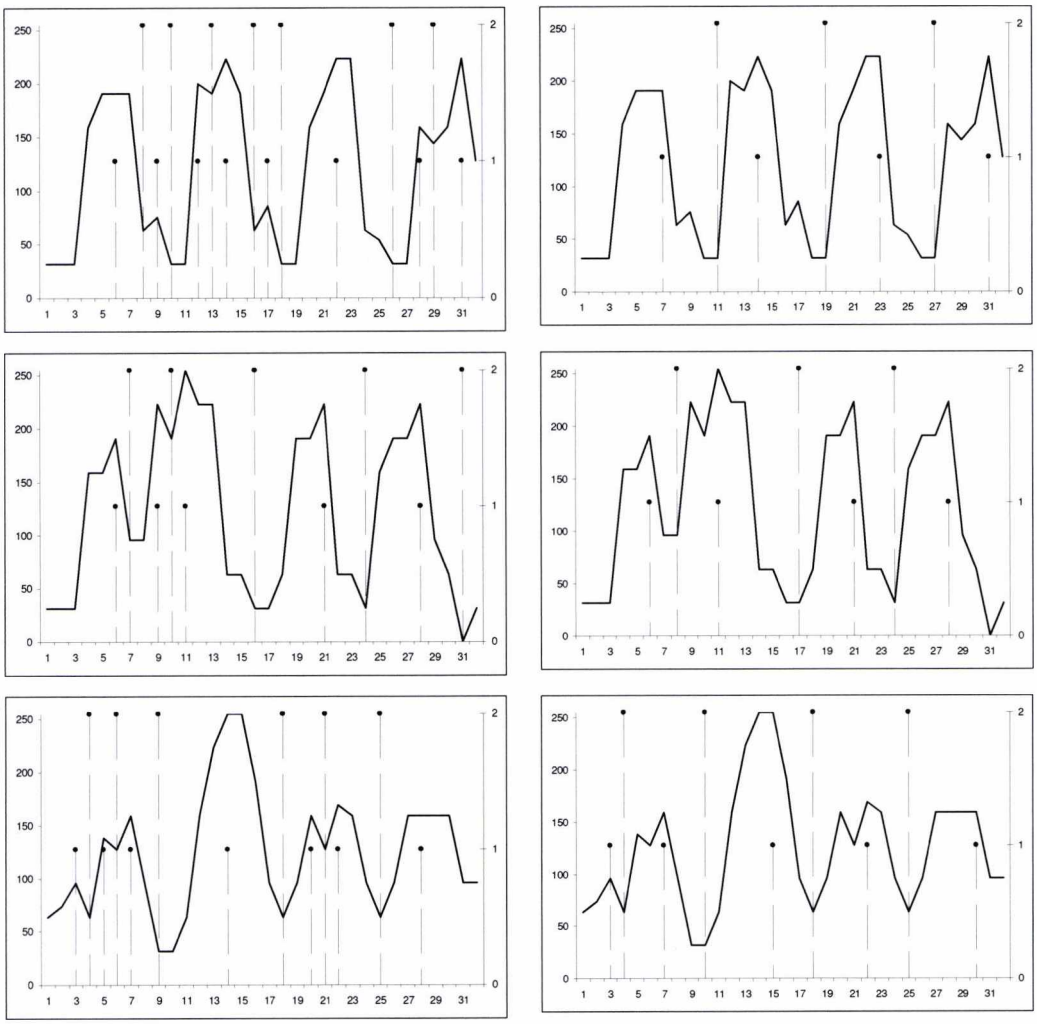


Figure 2-17: Example X-Signatures and their Detected Peaks and Troughs

The left and right columns of Figure 2-17 present the resultant peak-trough locations detected by the traditional and the proposed momentum-based approach respectively of 3 actual X-Signatures. It is clear that the traditional approach detects every peak and trough, which is not beneficial when an estimate of wavelength and amplitude is required. This is because there are many minor peaks and troughs which will adversely affect these estimations. However, the proposed method is able to bypass these insignificant peaks and troughs as the momentum generated is great enough to continue over these. Therefore, the subsequent estimations of the average wavelength and amplitude of the signal are more accurate.

The signals presented in Figure 2-17 are typical of X-Signatures, which very rarely have any prolonged periods of near uniformity. In Figure 2-18, a more complex signal is presented, which includes periods of near uniformity. It is evident from the left-hand graph that the traditional method picks up all the little fluctuations within the near-uniformity region. Conversely, the proposed method is able to disregard these because the momentum built up within these regions

does not exceed the initial threshold. Hence, the algorithm is not executing the peak-trough detection method during these periods, so only the prominent peaks and troughs remain.

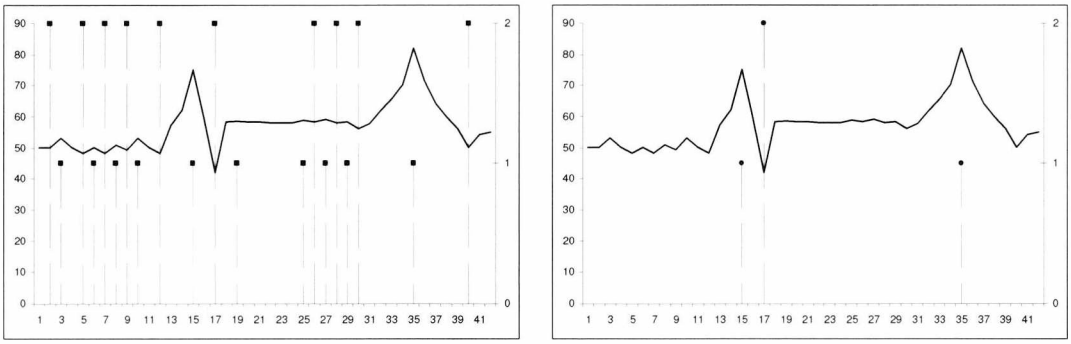


Figure 2-18: Resultant Peaks and Troughs of a Relatively Complex Waveform

Throughout these experiments, the friction parameter (μ) and the initial momentum threshold (*imt*) were kept constant at 0.3 and 3 respectively. Unfortunately, a method that could calculate these parameters automatically could not be established. In reality the friction coefficient is determined by slowly increasing the angle of descent and deriving the coefficient from the angle at which the body starts to move. This is analogous to a training stage where the parameter is determined a priori. However, a training stage is not ideal as signals can vary over time. Therefore, it is advantageous to view the friction parameter as being related to the rate at which momentum is lost.

A general rule-of-thumb was discovered during testing that can assist the estimation of the friction parameter. In the event that a wavelength within a signal (remember wavelengths can vary in a signal) is only 2 samples (i.e. peak (P_1) – trough (T_1) – peak (P_2)), then in order for the algorithm to detect the trough, the condition in Equation 2-7 must be satisfied. This is the worst case as the peak to peak distance is at a minimum so the momentum does not have time to reduce. Evidently, the required relative height of the subsequent peak reduces as the peak to peak distance increases. Therefore, an educated estimate of the friction parameter can be determined based on this knowledge.

$$P_2 >= (P_1 - T_1)(1 - \mu) \qquad \textbf{2-7}$$

As the proposed method does not recognise whether a peak or trough has occurred until the momentum is less than or equal to zero (backwards momentum), a decision is not made as promptly as the traditional method. The traditional method identifies a peak or trough one sample after the actual occurrence. However, it must be taken in to account that in order to retain meaningful peaks and troughs, pre- and/or post-processing stages may be required. This would require buffering the signal, which requires additional memory and will prolong the decision. Even though the proposed algorithm does not always identify peaks and troughs as promptly as the

traditional approach, it does provide a better judgement based upon contextual information (momentum) and the preset parameters.

Unfortunately, due to a certain degree of dependence on constant presets, it is possible that prominent peaks and troughs could be disregarded. For example, certain 1D Gabor functions may pose problems depending on how lax the presets are. If the friction parameter is set too low, the algorithm may ignore the latter half of the waveform. This is because the dominant peak may be considerably greater than the surrounding peaks (see left trace of Figure 2-19) and the rate that the momentum reduces would be too small to identify the troughs. A possible solution to this problem is to perform the proposed peak-trough detector in reverse (i.e. from last sample to first sample), as shown in the right trace of Figure 2-19, and combine the two results. This will make it a bi-directional approach rather than the current mono-directional implementation. Furthermore, by combining the two results, the plateau regions will now have a start and end location. Therefore the midpoint between these points can be taken as the peak or trough because the current mono-directional implementation will only identify the end of a plateau region. The drawback of this is addition is that it can only be implemented on a discrete number of samples, so buffering maybe required in a continuous application.

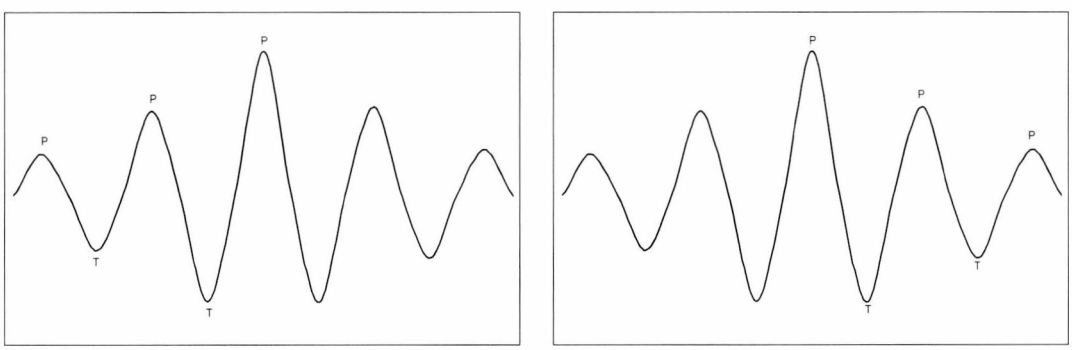


Figure 2-19: Example of Proposed Method on a Gabor-like Waveform

2.4.3. Evaluation Remarks

A conceptually simple alternative method to peak-trough detection has been proposed. The proposed momentum-based method has demonstrated its resilience to spurious peak-troughs, which are frequently present in most signals. Although, the experimental results concentrated on a discrete sample window, the algorithm can be easily converted from a discrete to a continuous implementation. This makes it flexible enough to be employed within any signal processing system.

Due to the contextual element momentum provides, many negligible peaks and troughs are disregarded. This removes the necessity of signal smoothing as the build up of momentum can be viewed as a dynamic smoothing parameter. However, depending on how noisy the input signal is, it

may be beneficial to implement a low-pass filter in conjunction with this algorithm in order to filter out the high frequency components. With the added controls of the friction and minimum momentum threshold parameters, the user can adapt or fine-tune the algorithm accordingly. Optimal selection of these parameters can eliminate the need for further post-processing.

Unfortunately, the objective of fully decoupling the peak-trough algorithm's dependency on prior knowledge was not completely attained. Ideally, the two preset parameters should either be well-defined or calculable rather than application-defined. However, this aspect is in need of further investigation because it is very complex determining which part of the signal constitutes noise and hence ascertaining the optimal input parameters to disregard the noise.

The proposed algorithm performs well for the implementation it was designed for, and it depends on the implementer to ascertain whether the algorithm is suitable for a specific application (i.e. to only retain prominent peaks and troughs). However, the algorithm is based on a very rudimentary model and not all forces are taken into account. Therefore, further evolution of this method could be carried out to fully remove the necessity of prior knowledge and to improve universality and functionality.

2.5. Fingerprint Orientation

Variations in the orientation of a fingerprint can adversely affect the performance of a fingerprint system. This is because the extracted features will not be identical between to acquisitions of the same fingerprint, which can increase the false rejection rate of a system. The general consensus for approaching this significant problem is to address the alignment issues at the matching stage. At the matching stage, a query fingerprint is compared with a fingerprint that is stored within a database to determine whether it matched. In the majority of cases, the fingerprint is stored in terms of its features in a template. Therefore, the matching algorithm would need to address the alignment issues otherwise the query template may not actually match with its respective stored template.

The simplest approach would be to generate a series of query templates by way of iteratively rotating the query image, which are then fed into the matching algorithm. In certain cases, the extracted feature vector can be transformed, which produces a feature vector that is equivalent to rotating the image. In either case, the rotated template with the largest matching score can be assumed to be the orientation closest to the stored images orientation. However, this approach is an exhaustive method, which requires additional computational resources and merely performs a rudimentary workaround to the problem rather than a legitimate method of accurate alignment. Alternatively, more complex methods, such as genetic algorithms, neural networks and classifiers can be implemented [109-116]. However, all these approaches require a matching stage in order to manage the rotational variation. In a template-free system, the matching stage is eliminated and is

consequently beyond the scope of this thesis. Therefore, appropriate methods of minimizing intra-sample variances at every stage of the template-free system is required in order to obtain feature values, which are highly stable, are required.

Unfortunately, only a small selection of fingerprint features can be classified as invariant. Invariant features generally refer to features that are invariant to scale, translation and rotation. The vast majority of features are susceptible to variance in at least one of the aforementioned categories, although most features are robust to most of theses categories to a certain degree. Scale and translation can be managed at the pre-processing stage, by the application of average inter-ridge distance (i.e. average frequency), reference point detection and segmentation algorithms. However, orientating a fingerprint at the pre-processing stage is an overlooked research topic. It is not surprising as it is by far easier, and potentially more effective, to address alignment at the matching stage where a query and a stored template can be directly compared. Furthermore, the vast majority of automated fingerprint systems available implement some derivation of a matching stage, so there is no urgency to manage alignment at the pre-processing stage. By orientating each fingerprint accordingly to a prescribed set of rules, it minimizes the effects of rotation prior to feature extraction, which reduces the variation exhibited by each feature.

Due to the elimination of a matching stage, the approach to fingerprint orientation becomes either a pre or post-processing problem. However, in many features, a transformation in the feature space that is equivalent to a rotation in the spatial domain could not be established. Therefore, in order to attain universality, it was decided that the fingerprint should be orientated at the pre-processing stage.

As mentioned previously, a simple approach to this would be to iteratively rotate the image and create a series of feature vectors. Unfortunately the drawbacks are still apparent, even in a template-free implementation. In addition to the extra timing and computational costs associated with producing numerous rotation-specific feature vectors, it is not feasible for a template-free biometric encryption system. In this implementation, an encryption key is required that is highly invariant. By extracting feature vectors for numerous orientations, multiple keys will be generated where one maybe the correct key, although it is not known which one.

In order to determine a consistently reproducible orientation of a fingerprint, certain features can provide information that can be employed to estimate the orientation, which is subsequently used to rotate the fingerprint so that its orientation is at 0^{o}. An estimation of orientation can be obtained by transforming a ROI, which is centred on a reference point (i.e. core singularity), into the Fourier domain, wherein the dominant direction can be extracted. However, it is more than likely that a clearly dominant orientation does not exist and, furthermore, it is also highly susceptible to noise because of the impact of noise on frequency. Therefore, the consistency of this approach is low.

Similarly, an estimation of fingerprint orientation can be obtained by first taking the second order central moments of a ROI centred on a reference point and then calculating the angle of the eigenvector associated with the dominant eigenvalue. This is also highly susceptible to noise as moments are statistical properties of an image, and the noise alters the images pattern that resides within the ROI.

It is also feasible to obtain an estimation of the orientation of the fingerprint by extracting the orientation field first and then generating a histogram of the quantized orientations lying within a ROI centred on a reference point. The modal value is the dominant direction and could be used as an estimation of the fingerprints orientation. Furthermore, the reflection symmetry of the orientation field of a ROI centred on a reference point can provide an estimation of the fingerprints orientation. This could be achieved by iteratively rotating an axis through the centre of the ROI and obtaining the sum of the difference between the orientations either sides of the axes. The orientation of the axis that results in the smallest sum is the orientation of the fingerprint. However, these methods are not universal. The modal value of the orientation field of a ROI would have problems with whorl and arch type fingerprints because there would not be a clearly dominant direction. Whereas the reflection-symmetry method would have difficulties consistently ascertaining the orientations of the whorl (if the whorl is almost perfect, i.e. tends towards being circular rather than ellipse) and double loop (if the two core singularities lie within the ROI) class fingerprints. This is because the symmetry values for all the axes would be similar.

The suggested ROI based methods clearly have many downfalls and have been rejected as viable fingerprint orientation methods. An ideal solution for fingerprint orientation is to view the problem as a function of the locations of singularities. Deriving an angle from the detected singularity locations is, in theory, a highly consistent approach. It has previously been identified that there is an equal number of delta and core singularities belonging to the vast majority of fingerprint. Therefore, the angle calculation could be similar to Equation 2-8. Unfortunately, more frequently than not, pairs of singularities do not reside in the acquired fingerprint region. Currently, techniques are not available that estimate the locations of the missing singularities, which prevents the employment of this approach.

$$\theta = \begin{cases} \arctan\left(\dfrac{y_{delta} - y_{core}}{x_{delta} - x_{core}}\right), & if\,(num\_pairs = 1) \\ \arctan\left(\dfrac{\left(\dfrac{y_{delta1} + y_{delta2}}{2}\right) - \left(\dfrac{y_{core1} + y_{core2}}{2}\right)}{\left(\dfrac{x_{delta1} + x_{delta2}}{2}\right) - \left(\dfrac{x_{core1} + x_{core2}}{2}\right)}\right), & if\,(num\_pairs = 2) \end{cases} \quad \textbf{2-8}$$

In order to provide an automatic estimate of the fingerprint orientation, the method proposed in [89] was implemented in further experiments where an automated orientation estimate was required. This method is implemented by first extracting the orientation field and estimating the primary core point (reference point), and then by analysing the orientation differences between the radial directions from the reference point and the local ridge orientations. A pictorial presentation is provided in Figure 2-20, although only 8 radial directions are used for this illustration, whereas in [89] 16 are implemented.

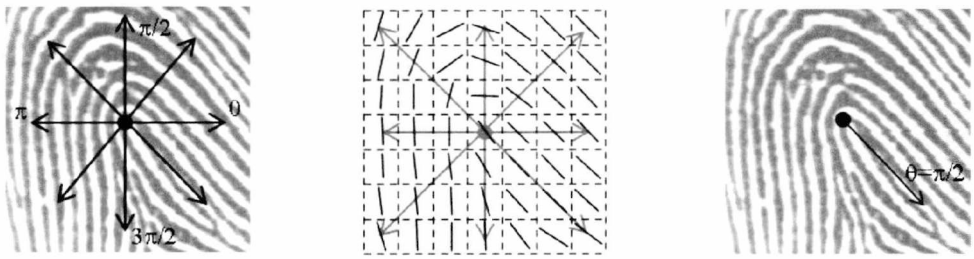


Figure 2-20: Fingerprint Orientation Estimation Example

In Figure 2-20, the left-hand image shows a reference point and the candidate radial directions and the centre image superimposes the radial directions on to the orientation field of the ROI. It is clear from the superimposed image, that the orientation that exhibits minimal deviation along its radial direction is $\pi/2$ radians (right-hand image). In order to achieve this orientation, an estimation of the orientation variation is determined by obtaining the mean of the absolute sine component of the orientation difference for each of the radial directions (see Equation 2-9). The radial direction with the minimum orientation variance is taken as the orientation of the fingerprint. However, the arch-type fingerprint is an exception to the rule, whereby two radial directions possess very small variations. In this case, the radial direction that is half-way between these directions is taken to be the fingerprint orientation.

$$V(k) = \frac{1}{N} \sum_{(i,j)\in\Omega_k} |\sin(\theta(i,j)) - \sin(\theta_k)|,$$

$$\theta_k = \frac{k\pi}{8}, \quad k = 0,1,...,15 \qquad \textbf{2-9}$$

Where:

$\theta(i, j)$ is the orientation within the orientation field at coordinates (i, j).

Ω_k is the set of orientations coordinates which lie along radial direction k.

Although this is the method employed for subsequent investigations, it does have certain drawbacks. The most important of which is that the reference point must be located by using an orientation field-based singularity detection method, such as the Poincare Index [6] or the sine-component map [24] algorithms. This is because the orientation estimation relies upon the structure

of the orientation field around these points. Alternative singularity detection approaches may detect a singularity at a different position where the orientation field does not exhibit the prerequisite structure. Another considerable disadvantage is that the algorithm does not perform well when the reference point is a whorl rather than a primary core and therefore all radial directions will exhibit large variations, limiting its universality. Unfortunately, not much research has been carried out with regards to fingerprint orientation as many systems employ a matching stage, which considers rotational variation. As the intended system is template-free (i.e. matcher-less), rotational variance needs to be suppressed to improve feature reproducibility. The pre-processing stage is ideal, although if multiple samples are to be used during normal operation of the system, then comparisons could be made to align the prints accordingly. However, if only one sample was to be used, which is the ideal scenario, the fingerprint would need to be rotated to a base orientation, prior to feature extraction, to improve the variant features reproducibility.

2.6. Conclusions

Many pre-processing methods have been highlighted in this chapter and many have also been implemented and tested. However, the methods which have been implemented did not always perform as claimed and the performance varied considerably from database to database. It was decided that if a concerted effort was made to commercialise the intended system, then the pre-processing algorithms must perform consistently for all types of fingerprint readers. Therefore an off the shelf software development kit (SDK) [59] was employed to enhance the image and the National Institute of Standards and Technology (NIST) Fingerprint Identification System (NFIS) [96] was used to provide quality estimation. As the intended application omits the template, no matching can be performed. Therefore, accurate pre-processing methods are essential in minimising intra-sample variance.

Chapter 3. Testing Framework

It is important for this research that the generated keys are stable. However fingerprint features can be subjected to large intra-sample variances which in conventional biometric systems are handled by matching templates of a query and an enrolled template. When templates are omitted, as in the desired application, it is essential to minimise these variances elsewhere. It is therefore important to identify suitable features, which exhibit the desirable properties of high inter-sample variance and low intra-sample variance. Hence, a significant part of the research has been dedicated to the selection and extraction of features from fingerprint images. As with the majority of pattern recognition applications, feature selection and extraction are recognized as the most crucial aspect of the system. Choosing the appropriate features for the task seriously affects the effectiveness of the system. Obviously, by using highly discriminatory features, a more accurate classification can be achieved whilst features which exhibit less discriminatory power would require sophisticated and complex classification techniques in order to obtain a satisfactory classification rate.

With this in mind, an extensive analysis of available features must be performed to assess the suitability of a feature to the system. By investigating features, the discriminatory strength can be obtained for individual features. This in itself is a major discipline. This will lead to informed judgements on the selection of features, which will in turn improve classification rates. The strength of an individual feature's discriminatory power can be defined by the inter-sample and intra-sample variances although, if multiple features are implemented, the correlations between these features are equally important. The most discriminatory features will exhibit little variance between feature vectors obtained from different samples of the same source. Conversely, feature vectors obtained from different sources should exhibit high variation. This scenario can be more eloquently expressed as a feature having very low intra-sample variance with a very high inter-sample variance. This type of behaviour in a feature is ideal, but for complex classification tasks can prove highly elusive. By analyzing these variances we can ascertain the information content of the feature and determine its quality.

In the existing literature, the features are presented within an entire system and the subsequent analysis is conducted on the system as a whole. The most popular representation of the performance of the system is the Receiver-Operating Characteristic (ROC) curve. The ROC curve's roots lie in signal processing (more specifically signal detection theory) and have migrated over to image processing. The ROC curve represents the sensitivity of a system, given a finite query dataset, by plotting the True Accept Rate (TAR) against the False Accept Rate (FAR). An example of a ROC curve is provided in Figure 3-1.

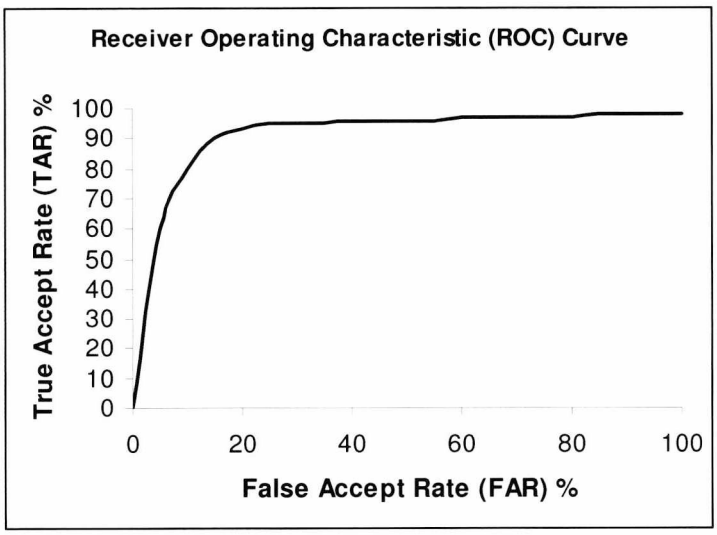


Figure 3-1: Example Receiver Operating Characteristic (ROC) Curve

The ROC curve provides a visual representation of the acceptance rate at a variety of decision thresholds. The *"elbow"* of the plot should ideally be located at (0, 100) (i.e. top left) resulting in a 100% TAR for all FAR. However in practice, this is near enough impossible to achieve and therefore the closer the *"elbow"* is to the top left of the plot, the better the performance of the system. The ROC curve is a representation of the overall performance of a recognition system. However it is not a great representation of the performance and variation of the feature itself. The ROC curve is an indirect measure of the feature's performance, whereby the accuracy of the system is an extension of how good the feature is. But it does not indicate the information content.

There are other, alternative representations of the overall system's performance in the literature. These include the Error Rate graph and the Genuine / Impostor chart. The Error Rate graph is akin to the ROC curve. The ROC curve plotted the FAR against the TAR, which were obtained using a variety of thresholds, whereas in the Error Rate graph the FAR and the False Reject Rate (FRR) are plotted as two separate entities over the threshold space. In other words, for a given threshold the error rates will be different and by plotting these error rates this way (see Figure 3-2), it is possible to obtain a suitable threshold depending on the error requirements of the system. The most commonly used statistic of this graph to depict the performance of the system is the Equal Error Rate (EER). This is simply the cross-over point of the graph where both the FAR and FRR are equal. The lower the percentage at which the two error rates cross implies better accuracy of the system.

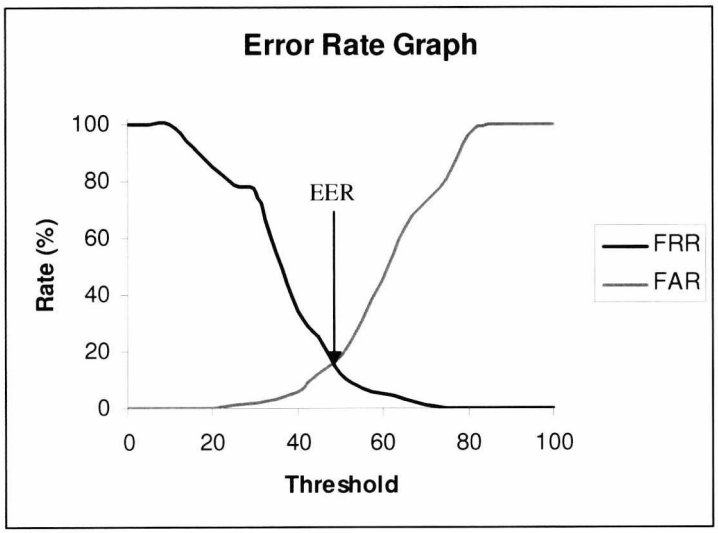


Figure 3-2: Example Error Rate Graph

The other common alternative is the Genuine / Impostor chart. This graph represents the distribution of the genuine and impostor samples over the normalized matching score space. An example is shown in Figure 3-3. If the genuine and impostor distributions are disjoint, then the system can correctly classify the samples in the dataset without error. However, this disjoining of the distributions is very rare in complex classification systems and therefore a certain degree of overlap is exhibited. From this plot, it is possible to estimate the classification rates (FAR, FRR, TAR, TRR – see Figure 3-3), which are obtained from varying the threshold of the normalized score. In this representation, the graph is independent of the threshold and therefore a useful tool to determine a suitable threshold based on the requirements of a system.

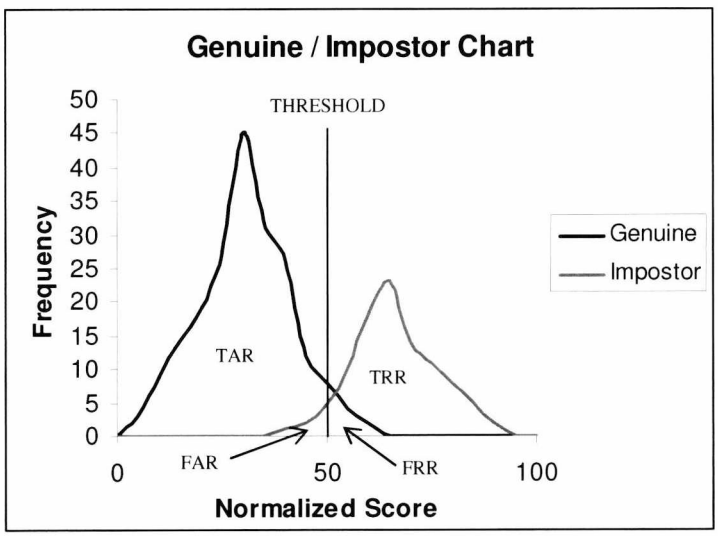


Figure 3-3: Example Genuine / Impostor Chart

Unfortunately, all the above representations rely on a template comparison stage. It is possible that, with sufficiently discriminatory feature vectors, the template matching stage is very intuitive and simple. This could be a simple Euclidean distance measure and a threshold. In this case, the

aforementioned performance characteristics may accurately portray the variability of the features. However in the majority of the fingerprint systems analysed, the template comparison stages often utilise sophisticated and complex matching techniques in order to accommodate the fuzzy nature of the fingerprint feature vectors and improve the performance of the system.

Many intelligent matchers implement classifiers, artificial neural networks (ANN) or genetic / memetic algorithms (GA / MA) in order to obtain a decision [110-112, 117]. The main drawback of using classifiers and ANNs is that a training stage is required. The training stage is a controlled calibration phase, whereby specific input feature vectors are fed into the matching / classification algorithm along with the desired result and algorithm adapts based on the difference between the desired and obtained result. In an ANN, the weighting of each of the nodes is altered, according to a specific learning algorithm, in order to converge on the desired result from the given feature vectors. Similarly, a classifier must be trained with the purpose of determining probabilities, or mappings, from the training set to assist in future decision-making. Furthermore, the resulting configuration must either be stored or implemented in software so training is not required every time the system restarts. Depending on the number of features and the ANN or classifier employed, this configuration could be quite large.

GAs / MAs approach fingerprint matching differently. GAs / MAs are commonly used to search irregular search-spaces for optimization purposes. With regard to fingerprint matching, these algorithms are often used to determine the optimum transformation between two data sets (query and template), which establishes how similar the data sets are. Therefore, what ultimately determines the matching outcome is a transformation that is initially created randomly, and then stochastically evolved until the transformation fulfils the fitness function criteria or exceeds the maximum number of generations. Hence, the information content cannot be quantified as the GA / MA does not relate to the actual fingerprint feature values but the transformation between a pair of feature vectors.

Determining the performance of features by employing them in a classification system is generally categorized as a "*wrapper*" approach to feature selection. The "*wrapper*" approach to feature selection essentially classifies a subset of features and results in a score, which is used to determine the subset's discriminative power [118]. However, this discriminative power alters with different methods of classification, which makes the discriminative power dependent on the classification technique employed. Seeing as there are many classification methods, it is not feasible to test every one. An alternative approach to feature selection is to rank each feature according to the strength of a discriminating measure, which is relatively independent of the classification method employed. This approach is classed as a "*filter*" method.

3.1. Discriminating Criterion

The "*filter*" methodology to feature selection approaches the multiclass problem as one rather than decomposing the multiclass problem into a series of binary-class problems. Although viewing the multiclass problem as multiple binary-class problems could be implemented in practice, it is not an ideal methodology for compiling a feature catalogue. When the multiclass problem is decomposed into multiple binary-class problems, different features are used to discriminatory between different classes. However, this implies the use of binary classifiers, which could be simply a decision plane, whereby overall classification is obtained by the composition of the outputs of these binary classifiers. It does not give any indication of the universality of individual features. For successful implementation in a cryptosystem, the key generation technique should have the means to accommodate unknown users, so universality is paramount.

There are several methods that select features based on a discriminating measure. These include correlation coefficients, Fisher's discriminate criterion, mutual information and statistical tests, such as the T-, F- and Chi-Squared tests [119]. For the feature catalogue, a number of discriminating measures were employed in order to account for the varying dimensionality of feature vectors belonging to different feature sets. For univariate feature sets, the F-Test [119] (obtained from the analysis of variance – ANOVA) is employed, whereas for multivariate feature sets, the Wilks' Lambda test statistic [120] (obtained from the multivariate analysis of variance – MANOVA) was employed as this can be perceived as a multidimensional F-Test. Furthermore, a statistic derived from the Mahalanobis distances is used to provide a dimension-independent, distance measure. It is important to note that all the extracted feature sets are assumed to be either univariate or multivariate Gaussian distributed (i.e. normally distributed). This assumption was based on the fact that there are only a small set of observations (at the most 8 impressions per fingerprint) and that there was no evidence to denounce the presumption that the features are not Gaussian (univariate or multivariate) distributed. Furthermore, many naturally occurring phenomena exhibit Gaussian distributions so it was a natural assumption.

3.1.1. F-Test

As mentioned previously, the desired characteristic of a feature is that the inter-sample variance should be very high, whereas the intra-sample variance should be very small. The F-Test is a variance ratio test, which tests the hypothesis that the feature vectors have equal variances. It provides a ratio between the intra-sample variance and the inter-sample variance, whereby a larger value indicates that the feature exhibits the desired characteristic. This was the primary reason why the F-Test was chosen over the alternative statistical hypothesis tests. Furthermore, the F-Test is well suited to multiple groups with multiple samples, which are assumed to follow the normal distribution (as in the subsequent experimentations), and is simple to implement. However, the F-

Test does not indicate the disjointedness of the distributions. The F-Test statistic is calculated by implementing Equations 3-1 – 3-3.

$$F = \frac{MS_A}{MS_E} \tag{3-1}$$

$$MS_A = \frac{SS_A}{df_A},\ MS_E = \frac{SS_E}{df_E}$$

$$df_A = N - 1,\ df_E = n(N-1) \tag{3-2}$$

$$SS_A = \frac{\sum_{i=0}^{i<N}\left(\sum_{j=0}^{j<n} X_{i,j}\right)^2}{n} - \frac{\left(\sum_{i=0}^{i<N}\left(\sum_{j=0}^{j<n} X_{i,j}\right)\right)^2}{nN}$$

$$SS_E = \left(\sum_{i=0}^{i<N}\left(\frac{\sum_{j=0}^{j<n} X_{i,j}}{n}\right)^2\right) - \frac{\sum_{i=0}^{i<N}\left(\sum_{j=0}^{j<n} X_{i,j}\right)^2}{n} \tag{3-3}$$

Where:

F is the F-test statistic.

$X_{i,j}$ is the feature value of the j^{th} observation from class i.

N is total number of classes.

n is the number of observations per class.

3.1.2. Wilks' Lambda

Wilks' lambda is a general statistical hypothesis test which is used in multivariate analysis of variance. The Wilks' lambda statistic is often calculated for differing numbers of dimensions to determine the likelihood (*p-value*) that the means lay within smaller subspaces. In order to determine the likelihood of the means lying within a subspace, a threshold is implemented to retain the subspaces that exhibit significant variation between groups. This significance level threshold is commonly set to 5% (i.e. 0.05) as the closer lambda is to zero, the better separated the groups are. Any lambda value below this value is judged to possess significant difference between at least two groups. Therefore, suitable features should exhibit significant subspaces of greater dimensionality as subspaces of higher dimensions should exhibit better group separation. Wilks' lambda can be calculated by implementing Equation 3-4.

The Wilks' lambda test is arguably the most popular hypothesis test for the multivariate feature selection. Wilks' lambda is very similar to that of the F-Test, so therefore is well suited to this implementation, especially as the features are assumed to follow a multivariate Gaussian distribution. Other multivariate tests, such as Pillai-Bartlett traces, Lawley-Hotelling traces and Roy's Greatest Root [120] could be implemented instead of Wilks' lambda test. All these tests

implement the same eigenvalues as Wilks' lambda, except Roy's Greatest Root is simply the greatest eigenvalue. In practice, these statistics perform similarly [121] and is recommended that all statistics together should be considered in unison as not one of them is consistently the best. However, ultimately a decision must be made to which statistic to abide by if they are not in agreement. This could be achieved by a voting algorithm derived from ranking the feature sets according to the statistical test used or a weighted sum, although it was decided that the Wilks' lambda of the largest significant subspace of a feature vector was to be used. This is because one of the objectives of this research is to determine stable, but also very long keys and a feature that, in theory, possesses the greatest effective dimensionality could potentially increase the number of effective bits and overall key length. Although Wilks' lambda was ultimately used for ranking the features, all the aforementioned multivariate tests are presented in Equations 3-4 – 3-7 for completeness.

$$LAMBDA = \prod_{i=0}^{i<d} \frac{1}{1+\lambda_i} \quad \textbf{3-4}$$

$$HOTEL = \sum_{i=0}^{i<d} \lambda_i \quad \textbf{3-5}$$

$$PILLAI = \sum_{i=0}^{i<d} \frac{\lambda_i}{1+\lambda_i} \quad \textbf{3-6}$$

$$ROY = \max(\lambda_i) \quad \textbf{3-7}$$

Where:

LAMBDA is Wilks' lambda.

HOTEL is the Hotelling-Lawley trace.

PILLAI is the Pillai-Bartlett trace.

ROY is Roy's greatest root.

λ_i is the i^{th} eigenvalue of A, where $A = W^{-1}B$ and W and B are the within-groups and between-groups sum of squares and cross-products matrix respectively.

d is the dimensionality of the subspace being examined, most commonly equals the number of eigenvalues.

3.1.3. Mahalanobis Distance Ratio (M-Ratio)

The Mahalanobis distance is an often-used distance measure which is employed for feature selection. The Mahalanobis distance provides a scale invariant distance measure that determines the distance of an unknown feature vector from that of a known one taking into account the underlying correlations of the already known feature sets. This distance measurement is essentially the Euclidean distance weighted by the inverse covariance matrix. By ascertaining the inter-mean Mahalanobis distance, it is possible to determine how well spread out the means are. The larger the

average inter-mean Mahalanobis distance indicates that there is significant distance between the means, which is a desired property. Unfortunately, this does not translate well when comparing feature vectors of varying dimensions. This is because there is a direct relationship between dimensionality and distance, i.e. that with greater dimensionality, it is expected that the distance will also increase. Therefore, it is not possible to directly compare a feature vector which only has 7 dimensions and that of a feature vector that has 32 dimensions.

This is also the case with the Wilks' lambda test. With features of high dimensionality, the primary lambda value (i.e. the lambda value that is calculated from all eigenvalues, not a subset) can be so close to zero that it becomes indistinguishable from other large dimensional feature vectors. This is the reason why the largest significant subset is used for comparing feature vectors. However, this breaks down when comparing feature vectors of varying dimensionality as a very high dimensional feature vector should have a significant subset greater than a lower dimensional vector. Therefore, Wilks' lambda, in this implementation, is primarily employed to compare feature vectors of equal dimensionality. In order to compare feature vectors of varying dimensionality, a Mahalanobis distance ratio is proposed.

As mentioned previously, the desired property of a feature is that the intra-sample variance is significantly smaller than the inter-sample variance. This can be roughly translated to that the average distance from a feature vector from a particular class to its class's centroid should be significantly smaller than the inter-mean distance. This is based on the ideal scenario where the feature vectors from each class are compact about the class's mean. This, in practice, is generally not the case as outliers often lie far from their class's mean, which causes them to be misclassified in a classification system. However, the ratio between the average inter-mean and the average intra-class (i.e. distance from a feature vector's class mean) Mahalanobis distances provides a useful dimension-independent discriminative criterion for feature selection. As both distances are derived from vectors from the same dimensionality, dividing one by the other should cancel out the dimensional dependency. The Mahalanobis distance ratio is presented in Equation 3-8. It should be noted that any average (i.e. mean, mode or median) can be used for the average inter-mean and average intra-class distance, although in this implementation the median was used to minimize the effects of the outliers.

$$MRatio = \frac{imd}{icd} \qquad \textbf{3-8}$$

$$imd = average\left(\sqrt{(\mu_i - \mu_j)^T C_\mu^{-1} (\mu_i - \mu_j)}\right) \quad i, j \in (0, N-1];\ i \neq j \qquad \textbf{3-9}$$

$$icd = average\left(\sqrt{(X_j - \mu_i)^T C_i^{-1} (X_j - \mu_i)}\right) \quad \nabla 0,..., j,..., S-1;\ j \in i \qquad \textbf{3-10}$$

Where:

average(X) is any one of the averages (i.e. mean, median or mode)

imd and *icd* are the average inter-mean and intra-class Mahalanobis distances respectively.

μ_i is the mean vector of class *i*.

C_μ and C_i are the covariance matrices of the means and the i^{th} class respectively.

X_j is the j^{th} feature vector, which belongs to class *i*.

MRatio is the Mahalanobis ratio (M-Ratio).

N and *S* are the total number of classes and the total number of samples respectively.

3.1.4. Directional Data Distance Ratio (D-Ratio)

Fingerprints are often represented by their orientation fields [122], which contain directional data. Feature vectors that consist purely of directional data cannot be analysed in the same way as non-directional data because directional data is circular (i.e. values that "wrap around" similarly to modular, or clock, arithmetic). The previously mentioned statistical tests assume that the data is continuous and therefore the means and covariances can be calculated using the standard methods. This is not the case with directional data so alternative methods have to be applied. When the feature vector consists only of directional data a test statistic that is similar to the Mahalanobis distance ratio (M-Ratio) is obtained. This statistic is essentially the ratio between the average Euclidean distance between mean vectors of different classes and the average Euclidean distance between the sample vectors and the mean vectors of their corresponding class. This was determined by transforming the directional data to a set of 2D coordinates, which was obtained by taking the cosine and sine of each orientation. The distance ratio can be obtained by implementing Equation 3-11.

$$DRatio = \frac{average(B)}{average(W)} \qquad \textbf{3-11}$$

$$B = \sum_{l=0}^{l<L} \left(\left(\sin\theta_{i,l} - \sin\theta_{j,l} \right)^2 + \left(\cos\theta_{i,l} - \cos\theta_{j,l} \right)^2 \right); \qquad \textbf{3-12}$$

$$i, j \in (0, N-1]$$

$$i \neq j;\ C(i) \neq C(j)$$

$$W = \frac{\sum_{\substack{i,j \in (0,N-1] \\ i \neq j;\ C(i)=C(j)=c}} \left(\sum_{l=0}^{l<L} \left(\left(\sin\theta_{i,l} - \sin\theta_{j,l} \right)^2 + \left(\cos\theta_{i,l} - \cos\theta_{j,l} \right)^2 \right) \right)}{n_c} \qquad \textbf{3-13}$$

Where:

DRatio is the distance ratio (D-Ratio)

average(X) is any one of the averages (i.e. mean, median or mode)

c is the class number (for 0,…,c,…,C_{tot}-1.)

N is the total number of feature vectors.

L is the length of the feature vector.

3.1.5. Summary

As mentioned previously, all features are assumed to follow either a univariate or multivariate Gaussian distribution. Therefore, the statistical tests chosen are, in general, well suited to these types of distributions. Other statistical tests could be implemented, although many factors attributed to their rejection. Some of these factors were that the tests do not perform well on multi-sample and / or multivariate feature sets, provided little, if any, additional benefits than the outlined tests or that they were just not suited to the Gaussian distribution assumption. It is possible that this assumption is incorrect. However, because the test databases contains fewer than 8 samples per class, it is not possible to confidently ascertain their true distributions, which is why it was considered that the Gaussian distribution is the most likely. Furthermore, it is very important to note that the results of these statistical tests will almost certainly be derived from unbalanced data.

The objective of the subsequent feature catalogue is to independently assess different feature vectors that can be extracted from a fingerprint image. Therefore it is important to highlight that the resultant statistical measures do not take into account any underlying correlations between different features. At this stage, where individual features are being assessed, identifying correlations, and the subsequent decorrelation of the correlated features, is not of great concern. However, when the features are to be combined in order to create an encryption key, it is highly recommended that the combined feature set should be analysed for correlations. Correlated data, due to the underlying relationship between features, potentially presents a weakness in the key generation stage because certain assumptions and approximations based on the relationships can be made to reduce the

search-space in which the key can be attacked. These aspects will be expanded in the Biometric Encryption chapter.

3.2. Common Tests

There are many tests, which can be applied to all fingerprint features that can affect the reproducibility of a feature vector. The image that is subsequently input into the feature extraction stage can undergo a number of prior manipulations. Prior to feature extraction, the image can be screened and / or pre-processed and the placement of regions of interest (ROI) can be different depending on the scheme employed. Therefore, it is important to investigate the effect that different input images / ROIs to the feature extractor have on the reproducibility of the resultant feature vectors.

3.2.1. Reference Point Detection and Fingerprint Rotation

Image-based feature extractors commonly employ region of interests (ROIs) that attempt to provide consistent reproduction of an area of the fingerprint. This is to minimize the adverse affects attributed to translational and scale variance, and inconsistent region overlap. Therefore it is vital that the ROI is consistently located at the same, relative position within the fingerprint. If the placement of the ROI varies between samples of the same fingerprint, then the features extracted from the region will be completely different as they refer to different regions. To facilitate consistent ROI placement, a reference point (i.e. landmark point that can be reliably located in all samples) is required to centre the ROI about.

There are many approaches to reference point detection, which are investigated in greater detail in the chapter on Level 1 Features, which can be used to centre the ROI. The reference point is commonly the primary core or whorl singularity. In most types of fingerprints, the reference point can be determined with a relatively high degree of accuracy (depending on the quality of the fingerprint image). However singularities do not exist in the arch fingerprint type, which seriously affects consistent ROI placement. Therefore in the subsequent experiments, the features will be extracted using a reference point detection algorithm (i.e. the difference in geometric regions of the sine component map method proposed in [24] and a reference point detected manually. It is expected that a significant boost in consistent feature extraction will be exhibited when the reference point is detected manually.

Consistent placement of a ROI is crucial in order to extract the same relative region every time. However, even with consistent region extraction, the feature vector may vary considerably on account of rotational variance. Rotational invariant features, such as frequency components invariant moments, can be implemented if there is only one ROI, although it is common for feature

extractors to partition the ROI into composite ROIs. The variety of sub-ROIs positioning is covered later in this chapter. However, the further partitioning of a ROI makes rotational invariant features susceptible to rotation because the sub-ROIs may not be consistently placed at the same relative location. Even if the feature vector is rotationally invariant, it is futile if the feature is extracted from different regions for every sample. Therefore, in addition to reference point location, the reference point orientation is also extracted as this will act as the approximate orientation of the overall fingerprint. To estimate the orientation of a reference point automatically, the technique outlined in [89] was employed. This method was chosen because the orientation is estimated from the orientation field that can then be reused to generate the feature vectors, which saves computational resources.

In order to investigate the affect inconsistent ROI placement has on the consistency of the extracted feature vectors, the worst case and the best case scenarios are investigated. The worst case scenario is where both the reference point location and orientation are obtained algorithmically, which is denoted as AUTO in further references. The best case scenario is where both the reference point location and orientation are estimated manually as this is far more consistent than the automatic approaches. The manual detection of the reference point and its orientation is referred to in future as MAN.

3.2.2. Region of Interest (ROI) Placements

In order to achieve consistent feature vectors between different samples of the same fingerprint image, feature extractors often restrict extraction to a sub-region of the image. This sub-region is coined as the region of interest (ROI) and by placing ROIs about the same relative position, ensures consistent overlap between different samples of the same fingerprint. The previous section approached the subject of detecting the same relative location and orientation of a reference point for which the ROI can be centred about. The reference point location and orientation essentially provides a rotational and translational invariant alternative origin, whereby further ROIs can be consistently placed by employing the polar coordinate system. However, instead of the origin being (0, 0) and the normal being 0^{o}, the origin will be the coordinates of the reference point and the origin its orientation. Using the polar coordinate system, any ROI can be consistently placed depending on the consistency of the relative location and orientation of the reference point. In this section, the different placement of ROIs employed by the feature extractors are presented and tagged to prevent repetition and to facilitate easy lookup of the ROI used in subsequent feature extractions.

The simplest ROI is the entire image, which will be tagged as FULL for future reference, although this provides minimal robustness to partial overlapping between samples of the same fingerprint. The FULL ROI allows the feature extractor to extract features on a global scale. An alternative

global ROI is a single, large ROI centred about the reference point. The shape of the ROI can be either a quadrangle or ellipse, although only two shapes are commonly used, which are the square and the circle (denoted by GBLSQ and GBLC respectively). These two ROI placements are demonstrated in Figure 3-4. For indication of the dimensions of the ROI, the radius value is used. Therefore, the width and height of GBLSQ is double the radius.

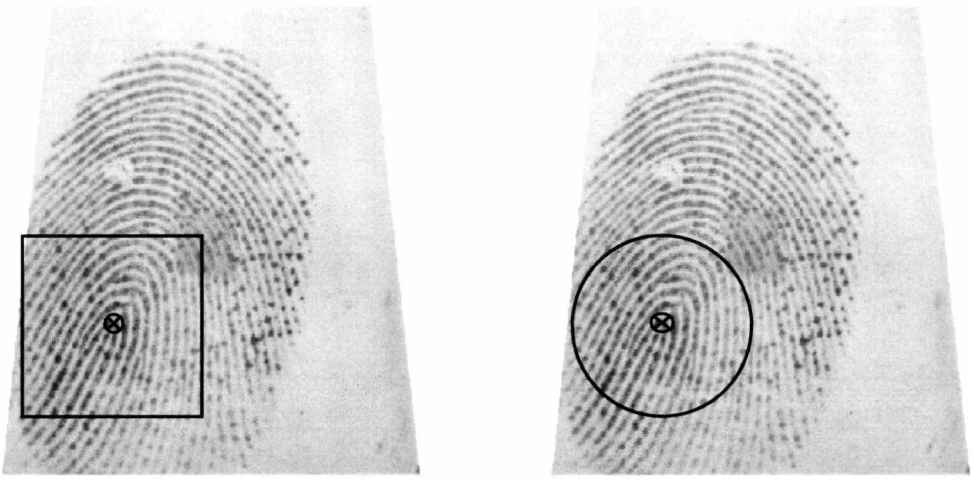


Figure 3-4: Example of global ROIs. GBLSQ (L) & GBLC (R)

The next form of ROI placement is based on a grid system, whereby either a circular ROI or square ROI is centred on the centre of each cell within a grid centred on the detected reference point. Therefore a larger ROI is being partitioned into composite ROIs (sub-ROIs), which fit into the individual cells of a grid. In Figure 3-5, square sub-ROIs (GRIDSQ) and circle sub-ROIs (GRIDC) are positioned within a grid system. This allows local neighbourhoods to be extracted independently from others.

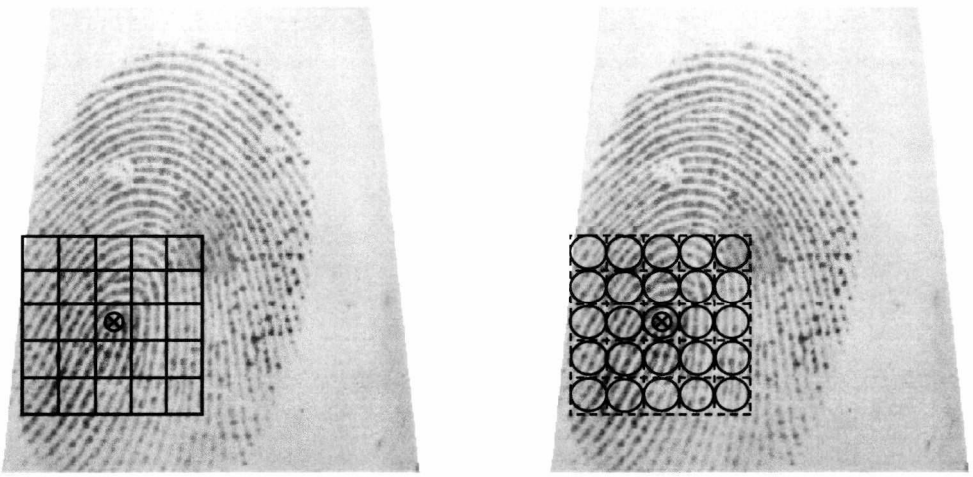


Figure 3-5: Grid Based ROI Layouts. GRIDSQ (L) and GRIDC (R)

The final two ROI placement systems are based upon the polar coordinate system in order to improve rotational robustness. However, if the reference point location and orientation are highly

consistent, the grid-based systems should perform similarly if the image is rotated accordingly prior to feature extraction. The ROI placement schemes presented in Figure 3-6, extract different shaped ROIs. One is tessellated like a Compact Disc (CD), whereby the sub-ROIs are shaped similarly to the blocks of a CD and the other places circular sub-ROIs radially, at predefined distances from the reference point, along a variety of orientations.

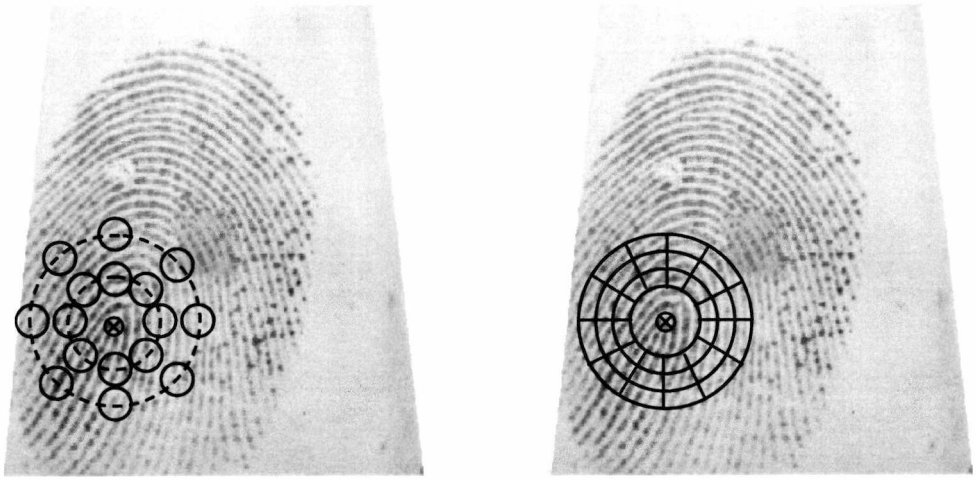


Figure 3-6: Polar Coordinate System Based ROI Schemes. RAD (L) and DISC (r)

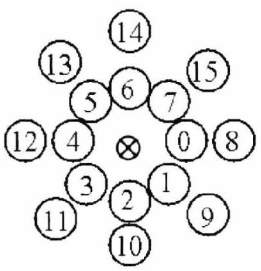
The ROI placement schemes shown in Figure 3-6 are interrelated. The left image of Figure 3-6 shows a simple radial scheme (henceforth denoted as RAD), whereby the sub-ROIs are placed equiangular along concentric circles radiating from the reference point. In the example presented, there are 8 directions and 2 concentric circle (denoted by the dashed circles), giving 16 circular sub-ROIs. This is similar in layout to the tessellated ROI scheme (henceforth denoted as DISC) in the right-hand image. However, the circular sub-ROIs in RAD are essentially centred on the 2D vertices of the CD block-like sub-ROIs of DISC. Therefore, the two schemes can be constructed similarly to that of a CD, whereby a spindle radius (i.e. the distance from the reference point to the closest concentric circle.), a track width (distance between concentric circles), the number of tracks and number of sectors can produce both schemes.

In some cases, the fingerprint image is transformed into a different domain or space. Orientation and frequency fields are perceived as different spaces or domains as they are an alternative representation of a fingerprint. As these fields are often computed for additional purposes within a fingerprint system (commonly in pre-processing methods) it would be beneficial to reuse the pre-existing fields, rather than recreating them at the feature extraction stage. Therefore, the centre coordinates of each sub-ROI can be used as an index into a pre-existing field to extract the value.

3.2.3. ROI Indexing and Feature Vector Construction

Indexing the individual sub-ROIs of a ROI placement scheme is an important part of constructing a feature vector. It facilitates the collation of the feature values in to an intuitive, organized feature vector. Figure 3-7 demonstrates the indexing of the sub-ROIs.

0	1	2	3	4
5	6	7	8	9
10	11	12	13	14
15	16	17	18	19
20	21	22	23	24



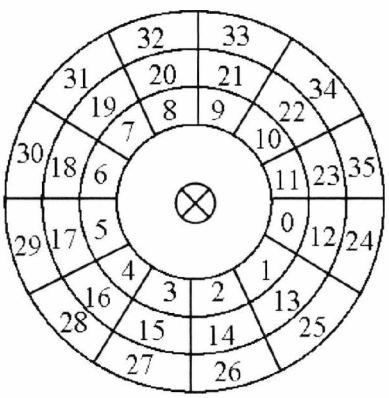


Figure 3-7: ROI Indexing. GRID (L), RAD (M), DISC (R)

The indexes of the sub-ROIs obviously alter when the number of rows / columns (in GRIDSQ and GRIDC) or the number of tracks / sectors (in RAD and DISC) are different. The examples provided in Figure 3-7 simply demonstrate the general organisational patterns and can be applied to all variations of the corresponding ROI placement scheme. In order to compile the feature vector, all the extracted feature values from each sub-ROI are placed in the vector according to its index. The final feature vector is constructed by implementing the ordering presented in Equation 3-15.

$$V = \begin{bmatrix} \begin{bmatrix} v_{0,0} & \cdots & v_{0,l} & \cdots & v_{0,L-1} \end{bmatrix}^T \\ \vdots \\ \begin{bmatrix} v_{i,0} & \cdots & v_{i,l} & \cdots & v_{i,L-1} \end{bmatrix}^T \\ \vdots \\ \begin{bmatrix} v_{N-1,0} & \cdots & v_{N-1,l} & \cdots & v_{N-1,L-1} \end{bmatrix}^T \end{bmatrix}; \quad \nabla l = (0, N-1], j = (0, L-1] \qquad \textbf{3-15}$$

Where:

V is the final feature vector

N is the total number of sub-ROIs.

L is the number of feature values extracted from each sub-ROI.

v is the l^{th} feature value of the i^{th} sub-ROI.

3.2.4. Pre-Processing

In addition to ROI placement schemes, features can be extracted from images that have undergone a variety of states of pre-processing. The ROI schemes are mechanisms that ensure consistent overlapping, although this is just a region within the fingerprint. This region could be enhanced,

rotated, normalized, etc prior to extracting the region's features. When features are extracted from different images obtained at different pre-processed states, the feature vector will also be affected. Therefore, images of differing pre-processed states are input into the feature extractor to determine which pre-processed state is optimal. A list of the different images, according to their pre-processed state, which are investigated follows.

- The ROI is obtained from the raw image (denoted by RAW).
- The ROI is normalised (NORM) using the z-score normalization method outlined in [24].
- The ROI is enhanced and binarized (BIN) using the Neurotechnolojia's Verifinger SDK [59].
- The ROI is obtained from a rotated raw image (ROT) about the reference point using the reference point's orientation.
- The rotated ROI is normalized (ROTNORM).
- The rotated ROI is enhanced and binarized (ROTBIN).

3.2.5. Screening

The primary purpose of a ROI is to ensure that the same relative region of the fingerprint is scrutinized between different samples acquired from the same fingerprint. This ensures that the features are derived from the same region, which in turn improves the reproducibility of the feature vector. However, the quality of the region is another important factor that severely affects the reproducibility. The ROI placement schemes only obtain a consistently reproducible region from the fingerprint image, although the region contains sub-regions of varying quality. Certain features are susceptible to noise, whereas others are more resilient. To investigate the extent of the affect image quality has on the reproducibility of the feature vector, a variety of quality maps of increasing severity are referenced in order to reject images which do not conform.

For this task, the MINDTCT algorithm from the National Institute of Standards and Technology (NIST) Biometric Image Software (NBIS) package [69] was employed to determine quality maps for each query image. The MINDTCT algorithm produces a map where a quality metric is assigned to local neighbourhoods within the fingerprint image based on directional, low flow, high curvature and low contrast maps. This algorithm assigns a quality metric ranging from 0 (background) to 4 (high quality foreground) and then a threshold is applied to the resultant map. In the subsequent experiments, the severity of the screening algorithm is depicted by 'Q' followed by the level of severity. For example, Q1 indicates that ROIs which contain regions whose index quality is less than 1 are rejected. Figure 3-8 presents the screening algorithm employed. It should be noted that the coordinates of the pixels that lay within the ROI of the fingerprint image are translated to the quality map. If any pixels of the translated ROI coordinates in the quality map are below the threshold, the algorithm terminates (i.e. reads in the next image).

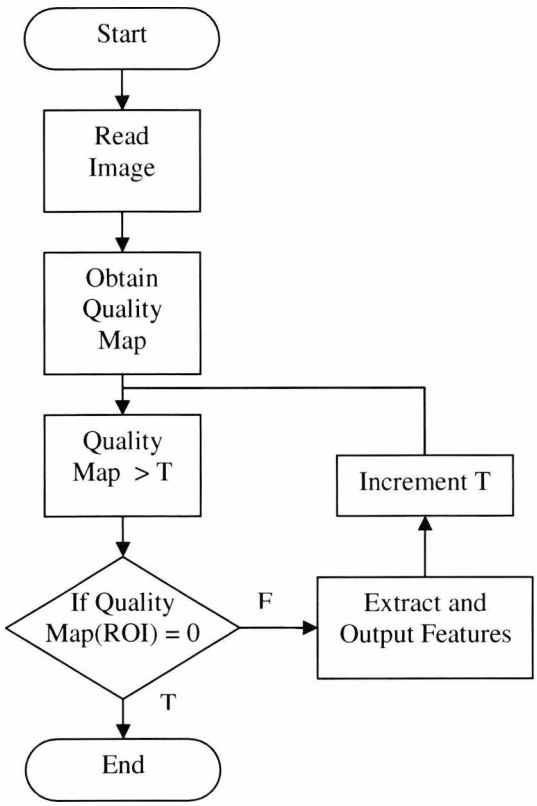


Figure 3-8: Screening Algorithm

3.2.6. Summary

It is clear that there are many factors, which are independent of the feature extraction algorithm (although ROI placement schemes are arguably part of the feature extractor), that can affect the reproducibility of the feature vectors. Therefore, additional test must be executed in order to ascertain the extent of their affects. As there are 2 reference point detection methods (AUTO and MAN), 7 ROI placement schemes (FULL, GBLSQ, GBLC, GRIDSQ, GRIDC, RAD and DISC), 6 states of pre-processed images (RAW, NORM, BIN, ROT, ROTNORM and ROTBIN) and 4 quality levels, there are many (336) permutations of tests. However, it is not necessary to conduct all permutations of tests for every feature because they are not all required. For instance, Zernike moments can only be determined from a circular ROI, therefore ruling out the entire set of square ROIs. Due to this, the name of the feature vector being investigated will be derived from its composite acronyms.

3.3. Test Databases

In order to investigate individual features, all features are to be extracted from databases 1 & 2 from the set of databases provided by the Fingerprint Verification Competition (FVC) 2004. This was decided because the FVC2004 database set is publicly available and is often used for benchmarking, providing standardisation for direct algorithm comparisons. Databases 1 & 2 were specifically chosen because the devices used for fingerprint capture were both optical sensors,

which are the most common form of sensor at present. Furthermore, the fingerprints from both DB1 and DB2 are captured at the same resolution, which removes scale variation so the feature extractors do not have to have their parameters re-tuned on account of scale. The databases each contain 800 images, where there are 100 different fingerprints and 8 impressions of the same fingerprint. This can be seen as a 100-class (i.e. multiclass) classification problem, for each database. Regrettably, the number of samples per fingerprint is quite low and it is foreseen that, due to insufficient intra-group samples, the feature distributions will not be accurately portrayed. Some example fingerprints from the 2 databases investigated are provided in Figure 3-9 and Figure 3-10.

Figure 3-9: Example Fingerprints From FVC2004 DB1

Figure 3-10: Example Fingerprints from FVC2004 DB2

Chapter 4. Level 1 Fingerprint Features

In human fingerprint matching, the features that are classified as level-one normally include the global ridge pattern, local ridge orientations, and singularities / reference points. These features are interrelated and fall in the category of *global features* as they represent the entire ridge structure of the fingerprint. Any feature pertaining to the overall structure of the fingerprint can be classified as level one. However, previous statistical analysis of level one features have shown that level one features, when solely employed, are not discriminatory enough for identification / verification purposes [8]. Therefore, many papers employ level-one features as a means to classify the fingerprint pattern [123-129]. As a result, these features can be seen as supporting, or even secondary, features. Whereas alone they are not effective enough, however it is anticipated that combining multiple features, even level-one features, may be synergistic and increase the effectiveness of the system exponentially. In this chapter, a selection of level-one features is investigated to determine suitability for use in a template-free biometric encryption implementation. This includes a comparison of orientation-based feature vectors, which results from a series of novel tests, and a novel core detection algorithm Although, the features are investigated independently within this chapter, these features will be combined in a later chapter to produce a biometric key.

4.1. Literature Review

The ridge pattern exhibited in a fingerprint is often described as being oriented and periodic, whereby the type and location of singularities dictate the general shape of the fingerprint. It is therefore the role of level-one features to provide a description of the ridge pattern and, considering that fingerprint ridge patterns possess high orientation components that are affected by singularities, these features provide a logical foundation on which to build.

4.1.1. Orientation Based Features

Due to its versatility, the dominant ridge orientations of local neighbourhoods within a fingerprint image are probably the most popular feature. Local estimates of the dominant ridge orientation not only provide a condensed representation of the pattern of the fingerprint but are often used as helper data for pre-processing methods (see Chapter 2). From the local dominant ridge orientation, further features can be derived.

Although the dominant ridge orientation of a local neighbourhood can be determined "*on the fly*" by treating the local neighbourhood as an independent image, orientation fields are more commonly determined. This is due to a number of reasons, of which the most important is to implement the contextual information residing in local neighbourhoods of the orientation field to

facilitate smoothing. The orientation field can be extracted by employing one of a multitude of methods. The most popular approach to determining the orientation field is outlined in [56] and presented in further detail later in this chapter. The main steps to this approach include the calculation of the local gradients by means of an edge detector (most commonly Sobel) and the estimation of the local orientation by employing the least squares estimate as a method of averaging the gradients over a local neighbourhood.

The result of the method presented in [56], produces a coarse orientation field. Alternative approaches that can be implemented to achieve this stage include the employment of slit-orientation masks [130], chain-coded contours [131], principal component analysis (PCA) of the gradients and coherence of a local neighbourhood centred on every pixel [132], and the transformation of local neighbourhoods into the Fourier domain (Short Time Fourier Transform) [1, 133]. However, these methods only provide a coarse orientation field, whereby many inconsistencies are attributed to noise rather than to legitimate changes in the ridge orientation. Therefore, the coarse orientation field requires smoothing, where a multitude of smoothing methods are available. The simplest of which is by employing a low pass filter (i.e. a Gaussian or mean filter) [24]. However, due to the circular, not linear, nature of directional information, a sine component map is derived from the orientation field to facilitate this.

The main drawback of filtering is that it is not known how much filtering is required. Iteratively filtering an image with a smaller kernel often provides better results than filtering once with a large kernel, although it is not known when to stop filtering. Some smoothing methods incorporate heuristics in order to prevent over-smoothing because over-smoothing can either filter important artefacts out completely or translate them. In [134], a contextual consistency measure was employed, whereby the effective neighbourhood of the smoothing operator was iteratively increased until the number of inconsistent blocks falls below a threshold. Alternatively, the smoothing can be terminated according to the number of singularities detected [135, 136] or when the correctness [73] or information reliability [137] measures exceeds a threshold. Furthermore, the entire orientation field can be enhanced by dynamically updating the field using the detected singularities [138], or a new orientation field can be determined by employing a polynomial and rational complex function model based on the type and location of these singularities [137]. This supports the fact that the orientation field is dependent on the location of singularities.

Once an orientation field has been established, a suitable descriptor is required. Using an entire orientation field as a descriptor is not a viable solution in a template-free implementation because the regions of the fingerprint captured will be different from the regions from the same fingerprint at a preceding acquisition. Therefore, alternative representations are required. The simplest form of an orientation field descriptor is just a restricted orientation field centred about a reference point [139]. An extension of this method is to determine the directional variance along the columns of

the restricted orientation field and use these variances as the descriptor [140]. This reduces the feature vector length and, in theory, provides additional robustness as the variance of individual neighbourhoods will be suppressed by taking the variance of each column.

The orientation field's dimensionality can be reduced further by obtaining local directional histograms [123, 141, 142], whereby the dominant orientations and their respective probabilities are used as the feature vectors. Alternatively, the orientation field can be transformed into a relational graph by segmenting the orientation field into regions, which is obtained by minimizing the variance of the element directions within the regions [125, 143, 144]. The relational graph descriptor was primarily employed for classification purposes by means of matching the graph with stored template graphs. However, useful information could be extracted from the connectivity of the regions. To enforce this hypothesis, the orientation field was represented as a planar graph (similar to a relational graph) in [145], whereby the graph edit distance was determined as the feature vector. The main challenge to overcome when employing graph representations of orientation fields is that the points of the graph are the centroids of the segmented regions. The centroids of the segmented regions can be highly variable because they are dependent on the area of the fingerprint that is captured.

An alternative representation of the fingerprint can be obtained from the orientation field. The general topology of the fingerprint ridge pattern can be obtained by obtaining the field flow curves from the orientation field [126, 128]. Salient features can then be extracted from the resultant field flow curves and used as feature of the fingerprint. However, these features are primarily used for classification as fingerprints of the same type will result in similar feature vectors.

4.1.2. Singularities / Reference Points

The analysis of singularities is essential to fingerprint systems because singularity points are landmark points, which are consistent for all impressions of the same fingerprint [24]. As with the orientation field, singularities are very useful throughout all stages within a fingerprint system. Singularity points are often used for alignment purposes, rudimentary classification, helper data for pre-processing and feature extraction and can be employed as features in their own right. Therefore, inaccurate singularity location detection has a huge impact on the performance of a fingerprint system. This is because singularities are often used to provide an origin that facilitates consistent placement of regions of interest (ROI) to enable feature extraction from the same relative region across different samples of the same fingerprint. When a singularity is employed as a point of reference for ROI placement, the singularity is therefore termed a reference point. However, reference points do not necessarily have to be singularities. Provided that the same relative location can be detected in each sample of a fingerprint, then any position can be employed as a reference

point. If the detected relative location varies then the features extracted from the ROI will be different because they would be extracted from different regions within the image.

In addition to inconsistent relative reference point locations, inaccurate reference point orientation estimation can also affect the system's performance when employed for alignment purposes. Poor estimation of singularity orientation can result in inconsistent alignment between successive acquisitions of the same fingerprint. In a template-free implementation, this could result in a poorer performance than not rotating the image to facilitate alignment. This is because raw images obtained from successive acquisitions will generally exhibit relatively small rotational variance, but if the raw images are rotated based on a poor orientation estimate of a singularity, then the rotational variance could be larger. These examples highlight the problems that can arise through poor singularity detection and therefore, it is crucial that the detection algorithm implemented provides consistent results. Not necessarily accurate, but more importantly consistent. Further impacts of singularities are discussed in [146].

Three types of fingerprint singularities exist, which are whorl, core and delta. The whorl is very rarely detected because it requires the ridges of a fingerprint to perform a full circle within a small area. Therefore, in many whorl type fingerprints, the whorl singularity is not detected due to the elliptical shape. Instead two core points will be generally be detected. A core is a focal point of the innermost re-curving ridgeline of the fingerprint. It can be thought of as a singular point of the flow field, which is the point of locally maximum curvature. Core points can be detected in loops, double loops and whorl type fingerprints. The final singularity is the delta point. A delta can be considered as the point at the centre of divergence of three sets of ridgelines. It can also be thought of as a singular point that is a point of locally indeterminate curvature [147]. Singularities are similar to minutiae as minutiae are seen as discontinuities in the ridge structure, whereas singularities are seen as discontinuities in the orientation field. Figure 4-1 presents regions that contain these singularities.

Figure 4-1: Example of Different Singularities. Whorl (l), Core (m) and Delta (r)

One of the most established techniques to determine fingerprint singularities is the Poincare Index (PI) [6]. By performing the PI on the orientation field, core and delta points can be located by the behaviour of the orientations within a closed square. The PI can also be applied on the squared orientation field [122]. The PI can consistently determine singularities, although this method requires an accurate orientation field as it is very susceptible to noise, which can cause spurious detections. Therefore, the orientation field is normally smoothed prior to executing the PI but this

can translate the singularity. Furthermore, post-processing may be required to remove spurious singularities. In [135], the PI is employed to identify candidate singularity points and then a further orientation field is computed based on the locations of the singularities, coined the global directional field. By using both the global directional field and the orientation field with the Difference of the ORIentation value along a Circle (DORIC) calculation, the spurious singularities were detected and omitted. Furthermore, a new flexible Poincare Index method, which determines the integral paths of the PI computation from homotopic properties, was presented in [135]. This technique was based on the premise that the orientation field is derived from singularity locations, whereby each fingerprint has an equal number of cores and deltas. Unfortunately, many singularities lie outside of the captured region.

There are numerous alternatives to the PI approach. Jain et al [24] presented a method which converted the orientation field, via a continuous vector field, into a sine component map. The integral of geometric regions centred on each element in the map was calculated, from which the maximum integral was used as the singularity. However, the sine component map approach only detects the primary core (commonly referred to as a concave core [24]), whereas the PI method will determine every apparent singularity and additional spurious points. Templates, or masks, are also popular approaches to singularity detection. The mask, or template, is applied to the orientation field and the regions which best fit the set of mask, or templates, is judged to contain singularity [122, 148-150]. Furthermore, singularities can also be determined from an extended relational graph of the orientation field [151]. In an extended relational graph, boundary and directional information is appended to the relational graphs which were briefly mentioned earlier. By incorporating this additional information, singularity points can be determined.

The aforementioned methods approached the detection of singularities as locating discontinuities in the orientation field and are hence extracted from the orientation field. However, the detection of singularities can be accomplished by analyzing symmetry within a fingerprint's orientation field. In [152] the core point was obtained by calculating the local axial symmetry within a pixel-wise orientation field, whereas in [153], two complex filters (one for core detection, one for delta detection) are applied to the orientation field at multiple resolution scales in order to determine areas of high rotational symmetry. However, Wang & Dai [154] approached singularity detection as a texture problem. As fingerprints can be perceived as oriented, periodic texture patterns, singularities can be viewed as different modes as the local texture breaks down at these points. Therefore, in [154], principal component analysis (PCA) was employed to estimate the distribution of Gaussian-Hermite moments of different orders, which represents the coherence of the image. Areas of incoherency are governed to contain singularities.

Unfortunately, many of the discussed singularity detectors are unable to locate a singularity in an arch type fingerprint. This is not surprising as arch fingerprints do not technically contain any

singularities, although a selection of these methods provide an estimate of a reference point in an arch [24, 152, 155]. However, a reference point does not have to be a singularity. The only prerequisite of a reference point is that the relative location must be consistently and accurately extracted. Therefore the reference point can, to the naked-eye, seem to be completely meaningless. At present there are not very many alternative reference points available because, unlike singularities, these points cannot be easily corroborated by manual detection. One example of an alternative reference point is presented in [89]. This method is based on multi-scale analysis of the orientation consistency and attempts to consistently locate a bespoke reference point regardless of the input fingerprint type.

Other than the type and location of the singularity, orientation is another useful feature. Reference point orientation is not employed very often as a feature for identification / verification as it provides minimal uniqueness. However, reference point orientation provides a method to rotate the image at the pre-processing stage, which essentially suppresses rotational variance as 0^{o} will be common to all samples from the same fingerprint. Unfortunately, a reliable estimation of orientation can only be determined for a core singularity because a delta point is the centre of divergence of three ridgelines, all at different angles. Therefore it is not known what angle to use as the orientation of the delta point. Besides, delta points are missed more often than cores. So, if the orientation is to be used for image rotation, then the core point (more specifically the primary core) orientation is the logical choice as core points are detected more frequently.

Unfortunately, core orientation, which can be used as an estimate of fingerprint orientation, is not that well researched because the majority of fingerprint systems employ a matching algorithm, which will handle alignment between the query and enrolled template. However, in a template-free implementation, alignment cannot be handled at the matching stage as there is no longer a matching stage. Therefore, to reduce rotational variance within the feature vectors, the fingerprint can be rotated prior to feature extraction, or the feature extractor can offset any orientation-derived feature vectors. For this, the orientation of the reference point is used as the orientation of the fingerprint. The orientation of a cores and reference points of an arch fingerprint can be obtained by employing the algorithm proposed in [152]. The local axial symmetry field extracted, when a threshold is applied, produces pairs of regions (i.e. two regions for each core or arch reference point) with low local axial symmetry, and the core or reference point is defined as the midpoint between the centroids of each pair. The orientation of the core or reference point is taken as the angle orthogonal to the direction of the line that joins the pair's midpoint.

An alternative approach to fingerprint core orientation is presented in [89]. In this method, the orientation is determined by examining the orientation field around the reference point. By tracing along radial directions from the reference point, the variation of the differences between the radial direction and the value of the orientation field along that radial direction can be used to estimate the

orientation. This was described in further detail in Fingerprint Pre-Processing, Chapter 2. Unfortunately both orientation estimation algorithms are unable to provide consistent estimations for a prefect whorl (i.e. the heart of the fingerprint is a circle not an ellipse). This is because features used for obtaining the orientation will be very similar.

Singularities are, more often than not, used as reference points to aid consistent placement of ROI. Furthermore, they can be used for alignment and orientation field enhancement, although they are not best suited to be used as features in their own right. When used as a feature, singularities are more commonly used for classification because the number of each singularity can be used to identify the class of fingerprint [6, 156]. However, the relative distances and angles can be employed as features [157], but singularities can easily be missed, which will seriously affect a system employing these features.

4.2. Orientation / Directional Fields

The ridge orientation is a very versatile feature. Many pre-processing techniques and other feature extractors utilise ridge orientation. However the ridge orientation can be used as a feature vector itself. Ridge orientation is a very descriptive feature, which represents the global ridge flow of the fingerprint pattern. It is also one of the easiest features to extract as well as, being one of the most consistently reproducible features, even in relatively poor quality images. As demonstrated previously, there is an abundance of literature on the extraction of orientation fields. However, many of them start off with a raw orientation field extracted in very similar ways. The methods only really differ in the way the orientation field is smoothed or modelled to remove discontinuities and maintain singularities. There are two main methods for extracting the raw orientation field. One is extracted in the spatial domain and the other in the Fourier domain.

4.2.1. Feature Extraction

The spatial domain method involves determining the greyscale gradients in the x (Gx) and y (Gy) axes of each pixel by using an edge detector like Sobel [25]. The orientation field can then be computed by either partitioning the image into non-overlapping blocks of a given width and height (known as block-wise) or by pixel-by-pixel (pixel-wise). Block-wise orientation fields are more popular than the pixel-wise because the extraction is quicker and they are more resilient to noise as the extraction is performed on a coarser scale. Anyway, in both cases a window of a specific size is centred about a focus pixel, x_0, y_0. The majority of the literature uses a specific window size, such as 8 x 8. However, the systems proposed are often only tested on one data set, whose images have constant resolution. In practice, this is not ideal as different systems will implement different sensors with varying resolutions. Therefore, setting the window size to a constant may not be suitable for all scanners as the feature becomes scale variant. Although the induced susceptibility to

scale would be small, it is ideal to make any feature robust to scale, rotation and translation variations. To counter this, the window size can be set to a factor of the average ridge distance by using a ridge distance estimation algorithm [158]. As this is calculated for every image, the information obtained from the window at differing scales should be approximately equal. In other words, the window will contain, on average, a ridge-valley pair regardless of the resolution of the image.

The greyscale gradients (*Gx* and *Gy*) of the window are then averaged. This is most commonly calculated by using the least square estimate, as presented in Equation 4-1.

$$O(i,j) = \frac{1}{2}\tan^{-1}\left(\frac{V_y(i,j)}{V_x(i,j)}\right)$$
$$V_x(i,j) = \sum_{u=i-h/2}^{i+h/2}\sum_{v=j-w/2}^{j+w/2} 2G_x(u,v)G_y(u,v) \qquad \textbf{4-1}$$
$$V_y(i,j) = \sum_{u=i-h/2}^{i+h/2}\sum_{v=j-w/2}^{j+w/2} (G_x(u,v)^2 - G_y(u,v)^2)$$

Where:

O(i, j) is the least square estimate of the ridge orientation of the window centred at pixel *(i,j)*.

w and *h* represents the width and height of the window respectively.

The aforementioned spatial method provides an orientation estimate which is orthogonal to the dominant ridge orientation and therefore must be rotated in order to represent the actual ridge direction. This can be achieved by applying a rotation of $\pi/2$, and is shown in Equation 4-2.

$$O(i,j) = \begin{cases} O(i,j) - \dfrac{\pi}{2}; & if\,(O(i,j) > 0) \\ O(i,j) + \dfrac{\pi}{2}; & otherwise \end{cases} \qquad \textbf{4-2}$$

An alternative approach to extract the orientation field is to transform the image into the frequency domain. This is achieved by implementing the Short Time Fourier Transform (STFT) as presented in [1]. The STFT is essentially a transform of a local neighbourhood of the spatial domain into the frequency domain by using the Discrete Fourier Transform (DFT), which is shown in Equation 4-3. Similarly to the spatial method, the original image is partitioned into either overlapping [1] or non-overlapping blocks [102]. The entire block is then transformed into the frequency domain. As this can get quite computationally expensive when using the DFT, the Fast Fourier Transform (FFT) is normally implemented. The FFT is a more computationally efficient method of calculating the DFT, but requires padding the signal to ensure its dimensions are of a power of 2.

$$F(u,v) = \sum_{y=-N/2}^{N/2-1} \sum_{x=-M/2}^{M/2-1} f(x,y) e^{-2\pi j\left(\frac{ux}{M}+\frac{vy}{N}\right)} \qquad \textbf{4-3}$$

Where:

M is the number of columns.

N is the number of rows.

$f(x, y)$ is the greyscale intensity of the pixel located at (x, y).

$F(u, v)$ is the corresponding Fourier coefficient.

It is quite logical to transform the image into the frequency domain because the main features that can be obtained are frequency and orientation. When neighbourhoods of a fingerprint are transformed into the Fourier the oriented, periodic nature produces an annular pattern of impulses. These impulses represent the different frequencies the neighbourhood is comprised of. The strength of the impulse represents how much of the neighbourhood is made up of waveforms of that frequency. From these impulses it is possible to obtain an estimate for the dominant orientation. When the neighbourhood is of good quality, two prominent impulses are produced on opposite sides of DC point. By drawing an imaginary line through the two greatest impulses (neglecting the DC. point in the centre), a direction orthogonal to the original image's local dominant direction is obtained. In order to obtain an estimate for the orientation, a Probability Density Function (PDF) for the Fourier coefficients and the Marginal Density Function (MDF) for the orientations are first determined. From these density functions, using Equation 4-4, an estimate of the orientation can be made.

$$O(i,j) = \frac{1}{2}\tan^{-1}\left(\frac{\sum_{\theta} p(\theta)\sin(2\theta)}{\sum_{\theta} p(\theta)\cos(2\theta)}\right)$$

$$p(\theta) = \sum_{r} p(r,\theta) \qquad \textbf{4-4}$$

$$p(r,\theta) = \frac{|F(r,\theta)|^2}{\sum_{r}\sum_{\theta}|F(r,\theta|^2}$$

Where:

$p(r, \theta)$ is the probability of the Fourier coefficient at polar coordinates (r, θ).

$p(\theta)$ is the probability of orientation θ.

$O(i,j)$ is the orientation estimation for the block centred at pixel (i, j).

Mathematically, the two outlined methods are identical [1]. However the result of the spatial method is derived from the greyscale gradients and these can be different depending on the edge detector implemented. In contrast, as there is not a variety of methods to transform from the spatial domain to the Fourier domain, there are not any external influences in the STFT method and can therefore be considered as the preferred choice for extracting the orientation. The STFT method

estimation is also considered more robust than the conventional spatial method due to the estimation being probabilistic [1]. However, these methods presented provide only an estimation of the orientation for the given block, or pixel, and many factors can adversely affect the result. These factors could be due to poor ridge structure within that region, scars or creases. Therefore, to reduce the spurious inconsistencies in the orientation field, smoothing is employed.

There are a multitude of smoothing methods [24, 89] which iteratively filter the orientations and use different features to determine whether the orientation field has been smoothed enough. In addition there are other methods which improve the orientation field based on singularity locations and function approximations [137, 138, 159, 160]. However, in the majority of cases, a simple low-pass filter (mean or Gaussian), as shown in Equation 4-5, can be applied to the field. If iterative filtering is required, it often indicates that a region of the image is of poor quality, in which case the image maybe rejected if a screening stage is implemented.

$$O'(i,j) = \frac{1}{2}\tan^{-1}\left(\frac{\sin(2O(i,j)) * W(i,j)}{\cos(2O(i,j)) * W(i,j)}\right) \qquad \textbf{4-5}$$

Where:

O(i, j) is the estimated orientation of the block centred at pixel *(i, j)*.

W(i, j) is the filtering kernel (either mean or Gaussian) centred at pixel *(i, j)*.

O'(i, j) is the smoothed orientation of the block centred at pixel *(i, j)*.

4.2.2. Experimental Results

In this section, a variety of orientation field derived feature vectors are investigated in order to determine the most suitable feature for use in a template-free implementation. Amongst these feature vectors, two are pre-existing feature vectors, which were highlighted previously in the literature review. A brief description of the feature vectors extracted follows.

- OFGRID - A restricted orientation field, as presented in [139], that created a feature vector length of 48. To obtain this feature vector, a 7x7 GRIDSQ ROI scheme was employed, with the centre cell neglected due to the volatility of direction about the reference point.
- SUMGRID - A novel representation that obtains the accumulated distance from the sum of unit length vectors with orientations from OFGRID. The accumulated distance is calculated using Equation 4-6. The distance of the vector sum provides a simple univariate feature vector, which is independent of the order of which the original feature vector is index and eliminates orientations from the feature vector. By removing orientations, it eliminates the necessity for modulus arithmetic to deal with the problems wraparound values can cause.

$$D = \sqrt{\left(\sum_{i=0}^{i<N} \cos\theta_i\right)^2 + \left(\sum_{i=0}^{i<N} \sin\theta_i\right)^2} \quad \textbf{4-6}$$

Where:

θ_i is the orientation at feature vector index i.

N is the length of the feature vector.

D is the accumulated distance.

- OFVAR – A feature vector proposed in [140], which is the variance of the columns within a restricted orientation field. To obtain this feature vector, the OFGRID was reused, but the variance of each column was employed to create a feature vector length of 7.
- OFRAD - A novel approach that uses the orientations extracted similar to that of a 3 x 8 RAD ROI scheme. Unlike OFGRID, where the orientation was extracted from individual ROIs, OFRAD employed a fully smoothed, orientation field and extracted the orientations from the field by indexing the field. This was obtained similarly to employing the RAD ROI scheme, but mapping the centre coordinates of each sub-ROI to that of the orientation field. The sub-ROI is not actually extracted. This method of extraction was chosen because, if any unexpected deviations are present, then they should be suppressed.
- SUMRAD – The same as SUMGRID but obtained from OFRAD instead of OFGRID.

Many experiments were executed in order to investigate a number of parameters. These parameters included whether the ROI was derived from manual or automatic detection (MAN or AUTO) of the reference point, the severity of screening (Q1, Q2, etc), and the stages of pre-processing the image has undergone (*PPS* – Pre-Processing Stage). These parameters and their respective enumerators were defined previously in the Testing Framework chapter. Furthermore, the neighbourhood size (*Size*) of which the orientation was calculated from was investigated in order to determine whether the size of the neighbourhood has a significant impact to the recreation of the feature vector.

It is expected that a larger neighbourhood should suppress the influence of noise within the local neighbourhood and therefore should provide improved consistency results. However, with the increasing severity of image screening, if the image surpasses the screening stage then in theory the orientations obtained should be of good quality. One drawback to a larger neighbourhood size though is that it increases the size of the ROI, which increases the possibility of the ROI extending past the boundaries of the image or into areas of poor quality. Therefore, this will result in fewer samples because the images will be rejected at the screening stage. With the reduction in samples, the number of classes residing in the feature space will also be reduced, which could possibly provide misleading results as the feature space will not be as densely populated as other tests. With this in mind, the results of each experiment includes the total number of classes (*C*) and samples (*N*) that advanced through the screening stage and were subsequently used in the statistical analysis. It should be noted that each database contains 800 images, which comprises of 100

classes. Furthermore, each table caption denotes whether the feature vector was obtained using a manually or automated reference point detection method and the severity of the screening algorithm.

4.2.2.1. Orientation Field Grid-based (OFGRID) ROI Descriptor Results

In this experiment, the OFGRID feature vector and the OFGRID-derived univariate feature vector, SUMGRID, were investigated. A 7x7 grid was centred on the reference point, as depicted by the GRIDSQ ROI scheme in the testing Framework chapter, whereby every cell of the grid was the size of the neighbourhood used to estimate the local orientation. Figure 4-2 and Figure 4-3 present the findings as a result of using manually detected reference point and orientation estimation (MAN). In all subsequent charts presenting the results of these experiments, the number of classes and the number of samples, which were used, are provided on the x-axis.

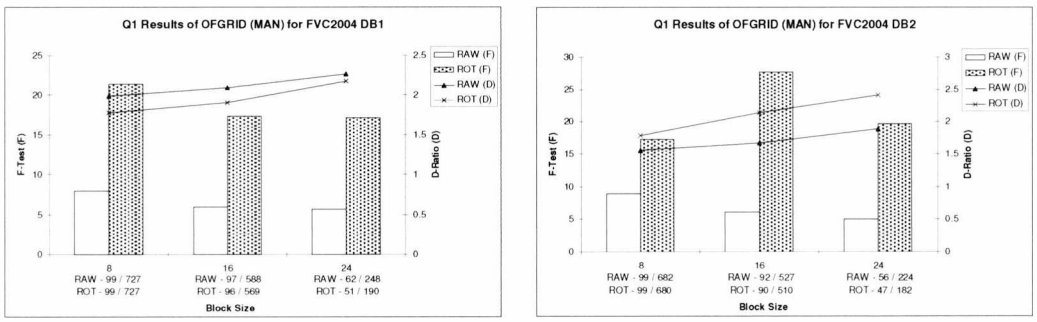


Figure 4-2: Q1 Results of OFGRID (MAN) for FVC2004 DB1 (l) and DB2 (r)

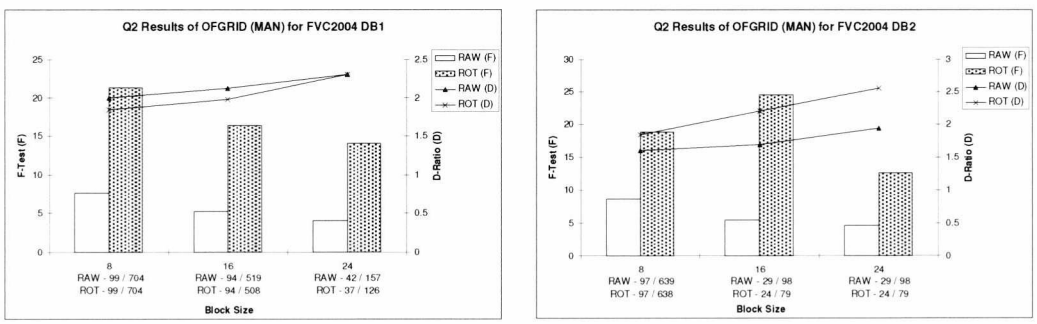


Figure 4-3: Q2 Results of OFGRID (MAN) for FVC2004 DB1 (l) and DB2 (r)

In Figure 4-2 and Figure 4-3, the F-test values (*F*) presented are the F-test values determined from the univariate feature vector, SUMGRID, and the D-Ratio is the distance ratio defined within the Testing Framework chapter for multivariate, directional feature vectors. From the results presented, it is clear that the F-test value is significantly greater for images that have been rotated prior to feature extraction. Therefore, the univariate representation of the OFGRID feature vector (SUMGRID) supports the hypothesis that rotated images improve the stability and reproducibility of the feature vectors. However, the multivariate representation (OFGRID) somewhat weakens the

validity of the hypothesis. The results for DB2 support the hypothesis fully, whereby the rotated image's D-Ratio is consistently greater than that of the non-rotated image, although the results for DB1 somewhat contradict the hypothesis.

The D-Ratio values of DB1 are consistently less, albeit by a small margin, for the rotated images. Therefore, this suggests that a grid ROI does not consistently capture the same regions through all rotations. However, the D-Ratio results do conform to an expected trend. This trend was that the D-Ratio should increase as the neighbourhood size and the severity of the screening algorithm increases. A larger neighbourhood size, as mentioned previously, is expected to suppress the influence of noise and hence improve the reproducibility of the feature value. Likewise, a severe screening algorithm should reject any images with regions of poor quality. Therefore the reproducibility should improve as these parameters increase. However, this trend is not exhibited by the F-test value, where, in fact, it generally deteriorates. A possible explanation for this phenomenon is that when these parameters increase, the reproducibility may increase but more classes may overlap due to the higher degree of smoothing and the reduction of the feature space (due to more samples being rejected). Coupled with the reduction of the feature vector's dimensionality, decreases the features individuality.

The results obtained from the MAN approach to reference point detection and orientation are encouraging and somewhat conform to expectations. However, the AUTO approach was expected to be inferior to MAN, but the results are far from encouraging. Figure 4-4 and Figure 4-5 present the findings for the AUTO approach.

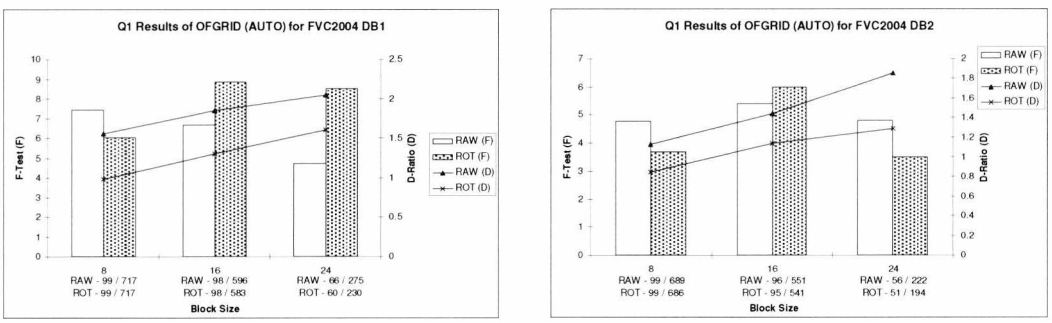


Figure 4-4: Q1 Results of OFGRID (AUTO) for FVC2004 DB1 (l) and DB2 (r)

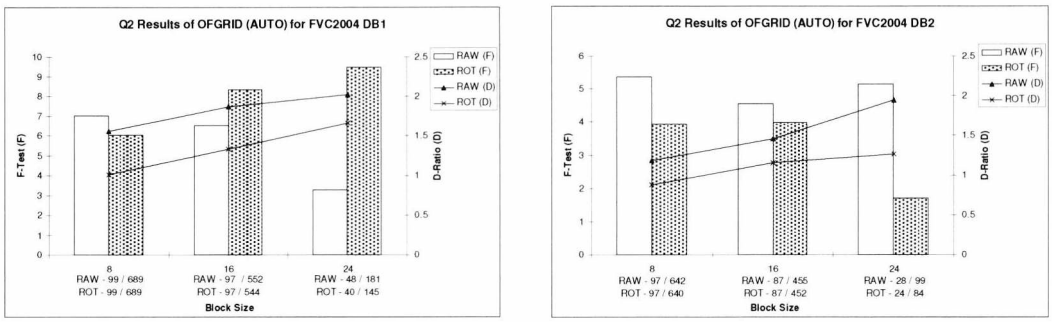


Figure 4-5: Q2 Results of OFGRID (AUTO) for FVC2004 DB1 (l) and DB2 (r)

It is immediately evident from the results presented in Figure 4-4 and Figure 4-5 that the F-test value for both the RAW and ROT images are approximately of the same order and are significantly less than that of the same tests but for MAN. In terms of the univariate representation, by employing automatic methods of reference point detection and orientation estimation, rotated images perform similarly, and in many cases worse, to that of non-rotated images. This is supported by the multivariate representation where the D-Ratio of the rotated images are consistently and significantly less (compared with the difference between corresponding D-Ratios from the MAN experiments) than that of the non-rotated images. This suggests that there are no benefits from rotating the image prior to feature extraction when the reference point and orientation estimation are automatically determined. However, the trend of the increasing D-Ratio is maintained in the majority of the AUTO experiments, but the F-test value for the rotated images is volatile and is not monotonic. The reasons expressed previously for the MAN experiments, coupled with inconsistent reference point location and inaccurate orientation estimation, are potentially accountable for these events.

4.2.2.1.1. Summary

Both the univariate (SUMGRID) and the multivariate (OFGRID) representations of a restricted orientation field descriptor have been investigated and assessed. It is clear that rotating the image prior to feature extraction affords significant improvements to the univariate representation, although this is the case only when the reference point and the orientation is estimated manually. When automatic methods are employed, a significant deterioration in the performance of both the univariate and multivariate descriptors is exhibited.

With respect to the impact of neighbourhood sizes and screening algorithm severity, the performance of the multivariate descriptor increases with the increase in neighbourhood size and screening algorithm severity. This was expected, although the F-test value decreased instead. It has been speculated that this is attributed to the reduction in dimensionality, reduction to the feature space (with respect to screening algorithm severity) and the similarity of different fingerprints at

larger resolutions (through increasing the neighbourhood size). However, the univariate descriptor is more appropriate for a template-free implementation on two accounts.

- The F-test values of the rotated images in the MAN test are significant.
- The univariate descriptor eliminates directional data, facilitating normalisation and quantisation for future processing by transforming wraparound values to linear values.

Taking these points into account, the best descriptor, based on the experiments executed, across the two databases is the univariate (SUMGRID) descriptor, which is rotated prior to feature extraction with orientations obtained from a neighbourhood of 16 pixels.

4.2.2.2. Column Variance of Orientation Field Grid-based (OFVAR) ROI Descriptor Results

For these experiments, the variance of the orientations along the columns of OFGRID is used as the feature descriptor. The original feature vector outlined in [140], obtained an estimation of variance along the columns by applying equation 4-7. However the method is ambiguous as it is not made clear how the authors handled the wraparound nature of directional data because Equation 4-7 is commonly used for linear data. Therefore, to clarify, for these experiments the mean orientation was obtained using Equation 4-8, which is a method for calculating the mean of directional data, and the difference between orientations is calculated by implementing Equation 4-9. Equation 4-9 is an alternative approach to calculating the difference between two angles, except that both the anti-clockwise and clockwise differences are taken into account. The difference is restricted to less than π because this is the orientation range of most orientation field extractors.

$$\sigma_k^2 = \sum_{i=0}^{i<N} \left(\delta(i,k)^2\right), \quad \delta(i,k) = \theta_{i,k} - \mu_k \qquad \textbf{4-7}$$

$$\mu_k = \arctan\left(\frac{\frac{1}{N}\sum_{i=0}^{i<N}\sin\theta_{i,k}}{\frac{1}{N}\sum_{i=0}^{i<N}\cos\theta_{i,k}}\right) \qquad \textbf{4-8}$$

$$\delta(i,k) = \min\left(\max\left(\theta_{i,k},\mu_k\right) - \min\left(\theta_{i,k},\mu_k\right), \pi - \left(\max\left(\theta_{i,k},\mu_k\right) - \min\left(\theta_{i,k},\mu_k\right)\right)\right) \qquad \textbf{4-9}$$

Where:

σ_k^2 is the variance of column k.

$\delta(i,k)$ is the angular difference between $\theta_{i,k}$ and μ_k.

$\theta_{i,k}$ is the orientation of row i, column k of OFGRID.

μ_k is the mean of the orientations in column k of OFGRID.

N is the number of rows, which in these tests are 7 as OFGRID is reused.

$\min(x, y)$ and $\max(x, y)$ returns the minimum and maximum value between x and y respectively.

As the feature vector contains no directional data, Wilk's Lambda and the Mahalanobis distance ratio (M-Ratio) was employed to assess the multivariate feature vectors. These statistical values were defined in the Testing Framework, Chapter 3. Figure 4-6 and Figure 4-7 present the results obtained by using a ROI placed according to a manually detected reference point and orientation (MAN).

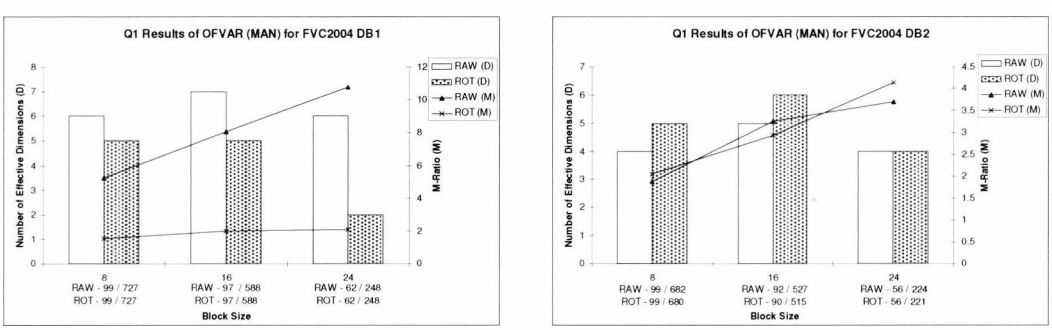


Figure 4-6: Q1 Results of OFVAR (MAN) for FVC2004 DB1 (l) and DB2 (r)

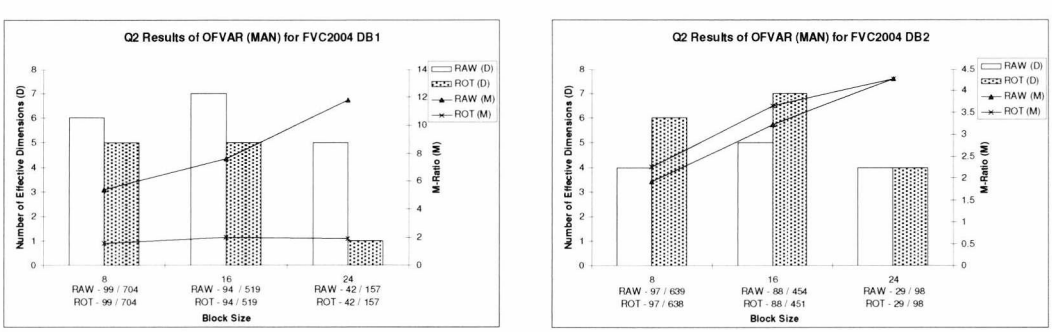
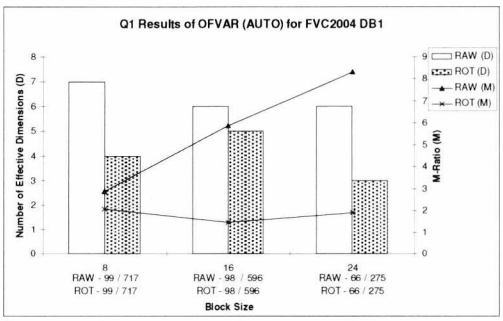


Figure 4-7: Q2 Results of OFVAR (MAN) for FVC2004 DB1 (l) and DB2 (r)

Figure 4-6 and Figure 4-7 tell a similar story to the results of the OFGRID descriptors with the same experimental setups, although this was somewhat expected as the OFVAR feature vector is determined from OFGRID. For both descriptors, the rotated images were outperformed by the non-rotated image for database 1, although the converse was true for database 2. Furthermore, the results conform to the expected trend whereby the distance ratio increases as the neighbourhood size and the screening algorithm's severity increases. However, there was a possibility that the number of significant dimensions will not follow the same pattern, because as the size and severity increases, fewer samples will remain. As there are fewer samples, the probability of the classes inhabiting in all dimensions reduces.

With reference to Figure 4-6 and Figure 4-7, it is clear, with the exception for the results of the rotated images in database 1, that the distance ratio metric has markedly improved, even though the OFVAR descriptor has significantly less dimensions. A potential cause of this observation could be due to the dimension-space of the OFGRID feature vector being restricted to a range of π and within that range, up to 100 classes maybe present in these experiments. Taking into account that there are 48 dimensions within the OFGRID descriptor indicates that the majority of dimensions will have significantly overlapping classes. Therefore, the problem arises due to a densely populated dimension-space, and consequently feature-space, where distance between the means of classes is close together as the space is restricted. This is why alternative representations of the orientation field (such as OFVAR and SUMGRID / SUMRAD) are investigated in order to expand the dimension-space, fostering sparsely distributed classes. Evidently, the result of this test (OFVAR) supports the concept of increasing the dimension-space because the distance metric is much improved, albeit at a cost to dimensionality.

Another prominent feature of the results presented in Figure 4-6 and Figure 4-7 is that the means of the classes frequently lay within either 6 or 7 dimensions with very high significance. This demonstrates that up to 100 classes can be separated within either 6 or 7 dimensions. Good class separation within a high multidimensional space is a desired property because the classes are well defined, potentially providing increased stability, whereas high dimensionality facilitates longer keys with a greater number of effective bits. These desired properties however, are not as prominently exhibited when the tests are conducted using automatic methods of reference point and orientation estimation is employed. Figure 4-8 and Figure 4-9 present the results of the OFVAR descriptor when automatic methods are applied.



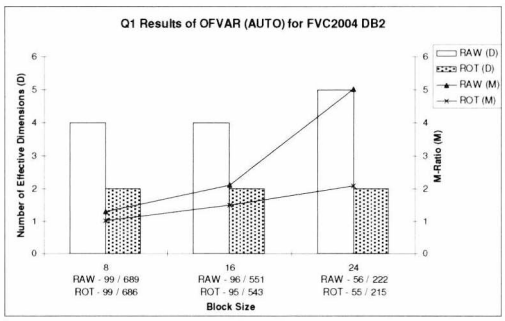
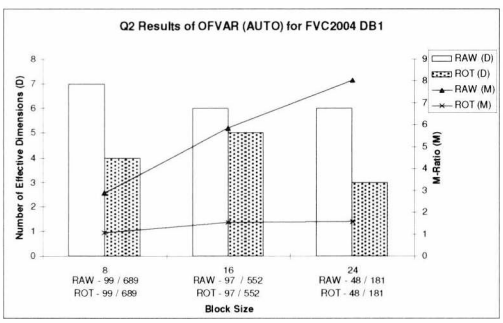


Figure 4-8: Q1 Results of OFVAR (AUTO) for FVC2004 DB1 (l) and DB2 (r)



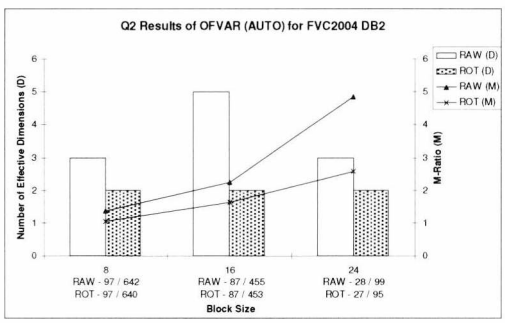


Figure 4-9: Q2 Results of OFVAR (AUTO) for FVC2004 DB1 (l) and DB2 (r)

As expected, Figure 4-8 and Figure 4-9, in general, exhibit a slight deterioration in both significant dimensions and the M-Ratio. However, the M-Ratio in some cases, such as that for database 2, is uncharacteristically large compared with other results from the same test, even though the number of significant dimensions is fewer. On the other hand, the unexpectedly large M-Ratio coincides with a large M-Ratio for its MAN equivalent. This is accountable to the reduction of samples in the feature space, so the feature-space is sparsely populated and, just by chance, the feature space is larger because this is determined from the range of feature values exhibited by the samples.

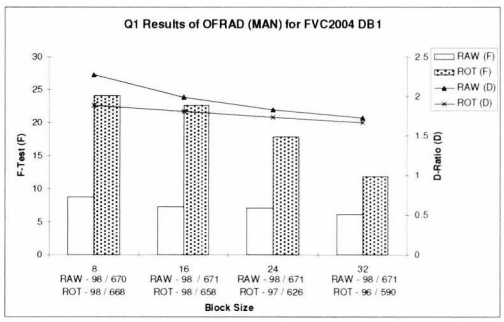
4.2.2.2.1. Summary

The OFVAR feature vector has been investigated and it has been determined that slight improvements over the OFGRID descriptor are attained by simply calculating the variance of the orientation within each column of the grid. This has essentially increased the dimension-space, and in turn the feature-space, which provides further space for more classes to be separated. Although it is desired that the feature vector be highly dimensional, there is not any benefit in a large feature vector that cannot be reproduced effectively. By using OFVAR, the dimensionality has been reduced from 48 dimensions in OFGRID to 7 dimensions, although the distance ratio has vastly improved with this reduction. Furthermore, the OFVAR descriptor is much more efficient in terms of dimensionality. OFGRID employed a 48-dimension feature vector, although the effective dimensions (dimensions that are deemed significant) are undoubtedly much less. Whereas OFVAR utilises a greater percentage of its dimensionality, in some cases 100%.

As the OFVAR descriptor is derived from the OFGRID feature vector, it is not surprising that it has similar characteristics as OFGRID. For both descriptors, the distance ratios increase with the increase of orientation field neighbourhood size and screening algorithm severity. It is also clear that rotating the image prior to feature extraction provides no additional benefits, even when manually estimated. However, slight performance boost was exhibited by both methods when the reference point was determined manually. Hence the optimum feature vector across both databases was a non-rotated image, with the grid centred on a manually estimated reference point and the orientation extracted from a neighbourhood of size 16. It should be noted that this deduction is based on the maximum significant dimensionality as the distance ratios are less robust to the number of samples and classes. Therefore, larger distance ratios can be attributed to fewer samples populating the same space.

4.2.2.3. Orientation Field Radial-based ROI (OFRAD) Descriptor Results

The OFRAD descriptor is a novel feature vector comprising solely directional data, although a univariate representation (SUMRAD) can be obtained by calculating the distance of the vector sum. Therefore, the statistical tests performed on both the univariate and multivariate feature vectors are the same that were performed for the OFGRID experiments. It is important to note that for these experiments a fully smoothed orientation field was extracted and polar coordinates, derived from the reference point location and orientation, were used as an index into the orientation field to obtain the orientation estimate at those coordinates. These coordinates are determined similarly to that of the RAD ROI scheme, whereby the coordinates are the centroids of the individual sub-ROIs. The actual extraction of the sub-ROIs is unnecessary because only the centre coordinates are required. The coordinates were determined by using a track width and a spindle radius (track width and spindle radius are defined in the Testing Framework chapter under the RAD ROI scheme) of the size of the neighbourhood. It was decided to implement a spindle radius to neglect the region which the reference point lies in because the orientation within this region is unstable. For these tests, the 24 coordinates, based on 8 radial directions and 3 concentric circles, was used to create a feature vector of length 24. The results for OFRAD, where the ROI scheme is placed according to manually estimated location and orientation (MAN) are presented in Figure 4-10 to Figure 4-12.



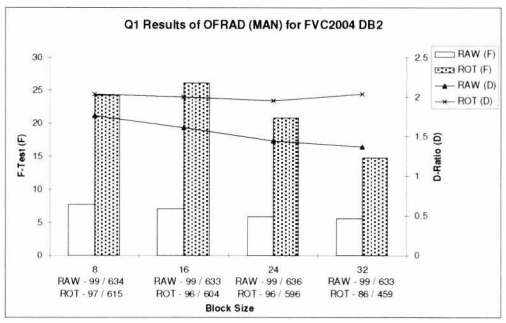
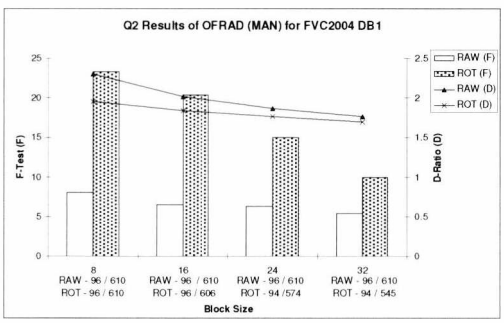


Figure 4-10: Q1 Results of OFRAD (MAN) for FVC2004 DB1 (l) and DB2 (r)



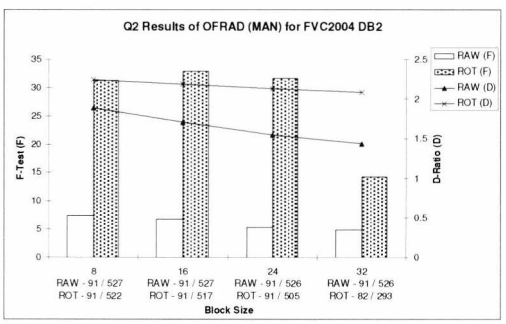
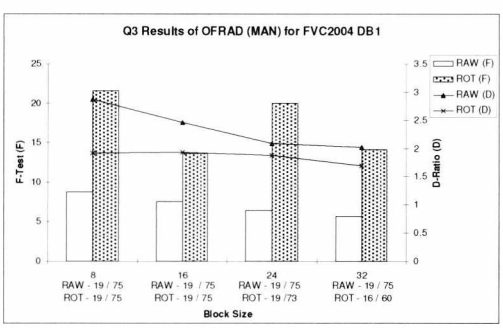


Figure 4-11: Q2 Results of OFRAD (MAN) for FVC2004 DB1 (l) and DB2 (r)



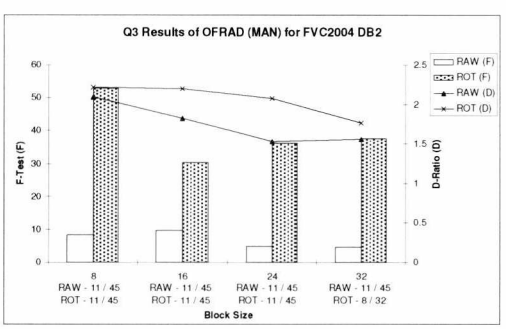


Figure 4-12: Q3 Results of OFRAD (MAN) for FVC2004 DB1 (l) and DB2 (r)

It is clear from the results presented in Figure 4-10 to Figure 4-12, is that additional tests have been carried out. Due to the adoption of the RAD ROI scheme, the effective ROI (i.e. the region which encompasses all the sub-ROIs) is reduced. As the effective ROI is smaller, the probability of regions marked "*poor*" by a quality estimation algorithm being present in the sub-ROIs, or any sub-ROI exiting the boundaries of the image, is reduced. Therefore, a further screening stage severity level and a larger neighbourhood can be investigated, before the number of remaining classes is less than 2.

From comparing the results in Figure 4-10 to Figure 4-12 to the results of the same tests for the OFGRID experiments, it is immediately evident that the values for the F-test and the distance ratio metric are slightly greater for OFRAD. Therefore, a slight performance increase is exhibited just by adopting a different ROI scheme. In addition to this slight performance increase for both the

univariate and multivariate representations, it seems that, in general, that both the D-Ratio and F-test value decrease as the neighbourhood size increases. This trend is different from that exhibited in the OFGRID experiments, whereby the D-Ratio increased with the neighbourhood size. A possible explanation for this inverse relationship is that the OFRAD descriptor is obtained from a smoothed, complete orientation field. When the neighbourhood size is large, it is possible that repeat values occur because different coordinates may result in indexing the same neighbourhood in the orientation field. Furthermore, a larger neighbourhood suppresses noise as the orientations are averaged over a greater area, which is then smoothed again upon completion of the orientation field. This additional smoothing could result in an increase in similarity between different classes. However, no performance boost is exhibited in Figure 4-13 to Figure 4-15, which presents the results of the OFRAD descriptor that is determined from automatically estimating the reference point and orientation (AUTO).

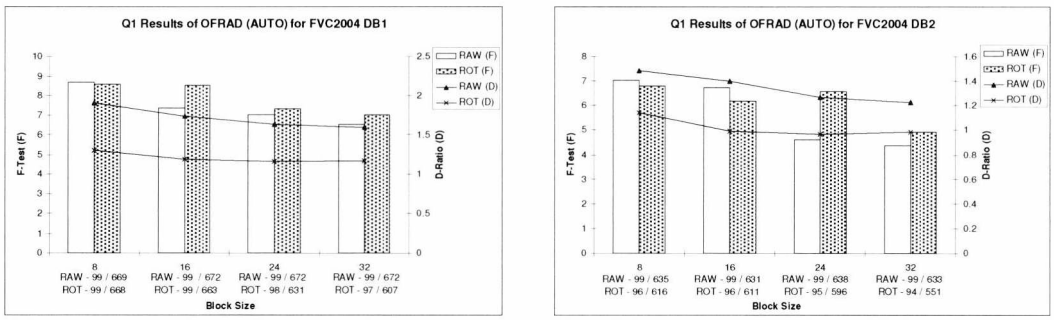


Figure 4-13: Q1 Results of OFRAD (AUTO) for FVC2004 DB1 (l) and DB2 (r)

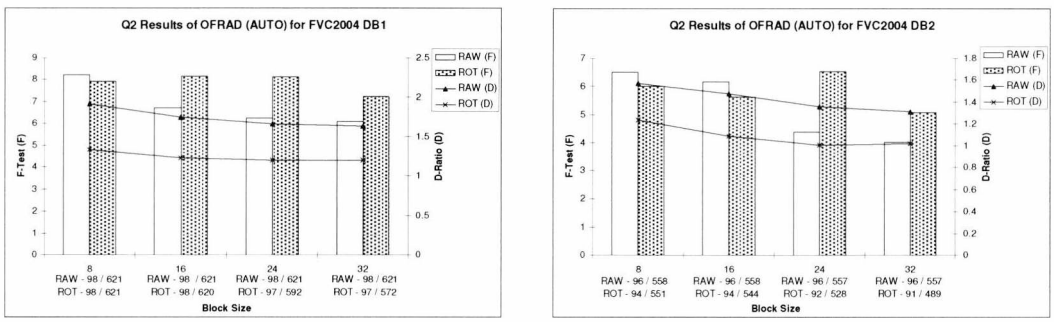


Figure 4-14: Q2 Results of OFRAD (AUTO) for FVC2004 DB1 (l) and DB2 (r)

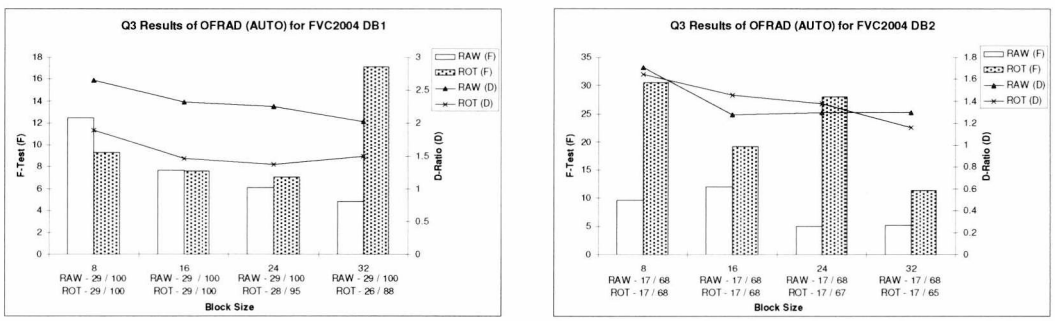


Figure 4-15: Q3 Results of OFRAD (AUTO) for FVC2004 DB1 (l) and DB2 (r)

Figure 4-13 to Figure 4-15, similar to the OFGRID and OFVAR experiments, show that in the majority of instances, automatic estimation of the reference point location and orientation result in slightly inferior performance. Even when the image is not rotated, the test statistics of AUTO are slightly lower than that of the MAN results. This is attributed to inconsistent location of the reference point due to errors in the orientation field or the true point lies too close to the image boundaries. The results also show that, when the image is rotated prior to feature extraction using an automatic estimate of orientation, in most cases the results slightly deteriorate. This agrees with the findings of the other experiments where rotating the image, based on automatic methods, provides no benefit over the non-rotated image. This is credited to inaccurate and inconsistent estimations in the whorl and arch fingerprint types.

4.2.2.3.1. Summary

As with the earlier experiments, when the reference point location and orientation estimation is performed manually, both the univariate (SUMRAD) and multivariate descriptors perform well. Similarly with the OFGRID experiments, SUMRAD exhibits a significant boost in its F-test value. However, the F-test value for OFRAD is slightly better than that of OFGRID. Furthermore, as with OFGRID and OFVAR, if the ROI placements are determined from an automatically estimated reference point and orientation, then there is no advantage to rotating the image as the errors induced reduce performance significantly.

4.2.3. Evaluation of Orientation-based Feature Vectors

From the experiments conducted, it is clear that the best results are obtained when the reference point is manually estimated. However, it is not immediately obvious whether there is any benefit to rotating the image prior to feature extraction. Significant gains can be achieved from a rotated image only when a multivariate descriptor, which comprises solely directional data (OFGRID and OFRAD), is reduced to a univariate feature value by means of a vector sum (SUMGRID and SUMRAD), provided the orientation is estimated manually. With regard to OFGRID and OFRAD, the distance ratio metric is only slightly greater for non-rotated images. This may be attributed to the poor choice of test statistic (D-Ratio) or the restricted range of each dimension-space. The restricted range is certainly a factor as up to 100 classes lay within a range of π for each dimension. Since the orientation of a region is expected to have a certain amount of variation, it is highly probable that there will be a significant number of overlapping classes.

Transforming the feature vector, so that the dimension-space is larger, seems promising because the number of classes should be able to fit in the space with minimal overlapping. The results seem to support this concept as OFVAR, SUMGRID and SUMRAD produce values that are closer to the desired properties. These properties include greater F-test values, more significant dimensions and

higher distance ratios. However, this has come as a result to the number of dimensions. By reducing dimensionality potentially adversely affects the number of effective bits and length of a key. It was decided that this is a necessary sacrifice as the stability of individual features is more important because additional features can be added to increase the length of the key.

The results have also shown that slight improvements can be attained by adopting a different ROI scheme. It was determined that a RAD ROI scheme produces marginal gains over that of a GRID ROI. This was somewhat expected as a GRID ROI does not extract the exact same regions when the grid or image is rotated, unlike the RAD ROI scheme which is based on the polar coordinate system and has circular sub-ROIs. Furthermore, the results, based on the distance ratio metric, generally depict a slight improvement as the size of the neighbourhood the orientation is extracted from and severity of the screening algorithm increase. However, this trend is not mirrored in either the F-test value or the greatest significant dimension statistics. In fact, in most cases these values decrease. This indicates that, although the distance ratio increases, the actual overlapping of the classes increase. This could be because the larger neighbourhoods are suppressing the noise excessively and that many images actually look the same at this level of resolution. It was determined that a neighbourhood size of 16 pixels on average produced the best results across both databases, but to determine the best feature descriptor out of the 3 is not straightforward.

Previously it has been ascertained that the univariate SUMRAD is slightly better than that of SUMGRID, although it is not easy to compare the OFVAR descriptor to SUMRAD due to the difference in dimensionality and test statistics. However, it was decided that dimensionality would be sacrificed over stability and that the M-Ratio was questionable in terms of its uniqueness measurability. Therefore, the SUMRAD feature vector of an orientation field, which used a local neighbourhood of 16 pixels and been rotated based on manually estimated reference point and orientation, was deemed to be most appropriate for a template free implementation.

4.3. Singularities

Singularities, as with orientation fields, are an important feature of fingerprints and are used for a variety of purposes. Singularities can be used as a feature themselves, for enhancing orientation fields or for reference points. It is important to note that, although singularities are reference points themselves, reference points do not necessarily mean singularities. There are other bespoke reference points available [89, 152, 161]. However, what makes singularities important is that they essentially dictate the flow of the fingerprint ridges. This can be clearly seen in the orientation field. As a result of the determined orientation fields, singularities are generally defined as discontinuities within the orientation field [6]. An example of a *core* and a *delta*, with their corresponding orientation fields are presented in Figure 4-16 and Figure 4-17.

Figure 4-16: Example of Fingerprint and Orientation Field Core Point

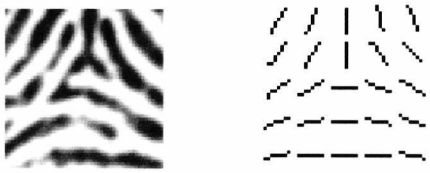


Figure 4-17: Example of Fingerprint and Orientation Field Delta Point

Although the locations of singularities can be used as a feature, they are more often used as reference points or for orientation field enhancement. The reason for this is that, in the vast majority of fingerprints, there exist no more than four singularities in a fingerprint. This is by no means discriminatory enough to identify an individual. Coupled with the fact that, in many fingerprints, singularities often lie outside the acquired image, this potentially reduces their discriminatory power even further. If all singularities are present, they are often used to classify the type of fingerprint [6, 156], which is useful for reducing the search-space of a database. Although classifying a fingerprint can provide a few *bits* of information (fingerprints can be classified into seven types [whorl, left loop, right loop, arch, tented arch, double loop and other], which can provide 4 *bits* of information), there are better classification techniques available. Therefore, in this section, only the locations of singularities are extracted and analyzed with the intention to use them for reference points.

Reference points provide a solution for alignment and rotational issues in automatic fingerprint systems. Every time a fingerprint is presented to the system, there will be varying degrees of rotational variance between each presentation. This effectively makes the traditional coordinate system redundant as it is highly improbable that the origin (0, 0) will be in the same position for each presentation. Extracting a consistent reference point alleviates this problem by providing an alternative origin. Although the Cartesian coordinate system is still ineffective, the polar coordinate system can be implemented as an alternative origin can be located. This assists alignment for matching purposes but also allows features of interest in the fingerprint to be located at a less-variable position. In addition, grids for feature detection can be positioned more consistently.

Singularities have the advantage over other reference points because they are easily identifiable with the naked eye, therefore allowing the locations to be verified by and compared with human recognition. Other reference points can be used, although these points may not be easily identified by humans, which make ascertaining the reproducibility of the locations very difficult. This is mainly due to the reliance of a computer for the extraction of the locations and the inability to ascertain *ground truth* locations to compare to.

In the following section, a novel detection algorithm is proposed. As mentioned previously, the singularities are extracted with the intention of using them as reference points and not for feature descriptors. Therefore the analysis and results will be focussed on the reliability, reproducibility

and accuracy of the extracted locations of the singularities. It is important to note that the accuracy of the extracted location is not the most crucial factor. For any subsequent feature extraction algorithm that requires a reference point, the most vital aspect of a singularity is that the position of the singularity / reference point is consistently located at the same point on the fingerprint image.

4.3.1. A Novel Fingerprint Core Detection using Orthogonal Tracing

It is generally accepted that the definition of singularities is that they are landmark structures which arise from discontinuities in the orientation flow. Therefore the effects of these points are witnessed in the orientation field, which is exhibited in Figure 4-18. The regions in which the singularities occur are highlighted within the figure (top region is that of a core and the bottom is that of a delta) and it is the pattern, which lay within these regions that many singularity detection methods, such as the Poincare Index [6] or Sine Component Map approach [24], utilize in order for detection. These methods investigate local neighbourhoods of the orientation field and determine the presence of singularities using the neighbouring orientations. However it is possible to determine a core singularity without the means of an orientation field with the subsequent investigation of local neighbourhoods, or at least not directly as orientation change being a very useful feature for singularity detection.

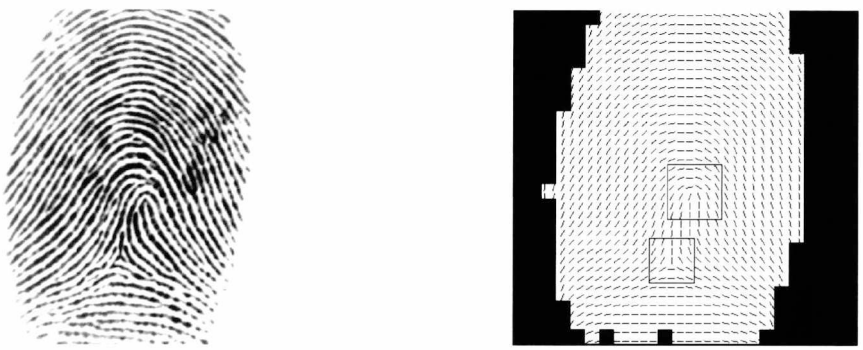


Figure 4-18: Example of a Primary Core and a Delta Singularity

As with many core detection methods, the origins of this algorithm lie within the orientation field. By looking at the orientations which lie above the core point, it is clear that if all the orientations were rotated by π (orthogonal to the orientation), then the lines representing the orientations would flow directly to the core. However, following the orthogonal orientations to the core point in the orientation field will not provide any benefits over the Poincare Index. This is because the orientation field will be the same in both methods and therefore the detected locations will be the same. However, if the ridges of the actual fingerprint were traced orthogonally, certain traces will lead directly to the core point. The resultant traces obtained by following orientations orthogonal to ridge structure is shown in Figure 4-19.

Figure 4-19: Example of Core Detection using Orthogonal Tracing

It is clear from Figure 4-19 that the traces which start above the core point all lead directly to the core before continuing along similar trajectories. All the other traces do not approach the core and it is only the traces which lead to the core that are required. It is also evident that the delta point repels the traces and hence the delta point cannot be extracted using this method. It is therefore the objective of the orthogonal tracing algorithm to extract the traces which are evident in Figure 4-19. The proposed method can thus be split into two modules. The first module is to actually extract the traces and the second is to determine feature points along the traces to be used for calculating the core points.

4.3.1.1. Orthogonal Tracing

To extract the traces, an iterative method akin to the tracing method employed in [162] is applied. However the proposed method is simpler because the traces bisect the ridges rather than run parallel. This means that the additional checks used to ensure that the trace stays on a ridge and when it ends or bifurcates are redundant. The algorithm takes either a segmentation or quality map as an input to determine the initial starting points of the traces. The quality map approach employed by the MINDTCT program of the NBIS (NIST Biometric Image Software) [69] was used to determine the fingerprint foreground region. The quality map was binarized using a threshold of 1 and enlarged to the dimensions of the original image, I. This means that any regions of quality 1 or less are set as background and the rest as foreground. However any alternative segmentation algorithm may be substituted at this stage. The number of traces, N, is also user-defined and denotes the total number of traces. The output of this algorithm is N lists of coordinates which make up the traces. The orthogonal tracing algorithm can be further expanded into a number of processes.

1. Create N lists in order to store the coordinates of all the points of each trace.
2. Obtain the centroid of segmentation map S by using Equations 4-10 and 4-11.

$$x_c = \frac{\sum_{S(x,y)>0} x}{n} \quad \textbf{4-10}$$

$$y_c = \frac{\sum_{S(x,y)>0} y}{n} \quad \textbf{4-11}$$

Where:

$S(x, y)$ denotes whether the pixel (x, y) is located in the background or foreground in the segmentation map.

n is the total number of pixels which lie in the foreground of segmentation map S.

(x_c, y_c) are the coordinates of the centroid.

3. Obtain N starting points by traversing radially from the centroid until it exits the image or foreground region. This is demonstrated in Figure 4-20, whereby $N = 16$ starting points are obtained. The centroid of the segmentation mask is denoted by the black circle in the centre of the foreground and the starting points by the black crosses near the border with the background.

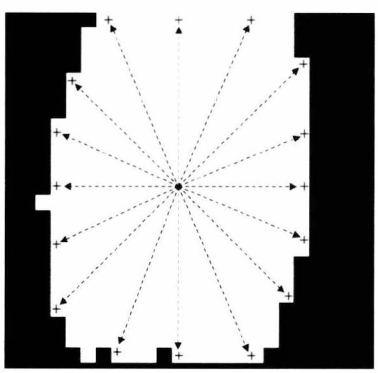


Figure 4-20: Obtaining the Initial Starting Points

4. If the number of traces completed equals N, then exit the algorithm. Otherwise set (x_o, y_o) to the coordinates of the next start point, set r to δ which is set to an initial constant (20) to get the trace started and θ_o to the angle which points from the start point to the centroid.
5. Move forward by implementing Equations 4-12 and 4-13

$$x_n = x_o + r\sin\theta_o \quad \textbf{4-12}$$

$$y_n = y_o + r\cos\theta_o \quad \textbf{4-13}$$

Where:

(x_o, y_o) are the coordinates of the starting point.

r is the distance setting.

θ_o is the angle in which the trace should move.

(x_n, y_n) are the new, current coordinates of the trace.

6. Check to see if (x_n, y_n) lie within the image and the foreground region. If not terminate the trace and go back to step 4 for the next starting point. However, if the point is valid, add it to the list of points that make up the trace.
7. Extract a window, whose dimensions are a power of 2 (e.g. 16), from the fingerprint image I, centered on (x_n, y_n).
8. Transform the window using the Fast Fourier Transform (FFT) and calculate the orientation orthogonal to the dominant ridge direction as in [1]. The discrete representation of this calculation is presented in Equations 4-14 and 4-15.

$$\Theta = \frac{1}{2}\tan^{-1}\left(\frac{\sum_{\theta} p(\theta)\sin(2\theta)}{\sum_{\theta} p(\theta)\cos(2\theta)}\right), for\ 0,...,\theta,...,\pi \qquad \textbf{4-14}$$

$$p(\theta) = \sum_{r}\left(\frac{|F(r,\theta)|^2}{\sum_{r}\sum_{\theta}|F(r,\theta)|^2}\right), for\ 0,...,r,...,d \qquad \textbf{4-15}$$

Where:

d is the dimension of the window, for example 16.

Θ is the orthogonal orientation.

$F(r, \theta)$ is the Fourier coefficient at radius r and orientation θ.

9. Determine appropriate direction. The orthogonal orientation calculation returns an angle between ± π/2. This means that the trace can go in two possible directions and, if the raw value is always used, the trace will always go towards the right. Therefore, it is important to determine which of the two directions is most likely. As the input direction is known (i.e. the direction from the previous point to the current point), it is simply achieved by finding the maximum between the two possible angles (α_0 and α_1). This problem is illustrated in Figure 4-21, whereby the left-hand diagram shows the previous point going into the current point and the two possible next points. The middle and right-hand diagrams show each of the two possible directions and their respective angles. The output angle is defined as the greater of the two possible angles (α_0 and α_1) and this is shown in Equation 4-16.

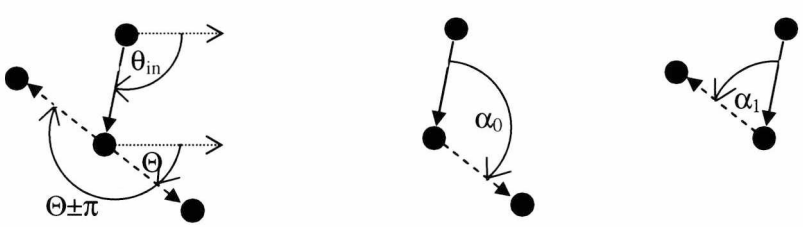


Figure 4-21: Example of Direction Decision

$$\theta_{out} = \begin{cases} \Theta, & if\,(abs(\alpha_0) >= abs(\alpha_1)) \\ \Theta + \pi, & otherwise \end{cases} \quad \textbf{4-16}$$

Where:

α_0 and α_1 are the angles between the possible output directions.

Θ is the orthogonal orientation.

θ_{out} is the output direction.

10. From the current point, obtain a ridge projection. The ridge projection is similar to the x-signature extracted in [2] except the ridges are projected from the current point along the output orientation (θ_{out}). The ridge projection is obtained by implementing Equations 4-17 to 4-19.

$$prj(i) = \frac{1}{fltr}\sum_{j=0}^{fltr} I(x(i,j), y(i,j)),\ 0,...,i,...l \quad \textbf{4-17}$$

$$x(i,j) = x_n + i\cos\theta_{out} + (j - \frac{fltr}{2})\sin\theta_{out} \quad \textbf{4-18}$$

$$y(i,j) = y_n + i\sin\theta_{out} + (j - \frac{fltr}{2})\cos\theta_{out} \quad \textbf{4-19}$$

Where:

l is the length of the projection, which is 32 as in [2].

$I(x, y)$ is the intensity of the pixel located at (x, y) in the fingerprint image I.

$prj(i)$ is the projected intensity value i pixels away from the origin.

(x_n, y_n) is the coordinates of the origin.

$fltr$ is the size of the average filter. For this implementation, a filter size of 8 was used as these sizes are the same as in [2].

θ_{out} is the orientation orthogonal to dominant ridge direction.

11. Determine whether the projected waveform is of acceptable quality. If the ridge projection is of poor quality then terminate the trace and go back to step 4 to start the next trace. The quality of the waveform is determined similarly to that outlined in [2], whereby the average peak-trough amplitude and the average wavelength are taken into account. If the average peak-trough amplitude is less than a given threshold and the average wavelength does not lie within a certain range, then the quality is judged to be poor and the trace is terminated. Other methods of assessing quality of the ridge structure can be substituted for this method.
12. Set r to the distance from the origin to the first ridge (i.e. the first trough after a peak in the ridge projection), θ_o to θ_{out} and (x_o, y_o) to (x_n, y_n).
13. Go back to step 4 to continue the trace.

4.3.1.2. Locating the Core Points

The result of the tracing algorithm is N lists of coordinates which make up each trace, from which the core points may be derived. The core points are detected by using the centre of gravity of feature points along the traces. These feature points are made up of 4 different types, which are points of high curvature, apex points, intersections and points of near-parallelism. Each trace has two accumulators, which contain the summations of the x and y coordinates respectively for the centre of gravity calculation. In addition to the accumulators, there is also a counter for each trace which keeps track of how many points have been added to the accumulators. The first category of feature points extracted is the points of high curvature.

4.3.1.2.1. Points of High Curvature

From inspecting Figure 4-19, the traces which lead towards the core points undergo a significant change in direction as they approach core points. This feature is probably the most important because most of the core detection methods available make use of this feature in some way or another. The significant direction change is very useful as a feature point but they can also be used to reject traces. By rejecting traces that do not turn enough, prevents them from influencing the extraction of intersections later on. Therefore, points that undergo a significant change in direction can be added to the accumulator for that trace. The majority of traces that lead to core points, turn through an angle close to 90 degrees so the algorithm attempts to find these points by calculating the curvature at every point along each trace using Equations 4-20 and 4-21.

$$C(n,i) = \arccos\left(\frac{a^2 + b^2 - c^2}{2ab}\right) \qquad \textbf{4-20}$$

$$a = \sqrt{(P(n)_x[i - fltr] - P(n)_x[i])^2 + (P(n)_y[i - fltr] - P(n)_y[i])^2}$$

$$b = \sqrt{(P(n)_x[i + fltr] - P(n)_x[i])^2 + (P(n)_y[i + fltr] - P(n)_y[i])^2} \qquad \textbf{4-21}$$

$$c = \sqrt{(P(n)_x[i - fltr] - P(n)_x[i + fltr])^2 + (P(n)_y[i - fltr] - P(n)_y[i + fltr])^2}$$

Where:

$P(n)_x[i]$ and $P(n)_y[i]$ are the x and y coordinates respectively of the i^{th} coordinate in the list of coordinates which make up the trace n.

$C(n, i)$ is the curvature of the i^{th} coordinate of trace n.

From the estimated curvature of every point, a set of accumulators are incremented according to Equations 4-22 to 4-24.

$$xAcc[n] = \sum_{i=fltr}^{nP[n]-fltr} val = \begin{cases} P(n)_x[i], \; if \;\; C(n,i) <= \dfrac{\pi}{2} + db \\ 0, \qquad\qquad otherwise \end{cases} \quad \textbf{4-22}$$

$$yAcc[n] = \sum_{i=fltr}^{nP[n]-fltr} val = \begin{cases} P(n)_y[i], \; if \;\; C(n,i) <= \dfrac{\pi}{2} + db \\ 0, \qquad\qquad otherwise \end{cases} \quad \textbf{4-23}$$

$$cnt[n] = \sum_{i=fltr}^{nP[n]-fltr} val = \begin{cases} 1, if \;\; C(n,i) <= \dfrac{\pi}{2} + db \\ 0, \qquad\qquad otherwise \end{cases} \quad \textbf{4-24}$$

Where:

C(n, i) is the curvature of the i^{th} coordinate of trace *n*.

xAcc[n] and *yAcc[n]* are the accumulators for the *x* and *y* coordinates respectively of trace *n*.

cnt[n] is the count of points added to the accumulators for trace *n*.

db is a deadband to allow flexibility in the degree of curvature.

nP[n] is the number of points in the list of coordinates making up trace n.

$P(n)_x[i]$ and $P(n)_y[i]$ are the *x* and *y* coordinates respectively of the i^{th} coordinate in the list of coordinates which make up the trace *n*.

n is the trace number from 0 to *N*.

fltr is the distance between the focus coordinate and the supporting coordinates. This acts as a filter because all the coordinates that lie between the focus and supporting coordinates are neglected so it is analogous to a straight line between points.

4.3.1.2.2. Apex Points

After obtaining the points of high curvature, the traces whose count is zero can be disregarded from further investigation as they do not exhibit enough curvature. In addition to the points of high curvature, the apexes of the traces are also a useful feature, not just because they are normally a point of high curvature but they are useful for determining which side of the apex an intersection lie. The apex is calculated by using a very simple calculation and is simply the point which is the furthest from both the start and end of the trace. The calculation to determine this point is shown in Equation 4-25.

$$A[n] = i,\ if\ \ a(i) + b(i) = \max(a(i) + b(i))$$

$$a(i) = \sqrt{\begin{matrix}(P(n)_x[i] - P(n)_x[1])^2 \\ +(P(n)_y[i] - P(n)_y[1])^2\end{matrix}} \qquad \textbf{4-25}$$

$$b(i) = \sqrt{\begin{matrix}(P(n)_x[i] - P(n)_x[nP[n]-1])^2 \\ +(P(n)_y[i] - P(n)_y[nP[n]-1])^2\end{matrix}}$$

Where:

A[n] is the index into the coordinate list of trace *n* where the apex point lies.

$P(n)_x[i]$ and $P(n)_y[i]$ are the *x* and *y* coordinates respectively of the i^{th} coordinate in the list of coordinates which make up the trace *n*.

n is the trace number from 0 to *N*.

i is the index into the current coordinate list

The apex points of all the remaining traces are then added to the accumulator by using Equations 4-26 to 4-28. As apexes of remaining traces will generally have a large degree of curvature, these points are added to the accumulator again here as they are important. Adding the same point multiple times ensures that the centre of gravity calculation is slightly biased towards these important feature points.

$$xAcc[n] = P(n)_x[A[n]] \qquad \textbf{4-26}$$

$$yAcc[n] = P(n)_y[A[n]] \qquad \textbf{4-27}$$

$$cnt[n] = cnt[n] + 1 \qquad \textbf{4-28}$$

Where:

xAcc[n] and *yAcc[n]* are the accumulators for the *x* and *y* coordinates respectively of trace *n*.

cnt[n] is the count of the points added to the accumulators for trace *n*.

nP[n] is the number of points in the list of coordinates making up trace *n*.

$P(n)_x[i]$ and $P(n)_y[i]$ are the *x* and *y* coordinates respectively of the i^{th} coordinate in the list of coordinates which make up the trace *n*.

4.3.1.2.3. Intersection Points

Other points of interest include points where effective traces intersect. However some intersection points are counter-productive, because the traces may intersect far away from the core points. Therefore the only intersection points that are used are when a trace intersects with another trace, but the intersection occurs on different sides of the apex of each trace, or when one of the intersection points is adjacent to, or is, an apex point on either trace. Figure 4-22 demonstrates the intersection of two traces, which crossover on either side of their respective apexes. Figure 4-22 depicts two traces which crossover twice and it is clear that an intersection lies on the left side of trace t_1's apex and on the right side of trace t_2's apex. An additional intersection that lies on

opposite side of the traces apexes is shown and both points should be added to the accumulators of t_1 and t_2. The dotted, arrow lines in Figure 4-22 depict the orientation of their respective trace. It is simply the angle between the apex point and the mid-point between the first and last coordinates of the trace.

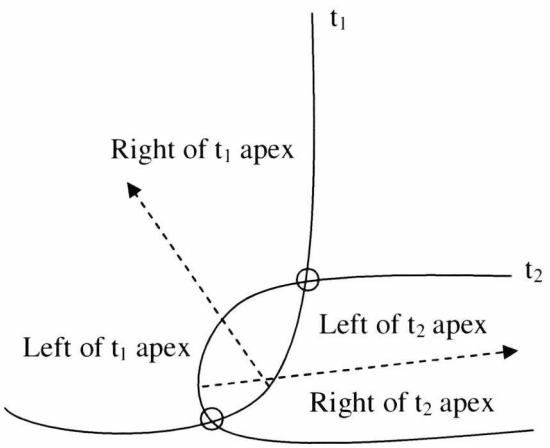


Figure 4-22: Example of Valid Intersections

In order to determine these points, the equation of every line segment (i.e. the connecting line between two consecutive points in the trace) within a certain distance from the apex point, for each remaining trace, was calculated. The distance is the number of points away from the apex within the trace. So if the apex occurs at the 10^{th} point on the trace, a distance of 4 will mean equating all the line segments between the 6^{th} and the 14^{th} point on the trace. The intersection points were determined by solving simple simultaneous equations for all the line equations belonging to different traces. The points that satisfy the restrictions previously outlined are added to the intersecting traces respective accumulators. However, there is another reason why detecting intersections is important. The intersections can be used to determine how many cores have been detected and which trace belongs to which core. This was achieved by implementing a union-find algorithm [25] in order to determine which traces interconnect with others indirectly. By keeping track of which traces intersect with which, an estimate for the number of cores can be determined. The number of disjoint roots resulting from applying the union-find algorithm is the number of cores which have been detected. This is because fingerprints that have two core points will not have any intersections between traces which belong to different cores, except when the cores are very close together. An example of this is shown in Figure 4-23, whereby the traces that lead directly to the core points are disjoint. Although not shown in this example, the trace running through the gap between the core points is actually rejected from the core point location algorithm.

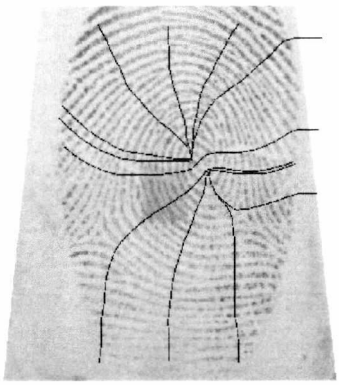


Figure 4-23: Example of Orthogonal Tracing on a Fingerprint with 2 Cores

Unfortunately, this does not guarantee to detect the correct number of cores. Occasionally it was found that spurious cores were detected where the traces ran parallel and did not intersect. Therefore to counter this problem, two additional tests were executed. The simplest case is by establishing the proximity of line segments from a trace to those of a different trace. If two traces pass within a few pixels of each other, then they are judged to belong to the same core. An alternative approach would be to use the orientation of the trace (i.e. the direction from the apex point to the midpoint between the first and last points on a trace) as the difference between trace orientations of traces belonging to different cores is usually large. However, this is not always the case and therefore this approach was rejected. The proximity measure was implemented as it is conceptually simple. Instead of line intersections, the algorithm locates intersections of rectangles.

4.3.1.2.4. Points of Near-Parallelism

As the traces pass through core points, the traces will often follow a similar trajectory. Therefore the point at which the trajectories start to be similar to that of other traces belonging to that core is a very useful feature point. These points are usually very close to the true core point so it is important that they are included. An example of these points is shown in Figure 4-24, whereby the points of near-parallelism are depicted as black circles and the trajectory of which the traces follows are shown by the black arrows.

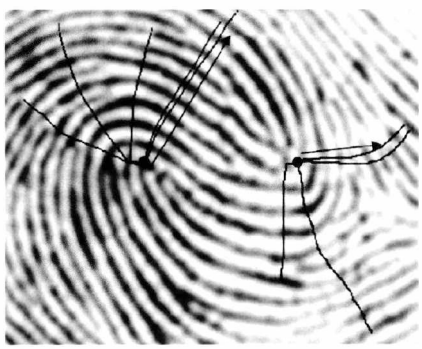


Figure 4-24: Example of Near-Parallelism Points

The near-parallel points are calculated by investigating points close to the apex of the traces that belong to the same core. For each trace and for coordinates which lie close to the apex point, two angles are determined by using Equations 4-29 and 4-30.

$$\alpha_{c,n,i,1} = \tan^{-1}\left(\frac{P_{c,n,y}[i-fltr]-P_{c,n,y}[i]}{P_{c,n,x}[i-fltr]-P_{c,n,x}[i]}\right) \quad \textbf{4-29}$$

$$\alpha_{c,n,i,2} = \tan^{-1}\left(\frac{P_{c,n,y}[i+fltr]-P_{c,n,y}[i]}{P_{c,n,x}[i+fltr]-P_{c,n,x}[i]}\right) \quad \textbf{4-30}$$

Where:

$P_{c,n,x}[i]$ and $P_{c,n,y}[i]$ are the x and y coordinates respectively of the i^{th} coordinate in the list which makes up trace n that belongs to core c.

$\alpha_{c,n,i,1}$ and $\alpha_{c,n,i,2}$ are the two query angles for trace n which belongs to core c located at the i^{th} coordinate.

n is the trace number from 0 to N- 1.

i is the current index into the list of coordinates which make up trace n. i ranges from $A[n] - d$ to $A[n] + d$, where d is a constant distance of 3 points. This restricts the search space to close to the apex.

$fltr$ is a constant distance which is greater than d to ensure that one angle points to the left side of the apex and the other to the right side. In this implementation a distance of 7 was used.

For each trace under investigation, only a total of two points of near-parallelism can be attained. These are the points which are deemed to be near-parallel and are closest to the apex point that are to the left and the right. To determine whether a point on a trace is near-parallel with that of another point on a different trace, the decision presented in Equation 4-31 is employed.

$$isPar = \begin{cases} true, & if\ abs(\alpha_{c,n,i,p} - \alpha_{c,t,j,p}) < thr \\ false, & otherwise \end{cases} \quad \textbf{4-31}$$

Where:

$\alpha_{c,n,i,p}$ is the angle of the i^{th} coordinate of trace n belonging to core c.

$\alpha_{c,t,j,q}$ is the angle of the j^{th} coordinate of trace t, which must not equal n, belonging to core c.

i is the index in the range of $A[n]$ - d to $A[n] + d$.

j is the index in the range of $A[t]$ - d to $A[t] + d$.

d is a constant distance as defined for equations 4-29 and 4-30.

p represents either α_1 or α_2 as depicted in equations 4-29 and 4-30. The calculation is the same for both.

thr is a threshold to judge whether the difference in angles is sufficiently small.

The near-parallel points which are closest to the apex are then added to the traces respective accumulators and the final core points can be determined by taking the centre of gravity of all the extracted feature points which lie on traces belonging to each core. These core points are calculated by implementing Equations 4-32 and 4-33.

$$x_c = \sum_{n \in c} \frac{xAcc[n]}{cnt[n]} \qquad \textbf{4-32}$$

$$y_c = \sum_{n \in c} \frac{yAcc[n]}{cnt[n]} \qquad \textbf{4-33}$$

Where:

xAcc[n] and *yAcc[n]* are the accumulators for the x and y coordinates respectively for trace n which belongs to core c.

cnt[n] is the number of coordinates added to the accumulator of trace n.

x_c and y_c are the final coordinates of core c.

4.3.2. Experimental Results

In order to ascertain the effectiveness of the proposed algorithm, tests were performed on both database 1 and 2 of the FVC2004 collection. Each database contained 100 different fingerprints and there were 8 images of each fingerprint. In addition to performing the orthogonal tracing method, the Poincare Index (PI) [6] and the Sine-Component Map (SCM) [24] approaches were also performed on the same databases. Two variations of the proposed algorithm were performed, which had a different number of traces (OT16 and OT32). These were included in order to determine the performance of an implementation with fewer traces to see whether the cores can be extracted as reliably as that of an implementation with a larger number of traces. This is because the larger number of traces required to reliably detect cores, the slower the algorithm.

The effectiveness of each algorithm across the databases was compared. In order to achieve this, the ground truth (manually determined) coordinates of every core and an estimation of the orientation of the fingerprint were determined for each fingerprint in the databases. The features used for the statistical analysis were the distance of the detected cores from their respective ground truth cores, execution time and a novel measure to determine how reproducible the core placement was for each set of fingerprints.

In order to extract these statistics, detected cores were matched up with corresponding ground truth cores by determining the distance between each detected and ground truth core and then minimizing the sum of the distances. This was achieved by placing the distances within a table and determining the shortest path from the left side of the table to the right side, without visiting a row or column twice. This is similar to the shortest path algorithms. The shortest path represents the

optimum match of the core points. Now that the core pairs (i.e. corresponding detected and ground truth core points) are known, the vectors between them can be extracted and statistical analysis can be performed.

To determine how accurate the methods are at locating the core point, the distance between the detected cores and their respective ground truth cores are used. The distributions of the distances for databases 1 and 2 are presented in Figure 4-25 and Figure 4-26 respectively.

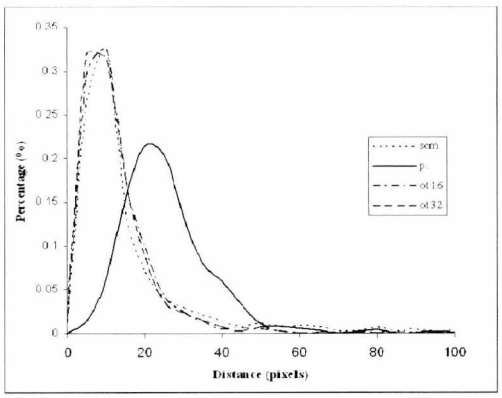


Figure 4-25: Distribution of Distances - FVC2004 DB1

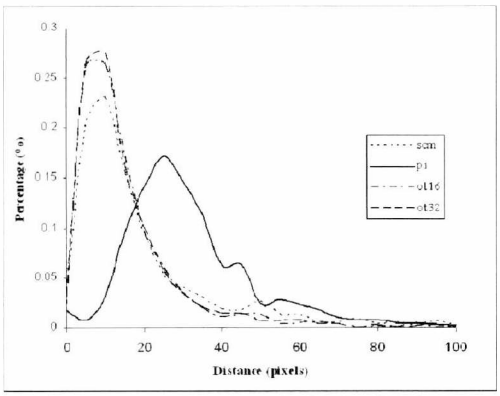


Figure 4-26: Distribution of Distances - FVC2004 DB2

The distance distributions presented in Figure 4-25 and Figure 4-26 highlights that the proposed method performs similarly to the Sine Component Map method regardless of the number of traces used. However the Sine Component Map detected many core points at a considerable distance from the true core point. This is not shown in the distributions as the scale of the x-axis has been reduced for presentational purposes, although this is reflected in Table 4-1, where the mean and standard deviation are higher than expected.

Table 4-1: Results of Distance Test

	FVC2004 DB1		FVC2004 DB2	
	μ_{dist}	σ_{dist}	μ_{dist}	σ_{dist}
SCM	19.273	37.61	23.921	36.151
PI	24.001	15.553	31.097	20.511
OT16	10.835	12.231	13.674	16.073
OT32	10.295	11.225	13.656	16.016

The main reason for the sine component map method producing a larger than expected mean and standard deviation is that the method does not know whether the point is a core or not as it takes the maximum of the difference of geometric regions in the orientation field. In most cases the maximum is a core point but in noisy images or when the core point is close to the top of the image, the maximum may not be the core. Therefore, when the sine component map method

produces an error, the error is generally quite large, which upsets the statistics as it is generally accurate.

Although accuracy is essential if the core points are to be used as features, it is not important if the core point is to be used as a reference point. Therefore, a novel measure, which attempts to quantify the variation of the positioning of the detected core point across different samples of the same fingerprint, is proposed. This is essentially a measure of the intra-sample variations. All the fingerprints in the same database were captured using the same scanner at the same resolution so in theory the vector between two points in one fingerprint should be similar to that of the vector between the same two points with a different acquisition of the same fingerprint within the same database.

Unfortunately, this is not corroborated in practice as deformations due to the elasticity of skin can have an adverse affect on these vectors. However, the vectors between the detected and their respective ground truth core points provide an estimation of the intra-sample variance. In addition to the ground truth location of the core points, an estimation of the orientation of the fingerprint was also determined manually so the vectors can be rotated accordingly, suppressing rotational variations.

In order to calculate this measure, the vectors are categorized with respect to the ground truth core to which they belong. The respective ground truth core points then act as an origin to the vectors which relate to that core. The vectors are transformed from the polar coordinate system into the Cartesian coordinate system whereby the standard deviation of all the x-coordinates and the y-coordinates which are attached to the respective ground truth core is calculated. The measure is simply the area of the ellipse, which is calculated using Equation 4-34. Therefore the area of the ellipse represents the area within one standard deviation of the mean of the detected core points. The smaller the area, the smaller the variation of the core point.

$$area = \pi\sigma_x\sigma_y \qquad \textbf{4-34}$$

Where:

area is the area of the ellipse representing the variation of the detected core point.

σ_x and σ_y are the standard deviations in the x and y axis.

In addition to the area, the rates at which the core points are dropped and added are also investigated. The drop rate (DR) is basically the rate at which cores are missed and the add rate (AR) is the rate at which spurious cores are added. These are simply calculated as the number of observed missing / additional core points divided by the total number of ground truth core points. This is further expanded to primary cores as these are generally the most used reference point. Table 4-2 to Table 4-9 shows the extracted statistics for all the methods from both database sets.

The tables are further broken down into the different types of fingerprints, and it is noticeable that the arch-type fingerprint does not make an appearance. This is due to the fact that an arch-type fingerprint does not contain any singularities. Furthermore, in some table entries an additional value is presented within a pair of brackets. This value denotes the same test but only taking into account the primary cores. This means the detected core which matches the primary core is investigated and the others are rejected.

Table 4-2: Results for SCM Reference Point Detection Method - FVC2004 DB1

	Left Loop	Right Loop	Whorl	Double Loop	Total
μ_{SCM}	141.287	316.148	124.771	143.204	206.143
σ_{SCM}	215.278	588.485	255.072	125.053	388.785
DR_{SCM}	0.04	0.037	0.052	0.046	0.043

Table 4-3: Results for PI Singularity Detection Method - FVC2004 DB1

	Left Loop	Right Loop	Whorl	Double Loop	Total
μ_{PI}	82.752	164,569	169.517 (77.032)	115.707 (84.878)	170.755 (150.422)
σ_{PI}	49.359	233.852	172.275 (85.513)	80.79 (67.315)	428.536 (425.772)
DR_{PI}	0.04	0.067	0.11 (0.052)	0.123 (0.172)	0.086 (0.072)
AR_{PI}	0.015	0.122	0	0.006	0.008

Table 4-4: Results for OT16 Core Detection Method - FVC2004 DB1

	Left Loop	Right Loop	Whorl	Double Loop	Total
μ_{OT16}	44.89	56.306	119.956 (31.328)	87.004 (49.934)	64.732 (46.512)
σ_{OT16}	32.094	43.339	202.06 (26.528)	54.683 (47.271)	95.574 (37.506)
DR_{OT16}	0.184	0.154	0.325 (0.366)	0.392 (0.322)	0.255 (0.227)
AR_{OT16}	0.004	0.004	0	0	0.002

Table 4-5: Results for OT32 Core Detection Method - FVC2004 DB1

	Left Loop	Right Loop	Whorl	Double Loop	Total
μ_{OT32}	48.829	54.577	163.146 (30.812)	57.735 (37.383)	72.118 (45.804)
σ_{OT32}	38.52	49.887	218.603 (22.655)	44.502 (23)	107.864 (39.51)
DR_{OT32}	0.092	0.093	0.192 (0.172)	0.234 (0.183)	0.15 (0.123)
AR_{OT32}	0.018	0.02	0.007	0.087	0.013

Table 4-6: Results for SCM Reference Point Detection Method - FVC2004 DB2

	Left Loop	Right Loop	Whorl	Double Loop	Total
μ_{SCM}	222.708	163.799	121.758	80.201	172.974
σ_{SCM}	239.437	171.973	220.817	79.923	211.936
DR_{SCM}	0.029	0.037	0.074	0.085	0.047

Table 4-7: Results for PI Singularity Detection Method - FVC2004 DB2

	Left Loop	Right Loop	Whorl	Double Loop	Total
μ_{PI}	103.874	113.2	202.781 (64.111)	122.765 (53.776)	134.889 (92.097)
σ_{PI}	106.254	103.288	166.219 (43.303)	104.873 (21.684)	129.738 (89.582)
DR_{PI}	0.054	0.041	0.086 (0.045)	0.065 (0)	0.066 (0.043)
AR_{PI}	0.098	0.086	0.02	0.042	0.083

Table 4-8: Results for OT16 Core Detection Method - FVC2004 DB2

	Left Loop	Right Loop	Whorl	Double Loop	Total
μ_{OT16}	52.694	48.123	113.5 (33.96)	39.96 (25.294)	67.38 (44.015)
σ_{OT16}	36.974	33.402	95.941 (44.48)	34.39 (11.969)	64.749 (37.558)
DR_{OT16}	0.187	0.169	0.415 (0.287)	0.398 (0.447)	0.299 (0.227)
AR_{OT16}	0	0.029	0.005	0	0.009

Table 4-9: Results for OT32 Core Detection Method - FVC2004 DB2

	Left Loop	Right Loop	Whorl	Double Loop	Total
μ_{OT32}	50.625	48.428	87.573 (22.463)	43.869 (23.764)	59.088 (39.906)
σ_{OT32}	38.083	41.044	76.282 (12.202)	25.507 (12.65)	53.89 (34.795)
DR_{OT32}	0.125	0.144	0.248 (0.203)	0.204 (0.191)	0.187 (0.158)
AR_{OT32}	0.068	0.021	0.013	0.011	0.013

It can be seen from the tables that the OT16 and OT32 means and standard deviations outperform the other two methods. It can also be seen that, although the means and standard deviations of OT16 and OT32 are comparable, the drop rate is considerably lower in that of OT32. This is because, with fewer traces, there are subsequently fewer traces that will exhibit significant turning to prevent being rejected in the core calculation stage. With more traces, there is a greater likelihood that more traces will turn substantially and be used in the core calculation. However, comparing the drop rates of the orthogonal tracing algorithms with that of the other two methods, it is clear that it is considerably larger. This is one drawback to the proposed algorithm as it is liable to miss a core point more often than the others, although a greater drop rate is preferable than that

of an inconsistent location or spurious cores. This is because the image would need to be processed entirely to determine whether a poor core estimate has occurred, whereas if the core is missed and no primary core is present, the image would normally be rejected outright. Furthermore, it is also interesting to note that the mean, standard deviations and drop rate for the Poincare Index, OT16 and OT32 methods dramatically improve when only the primary cores are taken into account and that the primary core point is nearly always detected. It is usually the secondary core, in fingerprints that have two cores, which is usually dropped.

Another drawback to the proposed algorithms is the execution time. Due to the variability in the number of points on a trace and the number of traces extracted, this variability is also reflected in the execution times. The Poincare Index and Sine Component Map approached maintain the same number of iterations and accesses throughout images of the same database. This ensures that the execution time is consistent on a database basis. This is reflected in Figure 4-27 and Figure 4-28, where the box plots representing the proposed method have greater area and are consistently higher than the box plots of the other methods.

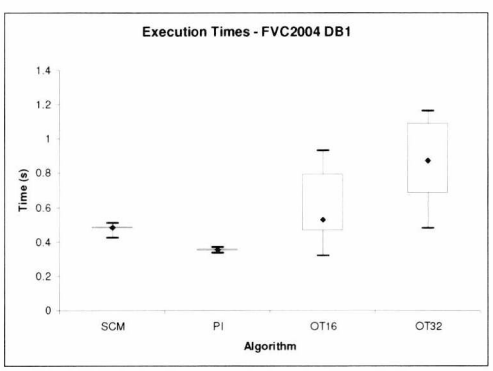


Figure 4-27: Execution Times for FVC2004 DB1

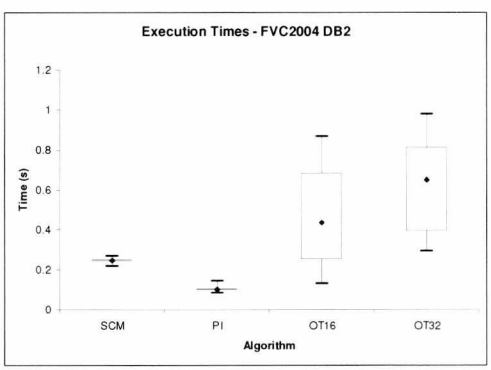


Figure 4-28: Execution Times for FVC2004 DB2

4.3.3. Summary

An intuitive and novel algorithm for the detection of core points within a fingerprint image is presented. The method exploits the nature of the ridges above a core point by tracing the ridges orthogonally as this will lead directly to the core point. Although the computational complexity of the algorithm is quite high, especially with regards to the two existing methods investigated, the algorithm is conceptually simple and does not run substantially slower than the existing methods.

The results of the algorithm demonstrate that the algorithm has outperformed the existing methods in both accuracy and reproducibility. This is due to the termination of traces "on the fly" depending on the quality of the region and the rejection of ineffective traces. By employing these processes prevents poor traces adversely affecting the core point calculation. This results in a better and consistent estimation of the core points.

The proposed algorithm detects all cores, with minimal variation regardless of the rotation of the fingerprint, as long as the traces turn through a significant angle. By increasing the number of traces ensures that the drop rate of cores is low. However, this incurs extra computational cost. The orthogonal tracing method may be able to extract cores from the majority of fingerprints, it is however unable to determine a reference point in an arch type fingerprint. This is because the arch-type fingerprint does not have any cores present so the traces will not turn enough. However, the traces extracted from an arch-type fingerprint may at some point be employed to determine a reproducible reference point.

4.4. Conclusion

In this chapter, a selection of level 1, fingerprint features are investigated and assessed. It has been determined that the orientation field derived descriptors benefit from reduced dimensionality and greater dimension-space. However, this deduction is based on the reproducibility of the descriptor on account of the separation of the sample class means and not on the potential effective bits these descriptors may provide. Furthermore, a novel core detection method has been proposed, which produces an accurate and consistent core point. Although, not all singularities have been extracted in the implementation proposed, the technique could potentially be transferrable to detecting delta points, arch-type reference points and even providing estimates of singularities which do not lie within the captured image.

Chapter 5. Level 2 Features

Level-one features are a coarse representation of a fingerprint and therefore are not considered discriminatory enough to be used for identification purposes. This is because different fingerprints, belonging to the same fingerprint type, may produce very similar feature descriptors. For identification purposes finer detail is required. While level-one features are derived from the overall ridge structure, level-two features are derived from the ridge formations themselves. Within a fingerprint, many ridges will contain discontinuities, whereby the ridge may split (bifurcations) or end (terminations). These discontinuities are more commonly known as minutiae.

5.1. Minutiae

Minutiae, also known as Galton features [8], are by far the most popular feature currently employed by automated fingerprint recognition systems. At present, these features offer unrivalled discriminatory power and have, so far, maintained individuality. That is to say that no two different fingerprints, which have been discovered, have ever had the exact same minutiae configuration. In fact, based on present literature, the probability of two different people having the exact same fingerprint is 1 in 65 billion [6]. However, as this figure is based on the fingerprint as a whole, it may not be a true representation of the individuality of just the minutiae configuration.

Minutiae are generally classified as one of two types, which are bifurcations and terminations. There are other classifications that are sometimes employed. However these are just local configurations of bifurcations and endings. Many of these classifications are shown in Figure 5-1.

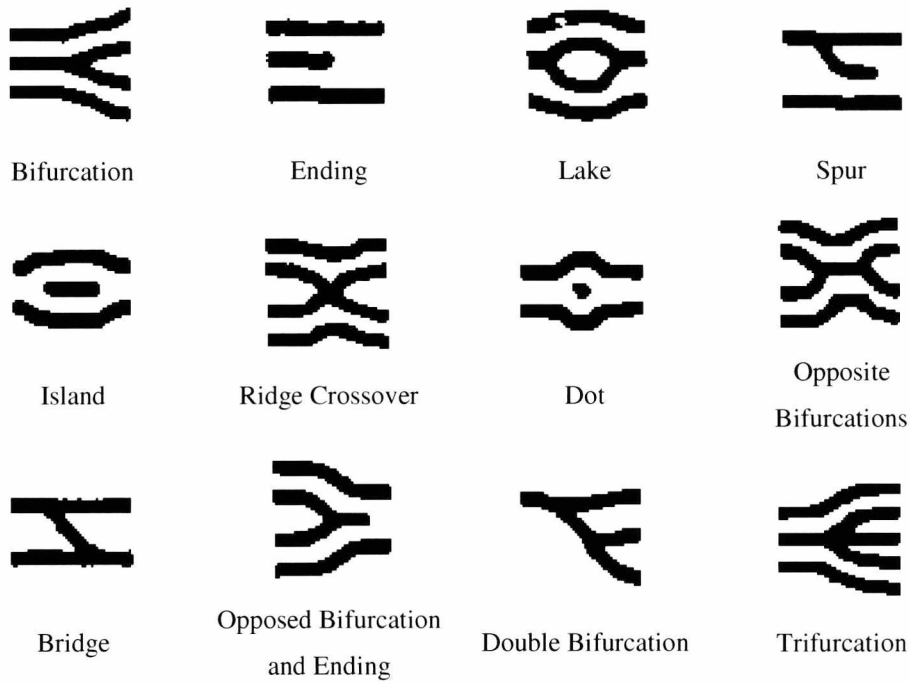


Figure 5-1: Example of Minutiae

It is evident from Figure 5-1 that all minutiae are combinations of the endings and bifurcations. In manual fingerprint matching, all these configurations can be taken into account. However, for automated fingerprint systems, only bifurcations and endings are normally detected. This is because noise in the image may cause spurious minutiae to appear. Minutiae which appear within a certain distance of each other are often a sign that the region is noisy and minutiae extraction about that area is unreliable. Therefore, many of the minutiae configurations presented in Figure 5-1 may have their quality rating degraded or they may be ignored entirely due to the close proximity of minutiae. Furthermore, bifurcations and ending are frequently exchanged (i.e. endings being detected as bifurcations and vice-versa due to noise / filtering methods), hence some systems do not discern between them [134, 163, 164]. In the following section, a brief explanation of popular minutiae extraction algorithms is presented.

5.1.1. Feature Extraction

Minutiae extraction methods are generally based on segmenting individual ridges from the background. This segmentation can be obtained either by adaptive ridge tracing [162] or by pixel-wise segmentation, which is traditionally attained by applying an adaptive threshold algorithm to the greyscale image [60, 165]. More recent methods have investigated the responses of complex symmetry filters [166] and principal Gabor basis functions [167] in order to detect minutiae. However, judging by the reviewed literature, ridge segmentation methods are the most prevalent approaches to minutiae extraction.

Different ridge segmentation-based minutiae extractors require varying degrees of pre-processing. The adaptive ridge tracing methodologies [162] do not require any pre-processing whatsoever as they can adapt to a certain degree of noise during the tracing. Ridge tracing is essentially a direct route to a skeleton image (i.e. an image where the ridges are only a pixel wide). The method presented in [162] extracts a skeleton image based on the concept that ridges are the local minima of points perpendicular to the dominant ridge direction.

The ridge tracing algorithm is an iterative approach. From a starting point, which is determined from the local minimum in an oriented window orthogonal to the local dominant ridge orientation, the ridges are traversed, step-by-step, in the direction of the local ridge orientation. After each movement to a candidate point, a ridge projection orthogonal to the local ridge orientation at this point is performed and the closest, local maximum of the projection from the old point is deemed the new point. The tracing is terminated and the algorithm moves on to the next starting point if one of three conditions is met. These are 1) the angle between the old point and the new point is greater than a threshold, 2) the trace intersects that of another trace and 3) the new point lies outside the ROI or image area. The distance moved each time is adaptive and this is based on the rate of change in the ridge orientation, akin to a car slowing down before a bend. This procedure is continued till the trace terminates, and the algorithm finishes when there are no other ridges left in the ROI to trace.

The remaining ridge segmentation based methods employ binarisation. Binarisation essentially separates the ridges from the valleys, whereby the ridges are represented by zeroes and the valleys by non-zero values (typically 1s or 255s). The main obstacle when applying a threshold to fingerprint images is that the background and foreground levels are not always constant throughout the image. In addition, the separation between the background and ridges may not be distinct. Therefore, algorithms which are derived from a binarised image will either need enhancing before binarisation to increase the distinctiveness or after binarisation to remove spurious artefacts.

There are a multitude of approaches to binarising an image, although due to the inconsistent threshold level of local neighbourhoods within a fingerprint image, a locally adaptive threshold method is often employed. This could be as simple as using the mean of a local neighbourhood as the threshold or using Otsu's threshold algorithm [3] on a local scale. However, it was found that these threshold methods provided inferior separation on low variance regions and, as a result, often produced "ghosting" effects (i.e. where only one level is apparent in a local region, but the method creates two levels). To counter this problem, an iterative Otsu's threshold algorithm based on the Lorentz Information Measure is presented in [60], and if insufficient information is present to confidently threshold a region, it is postponed until a larger resolution provides more information.

From a binarised image, without further pre-processing, minutiae locations can be obtained by using either chain-coded contours [131] or by run-length encodes [168]. A chain-coded contour is a representation of the contours of an image, and in the case of a binarised image, the ridge edges (i.e. where black pixels immediately neighbour white pixels). The contour is traced and for each point along the trace, certain information regarding that point is stored. The most important information stored are the x and y coordinate of the point, quantized slope pointing to the next point in the trace and the curvature. This information is required to obtain the minutiae locations.

The contours are traced anticlockwise from an arbitrary start point on each contour and the trace stops when the contour either exits the ROI, or the image boundary, or when the trace returns to the start point. For each point on the trace, the information stated previously is determined. It is not really necessary to store the information as the image does not need to be reconstructed. The objective is to determine the points of highest curvature on the trace as these points are the minutiae. To distinguish between bifurcations and endings, the direction of the turn is required. A bifurcation is denoted, when the initial transition is anticlockwise, as a right-hand turn, while a left-hand turn represents a termination. The converse is true when the initial movement is in a clockwise direction. A threshold to determine whether enough curvature is exhibited is used to determine whether a point lies about a minutia. This will produce a cluster of points belonging to the minutia, whereby the minutia location is set as the centre point.

The run-length codes method [168] is based on run length codes, which are primarily used to reduce the storage size of a binarised image. A run length represents the number of continuous black pixels along the horizontal scan line. However, in this implementation, the run length codes are determined along both the horizontal and vertical scan lines. This is to accommodate the variety of orientations on which minutiae can lie on. The terminations are detected by applying certain heuristic rules, which determine the adjacency of run lengths. If only one adjacent run length is present in the previous scan line or the next scan line, then a termination has been found and minutia location is taken as the centre point of the run length. In order to extract the bifurcations, the same process is executed. However the binarised image is inverted first. This is due to the duality of a fingerprint where ridge terminations are valley bifurcations and valley terminations are ridge bifurcations.

Another popular method for extracting minutiae is to determine the skeleton of the ridges. This can be achieved by using existing image processing methods [169], which can extract the ridge skeleton from binarised images, or by using the direct method of adaptive ridge tracing [162]. An example of the result of the ridge skeleton is shown in Figure 5-2.

Figure 5-2: Example of Ridge Skeleton

The minutiae are generally extracted from a skeleton image by examining the immediate eight-pixel neighbourhood, i.e. the eight pixels that are directly adjacent to a pixel. The majority of methods employ a count whereby only one adjacent pixel denotes a termination and three adjacent pixels represents a bifurcation [134]. However, using this approach causes many spurious minutiae. By examining Figure 5-2 closely, it is clear that the skeleton of many of the ridges produce spurious bifurcations and endings. These can occur due to a flaw in the skeleton extraction process or noise in the binarised image. Therefore, post-processing is required to remove these false minutiae, which generally consists of investigating minutiae proximity, quality of the regions in which minutiae lie and application of other heuristic rules [134].

The minutiae extractors that were reviewed all rely on the accuracy of the segmented ridges. Accurate ridge segmentation is a very difficult problem and additional artefacts that result from poor ridge extraction produce numerous spurious minutia. It is paramount to reduce the appearances of spurious minutiae as these can severely damage the reproducibility of the extracted feature vector. Therefore, in order to maintain the highest possible level of consistency an off-the-shelf minutiae extractor [59], which has performed consistently well in fingerprint verification competitions, has been employed.

5.1.2. Experimental Results

Minutiae are currently the most discriminatory feature that can be extracted using automated methods [170]. However they are also probably the most variant feature. Minutiae can be dropped (i.e. missing) or exchanged, which means the minutia type has switched from a termination to bifurcation or vice-versa. In addition, many spurious minutiae may remain even after post-processing. Therefore, in order to use minutiae effectively, a comparison phase is generally implemented, whereby a registered set of good quality minutiae are stored (template), in one of a variety of representations, and matched with a query set of minutiae. It is not necessary for all minutiae to match, which accommodates the variation induced during the extraction phase or the

influence of external noise. In the desired application however, template matching is omitted, which makes consistent minutiae-based descriptor difficult to obtain.

There are a multitude of minutiae matching technique available in the current literature, and many are reviewed in [170]. However, in this implementation, reliance on stored information is discouraged. This makes it very difficult to incorporate minutiae to their full potential. Unfortunately, due to the variation in the detected location and the number of minutiae, it is not a trivial task as minutiae maybe translated or missing between samples of the same fingerprint. A possible feature representation is to partition the fingerprint according to one of the ROI schemes presented in the Testing Framework, Chapter 3 and count the number of minutiae which are present within each sub-ROI. This was rejected because minutia could translate between neighbouring sub-ROIs in different samples of the same fingerprint if the minutia lay in close proximity to the sub-ROI borders. As the number of minutiae within each sub-ROI would most probably be in single digits, due to the proximity restrictions in many post-processing methods, the problem of floating / missing minutiae would not be handled effectively. Even if ranges were implemented to handle this problem, individuality of the feature would be severely affected. Therefore, an alternative, novel relational feature was devised.

For these experiments, it was proposed that the relational distances between all minutiae are employed, not discriminating between the different minutiae types to minimize variance. This is because the relational distance is robust to both rotational and translational variance (disregarding the small variations due to elastic distortion caused by inconsistent pressure between samples). Furthermore, by simply normalizing the distances using the average inter-ridge distance (i.e. average ridge width) the feature vector becomes robust to scale variations too. This is important because pixel distances are sensitive to the scale of the image. For instance the average inter ridge distance for a fingerprint image whose resolution is 300dpi is 4-5 pixels [171], whereas for a resolution of 500dpi it is approximately 10 pixels [24]. The final novel univariate descriptor is calculated by employing Equation 5-1.

$$r = \frac{1}{\delta}\sum_{i=0}^{i<N}\sum_{j=i+1}^{j<N}\sqrt{\left(M_{i,x}-M_{j,x}\right)^2+\left(M_{i,y}-M_{j,y}\right)^2} \qquad \textbf{5-1}$$

Where:

N is the number of minutiae detected.

$M_{i,x}$ and $M_{i,y}$ are the x and y coordinates respectively of minutia i.

δ is the average inter-ridge distance.

r is the final descriptor value.

As mentioned previously, minutiae are highly variable with regard to location and count. It is expected that when only a single image is used the feature will be highly unstable because a difference of only a couple of minutiae could significantly change the feature value. However, the number of samples taken at any one time is not restricted to 1. It was always intended that multiple samples would be used because, without templates, the intra-sample variances need to be suppressed as much as possible. Averaging features from multiple samples is another method of reducing the influence of the outliers. Although for experimental purposes there are no limits to how many samples can be obtained at one time, in practice users may become irritated if too many samples are required consecutively. Therefore, in the following experiments the number of samples is investigated, whereby n number of samples is taken and the average feature value is the final descriptor. An alternative method would be to summate only the common minutiae between the n samples. However, averaging the feature values was favoured over the common minutiae approach because it is simpler, more computationally efficient and that no additional benefit was expected from employing only common minutiae.

In order to execute these experiments, p permutations (producing a set of subsets containing n samples, P) was obtained for each of the 100 users (for both FVC2004 DB1 and DB2), whereby p was obtained by employing the binomial coefficient (N_i choose n, where N_i is the number of samples in class i) and the average from each permutation was used as an observation for that user. Equation 5-2 presents this calculation.

$$o_{i,j} = \frac{1}{n} \sum_{s \in P_{i,j}} r_s \qquad \textbf{5-2}$$

Where:

s is a samples from the j^{th} subset from the set of user i's permutations ($P_{i,j}$).

n is the number of samples in each subset.

r_s is the feature value of sample s.

$o_{i,j}$ is the j^{th} observation of user i. Where $0,\ldots,j,\ldots p_i$ and p_i is the number of permutations for user i.

The F-Test value (F) was then determined based on all observations for each user for an increasing subset size, whereby the subset sizes range from 1 to 7 samples (n). In this experiment, all minutiae detected in each fingerprint were used to calculate the final feature value (r). Therefore the FULL ROI (see Testing Framework, chapter 3) scheme was implemented. The results of this experiment, for both FVC2004 DB1 and DB2, are presented in Figure 5-3.

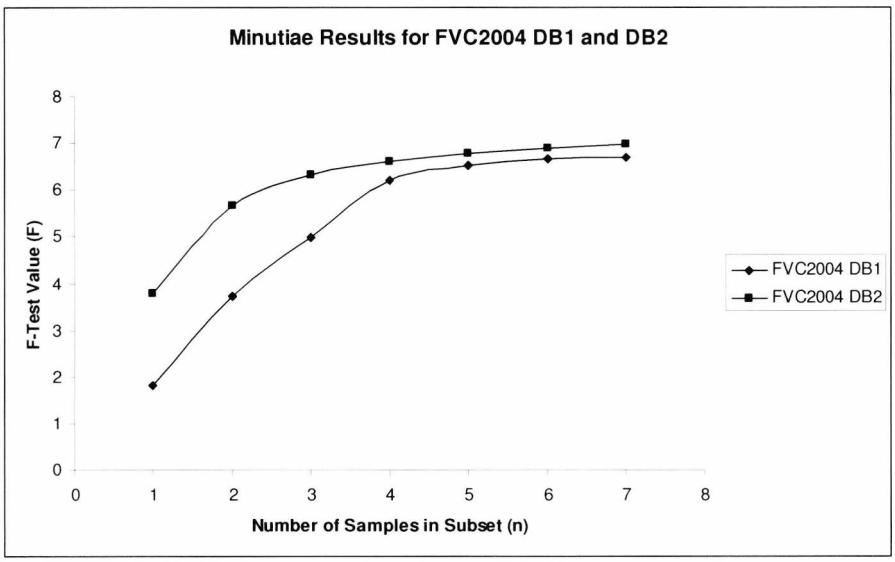


Figure 5-3: Minutiae Results using the FULL ROI Scheme for FVC2004 DB1 and DB2

The graph presented in Figure 5-3 depicts a common trend, whereby *F* increases when the number of samples in the subset and the number of observations increase. This confirms that when the average is calculated from a larger subset of samples, the feature value is slightly more stable as the variations are suppressed further. Furthermore, the shape of the graph is interesting in that both curves begin to flatten out when the number of samples in each subset exceeds 3 or 4. This could be accounted to the fact that different permutations, obtained from a maximum of 8 different samples, were implemented and that this trend might not flatten out as early if further samples were acquired. However, based on the available samples, a compromise between the subset size and the performance can be determined. It is clear from the curves that a subset of 4, for both databases, provides near-optimal performance (as determined from the available samples) from a feasible number of samples.

A supposition was made that, in a practical system, users would become irritable if they had to provide too many consecutive impressions and that the number of samples should be limited to 3. Obviously the ideal number of samples required would be 1, but it is evident from the curves in Figure 5-3 that increasing the number of consecutive samples to 3 provides over a 2-fold and a 1.5-fold increase in databases 1 and 2 respectively.

The previous tests were conducted using the FULL ROI scheme and it is expected, due to partial overlap (i.e. only a small common region is present between different samples of the same fingerprint), that this is not the optimal approach when implementing a minutiae vector. Therefore, to investigate the adverse affects attributed to partial overlap, experiments were executed that employed a GBLC ROI scheme of varying sizes. It is expected that the F-test values obtained from employing a GBLC ROI scheme should increase compared to that of the FULL ROI scheme. Furthermore, the ROI scheme can be placed according to a manually (MAN) or automatically

(AUTO) detected reference point, which can affect the descriptor. It is expected that MAN should outperform AUTO due to the difficulties of reference point detection in arch-type fingerprints and when the true reference point is located near the boundaries of the image. The results for these experiments are shown in Figure 5-4 and Figure 5-5, where Figure 5-4 and Figure 5-5 implemented a GBLC ROI scheme with a radius of 40 and 60 pixels respectively.

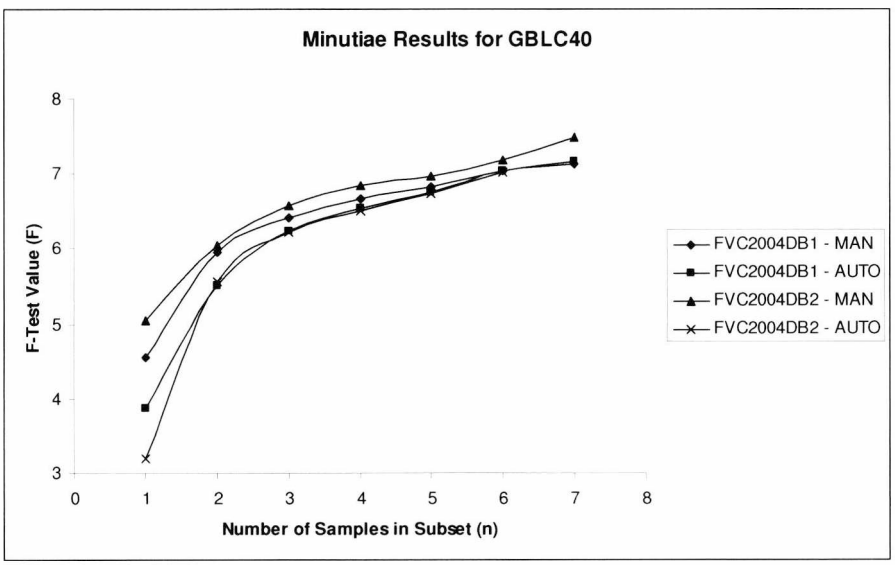


Figure 5-4: Minutiae Results for FVC2004 DB1 and DB2 using a GBLC ROI Scheme with Radius of 40 Pixels.

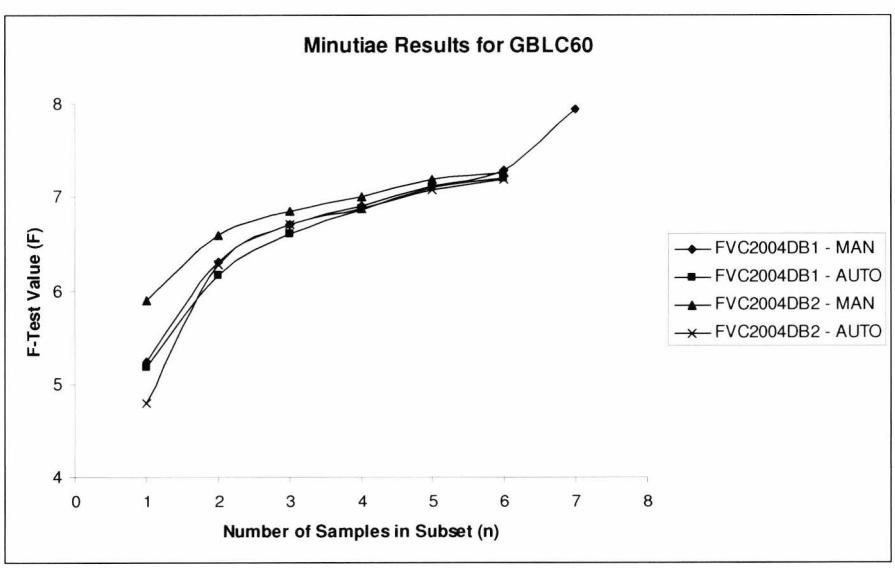


Figure 5-5: Minutiae Results for FVC2004 DB1 and DB2 using a GBLC ROI Scheme with Radius of 60 Pixels

From Figure 5-4 and Figure 5-5, it is apparent that the direct, seemingly-monotonic relationship between the number of samples within a subset and F is maintained. It is also clear that there is a small increase in F as the size of the ROI increases. However, it is difficult to determine whether

this increase is attributed to the slightly superior discriminating power of the feature value or to the reduction of samples being investigated. As the ROI increases, the probability of the ROI extending past the fingerprint's boundaries also increases, which increases the rejection rate. In fact, a GBLC ROI scheme with a radius of 80 pixels was also investigated, but due to the number of samples rejected there were insufficient samples that remained and a reliable estimate of F could not be established. It was decided that at least 3 samples were required in each class for that class to be included in the F-Test. This was decided because fewer samples would provide less information regarding the intra-sample variance, while permitting a greater number of classes to be included in the F-Test.

The rejection rate affects the results considerably and it can be seen that there are fewer tests conducted in each experiment because too fewer samples remained and a subset of 7 samples was not possible. It was determined that the number of classes and total number of sample remained relatively consistent until the number of samples in the subset exceeded 6 in Figure 5-4. This was, unsurprisingly, lower in Figure 5-5, where the number of classes and number of total samples plummeted when the number of samples in a subset exceeded 4. Therefore, the increase of F when n exceeds those values in their respective experiments could be attributed to a less-densely populated feature space.

Another significant feature of the results obtained from the GBLC ROI scheme experiments is the marked improvement of F when a single sample is employed. This supports the notion that a consistently placed ROI scheme improves the reproducibility of the feature value. Furthermore, a significant deterioration in F is not exhibited when an automatically detected reference point is employed, except when only 1 sample is used. This is attributed to the increased influence of a poorly detected reference point as it is not suppressed by averaging.

5.1.3. Conclusion

In this chapter a novel representation of minutiae is presented. The summation of the normalized relative distances between minutiae provides a rotational, translational and scale invariant feature. By averaging the sums of multiple samples, whatever variance is evident is further suppressed. From the experiments conducted, a slight improvement to the reproducibility of the feature value was apparent when a restricted region of interest about the reference point, rather than the whole image, was implemented. This further improved when the region of interest size also increased. However, the feature space became less-densely populated as the region of interest size increased as the region of interest extended past the boundaries of the image in more samples. Therefore, the best performance should not be measured on the F-Test value alone, but also on the number of classes and total

samples included in the test. It was found that the subset size that produced the best F-Test value, accounting for the number of classes in the feature space, was 3 with a global region of interest of 60 pixels, whereby 94% of the classes were represented in the feature space with at least 5 samples per class.

Chapter 6. Level 3 Features

The lower-level fingerprint features previously investigated are simple, intuitive and in most cases analogous to, or derived from, their respective levels originally defined by the manual detection precepts. Likewise, many level 3 features are algorithmic approaches to extracting features analogous to those extracted by expert (human) fingerprint analysts. Level 3 features are the fine details (i.e. pores, ridge widths, shapes and edge contours) of individual ridges and the location and identification of other artefacts (i.e. scars, creases, warts and insipient ridges). However these features are not apparent in all fingerprint images. Pores and edge shapes are features which all fingerprints possess but very high resolution scans are required in order to effectively extract them. Scars, deformations, creases and warts are permanent artefacts which do not occur in everyone's fingerprints, but possess high variability.

Although these artefacts are relatively discriminating, as not all fingerprints possess them, their appearance and the variations in the shape (due to elastic distortions and partial overlap) and locations (due to affine transformations) are highly volatile. One approach is to independently test for each of these infrequent and volatile features and extract if an image possesses one of them. This is not a preferred method as the high volatility will undoubtedly influence the intra-sample variance, which conflicts with the predefined desired properties of a feature. Furthermore, computational resources are wasted when the image does not possess one of these artefacts. However, an alternative approach is to extract features from an entire region as this would implicitly include any level 3 features (in fact all feature levels) apparent in that region, while providing additional resilience to the volatile nature of these features. These region-extracted features essentially extract characteristics based on the overall shape of the region. This is why level 3 features are often defined as features pertaining to shape [8]. In this chapter, previous work is initially outlined, and then a selection of level 3 features is investigated, and in many cases enhanced, to determine suitability for direct encryption key generation.

6.1. Literature Review

The underlying shapes, produced by the perturbations in individual ridge structures, of a fingerprint provide an alternative approach to characterising fingerprints. Even scars, creases and warts can be defined similarly as the inclusion of these artefacts disturb multiple, and frequently adjacent, ridges. These artefacts are simple to detect manually and provide a useful discriminating feature, similarly to how a tattoo or blemish / scarring of the skin can aid criminal identification. However, it is not as simple to consistently detect and extract these artefacts independently automatically. Therefore, it is simpler to determine shape / texture descriptors as the presence of these artefacts influence the underlying shape of the fingerprint pattern.

Given an image, or a sub-region of an image, shape / texture descriptors provide measurements that describe the composition of the overall pattern of the input. The overall pattern can be described by a multitude of components, ranging from primitive shapes (i.e. lines, circles and rectangles) to abstract texture features. One of the most versatile features that can be extracted from the fingerprint is the inter-ridge distance, and by extension, its frequency components. However, this feature could arguably be categorised as a level-1 feature, due to its connection with the orientation field. The inter-ridge distances of local neighbourhoods are primarily used as parameters to enhancement algorithms [30] and provide supplementary information for quality estimation [172]. However, a field of either the frequency or inter-ridge distances (frequency is just the reciprocal of the inter-ridge distance), similar to the construction of an orientation field, could be used as a feature describing the shape of the image.

The inter-ridge distance is essentially the width of the valleys, and due to ridge-valley duality, the ridge-widths are the inter-valley distance. This is measured in pixels and is often extracted using the x-signature approach defined in [2], whereby local ridge projections are attained by projecting pixel intensities from an oriented window based on the local orientation estimate. The ridge projection results in a sinusoid of varying deterioration related to the quality of the image. From the sinusoid, an estimation of the amplitude and wavelength can be determined, whereby the wavelength is an estimate of the ridge width. The amplitude can be used as an estimate of the strength of the sinusoid, which is primarily implemented as a means to provide an estimation of the quality and confidence of the determined wavelength. An alternative approach to ridge distance / frequency extraction is the Short-Time Fourier Transform method presented in [1]. By partitioning the image into overlapping blocks and using the Fast Fourier Transform (FFT) to obtain the blocks frequency components, an estimation of the local dominant frequency can be achieved. Further methods and a comprehensive review of ridge-width estimation algorithms are presented in [158, 173]. Furthermore, from the frequency / ridge-width estimations, ridge density / ridge count within local neighbourhoods can also be determined.

Due to the underlying oriented, periodic texture pattern of a fingerprint, it is not surprising that many methods look to capitalise on this. The drawback of the orientation and frequency fields is that the estimates used are single values, based on the dominant component within a local neighbourhood. These methods provide scant information regarding the shape / pattern residing in the local neighbourhood. However, additional features can be extracted from the STFT. When the FFT decomposes a 2D signal, the strength of its composite waveforms for each frequency and orientation is attained. This provides additional contextual information, not just the dominant direction and frequency. Other features can be extracted from the Fourier domain. Fourier energy is used frequently [1], although energy is affected by poor contrast and ridge structure and is therefore generally used at the pre-processing stage. Alternatively, a wedge-ring detector can be implemented [1, 150] to determine frequency and orientation variations. However, the Fourier

transform only provides robustness to translational variance. Therefore, in order to provide rotational and scale invariance, the result of the Fourier transform can be transformed again with the Mellin transform [174], from which invariant features can be extracted.

Although the Fourier transform is a useful transformation which provides directional and frequency information, the composite signals are not localised in space. Therefore, it is not known where in the image these composite signals occur. The STFT provides a rudimentary approach to localising the signal in space by transforming local neighbourhoods independently. However, the proportion of each composite signal is determined on a local scale and this does not reflect the true proportion with respect to the overall image. In order to localise the signal in frequency and time (i.e. space in a 2D signal) many methods employ wavelet analysis. Wavelet analysis decomposes a signal into different translations and dilations (daughter wavelets) of a short-lived oscillating waveform (mother wavelet), whereas the Fourier transform decomposes the image into sine and cosine waveforms of infinite length. Wavelet decomposition produces 3 details (obtained from the translations) and an approximation for each frequency / scale (obtained from the dilations), from which features can be extracted. These features include the mean and standard deviation [175], energy distribution [176] and the generalised Gaussian density [177] of the details.

Further image transforms can be implemented, from which additional features can be extracted. For instance, the Karhunen-Loeve transform (the discrete Karhunen-Loeve transform is more commonly known as Principal Component Analysis – PCA) is much used in biometric feature extraction and classification [178]. Primarily, PCA is used to reduce dimensionality of a feature vector, which could be the entire image [178] or another multi-dimensional feature, such as orientation fields [179] or Gabor features [180]. In most biometric applications, PCA is used for classification purposes, whereby a database of multiple samples from multiple classes is initially used to determine the eigenvectors, from which the eigenfeatures are derived. The feature vector is essentially the weight vector, which is the contribution (i.e. weight) of each eigenfeature that the query image is comprised of. This method requires the storage of the eigenfeatures, which is costly depending on the size of the initial feature vector and the number of eigenfeatures used, as the weight vector is determined from these. Although the storage of the eigenfeatures does not present any security concerns, as no individual user's data can be obtained if compromised, it does require maintaining backups because any unauthorised alteration to the eigenfeatures can render the system useless. Furthermore, the eigenfeatures are derived from an initial database and therefore, if additional users were to be included, the eigenfeatures may need to be recalculated. These are significant drawbacks to the PCA method when the intended application requires little, or no, storage of template or helper data.

An alternative transform that has been implemented in fingerprint feature extraction is the Hough transform. The Hough transform is primarily used for detecting and locating lines in images,

although the Hough transform can be adapted to determine other shapes. In [181], the fingerprint ridges are initially binarised and thinned, then for every pixel belonging to a given ridge, the accumulator bins are incremented corresponding to all possible straight lines that pass through the pixel to create the Hough space. The Hough space is then searched for peaks, which are feature points belonging to the ridge in the spatial domain. Although the matching approach compares the number of feature points of ridges in a query image with that of a template image, the sum of the inter-point distances (akin to the minutiae descriptor outlined in Level 2 Features chapter) could be used in a template-free implementation.

The fingerprint does not have to be transformed from the spatial domain in order to extract meaningful level-3 features. Fingerprint feature extraction is just a specialised field of image feature extraction and one of the most common shape descriptors used in general image processing is moments. Moments are statistical measures derived from the pixel intensities of an image although, due to rotational and translational variances, which are common to fingerprint acquisition, invariant moments [182] are required for this application. Moments maybe one of the most common descriptors in general image processing, but they seem underrepresented in fingerprint feature extraction. However, the current most popular level-3 feature descriptor for fingerprints is the Gabor-filterbank based FingerCode [24]. By employing Gabor filters, the resultant features are dependent on variations of both frequency and orientation, as these are the two main parameters of the Gabor filter.

The FingerCode approach has roots in the IrisCode [183] and it was initially used for pattern classification, although the technique has been further enhanced for identification purposes. In this approach, a ROI about the reference point is cropped and multiple filtered images are created, whereby the cropped region is filtered with a differently oriented Gabor filter. Each of the filtered images is tessellated similarly to the DISC ROI scheme (see Testing Framework, Chapter 3) and the average absolute deviation (AAD) of each block is extracted. When the Gabor kernel is oriented accordingly and its frequency is synchronised with the local ridge width, then the region is enhanced.

This provides a larger AAD because the variation between foreground and background would be greater than when the frequency and orientation are dissimilar to that of the local neighbourhood. This is because the foreground is smoothed into the background, resulting in a small AAD. Furthermore, the AAD will be relatively reduced when minutiae or other artefacts lie within that region (due to the variation in frequency and direction), providing a descriptor somewhat connected to the underlying texture. In addition to the FingerCode, other Gabor filterbank-based methods have been proposed, whereby the ROI has been changed to that of a GRID ROI scheme using the standard deviation as opposed to the AAD [184, 185], and additional features have been used in conjunction with the FingerCode [144, 186-188]

6.2. Frequency Field

The ridge pattern of a fingerprint is often described as an oriented, periodic texture pattern [28] and just by adopting this description, suggests potentially useful, measureable features. Orientations, on the local scale, have already been discussed and investigated and the orientation field exploits the oriented nature of the fingerprint. In this section the inter-ridge distance / frequency map, which exploits the periodic nature of the fingerprint, is investigated.

Local ridge frequencies (inter-ridge distance is essentially one wavelength, which is simply the reciprocal of frequency), similarly to local orientations, is a versatile feature. It is often used as helper information at the pre-processing stage for both segmentation and enhancement purposes. Furthermore, local frequency can also be extracted concurrently with the local orientation, with minimal additional computational costs.

6.2.1. Feature Extraction

The local frequency can easily be deduced from either the (see Orientation / Directional Fields section) spatial [2] or Fourier (STFT) domain [1] orientation field extraction methods. However, extraction of frequency from within the spatial domain is fraught with problems, whereas frequency extraction within the Fourier domain is straightforward as the Fourier domain localises the signal with respect to frequency. In the spatial domain, the local frequency is obtained by estimating the inter-ridge distance of local ridge projection (see X-Signature [2]) primarily using a peak-trough detector. However, the ridge projection rarely provides a clean signal so spurious peaks and troughs are present which affects the wavelength (inter-ridge distance) estimate. Insignificant peaks and troughs regularly appear within a ridge projection and this result in additional smaller wavelengths, which erroneously reduces the estimation of the average wavelength. Although these high frequency components will also appear in the Fourier domain, their relative impulse strength is negligible compared to that of the dominant frequency. This is because the impulse strengths are related to the amplitudes of the signal and the insignificant peaks and troughs have little amplitude. Therefore, in order to reduce the influence of the high frequency components, it was decided that the frequency map was to be determined using the following STFT (Short Time Fourier Transform) approach.

In [1], a frequency field was determined by partitioning the image into overlapping regions and transforming each region independently to the Fourier domain with the Fast Fourier Transform (FFT). However, given any region of the image, the region is transformed to the Fourier domain and the dominant frequency of that region is determined by implementing Equations 6-1 to 6-3.

$$\lambda = \frac{r.s}{N};\ if\ \ p(r) = \max(p(r)) \qquad \textbf{6-1}$$

$$p(r) = \sum_{\theta} p(r,\theta) \qquad \textbf{6-2}$$

$$p(r,\theta) = \frac{|F(r,\theta)|^2}{\sum_{r}\sum_{\theta}|F(r,\theta|^2} \qquad \textbf{6-3}$$

Where:

r is the distance. $r > 0$.

θ is the angle.

$F(r, \theta)$ is the Fourier coefficient at $r\angle\theta$.

λ is the dominant frequency.

N is the size of the window, which is a power of 2 (e.g. 32 in this application).

s is the sample rate, which is 1.

6.2.2. Experimental Results

In this section, numerous experiments are conducted in order to determine the suitability and the feasibility of implementing frequency (ridge width) in a template-free environment. A host of parameters are investigated by these experiments and the parameters are as follows.

- Neighbourhood size (8, 16, 24 and 32 pixels).
- Region of Interest (ROI) scheme (GRID and RAD – see Testing Framework, Chapter 3).
- Stage of pre-processing (BIN, RAW, NORM, ROT, ROTNORM and ROTBIN), denoted as PPS.
- Screening algorithm severity (Q).
- Manual or automatic detection of reference point and its orientation (MAN and AUTO).

Within a fingerprint, ridge widths suffer from both inter-ridge and intra-ridge variation. It is not feasible to determine the ridge width for each position along every ridge, therefore the dominant frequency of local regions is determined in order to suppress these variations. However, whilst conducting these experiments, it was witnessed that the ridge widths generally lay between 6 pixels and 15 pixels because the screening algorithm rejected images that exhibited poor ridge structure. This is a small range and any deviation will be significant. Coupled with the fact that the resolution is dependent on the size of the region being transformed with the FFT (i.e. only a distance of 4 pixels cover the range of 6 to 16 pixels in a 32 x 32 region and reduces further with the size of the region), may adversely affect uniqueness. It is therefore expected that the feature vectors extracted in these tests will not possess the ideal properties, which are desired.

6.2.2.1. Local Frequencies from GRID ROI Scheme (FREQGRID)

The grid ROI scheme is a popular scheme due to its simplicity. In this section, the performance of the GRID-based frequency descriptor is investigated across both databases 1 and 2 of the Fingerprint Verification Competition 2004 (FVC2004 DB1 and FVC2004 DB2 respectively) set. For these tests a 7 x 7 grid was implemented and the effect of the cell size (i.e. neighbourhood size) was investigated. A 7 x 7 grid was chosen as it facilitated concurrent extraction to OFGRID. These tests are separated into whether the reference point and its orientation was estimated manually (MAN) or automatically (AUTO). Figure 6-1 and Figure 6-2 present the results of this feature vector, whereby manual estimation was implemented, for screening severity of Q1 and Q2 respectively. The figures presented include the number of classes and the total number of samples populating the feature space for the stages of pre-processing undertaken, as the distribution of the feature space has a significant influence on the resultant statistics (denoted on the x-axis by the additional information, which is read as the PPS and the number of classes from the total number of samples). Furthermore, the left-hand scale belongs to the bar chart and the right-hand scale belongs to the line graph.

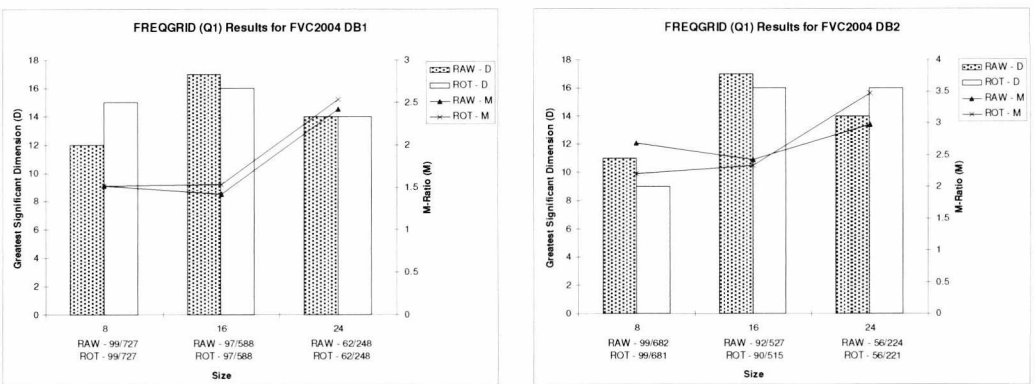


Figure 6-1: Q1 Results of FREQGRID (MAN) for FVC2004 DB1 (l) and DB2 (r)

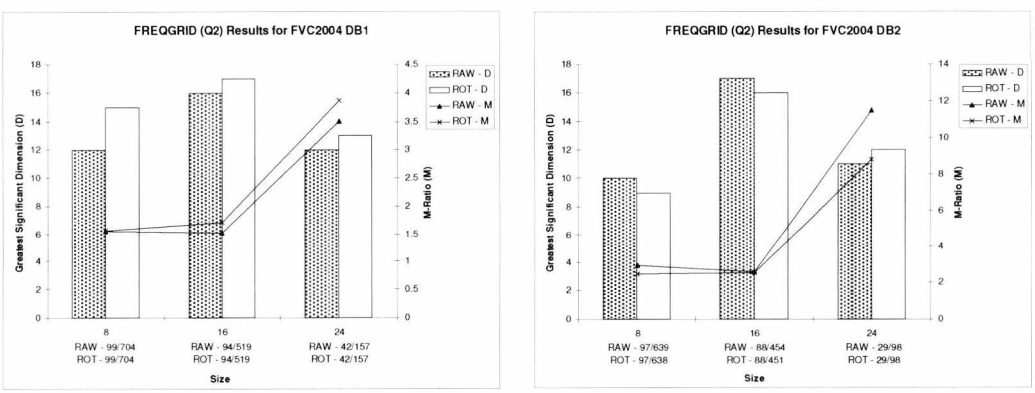


Figure 6-2: Q2 Results of FREQGRID (MAN) for FVC2004 DB1 (l) and DB2 (r)

The results depicted in Figure 6-1 and Figure 6-2 show that the greatest dimensionality, for both databases, occurs when the neighbourhood size is 16 pixels. As expected, the greater

dimensionality and improvement in the Mahalanobis ratio (M-Ratio) is exhibited as the screening algorithm gets stricter. However, due to sharp increase in the M-Ratio when the number of sample is significantly less, this is most probably attributed to the less-densely populated feature space. Although it is clear that the neighbourhood size peaks in terms of dimensionality at 16, the M-Ratio is not as obvious in regards to the size and stage of pre-processing. In the majority of cases, the untouched (RAW) image has greater significant dimensions, but the M-Ratio does not exhibit any dominance across all these tests and therefore, from the M-Ratio alone, no inferences can be made to which stage of pre-processing is most suitable. However, as there is not a considerable distance between the M-Ratios of RAW and ROT for the neighbourhood size of 16, the greater dimensionality indicates that RAW, ever-so-slightly, outperforms ROT. These highlighted trends are mirrored for the same tests that employed automatic reference point location and orientation estimation instead. The results of these tests are presented in Figure 6-3 and Figure 6-4.

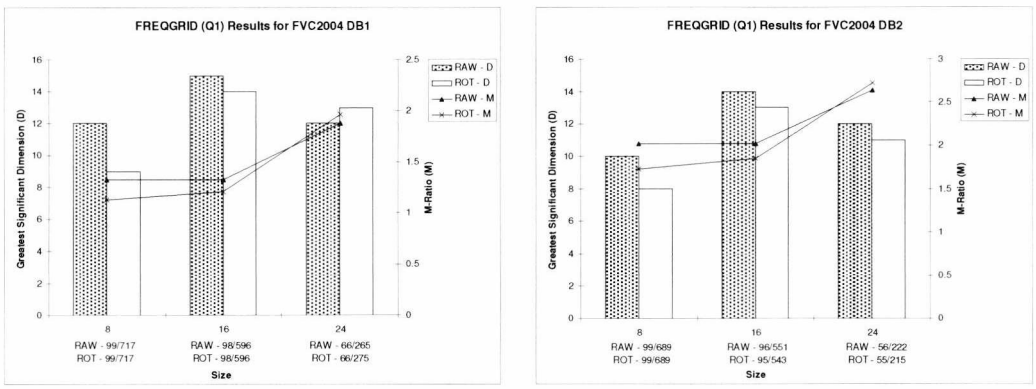


Figure 6-3: Q1 Results of FREQGRID (AUTO) for FVC2004 DB1 (l) and DB2 (r)

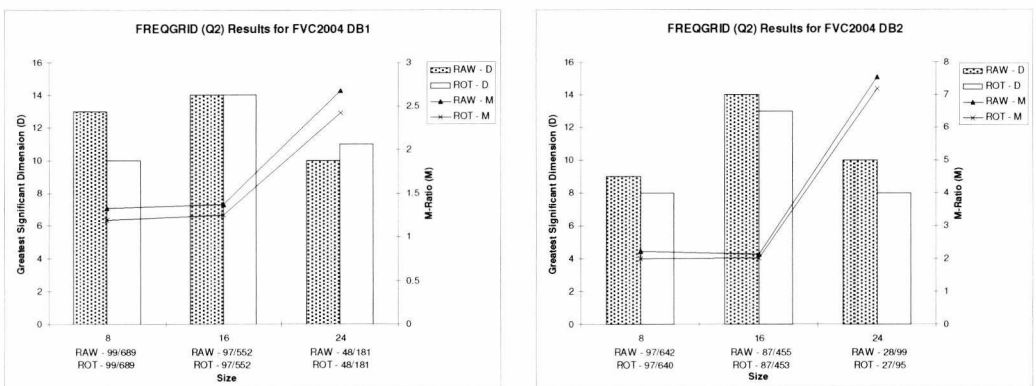


Figure 6-4: Q2 Results of FREQGRID (AUTO) for FVC2004 DB1 (l) and DB2 (r)

As mentioned previously, the results depicted in Figure 6-3 and Figure 6-4 are very similar to those presented in Figure 6-1 and Figure 6-2, except that both the effective dimensionality and M-Ratio are reduced. Both sets of graphs show that the greatest significant dimension occurs when the neighbourhood size is 16 pixels and that an untouched (RAW) image fairs slightly better than a rotated image (ROT). Furthermore, disregarding the results for when the neighbourhood size equals

24, the M-Ratio is consistently greater for the RAW tests than those of the ROT tests, albeit only slightly. This indicates that rotating the fingerprint prior to feature extraction does not provide any benefits, although it is more likely that the limited dimension-space, even before transforming to the Fourier domain, is causing significant class overlap.

6.2.2.2. Local Frequencies from RAD ROI Scheme (FREQRAD)

The previous section showed that minimal, if any, benefits were attained by rotating the image prior to feature extraction. This was unexpected and it could be due to the ROI scheme employed as a grid is not overly robust to rotational variance, even when an attempt to rotate the image to a base orientation is made. Therefore an investigation into an alternative ROI scheme was carried out. In this section the GRID ROI scheme is replaced with the RAD ROI scheme, which is based on the polar coordinate system and therefore should facilitate consistent feature extraction in rotated images. For these experiments, 3 tracks and 8 sectors that created 24 sub-ROIs (i.e. 24-dimension feature vector) were used as it facilitated concurrent extraction to OFRAD. However, a slight alteration from the RAD ROI scheme is made because the FFT requires a neighbourhood whose dimensions are a power of two. Therefore, instead of the circular sub-ROIs depicted, square sub-ROIs are employed. In order to compare this approach with that of FREQGRID, the same experiments as FREQGRID were carried out. Figure 6-5 to Figure 6-7 present the results of a RAD ROI scheme, whereby the reference point and orientation estimation was obtained manually, with increasing severity of screening across both databases.

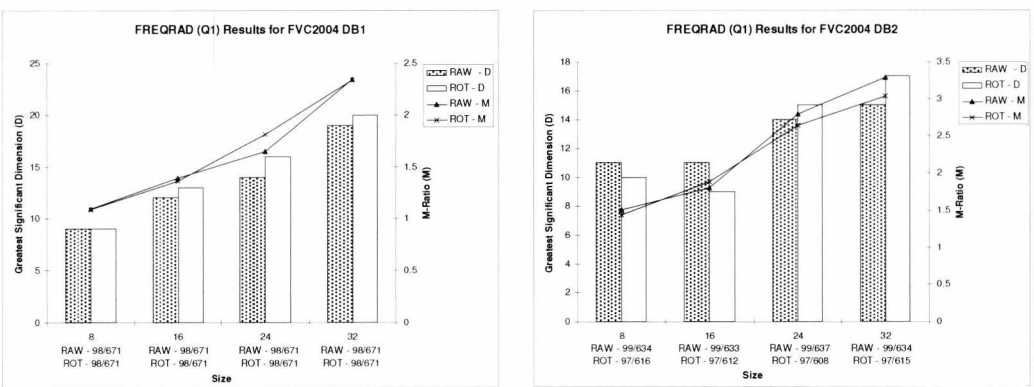
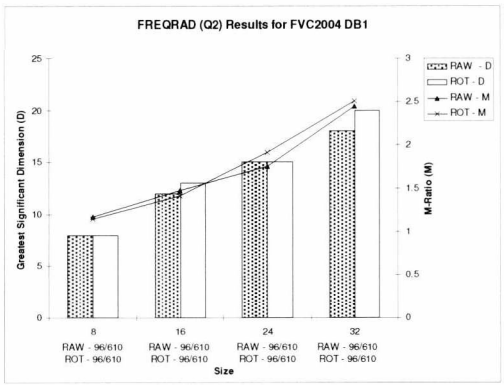


Figure 6-5: Q1 Results of FREQRAD (MAN) for FVC2004 DB1 (l) and DB2 (r)



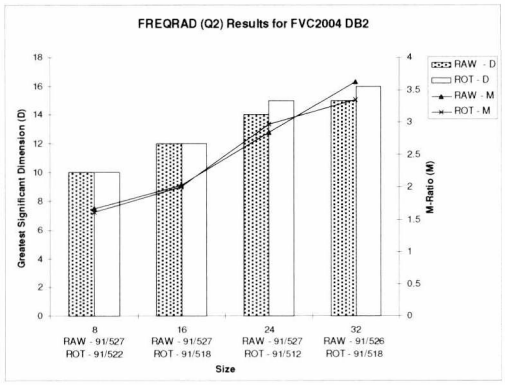
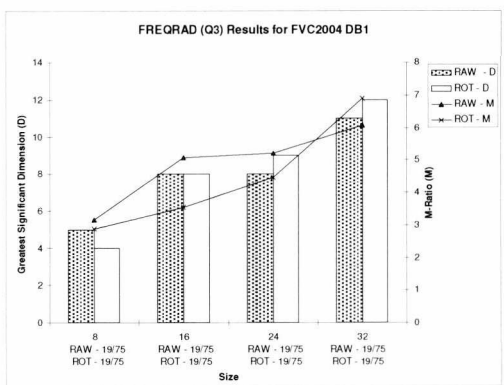


Figure 6-6: Q2 Results of FREQRAD (MAN) for FVC2004 DB1 (l) and DB2 (r)



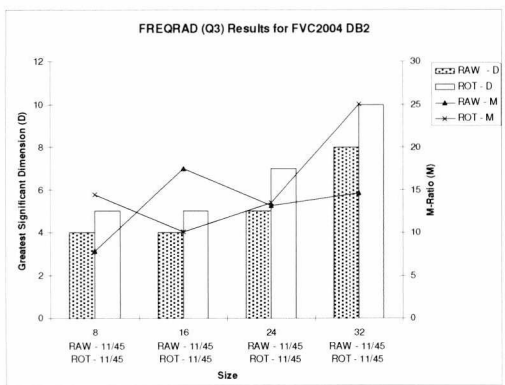


Figure 6-7: Q3 Results of FREQRAD (MAN) for FVC2004 DB1 (l) and DB2 (r)

From the results presented in Figure 6-5 to Figure 6-7, it is immediately evident that these experiments performed, in some respects, to expectations. As with many of the features, it is expected that as the size of the neighbourhood increases, the better the results should be. It was also expected that the ROT test should provide better results than the RAW experiments due to the ROI scheme's suitability for rotation. Both these expectations are supported by these results, whereby the greatest significant dimension increases monotonically as the neighbourhood size increases and that the results for the ROT tests, with regards to effective dimensionality, are slightly better. However, the magnitude of M-Ratio is disconcerting.

A small M-Ratio indicates that the distances between samples of the same class are similar to the distances between the means of different classes, which means significant overlapping is apparent. Identifying features that are well separated (i.e. with minimal or no overlap) is paramount, even more so than greater dimensionality as the stability and individuality of the subsequent encryption key is of the utmost importance. It is also evident from the results presented that, as the severity of the screening algorithm increases, the effective dimensionality decreases while the M-Ratio continues to increase. This is attributed to a less-densely populated feature space, where there are insufficient classes and samples per class to ensure the means of the classes are spread across all dimensions. Furthermore, the screening algorithm rejects more poor quality images reducing the number of samples within the feature space further while maintaining better quality descriptors.

These factors undoubtedly contribute to the improvements exhibited by the M-Ratio statistical test. Similar trends were also exhibited when the reference point location and orientation was estimated automatically although, as expected, a slight deterioration in the effective dimensionality and M-Ratio was also exhibited. Figure 6-8 to Figure 6-10 present the findings of the experiments conducted when automatic estimation is employed.

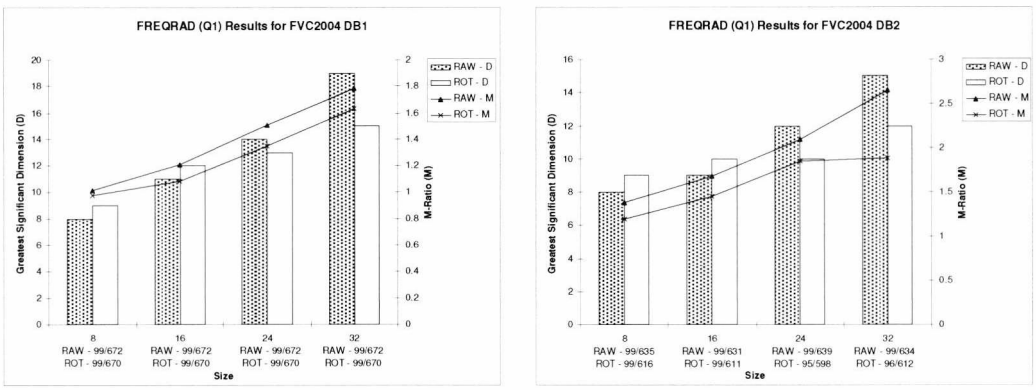


Figure 6-8: Q1 Results of FREQRAD (AUTO) for FVC2004 DB1 (l) and DB2 (r)

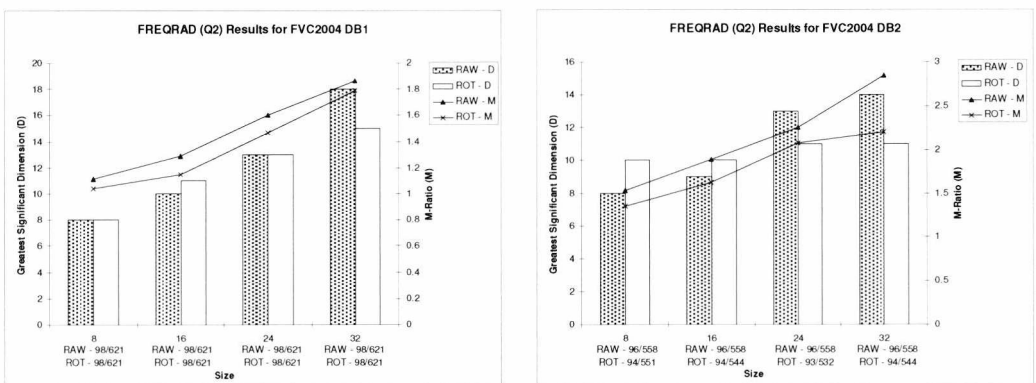


Figure 6-9: Q2 Results of FREQRAD (AUTO) for FVC2004 DB1 (l) and DB2 (r)

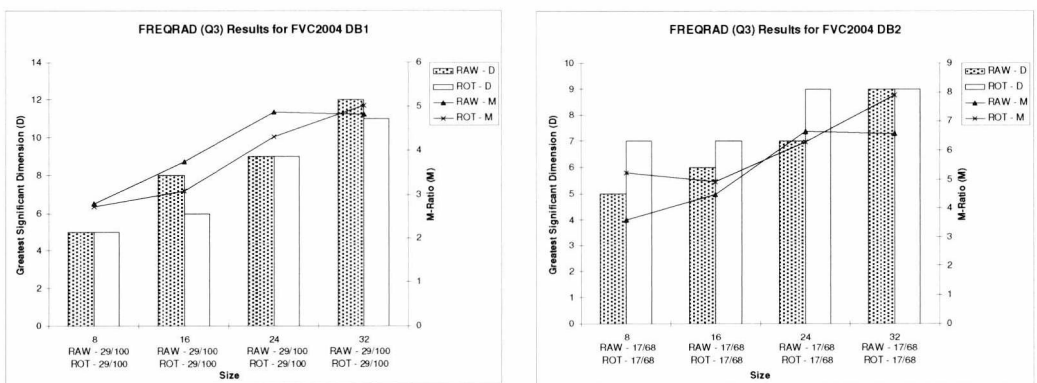


Figure 6-10: Q3 Results of FREQRAD (AUTO) for FVC2004 DB1 (l) and DB2 (r)

In addition to the experiments conducted, a further investigation into the influence of the number of sub-ROIs was executed. By varying the number of sub-ROIs, the length of the resultant feature

vector is altered. It is therefore obvious that as the number of sub-ROIs increase, the actual dimensionality of the feature vector and the effective dimensionality also increase. However, the tests previously carried out for both FREQGRID and FREQRAD are not entirely comparable because the feature vector lengths are significantly different. Therefore, experiments where the feature lengths are similar are executed. The results of which are presented in Figure 6-11, whereby the dimensional ratio (i.e. greatest significant dimension divided by the feature vector length) is depicted by the bars and M-Ratio by the line charts.

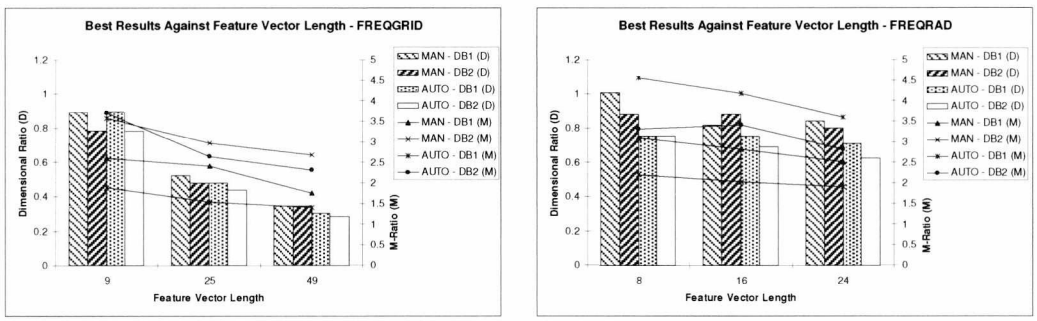


Figure 6-11: Influence of Feature Vector Length for FREQGRID (l) and FREQRAD (r)

Figure 6-11 depicts the change exhibited by the best M-Ratio and the best dimensional ratio as the feature vector length increases. The range of feature lengths was chosen because the lengths are similar for both FREQGRID and FREQRAD, from which a direct comparison could be made. Larger feature vectors were omitted on account that an estimation of frequency can only be obtained from neighbourhoods which possess a ridge and a valley (i.e. a full wavelength). Therefore, only experiments where the neighbourhood size is greater than or equal to 16 pixels were investigated. With the neighbourhood being quite large, a greater feature vector length requires more sub-ROIs, which inevitably covers a greater region of the image. With larger feature vectors, it was found that too few classes remained as many samples were rejected from the screening algorithm. It was decided that only experiments that retained over 90% of the classes were to be investigated to ensure the best representation of the feature based on the available samples.

From Figure 6-11, it is clear that both the dimensional ratio and the M-Ratio are greater for FREQRAD than that of FREQGRID, even when the feature lengths are similar. Furthermore, it can be seen that both statistics deteriorate as the feature length increases, although FREQGRID reduces more rapidly than FREQRAD. Therefore, it can be deduced from this experiment that the distribution of FREQRAD, due to the implementation of the RAD ROI scheme, is somewhat superior, in regard to class separation, than that of FREQGRID. Furthermore, a greater proportion of the dimensions are significant, which is beneficial for increasing key length.

6.2.3. Summary

In this section, frequency based descriptors of fingerprint images were investigated. The experiments conducted investigated the effects of varying many parameters and the analysis of the resultant statistical tests. It was determined that the descriptor performed best when the RAD ROI scheme was implemented and that the image was rotated to a base orientation prior to feature extraction. The greatest significant dimensions for both ROI schemes examined were of the same order but, due to the fact that the dimensionality of the FREQRAD descriptor is significantly less, the ratio between the actual dimensionality of the feature vector and its effective dimensionality is larger. For the best-performing FREQRAD feature vector, the means of the sample classes provided significant separation within 20 dimensions, which is similar to that of the FREQGRID descriptor. However, the FREQRAD descriptor consists of only 24 dimensions, whereas the FREQGRID descriptor contains 49 dimensions, indicating that the additional dimensions the FREQGRID descriptor contains provides little, if any, benefits.

It was determined from these experiments that, even though the effective dimensionality is relatively large, the M-Ratio is very small, which indicates that the feature space is very-densely populated. This will provide many difficulties when attempting to include this feature into the key generation stage as many classes will overlap. The M-Ratio begins to improve sufficiently at the strictest severity, but this is because only a small subset of classes and samples remain. When the severity is at its maximum, for both databases, the number of users is sub-20, which is not sufficient for a practical system. The greater number of samples used the more realistic and representative these statistical tests become. Since the M-Ratio is small when more than 90 users populate the feature space, it may not be suitable for key generation. It was also ascertained that a larger feature vector can be extracted, when the RAD ROI scheme is implemented, as the majority of the dimensions are significant. However, due to the pitiful M-Ratio, this may not present any additional benefits at the key generation stage.

6.3. FingerCode

The Gabor filter is one of the most popular filters used in fingerprint processing because the kernel shape is well suited to the oriented ridge structure. The Gabor filter is a linear filter whose impulse response is defined as a harmonic function (sinusoidal carrier) multiplied by a Gaussian Function [24]. The result of which provides the kernel shape depicted in Figure 6-12.

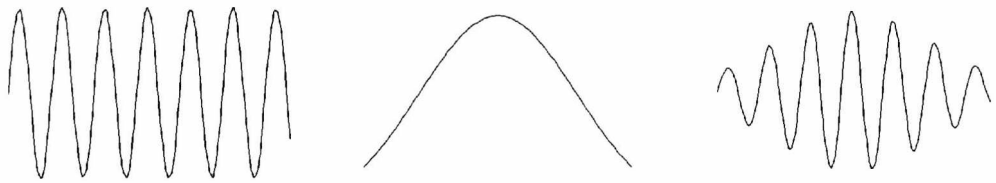


Figure 6-12: Gabor Waveform Composition. Sinusoidal Carrier (l), Gaussian Envelope (m) and Gabor Waveform (r)

It is evident from Figure 6-12 that by combining a Gaussian function with a sinusoidal carrier, the result produces a rippling effect whereby the ripples reduce at a rate dictated by that of the Gaussian function. It is clear, even from the shape of a 1D Gabor Kernel shown in Figure 6-12, that fingerprint ridges can be accentuated by assuring that the ripples of the Gabor kernel are spaced accordingly. However fingerprint ridges are oriented texture patterns in two dimensions. Therefore, the 2D Gabor filter kernel must be applied to a fingerprint. The Gabor filter function is presented in Equation 6-4.

$$G(x,y) = \exp\left(-\frac{1}{2}\left(\frac{x'^2}{\sigma_x^{\,2}} + \frac{\gamma^2 y'^2}{\sigma_y^{\,2}}\right)\right)\cos\left(2\pi\frac{x'}{\lambda}\right) \qquad \textbf{6-4}$$

$$x' = x\cos\theta + y\sin\theta$$

$$y' = -x\sin\theta + y\cos\theta$$

Where:

σ_x^2 and σ_y^2 are the variances of the Gaussian envelope in the x and y axes. These control the rate at which the ripples reduce.

G(x, y) is the Gabor kernel centered on pixel (x, y).

λ is the wavelength, which controls the spacing of the ripples.

γ is the spatial ratio, which controls the "*ellipticity*" of the function.

The filter function presented in Equation 6-4 requires multiple parameters. By tweaking these parameters it is possible to construct a filter kernel which is orthogonal to the orientation of the ridges, with the same inter-ripple distance as that of the inter-ridge distance. Some sample Gabor kernels are presented in Figure 6-13.

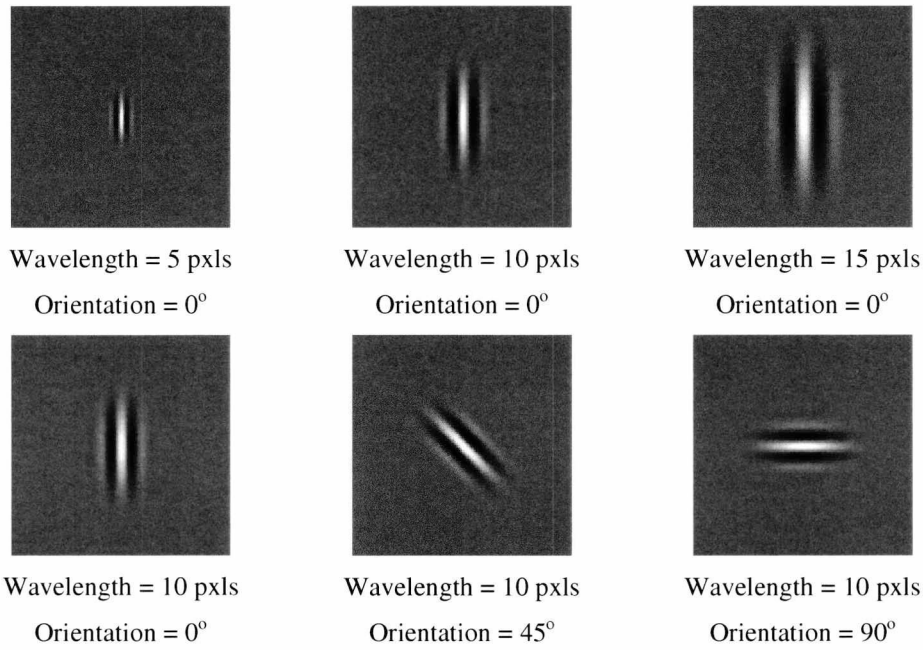


Figure 6-13: Example of Gabor Kernels at Varying Orientations and Wavelengths

When the kernel shape is similar to that of the orthogonal orientation and the frequency of the ridge structure, the distinctiveness between ridge and background is accentuated. Whereas dissimilar shapes will blur the ridges as the orientation will smooth ridges into each other. The Gabor filter can be applied either in the spatial domain, by convoluting the kernel with the image, or in the Fourier domain, by converting the image into the Fourier domain, multiply it by the kernel and performing an inverse Fourier transform to revert to the spatial domain.

6.3.1. Feature Extraction

Many enhancement methods employ 2D Gabor filters because they are very good at accentuating the ridges. However for identification, an appropriate measure needs to be identified. Variance-based measures are popular [24, 189] because the ridges that are in tune with the Gabor filters kernel (i.e. exhibiting similar orientation and frequency) will be accentuated. Therefore, the difference between background and foreground will be greater resulting in a larger variance. The FingerCode [24] is currently one of the most popular image-based features and this method implements an alternative variance calculation to the usual statistical variance. This method employs the average absolute deviation (AAD), which the authors claim to provide slightly superior results than that of the standard statistical variance. The primary steps for extracting the FingerCode are as follows.

- Locate reference point.
- Employ DISC ROI scheme to extract ROI. In [24], the DISC ROI was created using 5 *tracks*, with a *trackwidth* of 20 pixels (double the average inter-ridge width to encompass a ridge-valley pair), and 16 *geometric sectors*. This created 80 block-like sub-ROIs.

- Normalize ROI using the z-score normalization method outlined in Equation 6-5.

$$N_s(x,y) = \mu_0 + offset$$

$$offset = \begin{cases} \sqrt{\dfrac{\sigma_0^{\,2}(I_s(x,y)-\mu_s)^2}{\sigma_s^{\,2}}}, & if\ I_s(x,y) > \mu_s \\ -\sqrt{\dfrac{\sigma_0^{\,2}(I_s(x,y)-\mu_s)^2}{\sigma_s^{\,2}}}, & otherwise \end{cases} \qquad \textbf{6-5}$$

Where:

μ_0 and σ_0^2 are the expected mean and expected variance.

μ_s and σ_s^2 are the mean and variance of sector s.

$I_s(x, y)$ is the pixel intensity located at (x, y), which lies in sector s.

$N_s(x, y)$ is the normalized pixel intensity located at (x, y) which lies in sector s

- Filter the normalized ROI with a Gabor filter-bank containing 8 differently orientated filters, each with a radius of 16. The wavelength (10 pixels as this is approximately the average inter-ridge distance) and both the Gaussian envelope variances (empirically determined to be 4) were constant throughout. This creates 8 filtered ROIs.
- For each sector of each filtered ROI, calculate the AAD by using Equation 6-6.

$$AAD_{\theta,s} = \frac{1}{n_{\theta,s}} \sum_{x,y \in s} (F_\theta(x,y) - \mu_{\theta,s}), \quad for\ \ 0,...,s,...,N-1 \qquad \textbf{6-6}$$

Where:

$AAD_{\theta,s}$ is the average absolute deviation of sector s of Gabor filtered image θ.

$F_\theta(x, y)$ is the θ oriented Gabor filtered value at (x, y).

$\mu_{\theta,s}$ is the mean of the θ oriented Gabor filtered values within sector s.

$n_{\theta,s}$ is the number of values that lie in sector s of Gabor filtered image θ.

N is the total number of sectors. This is calculated by multiplying the number of concentric circles (tracks) by the number of sectors in a track.

- Collate all AADs to compile a feature vector containing 640 feature values.

6.3.2. Experimental Results

The FilterCode is often used as an alternative, or as a supporting, feature to minutiae. The results presented in [24] shows that desirable classification rates (for their target application) can be obtained by implementing a simple k-nearest neighbour (KNN) classifier, so it is therefore anticipated that this feature should demonstrate the desired characteristics. For the experiments carried out, the initial plan for testing was simply to extract the feature vector as outlined in the previous section from ROIs that have undergone varying stages of pre-processing. Additionally, an investigation was also to be executed into the influence of automatic estimation of the reference point's location and orientation. However, due to the size of the ROI (greater the area the greater the probability that the ROI will extend past the image boundaries or contain poor quality areas)

and the high dimensionality of the feature, not all tests were carried out as there were insufficient samples to carry out the MANOVA (Multivariate Analysis Of Variance) calculations. Figure 6-14 presents the results that were obtained when the standard FingerCode extraction algorithm was employed. Both the automatic (AUTO) and manual (MAN) estimation of the reference point location and orientation was investigated and are shown within Figure 6-14.

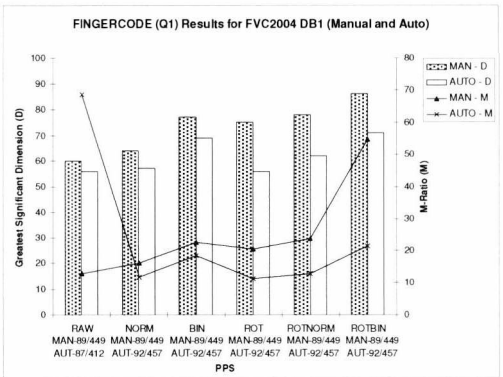


Figure 6-14: Results of the Standard FingerCode Feature Vector for FVC2004 DB1

Due to the factors previously mentioned, only FVC2004 DB1 with the lowest severity was able to produce results when the configuration outlined in [24] was implemented. However, based on this test it is apparent that the best results are obtained when the image is binarized and rotated (ROTBIN) prior to feature extraction. It is important to note that, due to the variety of pre-processing stages undergone is also under investigation, unless denoted by NORM, none of these tests underwent ROI normalization as outlined in the published paper. From Figure 6-14, it is clear that, for both rotated and non-rotated, binarization improves both the M-Ratio and effective dimensionality. However the effective dimensionality ranges from 45 to 70 dimensions. As the feature vector has 640 dimensions, this is a significant reduction. Taking this into consideration, and that there is insufficient samples to perform additional tests with this feature vector length, further experiments whereby the feature vector length is reduced were conducted.

The 640-dimension feature vector can be reduced by reducing either the number of tracks and/or the number of geometric sectors. However, by simply reducing the number of geometric sectors does not overcome the problems encountered when the ROI extends past the boundaries of the image or when poor quality is detected within the region. Therefore, in order to overcome the problem of the number of images rejected, the ROI size will also need to be reduced. By simply reducing the number of tracks automatically reduces the size of the ROI, but this is not the case when the number of geometric sectors is reduced. In order to attain this, not only is the number of geometric sectors is reduced, the track-width is as well. So, for the following experiments, two series of tests, which use alternative approaches to reducing the ROI size and the feature vector length, are investigated.

The first series of experiments investigates the reduction in the number of tracks from 5 to 3, whilst maintaining the initial number of geometric sectors and track-width. Therefore, this test configuration is denoted by *3T20*, which means 3 tracks and 20-pixel track width. The second series of experiments maintain the number of tracks, although the track-width is reduced from 20 to 10 pixels (to reduce screening algorithm's reject rate) and the number of geometric sectors is reduced from 16 to 8 (to reduce the feature vector length). Hence, using the same naming convention as before, this test configuration is denoted by *5T10*, meaning 5 tracks and 10-pixel track-width. The results of these series of tests for a manually estimated reference point location and orientation (MAN) are presented in Figure 6-15 and Figure 6-16. In these figures, the number of classes and the number of samples, which were used in the calculations, are presented on the x-axis. The convention for this additional data is the series (i.e. *3T20*) and the number of classes / the number of samples.

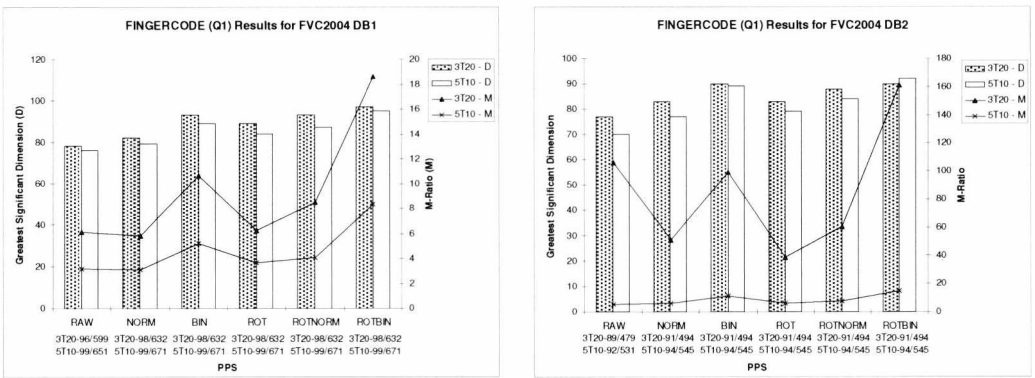


Figure 6-15: Q1 Results of FingerCode (MAN) for FVC2004 DB1 (l) and DB2 (r)

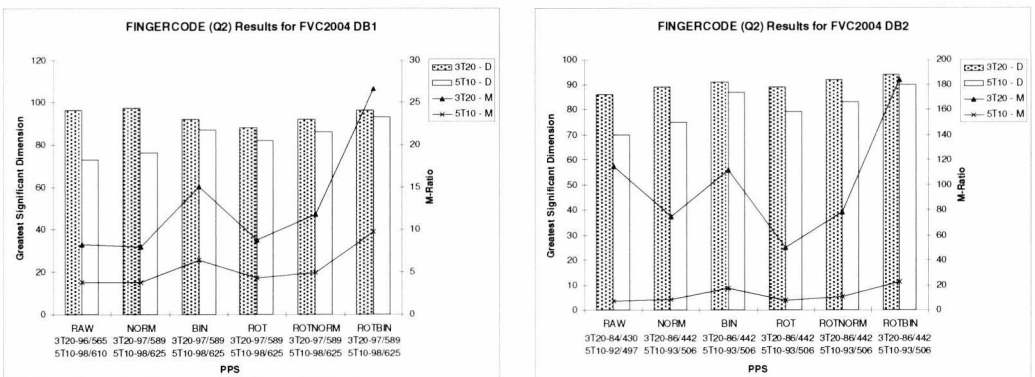


Figure 6-16: Q2 Results of FingerCode (MAN) for FVC2004 DB1 (l) and DB2 (r)

From the results presented, it can be seen that the effective dimensionality is influenced by which stages of pre-processing the ROI has undergone. It was expected that, by rotating the ROI prior to feature extraction, the effective dimensionality and the M-Ratio would be greater than that of any non-rotated ROI, even when normalized or enhanced. However this is not the case, whereby a non-rotated, enhanced ROI (BIN) results in better results than that of the rotated (ROT) and, in some cases, the rotated and normalized ROI (ROTNORM). It seems that, with reference to the effective

dimensionality, enhancing the ROI provides greater benefits than just rotating the ROI, but improvements are witnessed the rotated and non-rotated ROIs at the same pre-processing stage (i.e. NORM with ROTNORM, RAW with ROT and BIN with ROTBIN) are directly compared. Significant benefits can be achieved by simply enhancing the ROI, but to achieve optimum results, in both M-Ratio and effective dimensionality, rotating the enhanced ROI increases the effective dimensionality by 8 to 30% and results in a 2-3 fold increase in the M-Ratio.

The most significant characteristic of these results is the magnitude of the M-Ratio. The effective dimensionality is not as important as the reproducibility of the feature vector and the magnitude of the M-Ratios are far greater than witnessed previously. In fact, when compared with the actual dimensionality of the feature vectors, the ratio is very poor with less than 30% of the dimensions effective. However, the M-Ratios are impressive with the greatest M-Ratios for FVC2004 DB1 and DB2 being in excess of 25 and 180 respectively. This indicates that the distances between samples of the same class is significantly less than the distance between the means of different classes, which suggests the classes are well separated. Unfortunately, an explanation to why there is such a significant difference between the magnitudes exhibited by the two databases could not be established. The trends exhibited by the experiments, which implement automatic estimation (AUTO) are similar to that of the tests conducted with manually estimated reference point locations and orientations. The results of these experiments are presented in Figure 6-17 and Figure 6-18.

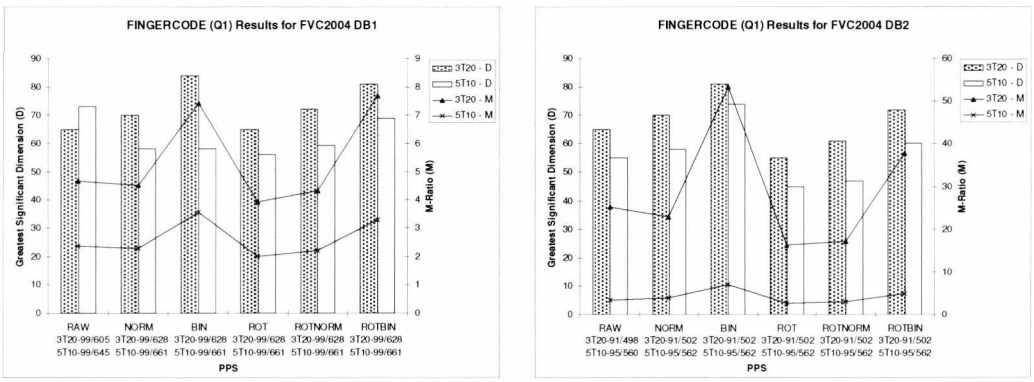


Figure 6-17: Q1 Results of FingerCode (AUTO) for FVC2004 DB1 (l) and DB2 (r)

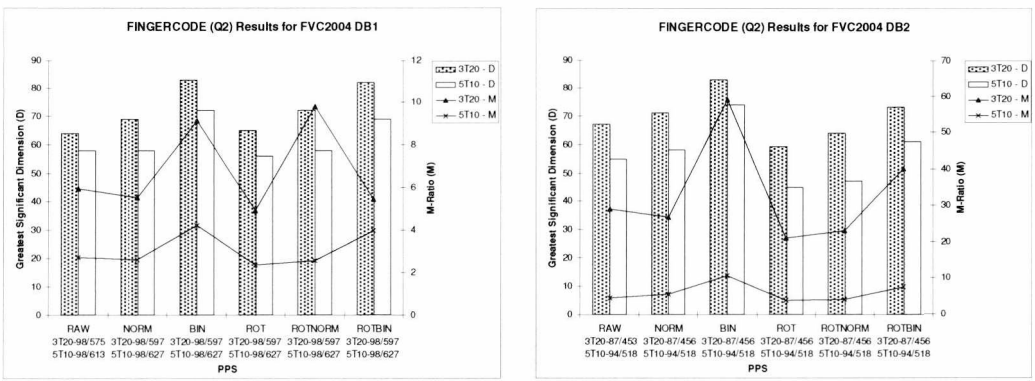


Figure 6-18: Q2 Results of FingerCode (AUTO) for FVC2004 DB1 (l) and DB2 (r)

It is clear from the results in Figure 6-17 and Figure 6-18 that in general the effective dimensionality and M-Ratio drops, although the experiments which implement a rotated ROI fair much worse. The obvious improvements obtained by ROI rotation cease and the non-rotated, enhanced ROI is the optimal solution when automatic estimation is employed. From all the experiments executed the *3T20* series of tests consistently outperform the *5T10* series. This suggests that a track-width of 10 pixels is not large enough to capture the ridge structure within the restrictive block-like sub-ROIs of the DISC ROI scheme, increasing the likelihood of overlapping within the dimension-space. Furthermore, it is interesting to note that the effective dimensionality remains similar and, in some cases, actually increases with the screening severity, which is a break from the indirect relationship exhibited by other feature vectors. This could be due to the fact that not many additional samples are rejected at the screening stage, but the samples retained are of better quality with a higher degree of separation.

6.3.3. Summary

The popular FingerCode fingerprint feature was investigated in this section. It was found that the extraction method outlined in [24] resulted in many prints being rejected due to the size of the ROI extending past the image boundaries and encroaching on regions of poor quality. Furthermore, due to the dimensionality of the feature vector, the results were incalculable for many experiments as the number of samples, which were accepted by the screening algorithm, was too few. Therefore, in order to evaluate the feature vector, a series of tests whereby the feature vector and ROI size was reduced.

These additional tests were complimented by the standard experiments investigating the influence of pre-processing the ROI prior to extraction. It was determined that the FingerCode obtained the best results when the ROI was rotated and enhanced in all experiments that incorporated the manual estimation of the reference point location and orientation. When automatic estimation was used, a non-rotated, enhanced ROI was deemed optimal. Furthermore, it was determined that simply reducing the number of tracks of the DISC ROI scheme was far superior than reducing the number of sectors and the track-width. The findings of these experiments suggest that the FingerCode provides high effective dimensionality, whilst maintaining a very high level of separation within the feature-space. These factors are highly desired and the FingerCode is a promising candidate feature for direct key generation.

6.4. Invariant Moments

Moments are descriptors that depict shapes with respect to the pixel intensities. There are some useful advantages to moments. They provide a compact descriptor, which is robust to noise as well

as variations in translation, scale and rotation. The most commonly used set of invariant moments are called Hu moments [190].

Although moments have been used for numerous image processing tasks for many years, moments seem underrepresented in the previous literature with regards to automated fingerprint systems. This may be due to the fact that moments may not be suited to fingerprint images or because only Hu moments offer the invariant characteristic which is demanded in automated pattern recognition system. However, Hu moments are not the only set of moments which offer robustness to all variations. Zernike moments are another set, which are insensitive to rotational, scale and translational variances. It is these two sets which will be investigated in order to ascertain their suitability in a template-free implementation.

6.4.1. Feature Extraction - Hu Invariant Moments

The only set of moments which have been implemented to date in a fingerprint system is the Hu moments [182]. The algorithm presented in [182] uses Hu's seven invariant moments to obtain a functional fingerprint verification system. For this system, the Hu moments are extracted from cells within a grid centred about the reference point (i.e. the GRID ROI Scheme – see Testing Framework, Chapter 3). However, from previous experiments, it has been determined that the GRID ROI scheme is not optimal, due to its inability to consistently extract from the same regions when the image is scaled, translated or rotated. Therefore, further ROI schemes will be investigated although, in general, the extraction is the same for all ROI schemes. Given a ROI, the Hu set of invariant moments are extracted from the region, which means that extraction is independent of the ROI scheme employed. The set of Hu moments are obtained from the central moments (see Equation 6-7) and then made scale and translation invariant by employing Equation 6-8. The Hu moments are then derived from these moments by using Equations 6-9 to 6-15.

$$\mu_{pq} = \sum_x \sum_y (x - \bar{x})^p (y - \bar{y})^q I(x, y)$$

$$\bar{x} = \frac{m_{10}}{m_{00}}, \quad \bar{y} = \frac{m_{01}}{m_{00}} \qquad \textbf{6-7}$$

$$m_{pq} = \sum_x \sum_y x^p y^q I(x, y)$$

Where:

$I(x, y)$ is the pixel intensity at (x, y), where x and y are coordinates which lie within the ROI.

μ_{pq} is the central moment of order $(p + q)$.

$$\eta_{pq} = \frac{\mu_{pq}}{\mu_{00}^{\left(1+\frac{p+q}{2}\right)}} \quad \textbf{6-8}$$

Where:

η_{pq} is the scale and translation invariant moment of order $(p + q)$.

μ_{pq} is the central moment of order $(p + q)$.

$$\phi_1 = \eta_{20} + \eta_{02} \quad \textbf{6-9}$$

$$\phi_2 = (\eta_{20} - \eta_{02})^2 + 4\eta_{11}^2 \quad \textbf{6-10}$$

$$\phi_3 = (\eta_{30} - 3\eta_{12})^2 + (3\eta_{21} + 3\eta_{03})^2 \quad \textbf{6-11}$$

$$\phi_4 = (\eta_{30} + \eta_{12})^2 + (\eta_{21} + \eta_{03})^2 \quad \textbf{6-12}$$

$$\begin{aligned}\phi_5 = {} & (\eta_{30} - 3\eta_{12})(\eta_{30} + \eta_{12})\left[(\eta_{30} + \eta_{12})^2 - 3(\eta_{21} + \eta_{03})^2\right] \\ & + (3\eta_{21} - \eta_{03})(\eta_{03} + \eta_{21})\left[3(\eta_{12} + \eta_{30})^2 - (\eta_{21} + \eta_{03})^2\right]\end{aligned} \quad \textbf{6-13}$$

$$\begin{aligned}\phi_6 = {} & (\eta_{20} - \eta_{02})(\eta_{30} + \eta_{12})\left[(\eta_{30} + \eta_{12})^2 - (\eta_{21} + \eta_{03})^2\right] \\ & + 4\eta_{11}(\eta_{30} + \eta_{12})(\eta_{21} + \eta_{03})\end{aligned} \quad \textbf{6-14}$$

$$\begin{aligned}\phi_7 = {} & 3(\eta_{21} + \eta_{03})(\eta_{30} + \eta_{12})\left[(\eta_{30} + \eta_{12})^2 - 3(\eta_{21} + \eta_{03})^2\right] \\ & + (3\eta_{12} - \eta_{30})(\eta_{21} + \eta_{03})\left[3(\eta_{30} + \eta_{12})^2 - (\eta_{21} + \eta_{03})^2\right]\end{aligned} \quad \textbf{6-15}$$

Where:

η_{pq} is the scale and translation invariant moment of order $(p + q)$.

ϕ_i is the i^{th} Hu moment.

6.4.2. Feature Extraction – Zernike Invariant Moments

An alternative set of invariant moments are the Zernike moments, which give an orthogonal set of rotation-invariant moments [191]. Unlike Hu's invariant moments, no evidence of the implementation of Zernike moments for fingerprint feature extraction has been discovered. This is surprising as Zernike moments are generally regarded as being superior to Hu moments with regards to rotational invariance. Therefore, the investigation into Zernike moments as a candidate feature for a template-free fingerprint system is entirely novel.

Zernike moments are constructed using a Zernike polynomial, instead of a monomial, which is the basis for Hu moments. The polynomials are based on the polar coordinate system in order to assure rotational-invariance. This means the Zernike polynomial will need to be mapped to its Cartesian equivalent. This bounds the polynomial in the Cartesian system to a unit circle, which means that Zernike moments can only be obtained from a circular ROI in the image. Other than the prerequisite that the ROI must be circular, the feature extraction is independent of the ROI scheme

and, therefore, given any circular ROI, any number of Zernike moments can be obtained. Although any number of Zernike moments can be calculated, it can be a time-consuming process for very high ordered moments due to the repeated calculation of the binomial coefficients and the subsequent factorial calculations.

Zernike moments, similarly to Hu moments, are derived from the centralised moments, which are then made scale and translation invariant [192]. However, as the ROI must be a unit circle, the coordinates have to be scaled accordingly so that $x^2 + y^2 \leq 1$. Once the coordinates are scaled, Zernike moments are computed from the centralized moments (see Equation 6-8) and the Zernike polynomials coefficients (see Equation 6-17) by implementing Equation 6-16.

$$Z_{pq} = \frac{p+1}{\pi} \sum_{\substack{k=q \\ p-|k| \in evem\ AND\ >=0}}^{p} \sum_{l=0}^{t} \sum_{m=0}^{q} (-j)^m \binom{t}{l} \binom{q}{m} B_{p,q,k} \eta_{(k-2l-q+m),q+2l-m} \qquad \textbf{6-16}$$

$$t = \frac{k-q}{2}, \quad \binom{u}{v} = \frac{u!}{v!(u-v)!}$$

$$B_{p,q,k} = \frac{(-1)^{\frac{p-k}{2}} \left(\frac{p+k}{2}\right)!}{\left(\frac{p-k}{2}\right)! \left(\frac{k+q}{2}\right)! \left(\frac{k-q}{2}\right)!} \qquad \textbf{6-17}$$

Where:

Z_{pq} is the Zernike moment with a polynomial order of p and repetition q.

$B_{p,q,k}$ is the k^{th} Zernike polynomial coefficient.

$\eta_{p,q}$ is the centralised moment, which is translation and scale invariant.

6.4.3. Experimental Results

Moments are often employed in image processing solutions. However, the shapes created by the ridge structure of a fingerprint is complex and it is envisaged that, on the local scale (i.e. sub-ROI), there will be considerable overlapping because, on that scale, ridges often look very similar. Unless the sub-ROI contains insipient ridges or minutiae, it is anticipated that regions which possess no such anomalies will produce similar values. Furthermore, the edges of ridges are not that well defined and are highly variable between successive acquisitions of the same fingerprint. Therefore, it is anticipated that moments, on the local scale, will not possess the desired properties, although the moments extracted on the global scale maybe more suitable. Under investigation in this section are two different sets of invariant moments. For both these sets a series of experiments were conducted, which investigated the influence of the ROI scheme, the variety of stages of pre-processing that the ROI can be subjected to and the estimation of the reference points location and

orientation. The reference point orientation is only employed to aid consistent sub-ROI placement as the moments extracted from the sub-ROI are rotationally invariant.

The ROI schemes investigated included the GBLC, GRIDC and RAD ROI schemes. The existing Hu moments based descriptor employed the GRIDSQ ROI scheme. However, as Zernike moments are extracted from a circular ROI, only circular ROIs can be used for this set of moments. In order to draw direct comparisons, the Hu moments will also be extracted from circular ROIs. The GRIDC ROI scheme was implemented so direct comparisons can be made with the pre-existing Hu moment descriptor. In order to cast fair judgment, the dimensionality of the feature vectors obtained from both sets of moments should be equal. As this set contains 7 moments, only 7 Zernike moments should be used, although it is possible to extract numerous Zernike moments, so which 7 should be used? A preliminary test was carried out to find whether the 7 greatest Zernike moments was consistent across multiple samples. It seemed that all moments extracted varied at similar rates, so it was decided that the first 7 moments was used in subsequent investigations. The first series of experiments conducted assessed the performance of the GRIDC ROI scheme. The grid comprised 6 rows and 6 columns and each cell had a diameter of 16 pixels (as per [193]). The grid was positioned using both automatic and manual estimation, which resulted in a 252-dimensional feature vector. Figure 6-19 and Figure 6-20 present the results of the set of Hu moments, whereas Figure 6-21 and Figure 6-22 depict the results of the set of Zernike moments.

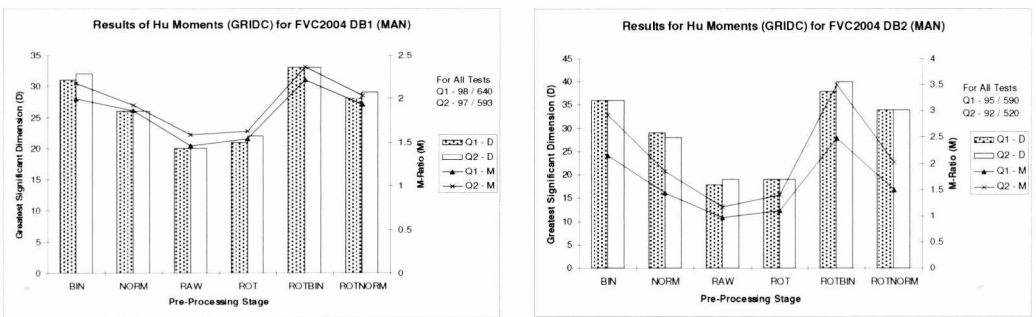


Figure 6-19: GRIDC Results of Hu Moments (MAN) for FVC2004 DB1 (l) and DB2 (r)

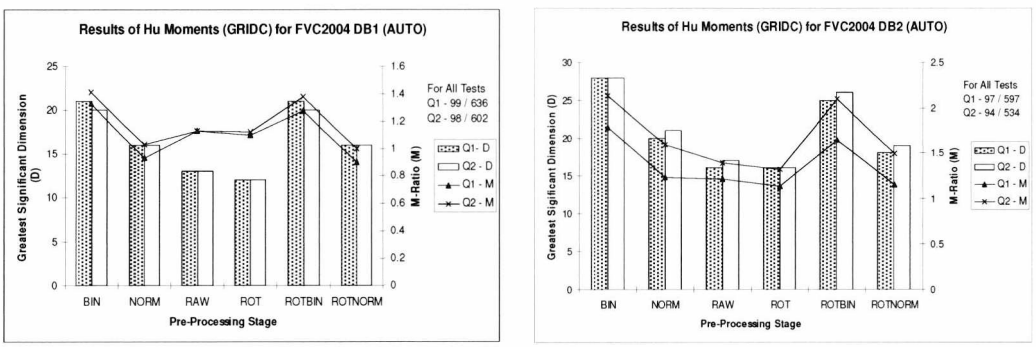
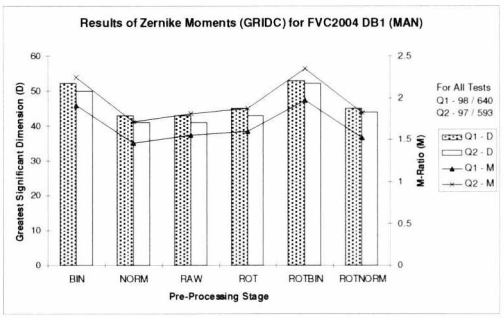


Figure 6-20: GRIDC Results of Hu Moments (AUTO) for FVC2004 DB1 (l) and DB2 (r)



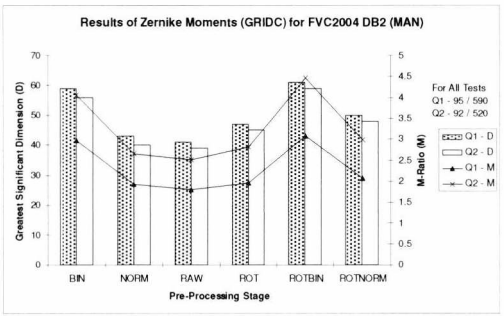
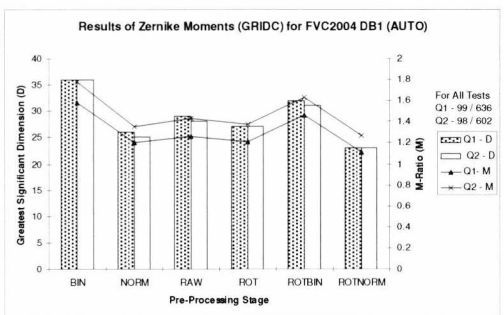


Figure 6-21: GRIDC Results of Zernike Moments (MAN) for FVC2004 DB1 (l) and DB2 (r)



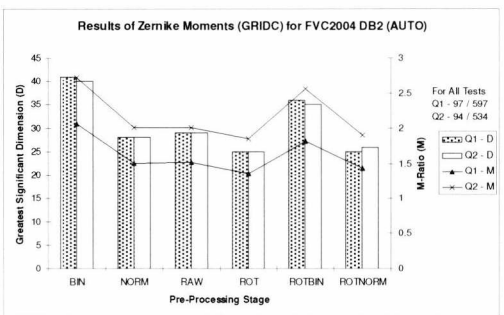


Figure 6-22: GRIDC Results of Zernike Moments (AUTO) for FVC2004 DB1 (l) and DB2 (r)

The results presented shows similar trends between both sets of moments. Both the M-Ratio (M) and the number of effective dimensions (D) both peak when the ROI has been enhanced. However, the results of the statistical tests for the set of Zernike moments show a significant increase. A >60% increase in the number of effective dimensions was exhibited, whereas the M-Ratio resulted in minimal, if any, improvement. Interestingly, the set of Zernike moments performed much better than the set of Hu moments on ROIs, which did not undergo any pre-processing, save of rotation. This seems to suggest that the Zernike moments are more robust to the variability in pixel intensity. Therefore the Zernike moments feature vector is more representative of the underlying shape within the ROI.

It is also clear from the charts presented that the series of tests, which assess the influence of automatic estimation of the reference point location and fingerprint's orientation, appear to mimic the trends of the manual estimation series of experiments. However, a noticeable deterioration of 30% was exhibited in both the number of effective dimensions and the M-Ratio.

Even though considerable improvement to the number of effective dimensions has been exhibited, the percentage of dimensions which are effective is low. The original dimensionality of the feature vector was 252, although the maximum number of effective dimensions obtained was only 62. Therefore the feature vector is considerably bloated. Due to this over-bloating, it was presumed that the same effective dimensionality could be ascertained from a feature vector of fewer dimensions by employing an alternative ROI scheme. The following series of experiments employed the RAD

ROI scheme, as the positioning of the sub-ROIs is more suited to rotation. Even when the image is rotated prior to feature extraction, a grid based ROI scheme, in theory, will not capture the same regions as consistently as the RAD ROI scheme. In this experiment a total of 24 sub-ROIs, spaced along 8 orientations with 3 radii. This produced a 168-dimensional feature vector. The results of the Hu moment series of experiments are presented in Figure 6-23 and Figure 6-24, and the results of the Zernike moments tests are presented in Figure 6-25 and Figure 6-26.

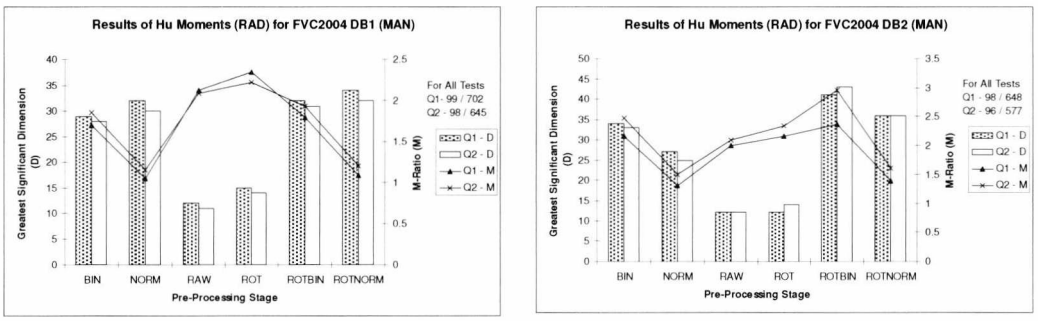


Figure 6-23: RAD Results of Hu Moments (MAN) for FVC2004 DB1 (l) and DB2 (r)

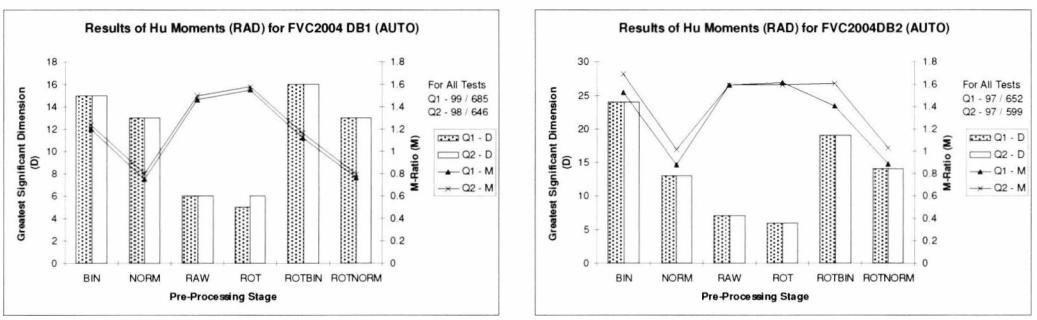


Figure 6-24: RAD Results of Hu Moments (AUTO) for FVC2004 DB1 (l) and DB2 (r)

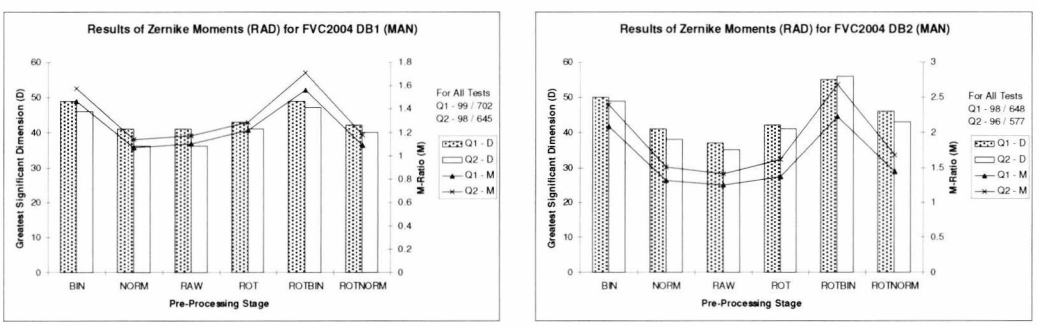


Figure 6-25: RAD Results of Zernike Moments (MAN) for FVC2004 DB1 (l) and DB2 (r)

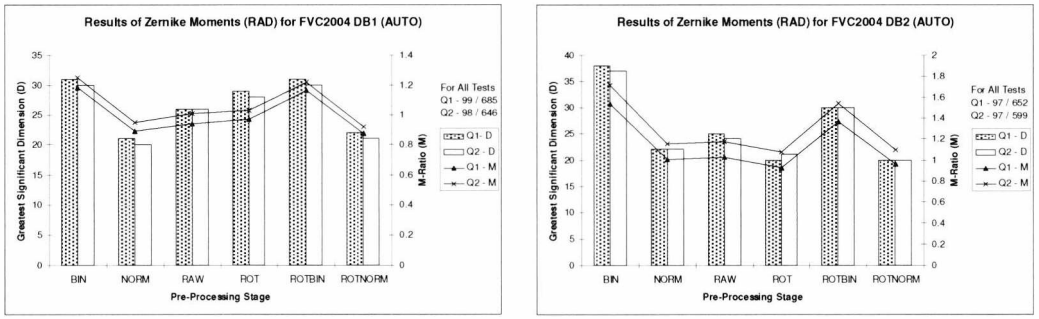


Figure 6-26: RAD Results of Zernike Moments (AUTO) for FVC2004 DB1 (l) and DB2 (r)

It is immediately apparent from the charts presented that the number of effective dimensions and the M-Ratio are of similar magnitudes to those presented for the GRIDC series of experiments. However, there is considerably less bloating in the feature vector as the means now lie within 33% of the total dimensionality, which is up from 23%. The most striking characteristic of these results is the unexpected improvement in the M-Ratio for the non-processed images (i.e. RAW and ROT) of the Hu moments series of experiments. This is contrary to the trend exhibited when the GRIDC ROI scheme was implemented. Although this unexpected improvement is negligible when taken into context, an acceptable explanation as to why this has occurred could not be established.

It was originally presumed that this phenomenon was due to the filtering out of descriptive information when effectively quantizing the pixel intensity by normalisation or enhancement. If more descriptive information remained in the non-processed ROIs, then this anomaly is conceivable. However, it is highly dubious because it was not exhibited in GRIDC series of experiments, which extracted ROIs of the same size. Another potential cause is due to the GRIDC ROI scheme extracting the region about the reference point. The first orbital path of the RAD ROI scheme is some distance away from the reference point, which was implemented because the region around the reference point is highly variable. However, this anomaly suggests that the region around the reference point may contain more descriptive information. If this was the case, why is this irregularity not reflected in the results of the Zernike moments? Further experiments, which increased the radius of the sub-ROI to 32 pixels was carried out. Partly to see whether this anomaly persisted and partly because it was purported that the probability of capturing a minutia with the sub-ROI would be increases.

Prior to experimentation, it was presumed that most sub-ROIs will result in similar descriptors because the region will normally be an oriented, sinusoidal pattern. However, orientation is no longer a factor as rotation invariant moments are employed, which reduces the descriptive information further. Therefore, the only sub-ROIs that will exhibit significant deviation from the norm are those which possess an artefact, such as a minutia. Unfortunately, negligible improvement was exhibited by the experiments and the peculiarity in the M-Ratio results of the Hu moments tests remained.

As moments are statistical features derived from pixel intensities, it was desired that moments will result in a feature vector that amalgamated the edge contours, curvature and ridge discontinuities (i.e. minutiae) of local regions into an elegant, compact feature vector. Unfortunately this has not been the case due to the considerable overlapping exhibited within the dimension-space, which may be, to some degree, attributed to the use of invariant moments. Perhaps, when employing an ROI scheme which comprises of sub-ROIs, standard centralised moments would perform better as the feature will become orientation dependent. The ROI scheme should ensure continual extraction of the same relative regions and, provided the image is rotated accurately beforehand, then the orientation, which can be obtained from the 2^{nd} order centralised moments, can also be extracted concurrently. However, the separation may improve when the invariant moments of a single, large region is investigated, as in the GBLC ROI scheme.

On a local scale, small deviation to the ridge edges can seriously affect the descriptor for that region. This can cause significant overlapping within the dimension-space. However, on a global scale, these deviations are negligible when taken into context. It is anticipated that the variance attributed to edge contours, and their subsequent influence on ridge widths, would be small compared to that credited to the frequency of minutiae, ridge direction, curvature and other artefacts such as scars. The actual pattern of the fingerprint can only truly be investigated when it is taken as a whole, in order to capture contextual information. Whereas, scrutinising local regions provide information about that region only. The only contextual element is the position of the sub-ROI in respect to the other ROIs. Figure 6-27 to Figure 6-28 presents the results obtained from Hu moments when a GBLC ROI scheme is employed and Figure 6-29 to Figure 6-30 depict those obtained from the Zernike moments test.

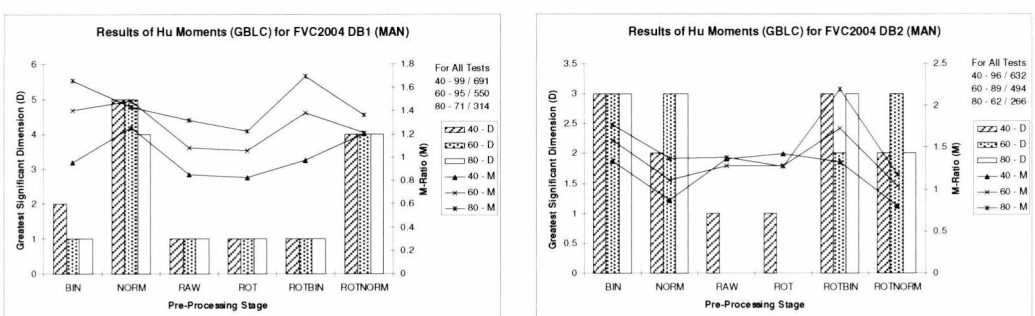
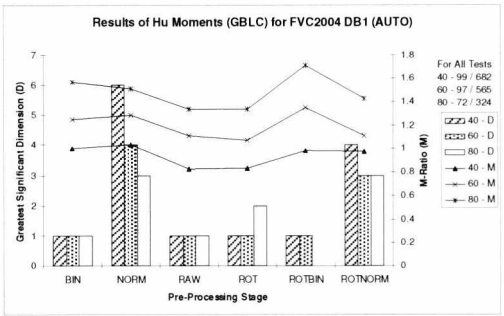


Figure 6-27: GBLC Results of Hu Moments (MAN) for FVC2004 DB1 (l) and DB2 (r)



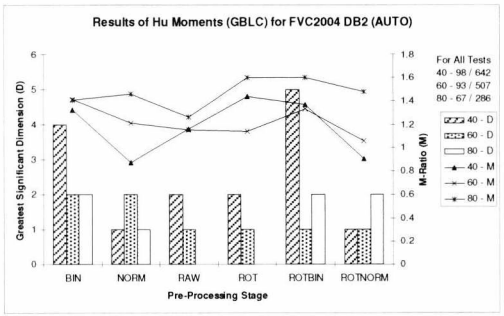
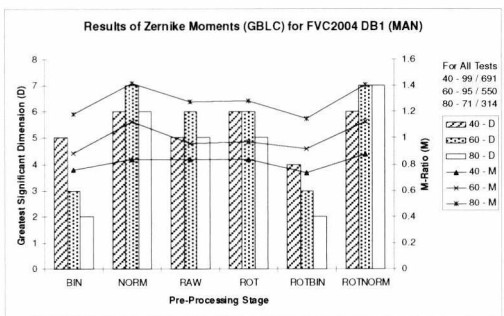


Figure 6-28: GBLC Results of Hu Moments (AUTO) for FVC2004 DB1 (l) and DB2 (r)



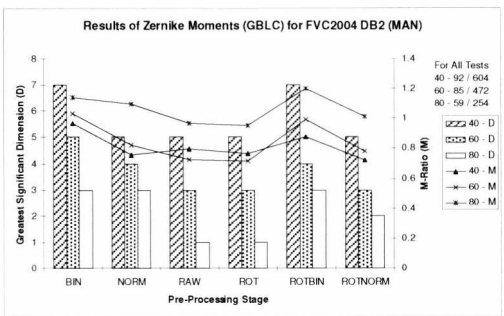
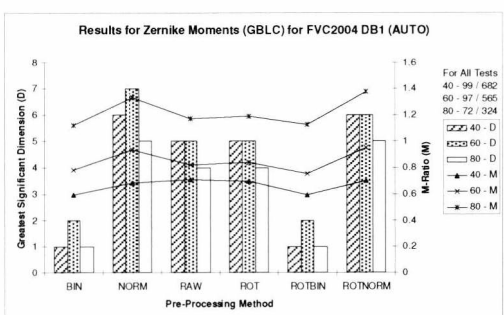


Figure 6-29: GBLC Results of Zernike Moments (MAN) for FVC2004 DB1 (l) and DB2 (r)



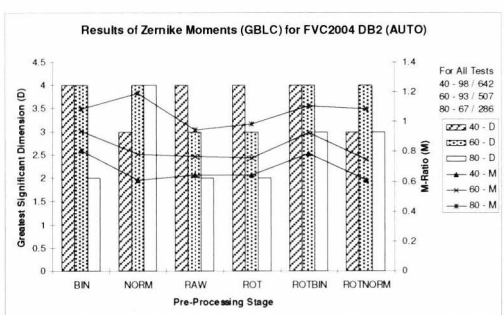


Figure 6-30: GBLC Results of Zernike Moments (AUTO) for FVC2004 DB1 (l) and DB2 (r)

For the GBLC series of experiments, the size of the ROI was gradually increased. However, only the M-Ratio seemed to be affected by this. The M-ratio increased with the size of the ROI, although this could be due to the reduced number of samples and groups within the feature space. As the size of the ROI increase, more samples are rejected as the ROI extends past the boundaries of the print. Even though the M-Ratio increases, it is still very small. Considering an M-Ratio of 1 indicates that the distance between group means is equal to the distance from a sample to its group mean, an M-Ratio of sub-2 means that substantial overlapping is present.

The most obvious property of the results, besides the poor M-Ratio, is that, for once, the databases disagree regarding which input configuration is best. For database 1, the best performance is achieved when the ROI is normalized, whereas database 2 maintains optimal performance when the ROI is enhanced. This was also demonstrated by the Zernike moments results, which initially

suggested robustness to pixel intensity changes. An acceptable explanation as to why this disagreement has occurred was not established. Regardless of this peculiarity, the percentage of dimensions which are effective is 100%, although the maximum possible was 7. It is obviously easier for a small dimensional feature to attain 100% effective dimensions, than a very large dimensional vector.

As highlighted, the M-Ratio for the GBLC series of tests is very poor. It may be better to use just a single moment as a feature of the entire ROI. Using the F-Test statistical test for each individual moment, it was determined that the 1st or the 4th Hu moments performed well for both databases with an F-Test value of between 3 and 4. However, for Zernike moments, the 2nd or 3rd moment were optimal with an F-Test value of 4. As with the M-Ratio, it is desired that these value be significantly greater than 1, so these results are not encouraging.

6.4.4. Summary

A number of experiments was conducted in order to evaluate the potential advantages of adapting a pre-existing grid-based, Hu moments descriptor. Although, this is a simple feature vector to extract, it was supposed that this may not result in optimal performance. A variety of adaptations was made to this feature vector, including the substitution of Hu for Zernike moments and adoption of different ROI schemes with varying degrees of pre-processing. Through thorough investigations, it was witnessed that the number of effective dimensions increased when Zernike moments were used, although the M-Ratio was slightly better when Hu moments were implemented. However, unlike Hu moments that are restricted to a maximum of 7, the number of Zernike moments can be increased. This allows additional features to be extracted for each region, which in theory provides greater the discriminatory information, and in turn increases the dimensionality of the feature vector.

The statistical analysis into the effects of increasing the number of Zernike moments was restricted due to the number of samples within the dataset. Although the number of moments was restricted, from the experiments conducted, it appeared that the M-Ratio would increase with the number of Zernike moments. However, with only a relatively small quantity of moments, it could not be determined whether this increase would be linear, or exponential, and continue indefinitely. As expected the number of effective dimensions would increase as well, although the percentage of the dimensions of the actual feature, which are effective, initially decayed rapidly but appeared to level out. The number of moments is limited only by the time and computational resources required to extract them. In these experiments, the number of moments was restricted due to the restriction imposed by the MANOVA calculation, but the feature vector could be much longer.

The experiments conducted demonstrate the effects of adopting alternative ROIs and the pre-processing of these regions. For the ROI schemes that are comprised of sub-ROIs, it was expected that the rotation of the image prior to feature extraction would be highly beneficial to facilitate consistent region extraction. However, no significant advantages were attained. Slight improvement was exhibited by the results of the rotated series of experiments for the GRIDC ROI scheme, but it has been suggested that this could be attributed to the inclusion of the region surrounding the reference point. If a region does not contain minutiae, singularity or high curvature (this is normally indicative of a singularity located within close proximity), then the region is essentially an oriented, periodic pattern.

As the moments investigated here are rotation invariant, the orientation is of little importance and it is expected that the moments obtained from these regions are very similar. However, regions about singularities generally possess high curvature and can possess minutiae. These two factors will produce a set of moments that should deviate significantly from the set obtained from a region possessing little of these characteristics. These regions are commonly located away from singularities and frequently occur when the RAD ROI is implemented. This is a possible justification as to why the RAD ROI scheme results in a reduced M-Ratio. It is envisaged that implementing a set of centralised moments may provide better separation (for ROI schemes, which employ sub-ROIs) as the moments will rely on orientation. Provided the image is rotated to a base orientation prior to feature extraction to ensure consistent sub-ROI placement, then the performance should improve. This would also facilitate the extraction of additional moments as centralised moments are not restricted to 7 and are not computationally expensive.

Throughout the experiments, the majority of trials extract 7 moments in order to ensure consistency and facilitate direct comparisons between Zernike and Hu moments. However, when different ROIs are employed the dimensionality of the feature vector changes and it is not possible to compare these based on effective dimensionality. The percentage of effective dimensions is not an ideal statistic for comparison purpose. This is because it is easier to obtain 100% effective dimensions when the feature vector has only 7 dimensions to begin with. The likelihood of a 252 dimension feature vector of obtaining 100% effective dimensions is very small. Therefore the M-Ratio was used for this occasion. It was found that, when only 7 moments was used, all M-Ratios obtained are sub 4.5, with the vast majority below 2. The optimum M-Ratios were actually obtained from the GRIDC ROI scheme, which obviously provides the greatest number of effective dimensions but the lowest percentage.

It was also determined empirically that by rotating the image and enhancing the sub-ROIs, generally produced the optimum results, although normalising sometimes was better. For both sets of invariant moments, this configuration (GRIDC + ROTBIN) was optimal for both effective dimensions and M-Ratio, although Zernike moments resulted in a significant increase in the

number of effective dimensions. Zernike moments consistently outperformed Hu moments with respect to the percentage of effective dimensions, which suggests that Zernike moments may be more useful for direct key generation as the difference in M-Ratios was negligible.

6.5. Wavelets

The 2D Discrete Wavelet Transform (DWT) is a solution to the shortcomings of the Fourier transform and other domains. The Fourier transform decomposes an image into its component frequencies, essentially transforming the image from the spatial domain to the frequency domain. However, the Fourier transform does not give any indication where these frequencies occur. The DWT attempts to solve this problem by analyzing an image at different scales.

6.5.1. Feature Extraction

In signal processing, the DWT is used for time-frequency analysis, whereby frequencies can be analyzed with respect to the time which they occur. This is similar to an image, although the DWT is used for spatial-frequency analysis. This is achieved by decomposing the image into varying degrees of resolution. The decomposition is accomplished by applying a filter-bank, whereby the input is subjected to low and band pass filtering and down-sampling. These processes are outlined in Figure 6-31 which was presented in [194].

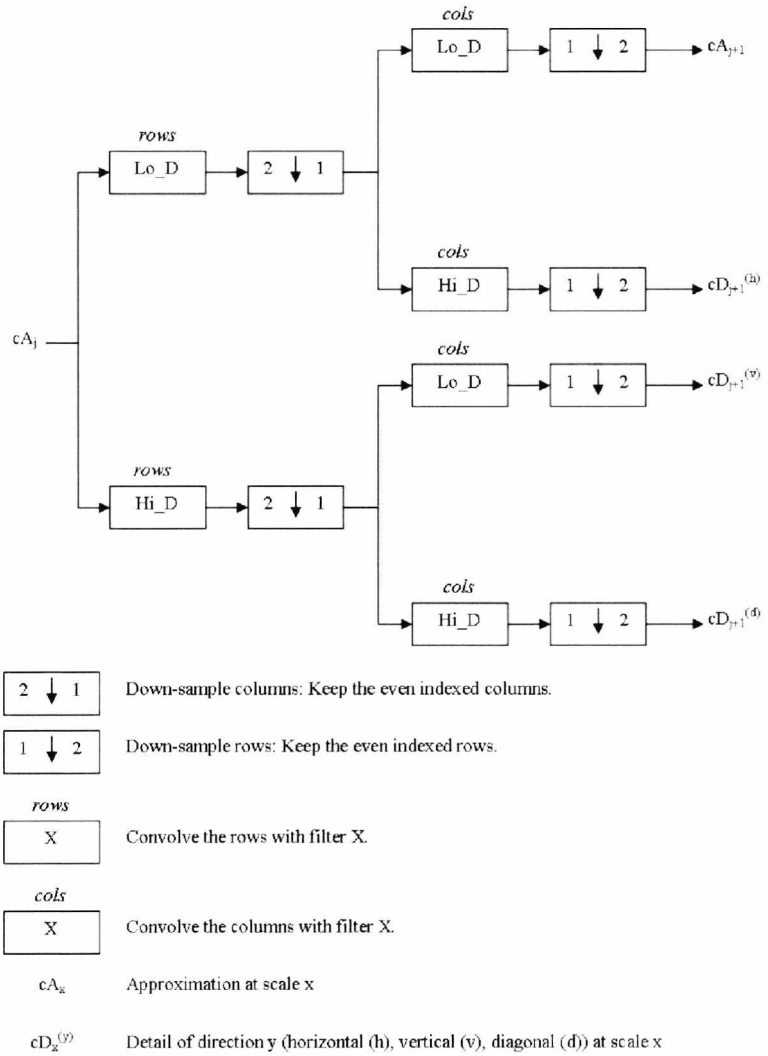


Figure 6-31: Wavelet Decomposition

The filter-bank uses a pair of filters. The Lo_D filter outlined in Figure 6-31 is determined by the coefficients of the Scaling function and the Hi_D filter is determined by the coefficients of the wavelet function. These coefficients are dependent on the family of wavelets used. However the filter-bank technique is a general method and is independent of the wavelet family being employed. The result of applying wavelet decomposition once is the production of four sub-images, which are the approximation (cA_{j+1}) and three directional details ($cD_{j+1}^{(v)}$) of the input at the next scale level. In order to further decrease the scale, the resultant approximation is used as the input to the next filter-bank. The result of wavelet decomposition over 3 scales is shown in Figure 6-32.

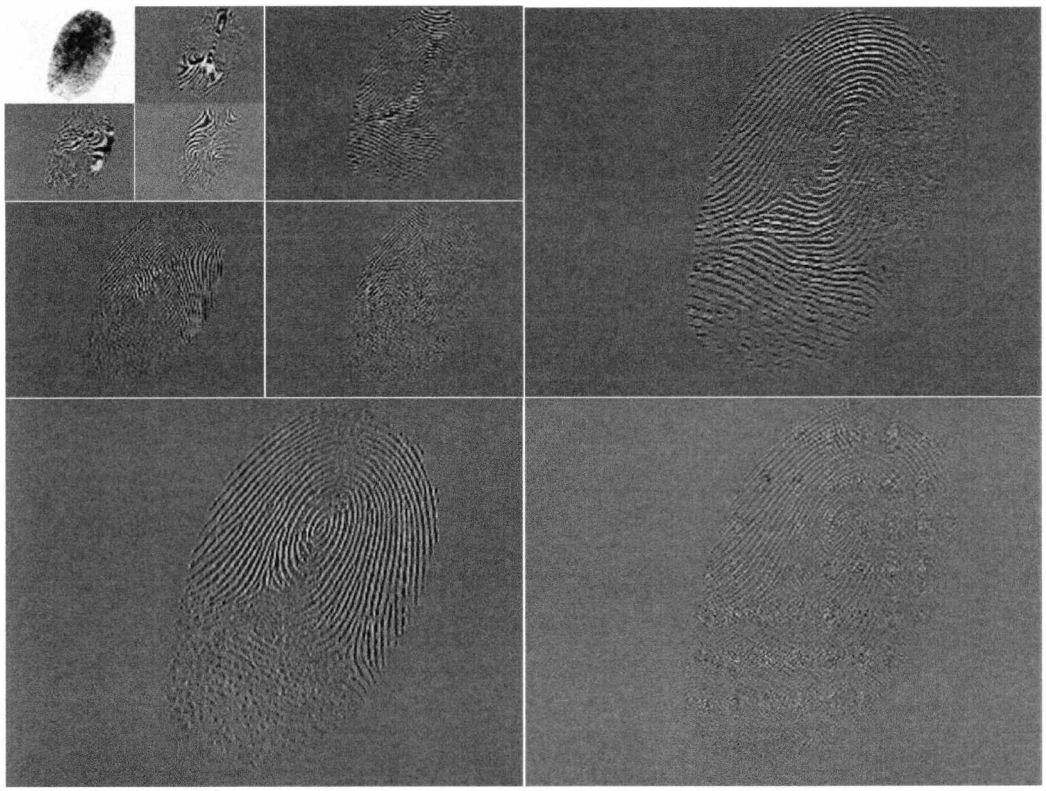


Figure 6-32: Example of 3-Scale Daubechies-4 Wavelet Decomposition

The wavelet decomposition illustrated in Figure 6-32 is presented in a generally accepted layout. The top-right, bottom-left and bottom-right images are the horizontal detail, vertical detail and diagonal detail respectively. The images in the top-left quadrant are the subsequent details of lower scales arranged in the same way. The final top-left image is the final approximation at the lowest scale. This example shows the result of the entire fingerprint decomposed using a Daubechies-4 wavelet with 3 scales. As mentioned previously, fingerprints are susceptible to partial overlap. Therefore, instead of the entire image being decomposed, a GBLC ROI scheme was implemented to ensure the same relative region is captured. This was obtained by simply cropping the image and setting the pixels that lay further than a predefined distance away from the reference point to 0.

It is clear from Figure 6-31 that multiple representations of the fingerprint are produced through cascaded decomposition over numerous scales and 3 directions. The result of single scale decomposition is 3 details, which represent direction (horizontal, vertical and diagonal) and an approximation that is to be used for further decomposition. From these details, descriptors can be extracted which represent the input image. The features extracted from the details are often statistical measures, which include the energy (L_2norm), mean and standard deviation. These features can be calculated using Equations 6-19 to 6-21.

$$\mu = \frac{1}{N} \sum_{x,y \in ROI} c(x,y) \qquad \textbf{6-19}$$

$$\sigma = \sqrt{\frac{\sum_{x,y \in ROI} (c(x,y) - \mu)^2}{N}} \qquad \textbf{6-20}$$

$$e_j^k = \frac{\left\| d_j^k \right\|_2}{\sum_{i=1}^{J} \sum_{l=1}^{3} \left\| d_i^l \right\|_2} \qquad \textbf{6-21}$$

$$d_j^k = \frac{\sum_{x,y \in ROI} \left| c_{x,y} \right|^2}{N}$$

Where:

$C_{x,y}$ is the coefficients located at *(x, y)* which lie within the ROI.

N is the total number of coefficients in the ROI.

e_j^k is the energy ratio of j^{th} scale and k^{th} directional detail.

6.5.2. Experimental Results

For the following experiments, many parameters were investigated in order to ascertain the optimal configuration. Similar to the experiments conducted for other features, the ROI scheme, the reference point estimation method and the pre-processed input image can be altered. However, for these experiments, different wavelets from a variety of families can also be implemented by simply substituting one for the other. This permits a multitude of test configurations to be created. Unfortunately, it would not be feasible to conduct every permutation so certain restrictions were made. For instance, only the GBLC ROI scheme was implemented as it was decided that wavelet features on the local scale would not be suitable as the increased smoothing will further suppress features. Furthermore, only the 3 most common wavelet families were implemented, in order to reduce the number of experiments. A list of the wavelets implemented, and their acronyms, is provided in Table 6-1.

Table 6-1: List of Wavelets

Haar (H)
Daubechies-4 (D4)
Daubechies-6 (D6)
Daubechies-8 (D8)
Daubechies-10 (D10)
Daubechies-12 (D12)
Daubechies-14 (D14)
Daubechies-16 (D16)
Daubechies-18 (D18)
Daubechies-20 (D20)
Coiflet-6 (C6)
Coiflet-12 (C12)
Coiflet-18 (C18)
Coiflet-24 (C24)
Coiflet-30 (C30)

As aforementioned, the input image can be altered due to what pre-processing methods have been employed. This is compounded further by the size of the GBLC ROI scheme used. As the ROI scheme radii are predefined (i.e. 40, 60 and 80 pixels) and there are 6 different methods of pre-processing stages under investigation, this means that there are 24 possible inputs to the feature extraction stage. Coupled with the 15 different wavelets investigated, produces 360 permutations. This is even before the number of scales is taken into account. In total, the first 5 scales were investigated, which substantially increases the number of test permutations. Furthermore, as the testing is conducted over two databases and two methods of reference point estimation, the final number of test permutations exceeds 5000. Therefore, only the input configuration (i.e. the reference estimation method, the ROI scheme and the stages of pre-processing the image has undergone) that was manually determined to perform the best for each feature will be presented.

The optimum results were determined by finding which tests resulted in the greatest number of effective dimensions. If the greatest number of effective dimensions has been determined by a multiple of tests, the decision as to which test is optimum is then based on the M-Ratio. Furthermore, it is important to ensure that the feature space is well populated as this provides a more accurate representation of the feature space. Therefore, it was decided that at least 80% of the classes must be represented in the feature space to be included in the analysis. This essentially omits the tests with the GBLC ROI radius of 80 pixels but, in the majority of cases, these tests would result in fewer effective dimensions than the other tests as a result of an under-represented feature space.

In this section, 3 feature vectors are investigated. The first descriptor is composed of the energy ratio for each detail, the second comprises the mean and standard deviation for each detail and the final descriptor incorporates the energy ratio, the mean and the standard deviation. The first series of experiments investigated the energy ratio descriptor [176] and the results of which are presented

in Figure 6-33 to Figure 6-36 for all the wavelet families for FVC2004 DB1 and DB2. In these tests, the dimensionality and, by extension, the scale is examined. As there are 3 details for every scale and only one feature (energy ratio) extracted per detail, the number of scales examined can be obtained from the charts by dividing the dimensionality by 3.

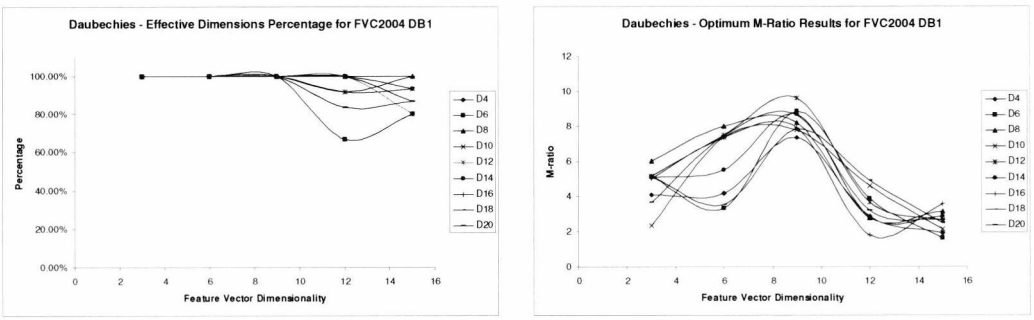


Figure 6-33: Optimal Energy Ratio Descriptor (Daubechies Wavelets) Results for FVC2004 DB1. Percentage of Effective Dimensions (l) and M-Ratio (r)

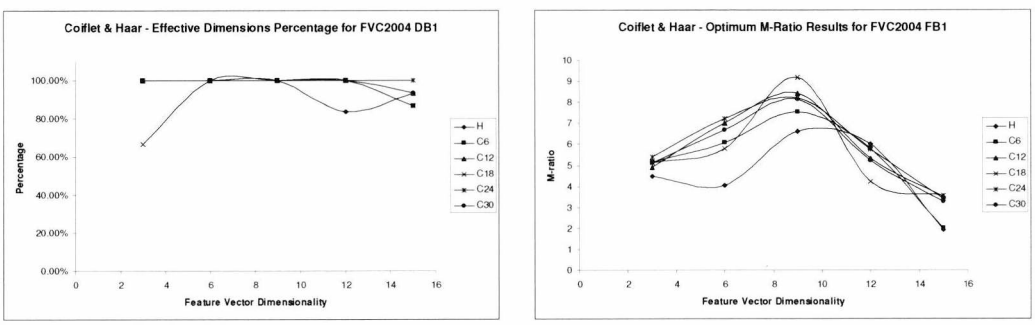


Figure 6-34: Optimal Energy Ratio Descriptor (Coiflet and Haar) Results for FVC2004 DB1. Percentage of Effective Dimensions (l) and M-Ratio (r)

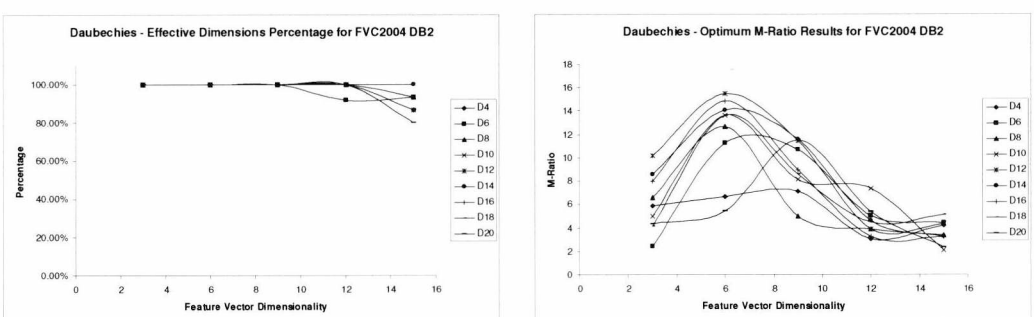


Figure 6-35: Optimal Energy Ratio Descriptor (Daubechies Wavelets) Results for FVC2004 DB2. Percentage of Effective Dimensions (l) and M-Ratio (r)

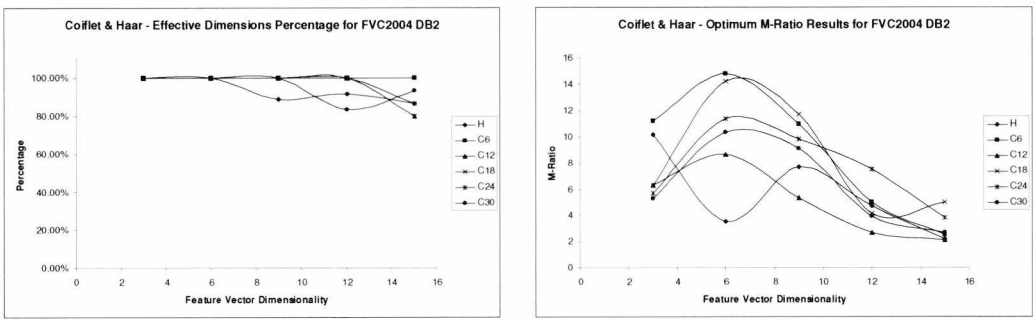


Figure 6-36: Optimal Energy Ratio Descriptor (Coiflet and Haar) Results for FVC2004 DB2. Percentage of Effective Dimensions (l) and M-Ratio (r)

From the left-hand charts of Figure 6-33 to Figure 6-36, it is clear that, for the majority of tests, 100% of the dimensions are effective until the feature vector length exceeds 9 (i.e. more than 3 scales), after which the percentage of effective dimensions begins to drop. As many of the tests performed result in 100% effective dimensions, the M-Ratio can be used to determine the optimum test configuration. The right-hand charts of Figure 6-33 to Figure 6-36 depict the M-Ratios obtained from the same tests that produced the optimal effective dimensionality.

The most striking characteristic of the M-Ratio charts is that many of the trends seem to be almost Gaussian distributed. However, the means of these pseudo-Gaussians is not consistent throughout both databases. The most frequent optimal feature vector length for FVC2004 DB1 is 9 (i.e. 3 scales) whereas, for FVC2004 DB2, the optimal length is typically 6 (i.e. 2 scales). Although there is a disagreement between databases as to which vector length is optimal, it is evident from the FVC2004 DB2 results that the M-Ratios for features of length 9 are somewhat better then the optimal results for FVC2004 DB1. Therefore, a feature vector of length 9 is a compromise where the feature will perform well across both databases, whereas a length of 6 does not perform as well for FVC2004 DB1.

From the results obtained, the optimal wavelet feature has been determined as Daubechies-12 with 3 scales of decomposition, resulting in 100% effective dimensions and M-Ratios 9.62 and 11.48 for DB1 and DB2 respectively. However, as expected, automatic estimation was significantly inferior. For automatic estimation, the number of effective dimensions reduces earlier (when length exceeds 6 dimensions), and at a greater rate than that exhibited when manual estimation is used. Furthermore, the M-Ratio is, on average, half (but can be much less) of that exhibited when manual estimation is used. Although the results are not as good Daubechies-12 was consistently the best performing wavelet.

The second series of experiments investigated the statistical measurements of each detail. Using the mean and standard deviation, as outlined in [175], a feature vector of length scale x 6 is extracted. As with the energy ratio feature vector, experiments were conducted by systematically altering the

radius of the ROI, its placement, and the pre-processing of the input image. The optimum results of these tests are presented in Figure 6-37 to Figure 6-40.

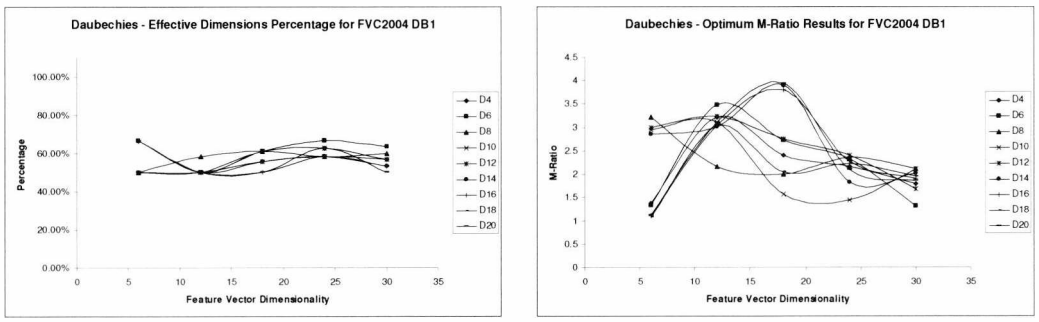


Figure 6-37: Optimal Mean/Stdev Descriptor (Debauchies Wavelets) Results for FVC2004 DB1. Percentage of Effective Dimensions (l) and M-Ratio (r)

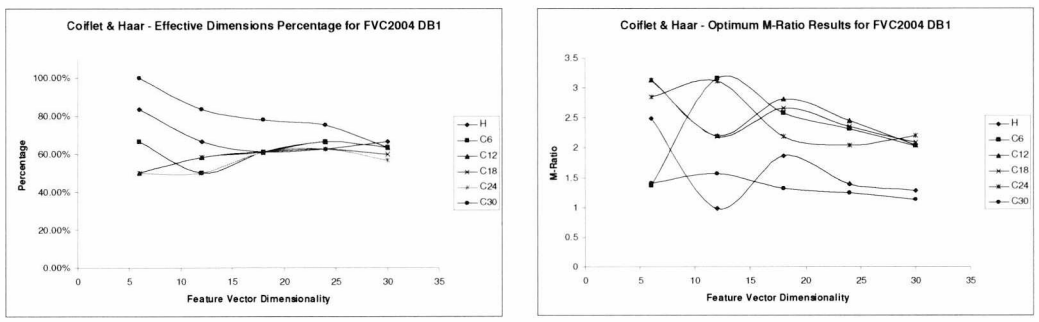


Figure 6-38: Optimal Mean/Stdev Descriptor (Coiflet and Haar) Results for FVC2004 DB1. Percentage of Effective Dimensions (l) and M-Ratio (r)

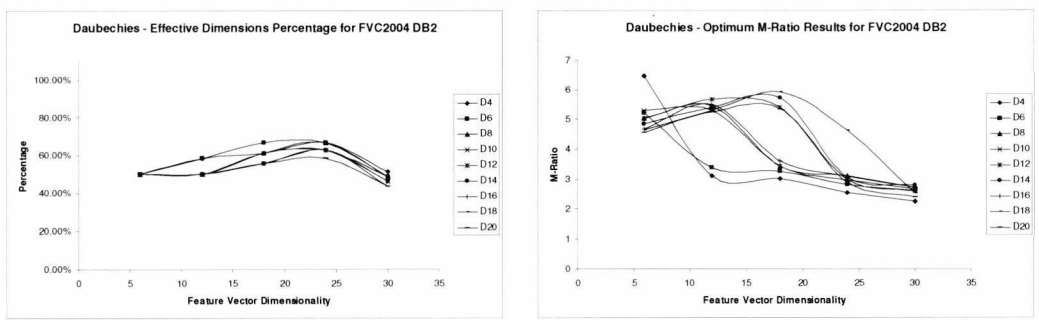


Figure 6-39: Optimal Mean/Stdev Descriptor (Daubechies Wavelets) Results for FVC2004 DB2. Percentage of Effective Dimensions (l) and M-Ratio (r)

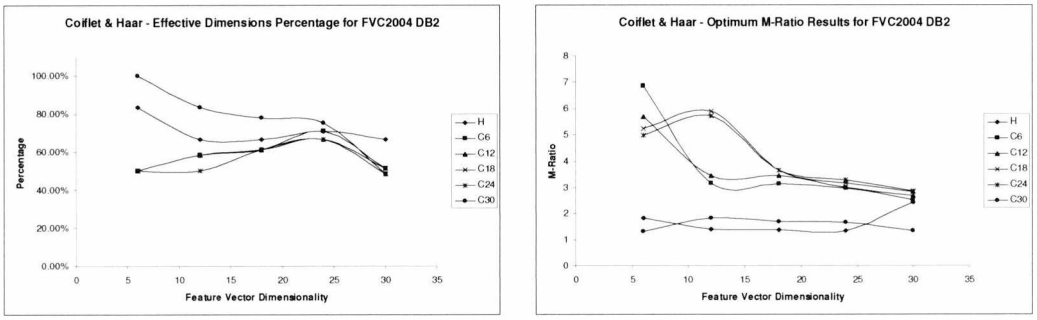


Figure 6-40: Optimal Mean/Stdev Descriptor (Coiflet and Haar) Results for FVC2004 DB2. Percentage of Effective Dimensions (l) and M-Ratio (r)

It is immediately evident from inspecting the charts in Figure 6-37 to Figure 6-40 that there is more bloating in the feature vector as, on average, 61% of the dimensions are effective. Therefore more dimensions are required in order to match the effective dimensionality of the energy ratio descriptor. It is also clear from the M-Ratio results, that the M-Ratios are significantly less than that of the energy ratio descriptor. In some cases, the M-Ratio deteriorates by a factor of 3. On average, the M-Ratio was approximately half of the M-Ratio of the energy ratio descriptor when the comparison was fair. The comparison is judged to be fair when the dimensionality, or the effective dimensionality, of the different descriptors is equal.

Although the mean/standard deviation descriptor produces inferior results, there are some similarities between the two feature vectors. Firstly, the input configuration of a 60 pixel radius GBLC ROI scheme, which was positioned using a manual estimated reference point, and rotated and enhanced prior to extraction was frequently the top performer. Secondly, in the majority of cases, the M-Ratio peaks when the number of scales implemented is either 2 or 3. However, this is not as well defined as the energy ratio descriptor. In fact it could be perceived as a momentary blip in an otherwise monotonically decreasing trend. This trend was well defined when an automatic estimation method was employed instead. Figure 6-41 presents the, almost linear, decreasing trend as a result of using an automatically estimated reference point location and orientation for both databases.

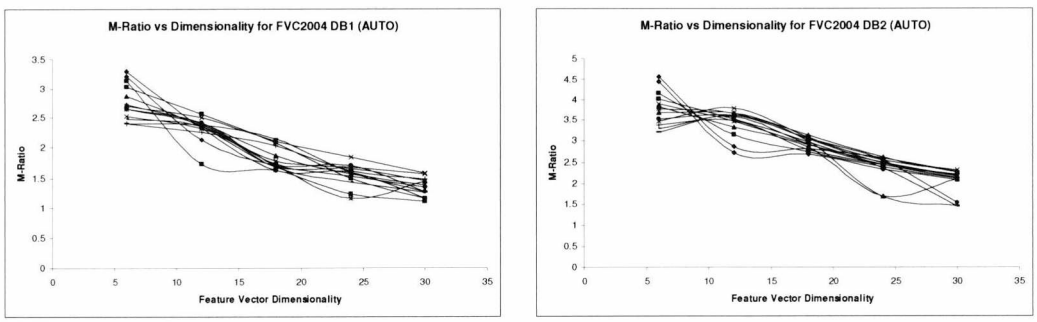


Figure 6-41: M-Ratio Deterioration for FVC2004 DB1 (l) and DB2 (r) for Mean/Stdev Descriptor (AUTO)

The results presented in Figure 6-41 show a consistent trend of deterioration for both databases at a seemingly consistent rate. The number of scales investigated was not sufficient to determine when, or if, the trend levels out. This deterioration is somewhat expected because wavelet decomposition smoothes and down-samples at every scale. As the number of scales increase, more and more features are suppressed and the image becomes increasingly blurred. This in turn reduces the standard deviation at each scale and the range in which the means lie will become increasingly restricted, which is essentially restricting the feature-space. As a result of these factors, at a greater scale, the features become increasingly similar, so the lower scale features are more significant.

As with the energy ratio descriptor, the effective dimensionality is also reduced when automatic estimation is employed. The effective dimensionality, when automatic estimation is employed, is on par with the results of the manual estimation tests (i.e. around 50% - 60% are effective dimensions) for the first 3 scales. However, it deteriorates at a faster rate after this point. This again suggests the ineffectiveness of decomposing the image past the third scale as the information lacks individuality.

In addition to these tests carried out, it was decided to conduct an additional series of experiments to determine whether any additional benefits can be achieved by amalgamating these two descriptors in to one. It was discovered that the M-Ratio was more similar in magnitude to that of the mean/standard deviation descriptor than that of the energy ratio descriptor. This was also evident in the percentage of dimensions which were effective. For simplicity, the averages of the M-Ratios and effective dimensions percentage for all wavelets examined are presented in Figure 6-42 for both databases.

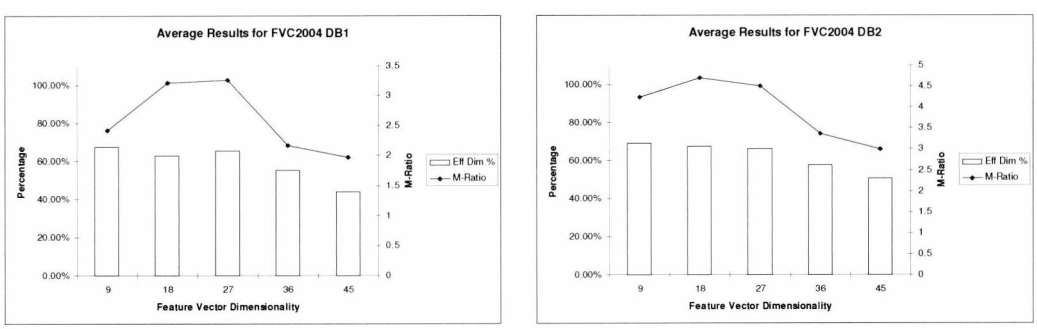


Figure 6-42: Mean Results of Energy Ratio/Mean/Stdev Descriptor for FVC2004 DB1 (l) and DB2 (r)

From the charts presented in Figure 6-42, it is immediately clear that when the feature vector dimensionality exceeds 27 (i.e. when the number of scales exceeds 3), both statistical tests deteriorate significantly. Even though the charts presented are derived from the average across all wavelets, the general trends are preserved. It is quite clear from these charts that the optimal M-Ratio occurs when 2 or 3 scales of decomposition is applied, although the average difference

between the two scales is marginal. Based on the individual results of each wavelet, other than a greater dimensional feature vector, the statistical tests do not show signs of any improvements.

It has been determined through the multitude of experiments conducted that the energy ratio descriptor outperforms the others when comparisons are made under fair conditions. However, there are many similarities. Throughout these experiments, it has been witnessed that the best results are obtained from the similar input configurations. For those tests that employ manual estimation of the reference point's location and orientation, it was found that rotating and enhancing the GBLC ROI prior to extraction was superior. For automatic estimation, best results were achieved when the ROI was enhanced only, although for the both cases, a ROI of 60-pixel radius was optimal. However, the M-Ratio can be influenced by the reduced population within the feature-space. Therefore it is difficult to establish whether the improvement in the M-Ratio is attributed to the improvement in the quality of the feature vector or that of the reduction in population. This same scenario is exhibited when the severity of the screening algorithm is increased. In theory the feature vector should be improved, although an aspect of that improvement could be due to fewer classes residing in the feature space.

As mentioned previously, the results provided in this section are derived from experiments that had more than 80% of the classes remaining after screening. Due to this, when the ROI radius is 80 pixels, too few classes remained. This is similar to the experiments where the severity of the screening algorithm was greater than Q2 (see Testing Framework, Chapter 3). Therefore, only the first two levels of severity were examined. However, it was found that the difference between the results of these two levels was minimal for a ROI radius of either 40 or 60 pixels, so no distinction is made. The breakdowns of the input configurations for both the manual and automatic estimation tests are provided in Figure 6-43.

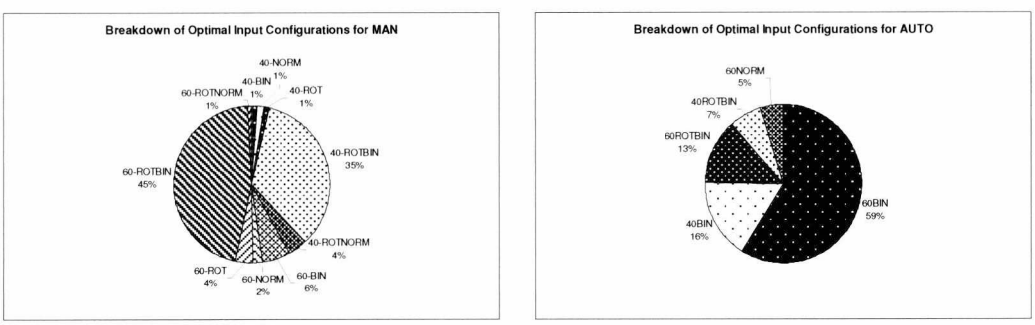


Figure 6-43: Breakdown of Optimal Input Configurations for MAN (l) and AUTO (r)

From the pie charts shown in Figure 6-43, it is clear that by rotating and enhancing the ROI prior to extraction produces optimal results in the majority of MAN tests. Furthermore, when automatic estimation is employed instead, enhancing the ROI is frequently the optimal input configuration. These findings were expected, although the ranking of the outcome of different pre-processing

methods was somewhat unexpected. It was expected that the ranking should be in the following order (best to worse):

- ROTBIN, ROTNORM, ROT, BIN, NORM, RAW (for MAN)
- BIN, NORM, RAW, ROTBIN, ROTNORM, ROT (for AUTO)

It was originally expected that when the reference point's orientation was manually estimated, the rotated series of tests should outperform the non-rotated ROI regardless whether the ROI was enhanced. Conversely, for the series of AUTO tests, it was expected that the non-rotated ROI would outperform the rotated, regardless of enhancement, because of the very inconsistent orientation estimate. This was because it was anticipated that, due to only 3 directions are represented in the decomposition, this would not be sufficient to handle the rotational variance. However, these initial expectations were denounced after testing suggested an alternative ordering. From analysing the results, it was determined that, in the majority of tests, the ranking of pre-processing methods was as follows (best to worse):

- ROTBIN, BIN, ROTNORM, NORM, ROT, RAW (for MAN)
- BIN, ROTBIN, NORM, ROTNORM, RAW, ROT (for AUTO)

This suggests that wavelet features are somewhat resilient to rotational variance otherwise significant improvement would have been exhibited. As the automatic estimation of the reference point's orientation is notoriously variable, the findings suggest that similar results are obtained by just enhancing the ROI.

6.5.3. Summary

In this section, numerous experiments were performed in order to determine the optimum wavelet and input configuration. Through exhaustive testing, it was determined that Daubechies-12, closely followed by Coiflet-18, consistently performed well for both databases and for all descriptors, so these wavelets were considered the best. However, significant differences were not observed between different wavelets. During testing, it also became apparent that significant degradation in the results is observed when the input image is decomposed with more than 3 scales. Therefore, the optimal results are obtained within 3 levels of decomposition.

Different features were investigated and it was found that the energy ratio descriptor was less bloated than that of the mean / standard deviation feature vector. This was because a greater percentage of the dimensions were effective and the M-Ratios were considerably higher. It was also found that no improvements were exhibited when the two descriptors were amalgamated into one. The input configuration was also discovered to have a significant impact on the results. Typically, by rotating and enhancing the ROI, which has been placed using a manual estimated reference point, produced optimal results. However, it was also discovered that rotating the ROI

does not provide any significant advantages, which suggests that wavelet features are somewhat rotation resilient, and that enhancing the ROI produces results which are only slightly inferior. The size of the ROI also has an effect although, from the variety of sizes investigated, the effect is minimal except that larger ROIs are more likely to be rejected due to the inclusion of background or poor quality regions. In general, a 60-pixel ROI was determined to be optimum, although it is not known whether it is due to the reduction in samples and / or to an increase in area that possesses additional features.

The results of these experiments have shown that the energy ratio descriptor is by far the more appropriate wavelet-based feature vector for use in generating encryption keys. By decomposing the image over 3 scales, a 9-dimension feature vector is created, which has 100% effective dimensions and an M-Ratio typically within double figures. These values are encouraging and it is anticipated that a significant contribution will be provided by this feature vector in generating an encryption key.

6.6. Conclusion

Many features and their variants were presented and investigated in this chapter. These features attempt to categorise the global structure and shape of the fingerprint. From the experiments conducted, it has been determined that there are many parameters that heavily influence the performance of these features. One of the most influencing factors is the input into the feature extractor. The feature itself is also important, but the performance of the feature is utterly dependent on capturing the same relative region of a fingerprint consistently. If the region interrogated is different on every acquisition, then the feature vector will inevitably be inconsistent. Therefore, the processing prior to extraction is just as important as the feature. Throughout the experiments, different ROI schemes have been implemented in order to determine which scheme consistently extracts the same relative region. The ROI scheme placement is also dependent on the location of the reference point, which is a significant problem to estimate consistently across all fingerprint types. The tests performed have shown that, unsurprisingly, considerable improvements are attained when the reference point is manually estimated. When automatic estimation is employed, severe deterioration is evident for all features, as a result of class overlaps and inconsistent feature vectors.

Throughout testing, it has been empirically determined that when sub-ROIs are implemented to capture local neighbourhood features, the RAD ROI consistently outperformed the GRID ROI in both the percentage of effective dimensions and M-Ratio. This suggests that the RAD ROI is more appropriate for neighbourhood features as it offers improved reproducibility. In essence the ROI schemes, which employ sub-ROIs, remove, to some degree, the necessity for the feature extractor to handle translational and rotational variations. However, to fully benefit from the sub-ROI based

schemes, the image needs to be rotated to a base orientation. When this is to be done automatically poses significant problems, which has been witnessed throughout the AUTO series of tests. Therefore, further research is required in order to develop a universal reference point location and orientation algorithm. This is because it has been shown that significant benefits are attained when the location and orientation estimates are consistent. Unless an accurate and consistent reference point estimation method is employed, it is recommended that rotating the image is omitted as not every fingerprint type will be handled effectively.

In addition to the RAD ROI scheme, significant improvements to the results of many features have been attained by rotating the image and enhancing the ROI prior to extraction. The act of rotating the image prior to feature extraction is essential for ROI schemes, which use sub-ROIs, to consistently extract the same region and rotation-sensitive features that employ a global ROI. Enhancing the ROI has also been determined to be optimal for all shape-based features (with exception to the GBLC invariant moment tests for FVC2004 DB1) as the ridge structure is well defined with contrasting background and foregrounds.

For this chapter, four different feature categories were examined. From these results, two categories provide encouraging results which, optimistically, will feature heavily in the key generation trials. These features are the Gabor-based FingerCode and the Daubechies-18 wavelet energy ratio descriptor. Although the FingerCode resulted in exceptional M-Ratio results, significant bloating was determined as a meagre percentage of the feature vectors dimensionality is effective. However, the energy ratio descriptor provided excellent dimensionality effectiveness and the M-Ratio was, in relation to the other features, markedly improved. The results of the other two features categories (frequency and invariant moments) were unremarkable, which, posted inferior M-Ratios and effective dimensionality, and it is anticipated that these features will contribute little to the key-generation stage.

Chapter 7. Biometric Encryption

In this digital age, an increasing amount of sensitive information is exchanged between parties across a variety of communication mediums. Much of this data is transported via the internet, which is fraught with many seemingly-anonymous malicious users who look to exploit the internet for their own gains. If precautions are not adhered to, this sensitive information maybe compromised. Therefore, it is essential that this classified data is protected. The most common method of ensuring the integrity of the information is by encrypting it. By creating an encrypted channel between parties, the possibility of the data being exposed is reduced. However, conventional encryption is essentially a knowledge-based authentication task, whereby keys, either known to the user or the system, are used to encrypt the data. As the old adage says "a chain is only as strong as the weakest link", so the security of the data is only as strong as the encryption key. If the key is weak, it can be easily compromised, hence the importance of choosing a strong key.

Unfortunately, it is not easy for humans to remember long and complicated keys so they are recorded, which is a further security risk. Therefore, biometrics provide an appealing alternative as they are characteristics that belong to an individual. The features obtained from the biometric samples can be used to strengthen security by substituting knowledge-based with characteristic-based authentication. This removes the onus from the user to remember keys as the keys are derived from the biometric samples. A number of biometric modalities (i.e. fingerprint, face, iris, keystrokes, etc) can be implemented, although different modalities are subject to a varying degree of intra-sample variation, which makes consistent key creation very difficult.

When used in traditional authentication systems, templates derived from the biometric samples are often employed to effectively manage the high intra-user variance. Many biometric encryption methodologies also make use of templates, whereby a key is bound to the template and is released upon successful authentication. This method is termed key-binding. However, by using templates, storage of said templates is required, which encumbers this approach with the same pitfalls of any template-based biometric authentication system. Alternatively, a key can be dynamically generated (direct key generation) from the biometric features. This methodology omits the template, which is primarily used to provide tolerance to intra-user variance. By removing the template, the task of handling intra-user variance is shifted to an alternative process. In this thesis, the pre-processing and feature extraction stages attempt to suppress this variance. Unfortunately the intra-user variance is still significant. Therefore it is vital that the key release, or generation, process of a biometric encryption scheme, takes intra-user variance, of varying degrees, into consideration. In this chapter, a review of the available literature is provided and novel approaches to key-binding and direct key generation using fingerprints are proposed.

7.1. Literature Review

Biometric encryption is a relatively new field of research where the discriminatory information possessed in biometric samples is employed to generate keys for pre-existing encryption techniques such as AES, triple-DES and RSA. It is important to note that the field of biometric encryption is somewhat misleading as biometric encryption is only a mechanism for key generation and is completely independent of any given encryption algorithm. The considerable advantage of this approach is that long and complex keys can be determined without the user having to be aware of the key. This strengthens the weakest link in the encryption chain as the key cannot be compromised through coercion or negligence, and the complexity ensures that brute-force cannot determine the key within a feasible timeframe. Even though the concept of biometrically-derived encryption keys has great potential, many difficulties impede successful application.

Encryption keys are required to have two vital attributes, which are stability and entropy. The key must be consistently reproducible (stability) and have a great number of effective bits, which ensures a large key-space. However a large key-space does not guarantee significant entropy. It is highly desirable that the distribution of keys within the key-space is such that each key has equal probability, which is indicative of high entropy as the key is seemingly random. This provides optimal security as a sub-key-space (as a result of significant clustering), which could potentially be exploited, cannot be determined from such a distribution. Currently, there are two schools of thought to deriving keys from biometric samples, which both have their own advantages and disadvantages. These are key-binding and direct key-generation, and overviews of current literature for both schools are subsequently provided.

7.1.1. Key Release (Key-Binding) Methodologies

As the title suggests, the encryption key for an individual is bound in some way to their biometric information, which is released upon authentication. Key-release methodologies are hugely popular and make up a significant proportion of the currently available literature. At a rudimentary level, key-release is similar to biometric authentication, whereby a user claims an identity and provides their biometric samples as proof. However, an n-bit key is acquired rather than a yes/no (i.e. 1-bit) answer.

Key storage and release are the most important aspects of a key-release system. A simple biometric authentication system could be employed to determine whether the user is who they claim and which obtains a key that is stored independently. This is akin to the claimant's identity being used as a primary key in a relational database. However, both the template and key storage requires the utmost security otherwise the compromise of the databases could potentially release the biometric information and encryption keys of all users [195]. Furthermore, if these databases are remotely located, a secure communications channel is required to ensure its integrity. Therefore additional

measures are employed to increase security of the templates, such as template protection [196, 197] that often employ one-way hash functions [198, 199].

One of the oldest and highly regarded key-release approaches was developed by Bioscrypt Inc (formerly Mytec Technologies Inc). Coined Biometric Encryption [13], Bioscrypt Inc outlined a secure key management scheme, which has been claimed to be applicable to any biometric image, not just fingerprints. Their method combines an arbitrary key with a biometric image to create a secure block of data, dubbed Bioscrypt™. The key can only be released when the Bioscrypt™ is presented along with its counterpart biometric sample. The Bioscrypt™ consists of a filter function, ID and linkage information. The linkage information is obtained from the linking stage which combines an arbitrary key with the biometric data. Following linking, the arbitrary key is used to generate the ID. When the ID of the Bioscrypt™ matches the ID obtained during verification, the key is released.

Alternatively the key can be hidden within the template using a secret, trusted bit-replacement algorithm, which extracts the key upon authentication [13]. This method can be applied to any template, so the template does not have to contain biometric data. If the template does not contain biometric information and the template is compromised, the biometric will not be compromised. However, this method implies that the bits of which the key is comprised will always be in the same location. Therefore, the bit locations could potentially be determined by an attacker. To improve the security of the key, "*Fuzzy Vaults*" [14] have increased in popularity.

The concept behind "*Fuzzy Vaults*" is that information, which is locked within the vault, can be revealed when a key (i.e. a feature vector) is sufficiently similar to the key that initially locked it. The important aspect of this approach is the fact that the keys do not have to be identical. Therefore, "*Fuzzy Vaults*" are suitable for biometric data, whereby the intra-user variance prevents consistent feature extraction. The initial "*Fuzzy Vault*" [14] employed error correcting codes (ECC), which are commonly used to correct data sent through a noisy communication medium. However, the term "*Fuzzy Vault*" can be applied to any approach that reveals secret information if the unlocking criteria is sufficiently similar to the locking criteria.

An alternative approach to creating a "*Fuzzy Vault*" is to employ Galois Field polynomials, as used within Shamir's Secret Sharing Scheme [200]. Given a secret value (i.e. key), a polynomial can be constructed using the secret value as the initial coefficient. By using feature values as a way to generate the additional coefficients of the polynomial, a "*Fuzzy Vault*" is created. The "*Fuzzy Vault*" can then be unlocked using only a subset of feature values. Provided that there are sufficient correct feature values, the original secret can be obtained through interpolation. "*Fuzzy Vaults*", in some form or another, have been employed using a variety of biometrics, such as fingerprint

minutiae [14, 16, 19, 201], face [20, 202], voice [17] and passwords [18]. However, the security of "*Fuzzy Vaults*" is questionable as research has suggested that they can be compromised [203, 204].

Key release/binding has significant benefits in that the key is completely independent of the biometric data and the key is only needed to be generated once. Provided that the key generation algorithm creates keys with high entropy, the desired properties of a key generation scheme are achieved. An additional benefit of decoupling the key generation from the biometric data is that the key can be easily changed if the key is ever compromised. This ability to change the encryption key is a desirable property and is major topic for biometric encryption researchers. Furthermore, biometric information cannot be stolen as it is never stored. Therefore, other than the key being compromised allowing access to previously encrypted data, the worst thing would be corruption to the Bioscrypt™/vault/template. This would be irritating but not catastrophic as they can be reconstructed with a different key and the user's privacy and security are unaffected. However, information still needs to be stored. Although to a certain extent the privacy and security concerns are allayed, people may still be uncomfortable with providing their biometric details but this could be mostly attributed to mistrust.

One of the main drawbacks of key release/binding schemes is that the user must have access to the Bioscrypt™/vault/template in order to obtain their key. Therefore, the information must be stored either locally or remotely. If the data is stored remotely, then the user must be connected to the server in order to obtain their key, which may not always be possible. Alternatively, the data can be stored locally, although the key release is restricted to the system on which the data resides. The data could be stored on a token, such as a smartcard, although the user will need to possess it whenever the key is required. By incorporating tokens, a method of reading the token is required whenever the user requires the key and the token can easily be lost, stolen or damaged. Therefore, having the key release mechanism stored somewhere restricts when and where it can be used. A potential solution to this restriction is to generate keys directly from biometric samples.

7.1.2. Key Generation

This school of thought towards key generation is not as highly researched as key release/binding. Key generation uses the features extracted from the biometric samples to create a key dynamically, without the need of storing sensitive biometric data or keys. Therefore, it is the intention that the same key can be generated anywhere, provided the software is installed on the system, which generates the keys.

One of the initial works that proposed a mechanism for generating encryption keys based on biometric data was presented in [205]. The concept was simple in that a user's biometric template was used as a seed value, from which a public and private encryption key can be determined. After

key generation, the template is discarded. This method was extended further [206] for application with the Public Key Infrastructure (PKI). However, this approach to key generation assumed that the biometric samples acquired were always the same. This in practice is not the case and hence the reproducibility of the keys is low. Therefore, an alternative approach to direct key generation is required that can consistently reproduce the same keys, given a relatively high intra-user variability.

A promising solution to this problem could be the generation of keys from the extracted features from the biometric samples. Although templates are determined from biometric features, it is the comparison of templates that provide intra-user variance management, rather than the template itself. Therefore, any features can be used to create a template provided the comparison stage is capable of managing the variability. By removing the comparison stage, the emphasis for reducing intra-user variability is shifted from template comparison to the feature selection and the key generation. To provide some resilience to the intra-user variability, some processing to the feature-space maybe required to ensure that the resultant feature value is reproducible. In [207], signatures are used to generate keys, whereby the feature boundaries are stored to assist the feature coding stage, which facilitates consistent key reproduction. This is one example of helper data being employed to enable consistent feature values to be extracted. Helper data does not contain any sensitive information regarding individual users.

Another example of helper data is presented in [208] where a codebook is used to ascertain the user. In this method, the feature space is quantised and the partitions in which the feature values lay represent code-words. The code-words are used in conjunction with the stored codebook to determine the user's identity. This method was expanded in [209], although encryption keys were determined by using a subset of multiple features from handwritten samples. Further biometric encryption methods, which employ helper data, are presented in [210]. Helper data can consist of normalisation, partitioning and mapping information, which allows consistent reproduction of feature values. This information is used to transform the input feature vectors to a domain, which minimises the intra-user variance. Essentially, any information that can assist repeatable key generation can be employed provided that no data, which can be used to circumvent the system or steal a user's identity, is stored. However, ideally, no information whatsoever should be stored. Even though an attacker should not be able to steal a user's identity or key, the helper data can still be corrupted, which means precautions, such as backups, must be adhered to, to ensure the integrity of the helper data. An attacker can tamper with the helper data, but a denial of service (DoS) attack can only be achieved. Although, helper data is stored, this approach to key generation is considered to be template-free as no sensitive (i.e. biometric data, keys and personal data) information regarding an individual is stored.

Key generation methods are still in their infancy due to the significant difficulties involved in generating reproducible and very strong encryption keys. Many of the methods outlined employ a handful of features, which restricts the entropy of the generated keys. As a consequence, the strength and the stability of the generated key from these methods are insufficient for use in a practical application. Stability can be attained but at a heavy cost to the strength and length of the key, although it has been suggested that by implementing more features, the number of effective bits should also increase. Furthermore, many approaches are so focussed on suppressing the influence of intra-user variance, that the inter-user variance is often neglected. If too much suppression is performed, individuality reduces, which also adversely affects the entropy of the key. However, probably the most significant drawback to key generation is that if the key is ever compromised, it would mean that the biometric that key was generated with would be irrevocably lost.

Unlike the key release/binding methodology, the key cannot be changed without changing the biometric. This leads on to cancellable/revocable biometrics [211, 212], which are currently undergoing research for application in template-based systems. The concept behind cancellable biometrics is that if a template is ever compromised, the biometric can still be used as a different transformation can be used to create a different template. This could be applicable to key generation in that different features or differently weighted features could be employed to change the key. Although cancellable biometrics may be a solution to this disadvantage, they are beyond the scope of this thesis and are hence not investigated further.

7.2. Novel Fuzzy Vault Fingerprint Smartcard Implementation using an Orientation-Based Feature Vector

Most applications employing biometric authentication store, in some form or another, feature data extracted from biometric samples. This feature data is commonly transformed into a template, which allows query sample to be identified or authenticated. However, there is a potential security flaw if the templates are compromised [11, 196]. Currently there are concerted efforts to try and move away from stored templates or even any form of stored data, especially in a centralised repository. These solutions are more frequently in the form of smartcards [19, 213, 214] or biometric cryptosystems [17, 18, 209]. However it is very difficult to remove the dependency biometric encryption methods have on stored data. Therefore, it is highly recommended that whatever is stored does not relate to the user's biometric data as the compromising of which could lead to identity theft.

Fuzzy vaults are a viable solution because the biometric data is not stored, but instead the biometric data is used to release secret information contained within a stored, locked vault. In this section, a novel biometric cryptosystem, which implements an orientation feature to lock / unlock a fuzzy

vault, is integrating into a smartcard solution is proposed. Using a similar approach described in [17], it is proposed that an encryption key is hardened using a fuzzy vault method based on Shamir's secret sharing scheme [200]. An orientation feature vector was employed as the locking / unlocking mechanism as it is arguably more reproducible and reliable when compared to that of minutiae points.

For this implementation, the entire vault is stored on a smartcard to reduce the possibility of circumvention. When the vault is downloaded onto a smartcard, all biometric information is deleted. This ensures there is only one vault in circulation for the user and can only be unlocked by the biometric samples of the smartcard carrier. This guarantees that the user must be present at the point of use.

The proposed biometric encryption scheme, as aforementioned, employs the Shamir's secret sharing scheme. Therefore a brief overview of Shamir's secret sharing scheme is provided. Following the overview, the processes involved in the vault construction are presented, which precedes the experimental results.

7.2.1. Shamir's Secret Sharing Scheme

The Shamir secret sharing scheme, as described in [200], is a well-known and commonly used cryptographic method, which when given a secret, divides the secret into a number of unique shares. The original secret can be retrieved by using a sub-set of these shares. This technique is also referred to as Shamir's *k-n* threshold scheme, whereby a secret is divided into *n* shares and *k* is the minimum subset size to retrieve the secret (i.e. at least *k* shares, from a maximum of *n* shares, are required to retrieve the secret). Being able to reconstruct the original secret from any subset of shares (provided that the subset is greater the k) is a very useful property and is an important aspect of this implementation. A basic overview of Shamir's secret sharing scheme follows.

7.2.1.1. Secret Division

Shamir's scheme is basically a two-stage method based upon polynomial interpolation. To divide the secret, a random polynomial of degree k-1 is defined using the secret as the initial coefficient. This polynomial is determined using Equation 7-1.

$$f(x) = \sum_{i=0}^{i<k} a_i x^i$$

$$a_i = \begin{cases} i = 0, \ \sec ret \\ i > 0, \ rand(\ ) \in Z_p \end{cases} \quad \textbf{7-1}$$

Where:

Z_p denotes that the value is an element in a finite (Galois) field [0, p) where p is a prime number larger that a_0 and n.

Once the random polynomial is defined, the shares are determined by taking n points sequentially along the polynomial. This will create tuples of [x, $f(x)$] where x = [1, n). Therefore the shares [D_1, D_n) can be evaluated by using Equation 7-2.

$$D_1 = f(1), \ldots, D_x = f(x), \ldots, D_n = f(n) \quad \textbf{7-2}$$

Where:

D_i is the i^{th} share.

Again the secret shares are elements of a finite field [0, p) and are therefore modulo p. This provides tuples of [i, D_i]. Obviously D_0 is the secret value, which should not be public and is hence not a share.

7.2.1.2. Secret Reconstruction

In order to recover the original secret, interpolation must be performed. As mentioned previously, $D_0 = a_0$ = secret and therefore can be retrieved by evaluating the polynomial to find $f(0)$. To determine the polynomial and subsequently evaluate it, Lagrange's interpolation algorithm was implemented as it is one of the most common methods to reconstruct the secret. Lagrange's interpolation determines each coefficient of the polynomial from their respective shares to reconstruct the polynomial. Lagrange's interpolation method is presented in Equation 7-4 and is employed in Equation 7-3 to evaluate the polynomial when $x = 0$.

$$D_0 = \sum_{i \in S} D_i \lambda_i \pmod p \quad \textbf{7-3}$$

$$\lambda_i = \prod_{j \in S, j \neq i} \frac{-x_j}{x_i - x_j} \quad \textbf{7-4}$$

Where:

S is the subset of shares.

D_i is the i^{th} share (i.e. a tuple).

7.2.2. Proposed Method

The proposed cryptosystem is essentially a two-stage process, whereby the orientation feature vector is extracted prior to the vault construction. The feature vector used in this implementation is orientation based and the orientations are obtained from determinable points within the orientation field. Therefore, to facilitate consistent location of these points, a certain amount of pre-processing must be carried out prior to feature extraction.

7.2.2.1. Pre-Processing & Feature Extraction

Before the vault can be constructed, a feature vector must be extracted to provide the locking mechanism. However, pre-processing, such as reference point detection, fingerprint orientation and segmentation, is required to achieve optimal reproducibility of the feature vector. The processing steps implemented to pre-process and extract the feature vector are subsequently provided.

- Image segmentation. The fingerprint is segmented (i.e. detection of background and foreground regions) using the STFT method outlined in [1].
- Reference point location. Any reference point can be implemented provided it consistently locates the reference point at the same relative point. Initially the Poincare index [6] was implemented, although the geometry of regions method, outlined in [24], was found to be more consistent.
- Estimate orientation of fingerprint and rotate image accordingly. Although not a necessary step, rotating the image to a base orientation suppresses rotational variance provided the estimation is accurate. For this step, the method provided in [89] was implemented.
- Extract orientation field. Many orientation field estimation methods are available, although the STFT method, detailed in [1], with a window size of 8, was found to be very simple to implement and efficient as it can be calculated concurrently to segmentation.
- Obtain inter-ridge distance. Can be estimated automatically using the frequencies of the frequency field obtained by the STFT method [1] or using a constant preset (i.e. 10 pixels for 500dpi image [24]).
- Extract orientations at specific points within the orientation field. The points to be extracted are determined from polar coordinates whose origin is the reference point. The points can be determined by using the Equations 7-5 and 7-6. The resultant location will need to be divided by the orientation field's window size to provide an index into the orientation field. Furthermore, if the points lie within the background of the image or outside the image's boundaries, then reject the fingerprint.

$$\Theta = \{\omega_0, ..., \omega_i, ..., \omega_{m-1}\}$$

$$\omega_i = \frac{2\pi \times i}{m} \quad \text{7-5}$$

$$X_j = RP_x + (r_j \cos\Theta)$$

$$Y_j = RP_y + (r_j \sin\Theta) \quad \text{7-6}$$

Where:

m is the number of angles.

r_j is the j^{th} radius in a list of radii. The list of radii contains n multiplicands, which are to be multiplied by the average inter-ridge distance.

Θ is a vector of angles.

RP_x and RP_y are the x and y coordinates of the reference point.

After following these steps, m x n points are determined, which together creates an m x n dimensional feature vector. From this feature vector, the fuzzy vault can be created.

7.2.2.2. Vault Construction

After determining the feature vector, the fuzzy vault is constructed by populating a table with true shares and dummy shares, which masks the true shares. It is this table that is downloaded onto the smartcard to be carried by the user in order to release the locked secret. The proposed method is similar to the techniques described in [17, 18], instead a fingerprint is used to lock and unlock the vault to obtain the original secret. In order to create the table, a certain number of pre-defined system-wide constants are required in order to ensure consistent feature extraction and secret decomposition and retrieval. The number of angles (m) and radii (n) in the list must be constant. Furthermore, the radii within the list must be kept constant, if not the orientations will be extracted from different points in the orientation field. One final pre-defined constant is that of the number of quantisation levels (q). The orientations are quantised to suppress the intra-sample variance, by using Equation 7-7, and the total number of quantisation levels is used to create the table. As the feature values are quantised into q levels, a table consisting of q columns and m x n rows is created, with all its elements initialised to 0.

Once the table is created, N training feature vectors are used to populate it. The orientation feature vector is extracted for each fingerprint and a range for each feature value is determined. Initially, the mean and standard deviation was used to determine a confidence interval, which would remove rogue values. However, orientations are cyclic (i.e. wraparound, akin to modulus arithmetic) and it was later discovered that feature values, whose orientations are near-vertical, result in an excessive range. This will populate an entire row of the table with key shares, which was found to increase the false acceptance rate (FAR) considerably. This was because the group of training feature vectors may have many rows within the table fully populated with shares, which results in other

fingerprints unlocking the vault. Therefore, an alternative approach to calculating the range was adopted.

To calculate the range using this new approach the training feature vectors need to be lined up side by side to create a matrix of L x N, where L is the length of the feature vectors and N is the number of training vectors. Each row of the matrix is then sorted (s_r) independently and the minimum and maximum values (min_r and max_r respectively) are determined using the algorithm in Figure 7-1.

- for r = 0 to L - 1
  - $max_r = s_{r,N-1}$, $min_r = s_{r,0}$
  - $\delta = max_r - min_r$
  - for c = 1 to N – 1
    - if $\delta > \pi - (s_{r,i} - s_{r,i-1})$
      - $\delta = \pi - (s_{r,i} - s_{r,i-1})$
      - $max_r = s_{r,i-1}$, $min_r = s_{r,i}$

Figure 7-1: Min and Max Calculation for Row Population

The result of executing this algorithm should be 2 vectors, which are for the minimum and maximum of each row. Each value within these vectors is then quantised using Equation 7-7, which are then used to flag the positions in the table where the true shares should go.

$$x_q = \frac{\left(x + \frac{\pi}{2}\right)q}{\pi} \qquad \textbf{7-7}$$

Where:

x is a feature value, which is an orientation.

x_q is the quantised feature value.

Now that the minimum and maximum quantised values ($qmin_r$ and $qmax_r$ respectively) have been calculated, non-zero values (i.e. 1) are placed in the table at the row, which corresponds to the sample number, in columns $qmin_r$ through to $qmax_r$. As the minimum quantised value can be greater than the maximum, it must be cyclic. For example, if the sample number is 3, $qmax_r$ = 1 and $qmin_r$ = 6 with a total of 8 quantisation levels, then non-zero values will be placed in row 3, columns 6, 7, 0 and 1. These non-zero values are the positions in the table where the shares are going to be placed. From the table, the number of shares (n in a k-n threshold scheme) can be calculated using Equation 7-8.

$$n = \sum_{r=0}^{r<L} \sum_{c=0}^{c<q} t_{r,c} \qquad \textbf{7-8}$$

Where:

$t_{r,c}$ is value at row r and column c of table t.

As k is equal to the length of the feature vector and n is determined from the flag locations in the table, the shares can now be calculated by using the method described in Section 7.2.1.1. As this implementation is designed for encryption/decryption purposes, the secret to be split is an encryption key, which in this case is a random number in the field $[0, p)$, where the prime number p needs to be quite large to ensure a wide key-space. As the secret splitting is conducted within a finite field and all calculation are modulo p, the prime number p will need to be kept constant throughout this implementation. This should result in n shares. However, it is not possible to recover the secret from the share alone. In order to reconstruct the polynomial both the share ($f(x)$) and its respective variable (x) are required. Therefore the share needs to be bound to its corresponding variable. To accomplish this, the share value was simply shifted to the left by 3 digits (i.e. multiplied by 1000) and its corresponding variable was added to the share to create a combined share value. By allocating the final 3 digits to represent the function's variable, this allows are to 999 shares, which should be plenty for most applications. These combined share values are distributed randomly in the table at the positions flagged. The rest of the table (i.e. the zeroed elements) is populated with a combined random share value and dummy variables, which are calculated using Equation 7-9.

$$dummyShare = (rand(1, p) * 1000) + rand(1, n) \quad \textbf{7-9}$$

Where:

p is the upper boundary of the finite field, which is a large prime number.

n is the number of legitimate shares.

The dummy shares are placed randomly about the table where the zeroed elements are in order confuse any potential attacker who somehow obtains a copy of the table. As with all cryptosystems, it is best assumed that the attacker will know everything about the system. Therefore they would know that k shares will unlock the vault and that they are hidden within the table. Furthermore, they would know that at least one element in each row of the table contains a share and that unlocking vector must contain unique variable values (x). So, in order to improve the security, multiple entries with the same variable value must be included. Assuming that on average there will be 3 entries with the same variable value within a table and using a 24-dimension feature vector, this would result in 3^{24} combinations. This is much stronger than a PIN number and a 6 digit alphanumeric (including combined upper and lower case letters using an English language keyboard) password, which have 10^4 and 62^6 permutations respectively. However, if some single entries are present within the table, the attacker would know that these entries are legitimate shares, which reduces the search space.

Once the table has been constructed, it is downloaded to a smartcard, or even transformed into a barcode. The table could be encrypted to increase security, although an attacker cannot obtain biometric information anyway because it is deleted as soon as the table is created.

7.2.2.3. Key Release

Releasing the key is very simple, provided the smartcard is presented along with a query fingerprint. Given a query feature vector, or even an average feature vector from a multitude of acquisitions, the key can be released by quantising the feature values and extracting the entries from the table on the smartcard. The selected table entries are the divided through by 1000, whereby the share value is its integral part and the variable value is its fractional part. These tuples are then used to interpolate the polynomial using Lagranges interpolation method outlined in Section 7.2.1.2, which is then evaluated to zero. If all the shares obtained from the table belong to the initial random polynomial, which was used to split the secret key, then the key will be retrieved.

7.2.3. Experimental Results

For the following experiments, databases 1 and 2 from the Fingerprint Verification Competition 2004 set of databases were used to assess the performance of the proposed method. Within each database there are 100 users, each providing 8 samples, which together creates a database of 800 fingerprint images. From each fingerprint image, a 24-dimensional feature vector was extracted using 8 angles and a list of 3 radii (i.e. OFRAD feature investigated in Chapter 4). In order to facilitate optimal feature vector consistency, a rudimentary screening stage was employed which rejected images when features were either extracted from the background or the calculated coordinates lay outside of the entire image.

As a screening stage was in effect, a list of radii needed to be determined in order to represent the global structure of the fingerprint, whilst retaining as many fingerprints as possible to provide a suitable representation of the feature-space. It was empirically determined that a list of 3 radii (3, 5 and 7) was optimal in that the features were extracted from regions, which covers a large proportion of the image, whilst retaining a significant percentage of the database. Furthermore, as there is considerable orientation change about the reference point, it was decided to extract features from regions at a relatively large distance from the reference point. This is why the first radius of the list is 3 inter-ridge distances from the reference point.

It is a desirable property of this implementation that only 1 sample is required to release the key. However, in order to construct the vault, a number of samples can be acquired, which allows the vault to manage perturbations in the feature vector. Although an average of multiple samples could be used to release the key, in the tests conducted only 1 sample was used to unlock the vault

because this is seen as the worse case scenario. When an average of multiple samples is used to unlock the vault, the performance should improve as variations will be suppressed. To assess the performance of the system, the number of training samples was varied. As there is only a maximum of 8 samples per user within the database, the number of training samples is limited. However, it was empirically determined that only a handful of user groups maintained 8 samples after screening. Therefore, the maximum number of training samples is restricted further to only 6 because at least one sample must be used for verification.

One benefit of having a small number of samples is that it becomes feasible to test every possible permutation. For example, if 3 training samples were required and there are 7 samples for a user in total, then there are 35 possible permutations. Verification was achieved by creating a vault for each possible permutation and using the remaining 4 samples for intra-user testing and the rest of the database for inter-user testing. The genuine acceptance rate (GAR) was determined from how many samples from the same user group released the correct key, and the false acceptance rate (FAR) was ascertained from how many samples from other users released the key. Figure 7-2 presents the receiver operating characteristic (ROC) curves for FVC2004 DB1 and DB2. The dotted line in the charts depict linearity because there are too few samples per user to test the system until FAR tends to 1. It is important to note that the desired result should see the elbow of the graph approach the top left (i.e. for all FAR, GAR = 1). Typically the ROC curve is generated by altering the decision threshold, but this is not the case in this implementation as there is no decision threshold. Therefore, the variable being altered here is the number of training vectors (N_t) that created the vault. This number is shown at its respective data point in the charts.

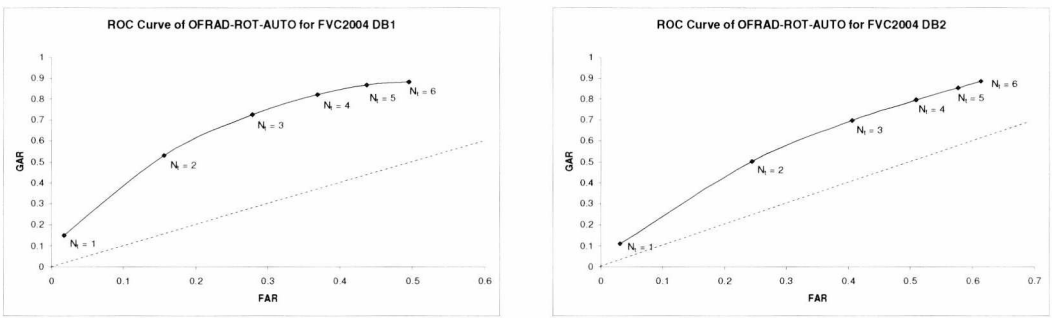


Figure 7-2: ROC Curve of OFRAD (ROT+AUTO) for FVC2004 DB1 (l) and DB2 (r)

It is clear from Figure 7-2 that the results are quite poor for both databases as there is hardly any discernable elbow in the curve. The GAR is only gradually increasing and does not approach a desirable rate (i.e. 90%) until 5 training samples are used to construct the vault. Furthermore, the curves are not significantly better than the line of linearity, which suggests that there are many false positives occurring. A significant factor, which has resulted in such poor results, is that the reference point location and image rotation was performed automatically. Consistent fingerprint orientation estimation is a difficult task, especially in the whorl and arch type fingerprints. The

results reflect this as the orientation algorithm was applied to all types. It is strongly suspected that if the whorl and arch types were removed from testing, then the performance should improve drastically. However, this is not an ideal operating scenario. Even though the whorl and arch fingerprint types are less common, a practical system cannot reject these tricky types. Therefore it is interesting to compare the results with non-rotated images to see what affect inconsistent orientation estimation has. Therefore, the same tests were performed with non-rotated images and the resulting ROC curves are shown in Figure 7-3.

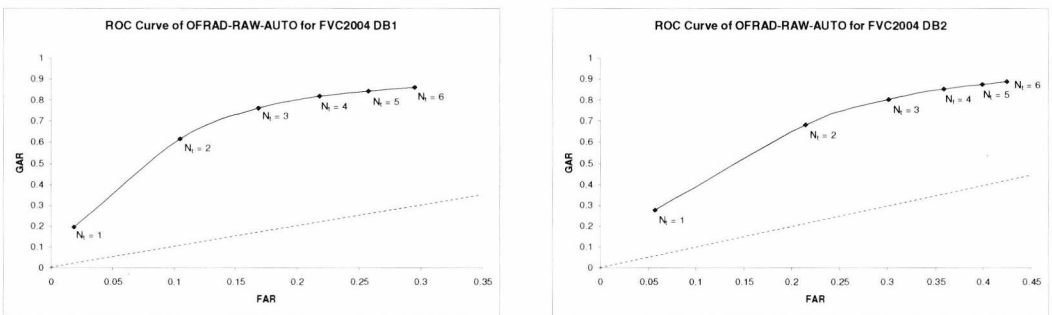


Figure 7-3: ROC Curve of OFRAD (RAW+AUTO) for FVC2004 DB1 (l) and DB2 (r)

The ROC curves presented in Figure 7-3 show a marked improvement over the ROC curves that were shown in Figure 7-2. This suggests that better performance is attained when the fingerprint is not rotated automatically. However, it looks as though that only the FAR has significantly improved and the GAR has hardly experienced any improvement. Therefore, by not rotating the image, the inter-user separation seems to improve, although minimal effects are exhibited by the intra-user separation. These results are attainable by employing existing pre-processing methodologies, which do not perform as well as claimed. The inconsistencies in reference point location and fingerprint orientation contribute heavily to the sub-standard performance metrics obtained. Therefore further tests were carried out, where the reference point location and the fingerprint orientation was estimated manually, to see how well the system would perform if the pre-processing methods performance matched that of a human. The results of these tests are shown in Figure 7-4.

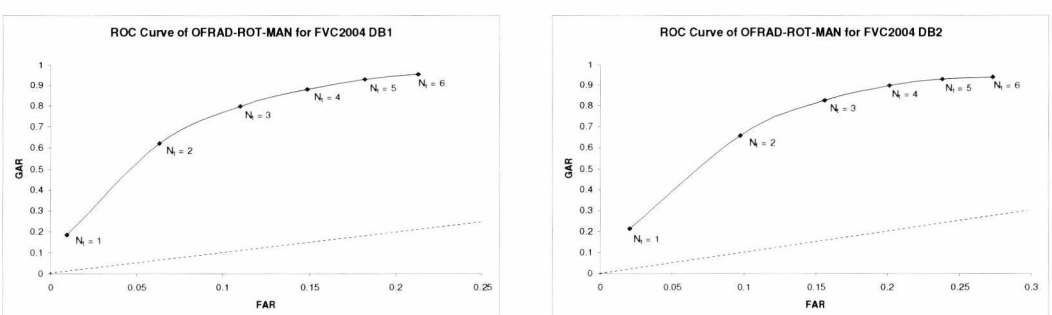


Figure 7-4: ROC Curve of OFRAD (ROT+MAN) for FVC2004 DB1 (l) and DB2 (r)

Figure 7-4 shows what sort of performance can be expected with near-human capabilities (i.e. an automated method that matches or exceeds the performance of human experts) of estimating the reference point and fingerprint orientation. The performance is not significantly better than that achieved by the non-rotated images series of experiments. Although the GAR does exceed 90% when 5 or more training samples are used to create the vault, the system investigated here does not truly reflect the performance of a practical system. Under normal operating conditions, poor images are rejected on the fly and another scan is requested until N decent fingerprints are collected. However, as the databases are static and have a limited number of samples per user, additional acquisitions cannot be obtained. Therefore a rudimentary screening method was implemented, whereby images were rejected where regions of low frequency and variance (below an empirical threshold) infringe the region of interest. During live scan operation, which employed quality control (using Neurotechnologija's Verifinger software development kit [59]), manual estimation of fingerprint orientation and reference point location, and a limited user base of 30, the optimum GAR was 98.7% with a FAR of 7.2% for 5 training samples and 5 testing samples. Although the GAR is good for most applications, the FAR is still too high for a large scale implementation.

7.2.4. Discussion

An easy-to-implement fingerprint smartcard solution using a unique feature vector has been presented in this section. The findings suggest that an implementation of the current system would be suitable only for small to medium scale projects. This is mainly due to a relatively large FAR. However, the large GAR obtained is encouraging and this is because the orientation feature vector is one of the most reproducible, which makes it an ideal basis candidate for use in a rudimentary biometric encryption system. The reproducibility of the feature vector is improved with the application of the fuzzy vault as the secret can be recovered even with a degree of variability within the feature vector.

This flexibility comes at a cost as the FAR increases because similar fingerprints can unlock the vault. Although, it was anticipated that the orientation feature vector would not be as discriminative as minutiae, the system's performance can be improved by adapting it to accept multiple features. For example, minutiae could be incorporated similarly by using a DISC ROI scheme and placing key shares in a table that relates to the sub-ROIs which contain minutiae. Other features can be incorporated in the same way as each row of the table corresponds to a feature value from a sub-ROI.

Implementing further feature vectors will also make the vault more difficult to attack. Not only because of the increase in permutations, but because a similar feature vector could potentially be estimated by a human. Assuming the attacker knew everything about the processes used in the proposed system and possessed a lifted print from the user, they would know from approximately

which region the orientation is obtained from and they could estimate the fingerprint orientation. From this information, orientations can be estimated and then quantised accordingly, giving them locations in the table. However, this problem is common to most fingerprint features, including minutiae, because the fingerprint is in the public domain. If enough information is known about the system, spoofing is a concern.

In the method proposed, the key being shared facilitates file encryption/decryption. However, this in itself is an indirect form of authentication, whereby only the legitimate smartcard holder can decrypt the file. An identification or verification system could be created based on this premise by either having a central repository of encrypted files, which were encrypted by the user at the time the user was enrolled, or by saving the encrypted file to a smartcard. By simply using the biometric and the obtained key to decrypt the file, upon successful decryption the user will be authenticated.

7.3. Template-Free Biometric Cryptosystem Based on the Fingerprint Modality

Previously, an approach to release an encryption key based on the fingerprint biometric of an individual was proposed. However an alternative approach to obtaining an encryption from biometric data is to generate an encryption key using the extracted features. This essentially binds a key with an individual, so the same key should be determined from every presentation of the biometric sample. This means that the key can be obtained without additional tokens, such as a smartcard or barcode, and a communications uplink to a central repository. This is a highly desirable property as the same key can be determined from any system in the world, provided that a means to acquire the relevant biometric data and the key generation software is installed. However biometric data possesses a high degree of intra-sample variance (varying between biometric modality), which subsequently makes the generation of a very strong and stable encryption key a considerably difficult task.

In order to attain the desired properties of encryption keys derived directly from biometric features, a mechanism for transforming the feature values from a noisy domain to a sufficiently stable domain is necessary. Certain pre-processing methods can suppress image noise and reduce the variance attributed to affine transformations and partial overlap, although significant intra-sample variance is exhibited in the extracted feature. Generating a key from these features, without further processing, will be highly unstable. Therefore, an approach is required that takes an arbitrary feature space and transforms it to an alternative feature space, which is resilient to perturbations.

In the field of biometrics, such a mechanism is currently underrepresented as current research favours key release biometric encryption schemes. This is because the key release methodology does not rely on the biometric data to generate very strong and stable keys, which is the biggest

challenge for direct key generation methods. Current approaches to direct key generation from biometric features employ mechanisms such as codebooks [215] and maps [210, 216], which are obtained from distributions within the feature-space. Further expansion to [216] was proposed in the ICMetric series [217-219], which generated keys directly from debug features of embedded electronic systems. Although direct key generation algorithms are not strictly in the scope of this thesis, a proof of concept was desirable to assess the application of fingerprint features within such a system. For this prototype system, the ICMetric approach [217, 218] was used as a starting point and to serve as a benchmark. This method of direct key generation is subsequently expanded and enhanced.

7.3.1. Direct Key Generation from Fingerprint Features

As the ICMetric approach [217, 218] is simple to implement, the initial investigation into key generation was centred on this method. The processing in this approach is minimal and consists of two stages. The initial stage, which is a calibration phase, creates maps that map every quantisation interval in the feature space to corresponding virtual modes. The number of virtual modes is equal to the number of groups used for calibration and the location of the virtual modes was kept constant at predefined intervals. The maps are stored, although no user-specific information is contained within the maps. The operation phase retrieves the maps and subsequently constructs the key from the feature values using the maps

7.3.1.1. Calibration

The purpose of calibration is to generate helper data that will assist the key generation stage by transforming the volatile feature-space to an alternative, stable space. In this investigation, calibration is employed similar to that presented in [217, 218], where the helper data takes the form of feature-maps. However, certain aspects of the method have been adapted in order to be more suitable for biometrically derived features.

Before any maps, or additional helper data, can be created, it is paramount that the feature vectors used for calibration are normalised. A feature-map is generated for each feature and, as multiple features can be implemented, it is necessary that they be combined in some way so only a single encryption key is constructed. Two methods of feature vector combination was discussed in [217, 218], whereby individual feature maps were used to obtain mapped values for all features, which are then either summated (*Combined (Sum)*) or used as individual components for subsequent concatenation (*Combined (Concat)*). A third method is to simply sum all features, which have been normalised accordingly, together prior to mapping, essentially creating a single-dimensional feature vector. If the feature vectors were to be combined by summation (denoted as *Feature Sum*), it is essential that each feature value is normalised to a standard range to ensure each feature vector has

equal influence. Otherwise features which have greater magnitudes will dominate and the features with small magnitudes will have minimal effect. Furthermore, if the *Feature Sum* approach is to be employed, then summation should take place immediately after the feature vectors have been normalised. This creates a single feature value, which can be used to create a single feature-map. *Combined (Sum)* and *Combined (Concat)* employs feature-maps for each individual feature, where the resulting mapped values are used to construct the key during normal operation.

Prior to normalisation, it is desirable that many samples are acquired in order to improve the representation of the feature-space. Therefore, normalisation is best performed when all calibration samples have been acquired in order to obtain good estimates of each features global maximums and minimums. Each feature's global maximum and minimum is stored so during operation the feature vectors can be normalised to the standardised range. Normalisation is quite standard and it essentially standardises the feature-space to a range of [0, 1]. However, it is far easier to work with discrete values (i.e. integers) when it comes to key generation rather than rational values (i.e. floating points), so the min-max normalisation method presented in Equation 7-10 was used to spread the range from [0, 1] to [0, 10000], with the values rounded to the nearest integer. This provides an accuracy of up to 5 decimal places. Increasing the range further will afford greater numerical accuracy, but at additional storage and processing costs when feature maps are generated and stored. The following sub-section detail how these maps are generated.

$$nval = \frac{(x - \min_f)(\max_c - \min_c)}{\max_f - \min_f} + \min_c \qquad \textbf{7-10}$$

Where:

x is the current value belonging to the feature vector

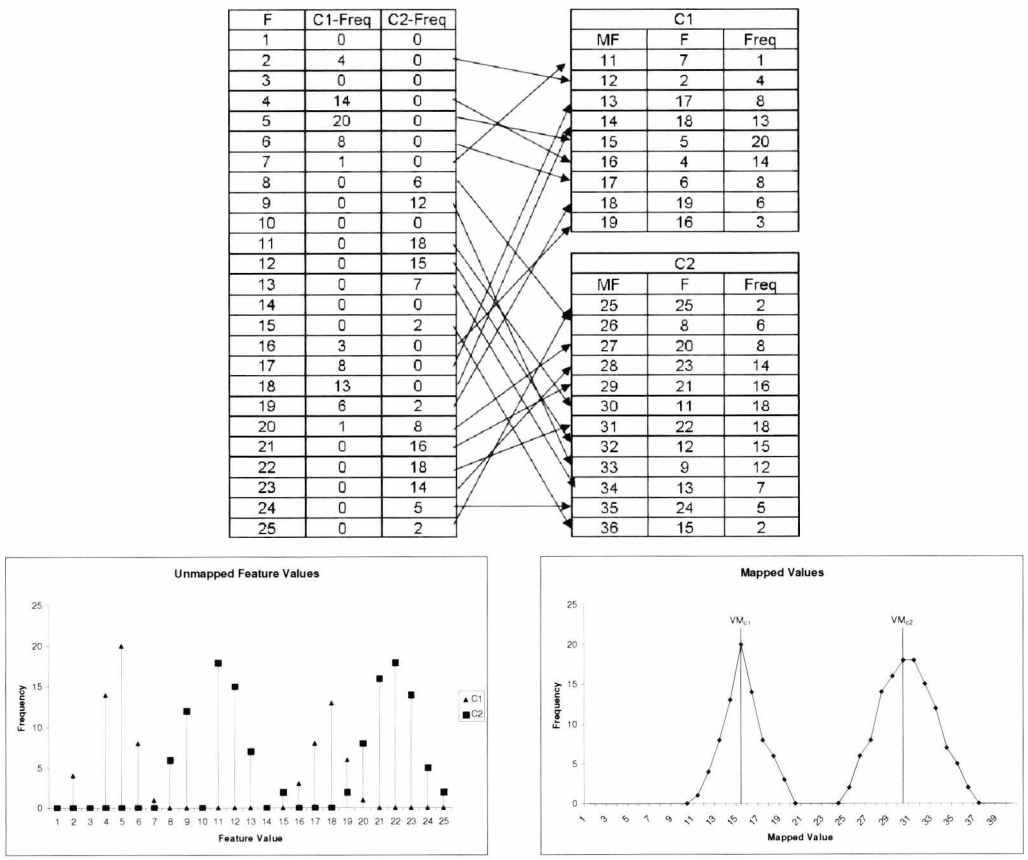
min_f and max_f are the minimum and maximum values of the entire feature space for all training samples respectively.

min_c and max_c are the constant minimum and maximum values of the normalised range respectively.

7.3.1.1.1. ICMetric Mapping

The mapping process outlined in [217, 218] is a simple process, which was originally devised in order to manage multi-modal distributions. It essentially creates a series of feature-space mappings, where each feature is mapped to an alternative mapped-space. Each calibration class is assigned a static virtual mode (sufficiently spaced to prevent overlapping) and for every feature value obtained, the feature value is mapped to the virtual mode of the class of which that value occurs most frequently. A list for each class is compiled containing the feature values and their respective frequencies. Based on the frequency of each feature value, each list, which contains all feature values judged to belong to that particular class, is sorted in descending order. The feature values are

then mapped in descending order to the region about the corresponding virtual mode's modal location. This is performed by mapping the most frequent value to the modal location and, as the frequency decreases, mapping the remaining feature values to positions increasingly deviating on either side of the modal location. A graphical representation of a simple example is provided in Figure 7-5, whereby two classes are mapped according to this method.



F	C1-Freq	C2-Freq
1	0	0
2	4	0
3	0	0
4	14	0
5	20	0
6	8	0
7	1	0
8	0	6
9	0	12
10	0	0
11	0	18
12	0	15
13	0	7
14	0	0
15	0	2
16	3	0
17	8	0
18	13	0
19	6	2
20	1	8
21	0	16
22	0	18
23	0	14
24	0	5
25	0	2

C1		
MF	F	Freq
11	7	1
12	2	4
13	17	8
14	18	13
15	5	20
16	4	14
17	6	8
18	19	6
19	16	3

C2		
MF	F	Freq
25	25	2
26	8	6
27	20	8
28	23	14
29	21	16
30	11	18
31	22	18
32	12	15
33	9	12
34	13	7
35	24	5
36	15	2

Figure 7-5: Example of ICMetric Mapping

At this stage, the mapping only maps the values which the calibration stage has seen. A lot of unmapped values remain because many feature values are not present in the calibration samples. Therefore it is necessary to map these unseen feature values to an appropriate location in the mapped space. In [217, 218], as the feature space for an individual user could be multi-modal, it is assumed that each mode exhibits a Gaussian behaviour, although these modes may not be disjointed. As the modes are treated independently, a mode-finding algorithm is employed to count the modes and locate the ranges in the feature space that each mode occupies. From analysing the feature values between these ranges, an estimate of that mode's probability density function (PDF) can be acquired. Once all PDFs for each calibration group have been determined, the probability of the unmapped feature values can then be calculated. Using these probabilities, the feature values are mapped similarly to the frequencies, whereby a list of probabilities are generated for the feature values belonging to each class and then sorted. Starting where the map finished for each virtual mode, the feature values, from each list of probabilities, are then mapped to their respective virtual

modes in a similar fashion to Figure 7-5, although other approaches to mapping unseen values can be employed.

As with most training stages in pattern recognition systems, the more samples that are obtained for calibration, the better the representation of the feature space. In [217, 218] the number of samples, for both calibration and normal operation, is not a factor because only synthetic samples are used. Even when the system in [217, 218] is implemented completely, the number of samples is only restricted by elapsed time as an electronic system will not tire and become irate with incessant sample requests. However, this is not the case when the same technique is applied to biometrics. If too many samples are required, this will inconvenience the user, which will only frustrate and irritate them. Therefore, the number of samples used during normal operation must be restricted to only a handful of acquisitions. For calibration purposes, the number of samples can be significantly increased to generate the maps according to [217, 218], but the fewer the training samples required the better in regards to the user's viewpoint.

When a restriction is imposed on the number of samples, mapping the feature space akin to that of the ICMetric method is fruitless as the vast majority of the feature space will remain unmapped. As the unmapped values are subsequently mapped based on probability, it is logical to map all values based on probability from the outset. Unlike the proposed debug features used in [217, 218], it is assumed biometric samples will follow a unimodal Gaussian distribution instead of multimodal. A Gaussian distribution is assumed because biometric features are typically distributed in this way and it is likely that fingerprint features will also be Gaussian distributed. Furthermore, the Gaussian distribution was chosen due to its desirability and simplicity. Assuming Gaussian distribution, the feature value probabilities can be mapped using the observed means and standard deviations of each group. This means that it is not necessary to generate complete maps as the mapping of feature values to virtual modes is determined based on probability. Therefore the means and standard deviations can be stored instead of full maps, which is much more efficient in terms of storage, although storing the statistical information of each users biometric features raise security concerns. Mapping is an important step to reducing the influence of inconsistencies between intra-sample feature vectors, although mapping in this way serves an additional purpose.

By mapping in this way an element of group belongingness is represented by the mapped value. If the frequencies and the subsequent PDF mapped only to a single value (i.e. the virtual mode) then this is, in essence, a classification task. However, mapping to a Gaussian distribution within the mapped space permits belongingness to be taken into account, whereby the closer the mapped value is to the virtual mode, the greater the confidence of that value belonging to the mode. A similar approach to mapping the feature-space was implemented based on the calculated probabilities rather than the observed frequencies.

To generate a feature map, the most probable user group for each value within the feature-space was determined using Equation 7-11. The values are grouped according to the most probable user group and then they are sorted based on the calculated probabilities. Then in the same fashion as in [217, 218], where an example is provided in Figure 7-5, the values are mapped to the group's respective virtual mode. The most probable value is mapped to the centre of the group's respective virtual mode and, as the probability decreases, map the remaining feature values belonging to that group to positions increasingly deviating on either side of the centre. Once all feature maps have been generated, they are stored so they can be retrieved when the key needs to be constructed during operation.

$$C = i;\ if\,(p_i = \max(p_x; \forall x = [0, N]))$$
$$p_i = \frac{1}{\sigma_{i,f}\sqrt{2\pi}} \exp\left(-\frac{(X_f - \mu_{i,f})^2}{2\sigma_{i,f}^{\ 2}} \right) \qquad \textbf{7-11}$$

Where:

C is the highest probable calibration set for feature f of feature vector X.

$\mu_{i,f}$ and $\sigma_{i,f}$ are the mean and standard deviation of training set i for feature f respectively.

X_f is the value of feature f of feature vector X

By mapping the feature-space in this fashion, thresholds based on the probabilities can be used to ensure only highly probable features are included when constructing the key during normal operation. If the probability of a feature value belonging to a group does not meet some predefined criterion, the sample is rejected and a new one is requested. This method of rejecting features was analysed in [219]. However this approach is only practical when many samples are available and where multiple replacement samples can be obtained. This is not a tenable solution for a biometric system as it is impractical to continually request another sample when one feature value is rejected. In order to improve the reproducibility of the biometrically-derived encryption key, an alternative approach is required.

7.3.1.1.2. Proposed Improvement to ICMetric Mapping

It is assumed that only a coarse representation of the entire feature space can be determined from calibration because the number of training samples is limited. Therefore rejecting samples based on thresholds derived from the PDFs is not appropriate because its accuracy is questionable, which may result in continual rejections. This is compounded when the feature-space is very densely populated, with multiple overlaps between intra-group distributions, as witnessed in many fingerprint feature vectors throughout the research conducted. Therefore, a method is proposed that permits multiple distributions to be mapped to the same virtual mode in the mapped space. This will allow similar feature values from different groups to produce very similar, if not the same, mapped value.

This is not possible in the ICMetric [217, 218] approach because every group has its own distinct virtual mode in the mapped space, which is placed incrementally along a linear scale. Imagine that distributions of calibration set 1 and 100 overlap. Based on the PDFs, the probability of belonging in each set can be high. However, as the mapped space is linear, where the virtual modes are positioned based on the order the calibration sets are presented, the distance between the virtual modes of set 1 and that of set 100 could be very large. Misclassification between the two, which in fingerprint features could be highly frequent, will significantly change the resultant key. In [219], a "feature trimming" algorithm, which rejected features that did not conform to some predefined criterion and acquired another sample, was implemented to reduce the misclassification rate. However, as mentioned previously, biometric systems do not have the luxury of acquiring fresh samples so this method is inappropriate for a biometrically derived key. Therefore, the ability to map to the same region is an important step if this approach is to be used in a biometric system.

To accomplish this approach to mapping, the virtual modes are determined on a feature by feature basis. For each feature, the mean and standard deviation for all training sets is determined. Using these values, the distributions, which overlap for that particular feature, are identified. Equation 7-12 presents the expression used to determine whether two distributions overlap.

$$Overlaps = \begin{cases} false, & if\,(lo_2 > hi_1 \;\; OR \;\; lo_1 > hi_2) \\ true, & otherwise \end{cases}$$
$$lo_1 = \mu_1 - (f\sigma_1);\; lo_2 = \mu_2 - (f\sigma_2) \qquad \textbf{7-12}$$
$$hi_1 = \mu_1 + (f\sigma_1);\; hi_2 = \mu_2 + (f\sigma_2)$$

Where:

μ_x and σ_x is the mean and standard deviation of distribution x respectively.

f is a factor that allows the significance of overlap to be altered.

Once it is known which distributions overlap, a union-find algorithm is used to group together the distributions which are connected through overlapping. The number of groups is then used as the number of virtual modes for that feature and then each grouping is assigned to its respective virtual mode according to its sorted probabilities. This allows the virtual mode, for each calibration set, to change according to the degree its distribution overlaps with that of other groups. Maps are generated similarly to the ICMetric mapping approach whereby each value in the feature space is mapped to a virtual mode according to the most probable distribution. A series of maps, one for each feature, is then stored for later use during normal operation.

7.3.1.2. Operation

Once the features have been extracted, the very next action to execute is feature vector normalisation. Each feature's global maximum and minimum were calculated and stored during

calibration so the features can be normalised to the same range using Equation 7-10. At this point, if the feature vectors are combined by summation (*Feature Sum*), then the summation should take place immediately after normalisation. A normalised feature value, or *Feature Sum*, is then used as an index into the feature's respective feature-map to retrieve the base number. This is shown in Equation 7-13. A single mapped value is retrieved when only the *Feature Sum*, or a single normalised feature, is used to generate the key, whereas a series of mapped values is retrieved when the feature vector is to be combined later using concatenation (*Combined (Concat)*) or summation (*Combined (Sum)*).

$$base_f = M_f(x_f) \qquad \textbf{7-13}$$

Where:

$M_f(x)$ is the mapped value from feature map f with a feature value x.

x_f is the feature value of feature f.

7.3.1.2.1. Key Construction

The mapped value is essentially a positive integer value, which is used to generate a component of the key. The mapped value itself cannot be used as a component of the key because it is not stable enough on account of two factors. Firstly, the mapped value is not a fixed number as it is derived from variable feature values (although can be if all mappings map to the virtual mode's modal value) and secondly, the natural binary representation is inappropriate for generating encryption keys. This is because the Hamming distance between two neighbouring binary values can be within a wide range, depending on the magnitude of the binary values, and it is not known which bits will be stable. This is typified by the following example of the natural binary values of 15 (i.e. 01111) and 16 (i.e. 10000).

It is clear that only a change of 1 decimal value has occurred yet this has caused a significant change in the bits used to represent these values. The Hamming distance between these values is 5, meaning that 5 bits have changed value. This means that there are not any stable bits. This highlights the problems with binary, where the lower order bits change frequently, which means in order to obtain a stable value, a large proportion of lower order bits may need to be discarded. However, by doing this, a significant range in the feature-space will obtain the same bit pattern and any subsequent key will not result in many effective bits. Therefore, an alternative representation of the mapped values is required. In [217, 218], Gray Codes were used for such a purpose.

Gray Codes [220], which are also known as reflected binary codes, are an alternative binary representation where the Hamming distance between two successive values is 1. The significance of using Gray Codes is that the Hamming distance between two arbitrary positive decimal values will be significantly reduced compared with natural binary. This means more common bits will be present from which a key can be constructed from. To calculate the Gray Code from a decimal

value, the value needs to be converted in to binary first and then Equation 7-14 is used to convert the binary value.

$$g = b \otimes \left\lfloor \frac{b}{2} \right\rfloor \qquad \textbf{7-14}$$

Where:

b is the binary representation.

g is the Gray Code representation.

Unfortunately, converting the mapped values to Gray Codes will still produce an unstable key as many bits will still change. Therefore it is important to localise these common bits and discard the unstable bits. An approach similar to that presented in [217, 218], was initially employed. This method involved converting a range of values, centred on the mapped value, into Gray Codes and determining which bits were stable. The range was determined from a mapped value ± a constant, which is determined empirically during calibration. A component of the final key is constructed from the stable bits and their respective bit location. Figure 7-6 presents a simple algorithm for constructing the key.

- bpos = 0, bval = 0
- min = base – m_c, max = base + m_c, range = (max – min) + 1
- for b = 0 to 32
    - $Acc_b = \sum_{i=\min}^{i \le \max} \left((D2GC(i) >> b) \,\&\, 0x01 \right)$
    - If Acc_b > T x range
        - bpos |= 0x01 << b
        - bval |= 0x01 << b
    - Else if Acc_b < (1-T) x range
        - bpos |= 0x01 << b
- keypiece = (bpos << 32) | bval

Figure 7-6: Range Conversion Algorithm

The algorithm presented in Figure 7-6 shows that every value from min to max is converted to a Gray Code by the D2GC function. For every bit of each value within the range, an accumulator counts how many "on" bits are present at each location. If the number of "on" bits is greater than a percentage (T) of the number of values within the range, then the bit is deemed stable with a value of 1. If the number is less than a percentage (1-T) then this bit is also stable, although with a value of 0. The component of the key (key-piece) is essentially the positions of the stable bits concatenated with the values of the bits at those positions. If the normalised feature vectors have

already been combined by summation (*Feature Sum*), or there is only a single feature, then this algorithm only needs to be run once as there is only a single basis-number (i.e. mapped values). Furthermore, if the *Combined (Sum)* approach is employed, then all the mapped values are summed together at this stage to create a single, larger base-number from which the key can be constructed by executing the algorithm once. However, for *Combined (Concat),* this algorithm would need to be run for every basis number (i.e. mapped value), with the components of the key (key-pieces) concatenated together. If there are many features, the computational costs could be significant.

An alternative to this key generation method is to discard lower order bits. The lower order bits tend to fluctuate within a given range, so if the volatile bits are removed, a stable, shorter bit pattern remains. For each feature, a certain number of lower order bits are discarded from the Gray Code representation of each base number. This is followed by concatenating the resulting bit patterns together. Obviously, if there is only a single feature, such as in *Feature Sum* or *Combined (Sum)*, then the algorithm is only executed once, although for *Combined (Concat)* it is executed for each feature. Figure 7-7 presents the algorithm for generating each key piece, whereby *n* depicts the number of lower order bits to discard and D2GC is a function that converts base number (*i*) from decimal to Gray Code.

- bpos = 0, bval = 0
- base = D2GC(i)
- for b = n to 32
    - If ((base >> b) & 0x01)
        - bpos |= 0x01 << b
        - bval |= 0x01 << b
    - Else
        - bpos |= 0x01 << b
- keypiece = (bpos << 32) | bval

Figure 7-7: Discard Lower Order Bits Algorithm

The number of lower order bits, which need to be discarded, can be determined at the calibration stage. To accomplish this, the number of lower order bits discarded is iteratively increased and the subsequent keys are constructed. On each iteration, the keys are compared and the number of discarded bits, which produce the best performance based on the calibration samples, can then be used during the operation stage. However, this approach would require additional storage, detailing how many lower order bits need to be discarded, which if compromised would provide information that could reduce the search-space for an attacker.

7.3.1.2.2. Novel Approach to Base Number Generation using Euclidean Distances

There is a fundamental problem with the map-based method previously outlined, which is that it is assumed that the features being used are 1-dimensional. This means that multi-dimensional feature vectors need to be collapsed to a single value before they can be implemented. As some fingerprint features are multi-dimensional, it is therefore essential that the features are represented by a single value so the map-based key generation method can be performed. Principal Components Analysis (PCA) can be used to collapse a multi-dimensional feature vector to a single dimension, although it is far simpler to sum the dimensions together. However, it is assumed that when a multi-dimensional feature vector is collapsed to a single value, its discriminatory power reduces significantly. Therefore, a method is proposed, which creates an alternative base number from multi-dimensional feature vectors.

In this method, it is proposed that the base number is determined from Euclidean distances obtained from the multi-dimensional feature-space, instead of mapping single-dimensional feature-spaces. It is assumed that this will preserve the discriminatory power of the feature as groups are generally better separated within multi-dimensional space. In this method, the calibration stage simply ascertains the mean vectors of each calibration set. These provide landmark points in the multi-dimensional feature space, whereby the location of any point in the feature space can be determined using the mean vectors for reference. Initially, the base number was to be determined from the vector sum of the vectors between the sample point and the means of the calibration sets. This would require obtaining a multi-dimensional direction, which is a difficult task when each dimension is dependent on each other. Therefore, dimension reduction would be required to bring the multi-dimensional feature-space down to 2 or 3 dimensions, in order to simplify direction extraction. However, this is exactly the opposite of what this method is trying to achieve because potentially useful, discriminatory information maybe lost during dimension reduction.

To retain the dimensionality of the features, it was decided that the Euclidean distances between a sample and the mean vectors would be sufficient for a rudimentary proof of concept. Although the Euclidean distance from a sample to the global mean would suffice, the range of base numbers was considered too small. However, by accumulating the Euclidean distance between the sample and all the calibration set's mean vectors, a larger range is obtained. A larger range is preferential as the potential of extracting more effective bits is greater and it is hoped that with a greater range, the groups will be better separated.

To determine a base number, it is important that before the means of the calibration sets are calculated, the feature vectors are normalised. For this method, many multi-dimensional feature vectors can be made in to a single multi-dimensional feature vector by concatenating the component vectors together *(Feature Concat)*. These component feature vectors are of varying

dimensionality and range. Some feature vectors may be of the magnitude of 10^6 whereas others might be of the order 10^{-6}. When the Euclidean distance is calculated, feature vectors that have very small magnitudes will become negligible and will have minimal effect on the base number. In order to ensure that each feature vector is represented equally, the component feature vectors must be normalised to the same range. Therefore, during calibration, and prior to calculating each user groups combined feature vectors mean, each component feature vector was normalised using the min-max normalisation method, which is presented in Equation 7-10. The normalised feature vectors can then be concatenated together to create the final feature vector. The means of each user group is then determined.

This approach to key construction requires that the calibration stage results in a collection of mean vectors and each feature-space's range, in order to normalise the sample's feature vectors prior to key generation. The base number is calculated using the Euclidean distance between the mean vectors and the normalised feature vector. This is obtained by using Equation 7-15. Consequently, the base number is converted to its Gray Code equivalent and final key is constructed using either the range conversion method, presented in Figure 7-6, or the discarding of lower order bits approach (Figure 7-7).

$$base = \sum_{t=0}^{t<N} \sqrt{\sum_{i=0}^{i<M} \left(X_i - \mu_{t,i}\right)^2} \qquad \textbf{7-15}$$

Where:

X_i is the i^{th} value in feature vector X.

N is the number of training sets

M is the dimensionality of the feature vector.

$\mu_{t,i}$ is the i^{th} value in the mean vector from calibration set t.

7.3.2. Experimental Results

The purpose of conducting these experiments is to assess the stability and reproducibility of the biometrically generated encryption key. For these experiments a selection of fingerprint features were used. The selection of these features was based on the results obtained from the previous chapters of this thesis. It was determine through numerous experiments that the best performing features were as follows:

- FingerCode with 3 tracks and a track width of 20 pixels.
- The directional sum of the orientations from OFRAD.
- The energy of Daubechies-12 wavelet across 3 scales. As multi-dimensional features are summed, to obtain a single dimensional feature for the map-based approaches, the energy ratio could not be used. This is because the sum will always equal 1. For the map-based approaches, the raw energy value is used, whereas the ratio is employed for other tests.
- Minutiae within 60 pixels of the reference point.

All these features were extracted from the images in FVC2004 DB1 and the reference point location and the fingerprint orientation was determined manually. The lowest level screening algorithm was also implemented to retain more samples as this will allow a larger training set, which is important to provide the most accurate depiction of the feature space. By employing these features, it ensures that the performance of the system will produce the best results possible for the database. Furthermore, it was decided in the interest of obtaining optimal results that multiple samples were to be used. From the minutiae experiments, which exhibit improvement of the statistical tests with a greater number of samples, the elbow of the graph was located when the mean vector of 4 samples was used. However, judging the results of a small opinion poll, people would prefer to submit 3 or less samples during normal operation. Therefore, the following experiments will employ 3 samples for the key generation stage as this is the most samples potential users would find acceptable. After screening the samples of FVC2004 DB1, it was found that 94 groups, which contained 3 or more samples, remained.

For calibration however, more samples can be used. Regrettably, the database used for testing consisted of only 100 users with 8 samples each. Although there are 800 images in the database, following the screening algorithm there are considerably less samples and there were very few groups (2), which still possessed all 8 samples. There was also too few groups remaining when 7 samples (12) were used, yet 51 groups remained when 6 samples were employed. Therefore, it was decided 6 samples would be used for calibration. In order to calibrate the system, the mean and standard deviation can be obtained from all 6 samples. However, during operation, 3 samples would be used, so it was decided that it would be best to calibrate using the same setup, which would be used for testing. To achieve this, all the permutations that contained 3 samples from the 6 training samples were obtained. This resulted in 20 permutations and the mean vectors for each permutation were subsequently used for the actual calibration.

In order to test the performance of the system, for all of the 94 groups containing 3 or more samples, every permutation of 3 samples within each group was determined. A key was then determined from the mean of each permutation using the three methods previously described (i.e. the ICMetric-based mapping [217, 218], the same mapping but with multiple distributions mapped to the same virtual mode and the alternative Euclidean distance method). The different methods and their variants are analysed in terms of the key stability, the number of effective bits and the number of unique keys. Key stability is measured by the percentage of keys per user which match (Match Rate) and the number of unique keys is obtained by determining the percentage of keys that are unique (i.e. number of different keys in key-space divided by the number of classes). As there are 94 groups within the key-space, there should ideally be 94 unique keys, which would result in 100% of the keys being unique. The number of effective bits (B_{eff}) was determined by calculating which bits within the key-space change. This was obtained by using Equation 7-16.

$$B_{eff} = CountOnes(xorkey)$$
$$xorkey = andkey \otimes orkey$$
$$andkey = key_0 \ AND \ key_1 \ AND .... \ key_{n-2} \ AND \ key_{N-1}$$
$$orkey = key_0 \ OR \ key_1 \ OR .... \ key_{n-2} \ OR \ key_{N-1}$$

7-16

Where:

CountOnes(x) is a function that counts the number of "on" bits in a binary value.

key_i is the i^{th} key from a list of generated keys (i.e. the key-space).

N is the number of generated keys within the key-space.

As many fingerprint features are multi-dimensional, it is a prerequisite of the map-based methods investigated that only single-dimension features were to be used. Therefore, it was decided that the dimensions of a multi-dimensional feature vector were to be summated to produce a single feature value. This approach to reducing the dimensionality was favoured over employing Principal Component Analysis (PCA) because there are too few samples used for both calibration and operation. It was assumed that, by employing PCA to reduce the dimensionality, with too few samples, would be highly inaccurate. Even if sufficient samples were used for calibration, the eigenvector corresponding to the greatest eigenvalue would need to be stored and retrieved later during operation. However, depending on the quantity and the dimensionalities of the feature vectors employed, this can significantly increase storage costs.

In order to summate the different dimensions, the dimensions are normalised accordingly. Many multi-dimensional features are comprised of dependent dimensions, which lay within a common range. However, some multi-dimensional features are constructed with independent dimensions that lie within vastly different ranges, such as Hu moments. Some Hu moments were of the order 10^{-6}, and some were much lower, in the order of 10^{-20}. When there are significant differences between the magnitudes of each dimension-space, it is not possible to sum these together as the value will be completely dominated by the much larger values. The dimensions with minute values would be overshadowed and therefore have minimal effect, which is why normalisation is crucial.

The first method investigated was the ICMetric-based mapping approach. Each group within the calibration set was mapped to virtual modes, where the first virtual mode was located at 10000 and every subsequent virtual mode was placed at intervals of 1000. The virtual modes begin at 10000 to primarily prevent mappings below zero and an interval of 1000 was used to prevent virtual modes overlapping. As a range centred about the base numbers is used, it is conceivable that no bits will be 100% stable, so a stability threshold (T) of 90% was used. This meant that the bit is deemed stable if the bit value is the same in more than 90% of the training samples for that group. The first series of experiments constructed the key from converting a varying range of values, centred on the base numbers, to Gray Codes and the stable bits were determined. Figure 7-8 and Figure 7-9

present the results of these experiments. In these experiments the range was iteratively increased by 1000, which is the interval between each virtual mode.

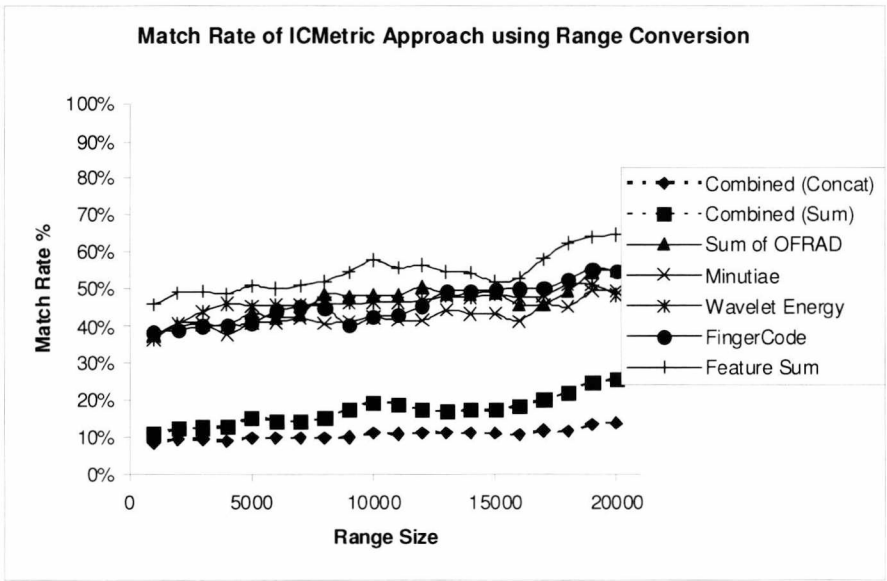


Figure 7-8: Match Rate of ICMetric Approach using Range Conversion Key Construction

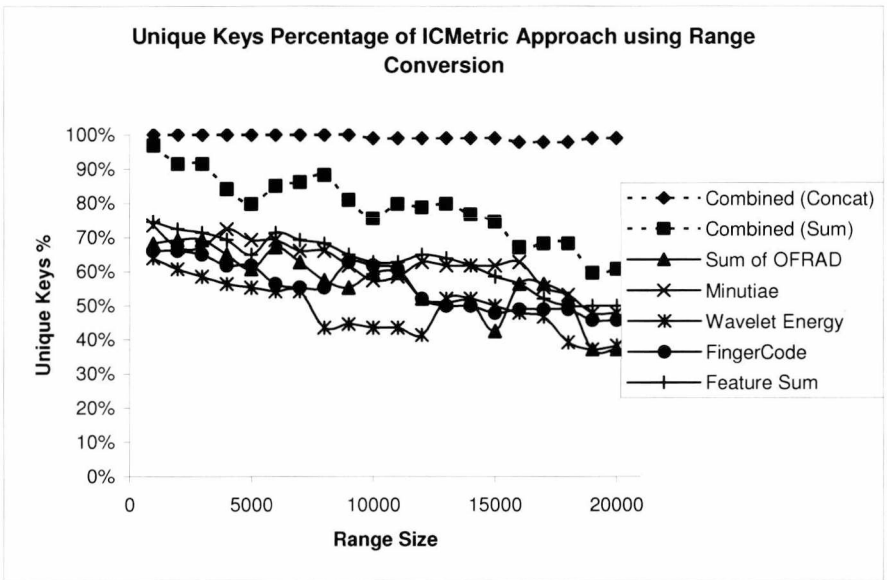


Figure 7-9: Unique Keys Percentage of ICMetric Approach using Range Conversion Key Generation

From the results presented, it is evident that combining the feature vectors by summation (*Feature Sum*) produces a slight improvement in the reproduction rate than when individual features are used to generate a key. However, as expected, *Combined (Concat)* performs significantly worse. This is because only one value needs to be wrong and the key cannot be reproduced. Hence the maximum possible reproduction rate attainable using *Combined (Concat)* is limited to the worst performing individual feature. Furthermore, the reproducibility of *Combined (Sum)* is also poor, but not to the

same extent as *Combined (Concat)* due to the same reason. It is also clear from the charts that, with the exception of the post-mapping combination results, the performance is actually quite similar. This is also evident in the number of effective bits. The number of effective bits starts at 16 bits, for a range of 1000, and then gradually decreases to 6 bits. The rate, at which the number of bits reduces, is almost identical for each test. In general, *Combined (Concat)* resulted in a similar pattern, although it was quadrupled as 4 feature vectors were used. There was no discernable relationship between the number of effective bits for *Combined (Sum)* and that of the other tests, although it can be assumed that it is attributed to the differing magnitudes and ranges of the base-number.

It is also noticeable from the charts that the match rate is far too low for practical applications, whereby the number of effective bits should be at least 50 and preferable greater than 80 [221]. This is because of the methodology of mapping to virtual modes. It is expected that, by using an initial range of 1000 (i.e. the interval between virtual modes), this should encompass all values which have been mapped correctly. As the range increases, more values from the neighbouring virtual modes are included in the calculation. There are 51 virtual modes (there are 51 calibration classes, although not all of them are used), spaced evenly along the scale. However, not all of these virtual modes are populated, so if these unpopulated modes are included in the range, the match rate will not improve, but it will decrease the number of effective bits. This means a larger range would be required so more samples will be included in the key construction stage. This range may be significant because, not only can unpopulated virtual modes be located anywhere in the mapped-space, erroneous mapping can result in a basis number being obtained from any of the populated virtual modes. Hence, in order to obtain a reproduction rate desirable for practical usage (i.e. where users will not become irritated by the continual resubmission of samples), a range encapsulating a large proportion of the mapped-space maybe required. Then again, this would result in very few unique keys and the computational costs and the execution times will rise. Alternatively, greater reproduction rates at lower computational costs can be attained by dropping the lower order bits. The results of applying this method are presented in Figure 7-10 and Figure 7-11.

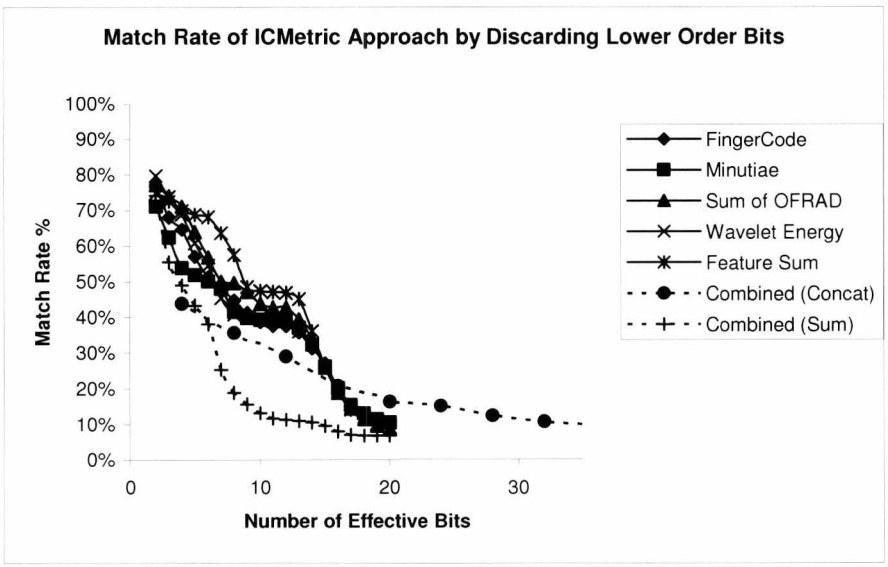


Figure 7-10: Match Rate of ICMetric Approach by Discarding Lower Order Bits

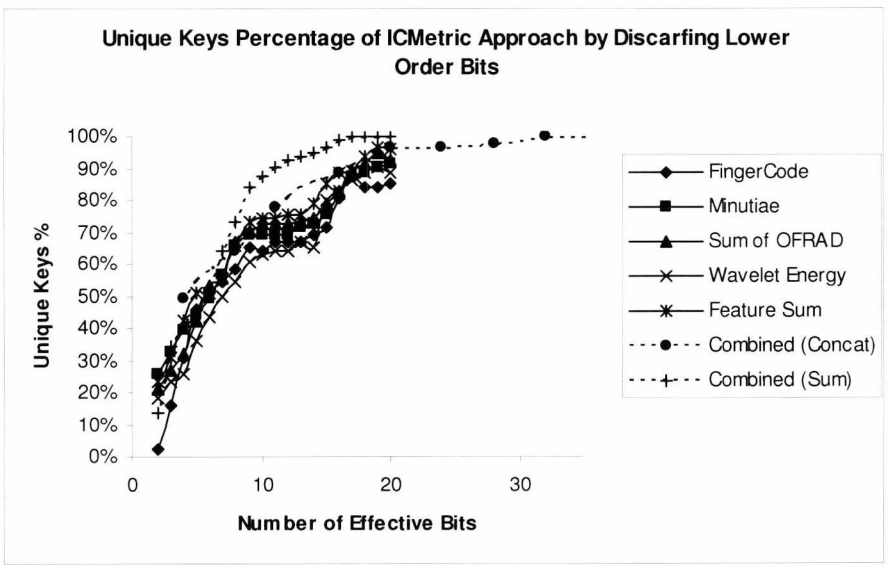


Figure 7-11: Unique Keys Percentage of ICMetric Approach by Discarding Lower Order Bits

The charts shown in Figure 7-10 and Figure 7-11 depict the reproduction and unique key rate against the number of effective bits. In both charts, the scale of the x-axis is reduced for clarity purposes, although the results for *Combined (Concat)* extend to 64 bits. It is immediately evident from these graphs that similar trends are obtained when a single feature vector or *Feature Sum*, is used. This again is attributed to the mapping methodology and it is suggested that the flattish regions within each trend is attributed to unpopulated or under-populated virtual modes. As with the range conversion approach to key generation, *Feature Sum*, marginally improves the reproducibility performance, whereas *Combined (Sum)* and *Combined (Concat)* performs noticeably worse. However, the results of the post-mapping combination approaches are improved, which suggests that discarding lower order bits would be the better approach to generating the key for these combination methods.

Unfortunately the performance obtained is still far below that required for a practical system. Not a single test broke the 80% match rate mark, even when only 2 effective bits were used! This suggests that the ICMetric mapping approach cannot be implemented directly in to a system whose features overlap considerably, such as a fingerprint biometric encryption system. This is mainly attributed to two significant drawbacks. The first weakness is that there is a virtual mode for every user that provided calibration samples. Some virtual modes may possess few, if any, values. This means that a large sub-space of the mapped-space is empty, which increases the distances between the effective virtual modes. Therefore, finding a suitable constant range is difficult because it may be larger for some features, but smaller for others, as it depends on the frequency and location of empty virtual modes within the mapped-space for that feature.

The second drawback is due to the mapping of overlapping distributions to alternate sub-spaces, especially when the equal probability point (i.e. crossover) between 2 overlapping distributions is high. This would cause many erroneous mappings. The ICMetric approach was able to reject samples, which did not meet some criteria, and request further samples. One such criterion was that the probability of the value belonging to the most probable virtual mode is sufficiently larger than the probability of it belonging to the second most probable. Unfortunately, the feature-space of most fingerprint features is densely populated with many overlapping distributions. It is not desirable to continually request the user to resubmit a replacement sample if it does not conform, especially as this can occur frequently. This is why the ICMetric mapping approach was improved by permitting sufficiently overlapping distributions to map to the same sub-space. However, it is not known to what extent of overlapping is judged to be sufficient. Therefore, this was ascertained empirically by altering factor f in Equation 7-12.

The factors investigated were 3, 2, 1, 0.5, 0.25, 0.125 and 0.0625. It was found that factors 3, 2 and 1 resulted in all distributions being mapped to the same virtual mode as each distribution was interconnected through overlapping. As the factor decreased, the reproduction rate also decreased, although the number of unique keys and the number of effective bits increased. This was to be expected because the greater the factor, the fewer virtual modes there will be as many distributions will be mapped to a common virtual mode. Therefore, for a highly reproducible key, the factor should be relatively high. However, if a highly entropic key is required the factor should be quite low. Although high entropy is greatly desired, there is not much point in having a strong key if it cannot be reproduced. So out of the two desirable key properties, the stability takes precedence over entropy. The best factor for key stability was found to be 0.5, which resulted in keys consistently exceeding 70%. Therefore, distributions are said to be overlapping if the point of equal probability occurs within half a standard deviation away from the means of both distributions. The results of the series of experiments for this setting are presented in Figure 7-12 to Figure 7-14.

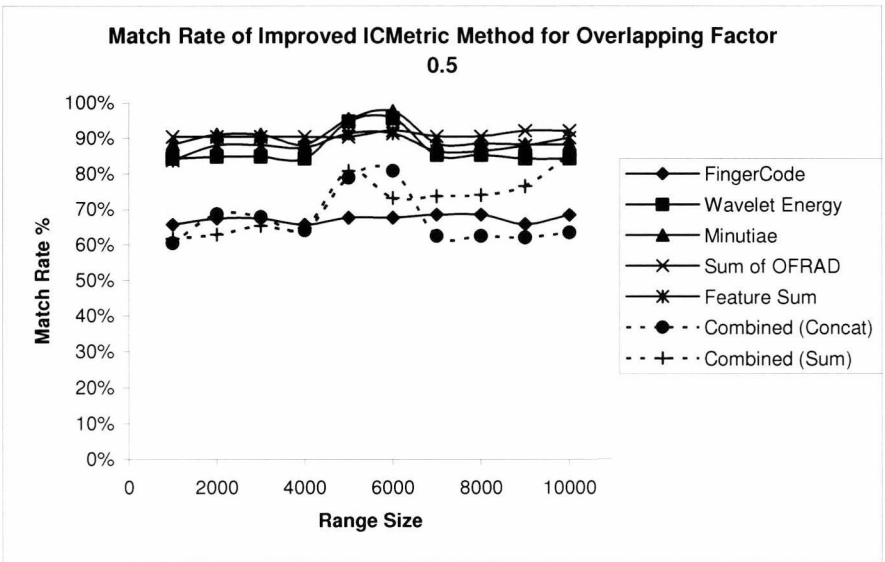


Figure 7-12: Match Rate of Improved ICMetric Method

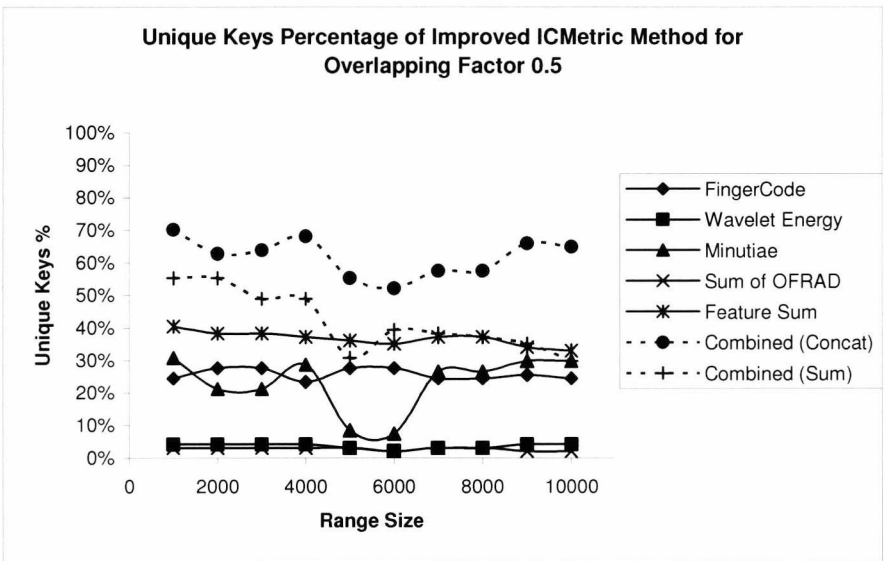


Figure 7-13: Unique Keys Percentage of Improved ICMetric Method

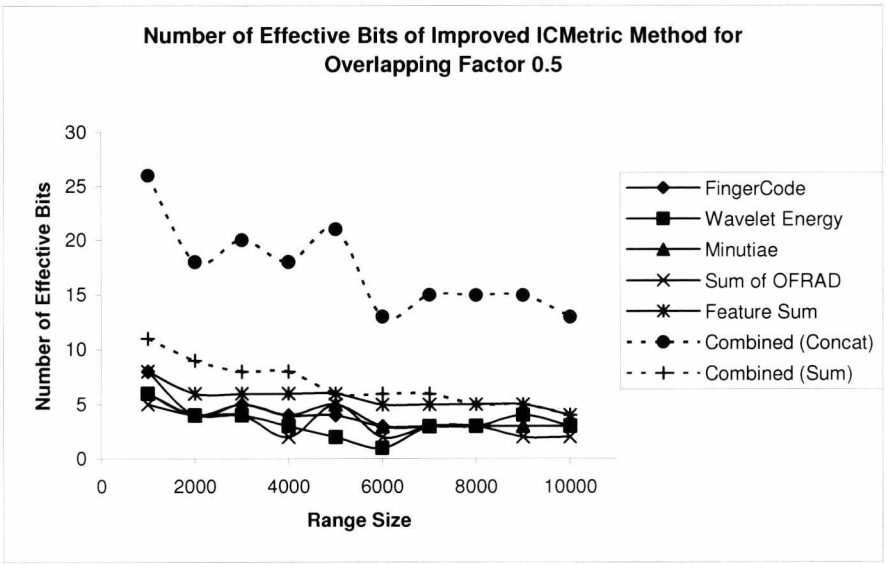


Figure 7-14: Number of Effective Bits of Improved ICMetric Method

The graphs for overlapping factor 0.5 suggest that reproducible keys can be constructed, although with fewer effective bits and reduced uniqueness. From Figure 7-12 it is clear that the wavelet energy, minutiae, sum of OFRAD and the Feature Sum result in a match rate in excess of 80%. However, only the wavelet and sum of OFRAD feature vectors result in extremely poor uniqueness, although the uniqueness of the minutiae feature drops considerably between range sizes of 4000 and 7000. Although the intended system is for multiple features, it is interesting to see the performance of the system for each individual feature as it suggests the contribution of that feature when the features are combined. However, it is only a suggestion of the contribution because correlations between multiple features have not been taken in to account. Evidently, in this system, much of the individuality seems to be attributed to the FingerCode and minutiae.

The main purpose of testing these key generation methods is to evaluate the combination of multiple feature vectors as a multi-feature application is intended. It was originally hoped that combining multiple feature vectors would be synergistic (i.e. the performance would be better than the sum of the individual feature vectors performance). However, this is not the case. In these experiments, it was found that when the *Feature Sum* method of combination was employed, the match rate does not always exceed an individual features match rate. Although the match rate is not always optimal, it does improve key individuality with a similar number of effective bits. Post-mapping combination techniques, such as *Combined (Sum)* and *Combined (Concat)*, were always expected to result in poorer reproducibility performance as erroneous mapping of a single feature prevents the key being correctly reproduced. However, the entropy of keys generated using these methods of combination is increased. Initially, it was assumed that post-mapping combination methods would have their reproduction rate restricted to that of the worse performing feature vector, although this is not always the case.

In the series of test, where the overlapping factor was set to 0.5, most individual feature vectors produce keys which are highly reproducible, but possess few effective bits and many repeated keys. However, post-mapping combination improves the entropy. This is reflected in the results, where the improvement is significantly noticeable. Coupled with high match rates of the component feature vectors, these methods also produced highly reproducible keys for this overlapping factor. By employing an overlapping factor of 0.5, and a range size of 5000, the best performance was attained using the *Combined (Concat)* method, which correctly reproduced a key, with 21 effective bits, 79% of the time. It also produced 52 unique keys out of a possible 94 (i.e. 55% unique key percentage). This balance of stability and entropy was not bettered by any other test. Therefore a significant improvement can be attained by simply allowing overlapping distributions to be mapped to common virtual modes. However, it is believed that this method can be slightly improved further by ensuring that all virtual modes are populated before including them in the map.

Currently, the virtual modes are designated a sub-space in the mapped-space without checking whether any values can actually be mapped there. A simple check can be performed at the sorting of probabilities stage, whereby the virtual mode is discarded if there are no probabilities to sort, which prevents gaping holes appearing in the mapped space. This means a smaller range will encapsulate more feature values. Although this improved mapping method reduces the occurrence of erroneous mapping, it is still possible. Since the virtual modes are evenly positioned along a linear 1-dimensional scale, erroneous mappings can induce a large error, rather than minimising the error.

As this method can only operate with single-dimension feature values, it is assumed that dimensionality reduction may indirectly affect the frequency of these erroneous mappings. This is because the collapsing of a multi-dimensional feature vector to a single dimension may significantly increase overlapping and, if a suitable overlapping factor is not implemented, many erroneous mappings can result. Therefore, an alternative approach to base number generation was employed, which employed multi-dimensional feature vectors. By keeping the feature in its multi-dimensional space, the separation between user groups should be better and it preserves all dimensional information. Furthermore, provided that the dimensional deviations are not huge, the distance from its respective groups mean should be manageable. In this method, the base number is generated from the sum of the Euclidean distances between a sample and each of the calibration set's mean vectors. The results obtained from this method, whereby the key was generated using the range conversion approach, is presented in Figure 7-15 to Figure 7-17.

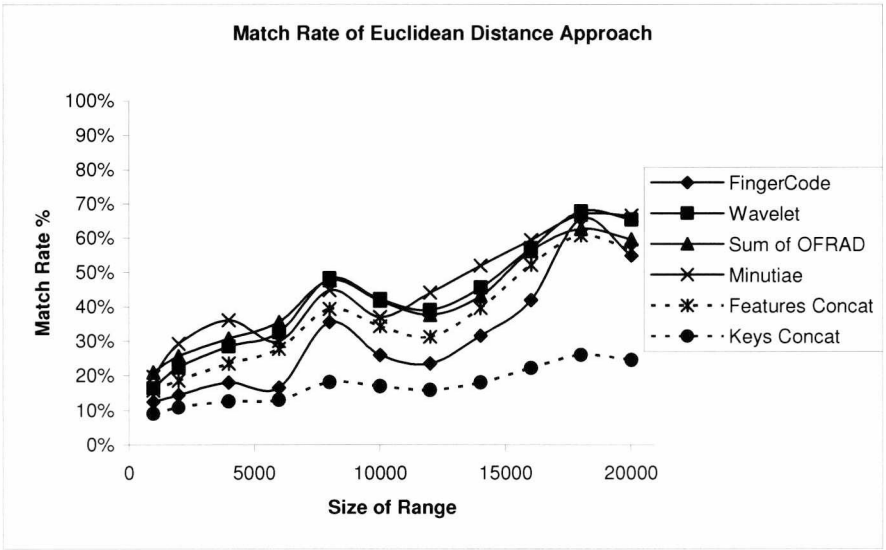


Figure 7-15: Match Rate of Euclidean Distance Approach

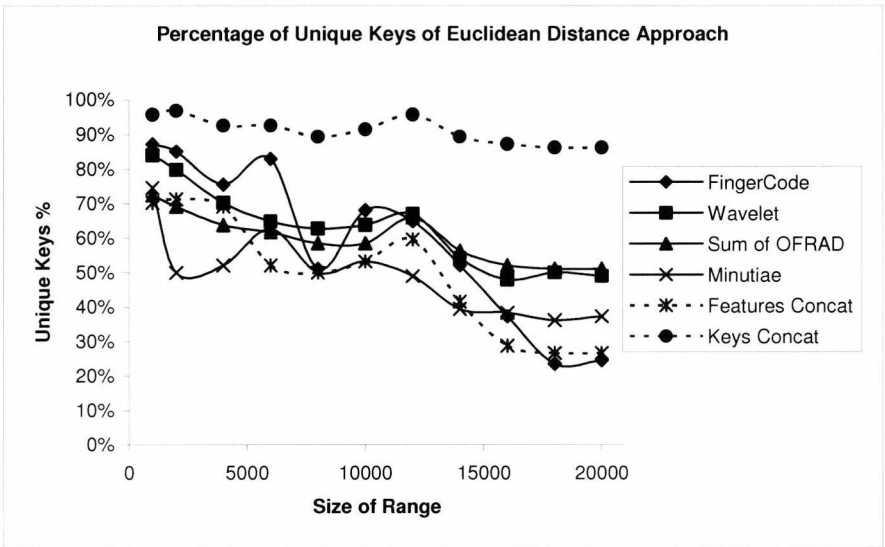


Figure 7-16: Unique Keys Percentage of Euclidean Distance Approach

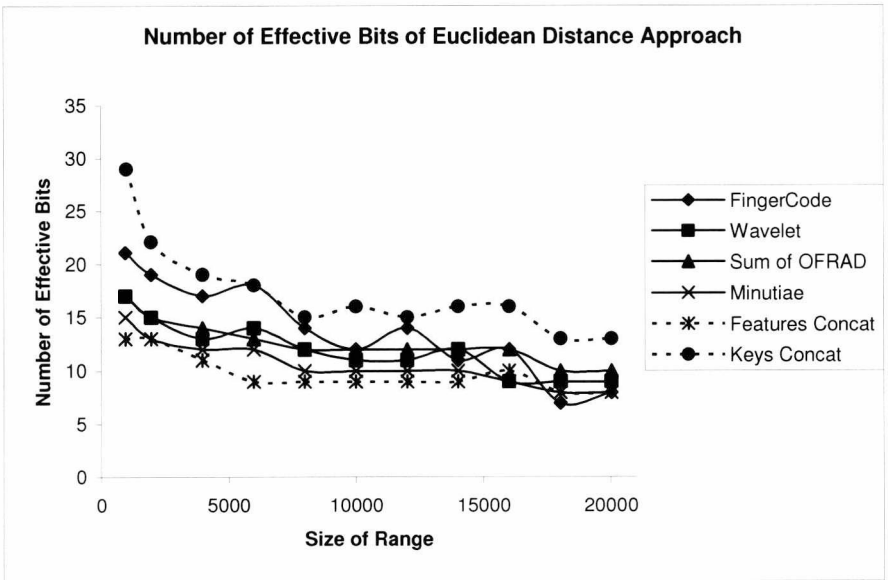


Figure 7-17: Number of Effective Bits of Euclidean Distance Approach

For these tests, feature vectors were combined slightly differently from the map-based methods. The *Features Concat* method is referred to the normalised feature vectors being concatenated together to create one large multi-dimensional feature, from which the Euclidean sums are calculated from that. The alternative approach is that the features are combined by obtaining component key-pieces from each individual feature vector and then concatenating the key-pieces to create the final key (*Keys Concat*).

Figure 7-15, which depicts correct reproduction of the key, shows in general, that as the range of values converted to Gray Codes increase, the match rate also increases. Although the increase is not monotonic, it is expected to increase further with the size of the range. However, the range size was restricted in these experiments because the execution time for the *Keys Concat* approach to combination was exceeding an acceptable time limit (i.e. 5s). This is because a range is converted to Gray Codes for each multi-dimensional feature vector independently, and if the range is large, the execution time is quadruple the time taken for a single feature. However, this can be overcome with concurrent programming. The range size was also restricted to prevent negative numbers (or overflow if unsigned data types are used) being included. Each feature was normalised to a range of [10000, 20000] so each feature has equal influence. If the range from which Gray Codes are determined exceeds 20000 (10000 either side of base number) then the range may extend past 0.

The results obtained from these series of tests are unremarkable and perform similarly to what was expected. The *Keys Concat* approach resulted in the lowest rate of reproduction, but created high entropy keys. Whereas when single base numbers were employed, the rate of reproduction increased as the entropy decreased. However, for the ranges investigated, considerable improvement was not witnessed. Furthermore, the tests which created the key from a single base number (i.e. individual features and *Feature Concat*) produced similar trends for the match rate,

which were not monotonic with local peaks at similar range sizes. It is presumed that the local peaks are located where the range includes values that are represented mostly by the same number of bits in binary. When the range consists of values that are made up of different number of bits, some bits are borderline stable (i.e. their presence is around the cut-off point, which in this case is 90%), which cause little peaks and troughs as the borderline bits fluctuate between stable and unstable. By discarding lower order bits, localising stable bits is omitted, so the trends are always monotonic. Another considerable benefit of discarding bits is that an improved rate of reproducibility can be attained without the need of converting vast ranges of values to Gray Codes, which reduces computational resources considerably. The results of discarding lower order bits are presented in Figure 7-18 and Figure 7-19.

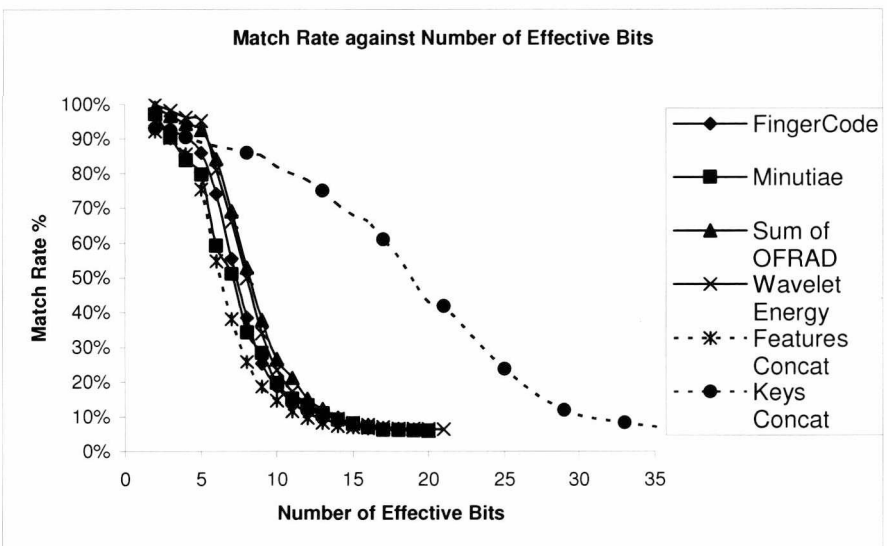


Figure 7-18: Match Rate against Number of Effective Bits for Euclidean Distance Approach

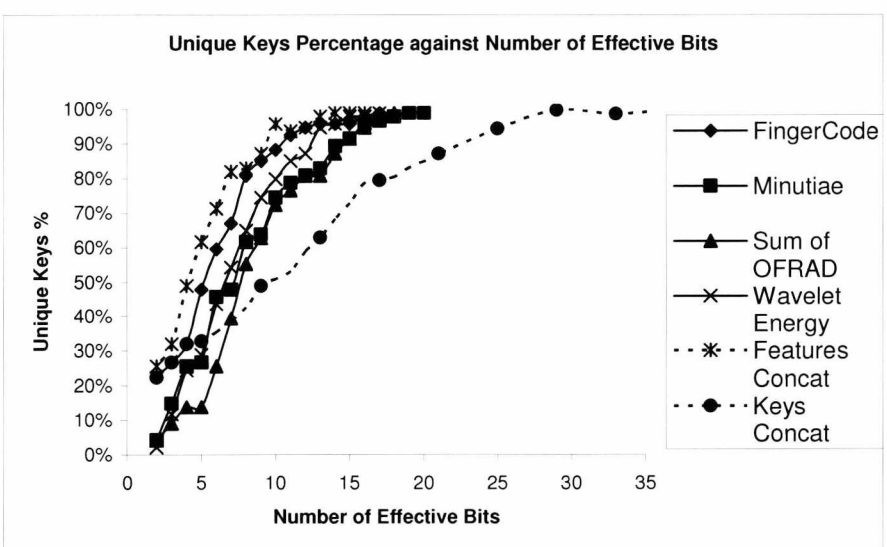


Figure 7-19: Unique Keys Percentage against Number of Effective Bits for Euclidean Distance Approach

From the results presented, the likeness between the trends of different feature vectors is immediately apparent for both the match rate (Figure 7-18) and the percentage of keys, which are unique (Figure 7-19). However, there is one exception. By combining the features together by concatenating their component key-pieces (*Keys Concat*), the performance is noticeably improved. Although the results do not suggest immediate implementation into a practical implementation, a relatively high rate (>80%) of correct key reproduction is achieved with the number of effective bits within double figures. Whereas the other feature vectors produce 80+% with around 4-7 effective bits, this approach to combining feature vectors exceeds the 80% mark with 11 effective bits. Furthermore, as anticipated, a greater number of unique keys (i.e. 50 unique keys out of 94) can be achieved using this method. Finally, it is important to note that the number of lower order bits, which was discarded in these experiments, was kept constant for all features. Although performance could possibly be improved by allowing different features to discard an optimal number of lower order bits, this would require the storage of this information. This gives potential attackers clues about the size of the feature space, which may be exploited or the attacked could alter the information so all bits, bar the most significant, will be discarded. This would essentially produce the same key for everyone. However, if a constant number of lower order bits to be discarded was to be used, this value can be hard-coded, which cannot easily be altered by an attacker.

7.3.3. Discussion

In this section, some simple approaches to key generation were investigated. Initially a pre-existing key generation method was adapted for implementation in a biometric system and an enhancement to this algorithm was proposed. This enhancement allowed multiple overlapping distributions to be mapped to the same sub-space (i.e. virtual mode), which significantly reduced erroneous mapping. Significant improvement was exhibited when this enhancement was incorporated and a key, with 21 effective bits, was reproduced 79% of the time. Considering the high intra-sample variance exhibited by most fingerprint features, this is quite some achievement and unexpected. Ideally the key should be reproduced 100% of the time, which was not attained. However, keys which are reproduced in excess of 90% can be achieved, but only with less than 11 effective bits and low uniqueness (i.e. ~40% unique keys). By using the improved mapping method, errors are infrequent, but large. This is due to the mapping process, whereby the distance between all virtual modes is not equal.

A further method, which generated a key from the multi-dimensional normalised feature space, was proposed. Unlike the map-based key construction, whereby the best results were obtained by converting a range of values to Gray Codes, for this method the optimal performance was achieved

by discarding lower order bits. However, it is envisaged that if the range was increased further, the reproducibility will continue to improve. In the map-based key generation method, the maps transform the values to a known sub-space, whereas in this approach the location and size of the region in multi-dimensional space is unknown. Therefore, it is possible that the range will need to be of considerable size in order to obtain a sufficient rate of key reproduction. By discarding lower order bits, high reproducibility can be achieved with significantly less processing. Using this method, a key, of 11 effective bits, with a rate of reproduction in excess of 80%, was attained. These results were obtained when the features were combined using the concatenation method.

Generally, the keys obtained using concatenation result in poor reproducibility. However, if the component features are highly reproducible, a stable key can be obtained using concatenation, which can significantly improve the key's entropy. Although a highly entropic key is desired, it is not necessary. For instance, many people have the same password or PIN. What maintains security is that the user is kept anonymous. This is why direct key generation methods permit helper data. The data stored is either global information, which when compromised releases no information regarding an individual, or the data belongs to an anonymous user. An attacker can use the information to generate a key, which belongs to someone else, but without knowing whom it belongs to is useless. As mentioned previously, there are duplicate passwords and PINs in circulation today but nobody knows who else has the same. Obviously, the percentage of unique keys must not be too low otherwise many people will have the same key within a small population. Therefore, analysing the percentage of keys which are unique is just as important as the reproducibility. However, there is a little more tolerance to the percentage of unique keys than there is to reproducibility.

The results obtained from these experiments suggest that the methods investigated are encouraging but as yet not suitable for implementation in a practical system. However, the results are promising and a number of positives can be taken from these tests. Firstly, only four features were used in the experiments. It is entirely plausible that implementing additional, suitable features can increase reproducibility while improving the uniqueness and increasing the number of effective bits of the key. New features and alternative representations of existing features, which offer improved stability, will be discovered as fingerprint research continues unabated. Secondly, the features of the fingerprint modality are regarded as one of the most variable of all biometric modalities. Obviously the key generation method did not perform exceptionally for fingerprint, but it is envisaged that a more stable biometric modality will result in significant improvements with regards to reproducibility. Thirdly, using a database restricts the number of quality samples. In a practical system, samples can be rejected and new ones acquired at the point of key generation. Acquiring high quality fingerprints will improve reproducibility as the features should be more stable. Furthermore, additional samples can be acquired for training and, if acceptable, normal operation. Finally, the tests conducted incorporated a significant number of unknown classes.

Training was preformed using 51 user groups, whereas the testing stage used an additional 44 users. It is encouraging that keys, although lacking in strength, can be reproduced for users that the system has never seen before.

At present, the methodologies and feature vectors employed in this proof of concept do not result in the desired performance metrics for the target application and implementation into such a system is currently not advisable. Expansion to the techniques and the addition of new feature vectors should improve performance. One such expansion should incorporate a decorrelation process, whereby at the training stage the features are decorrelated. This is one of the main attractions of employing maps as the feature values will be mapped to an alternative space, which is decorrelated. Many fingerprint features maybe highly correlated, which may cause determinable relationships between the feature space and the key-space. If an attacker knew that certain relationships existed, it is conceivable that an attacker could surreptitiously lift a fingerprint and exploit this knowledge to considerably reduce the search-space. However, as only four features were used, which were assumed to exhibit minimal correlation, a decorrelation process was deemed unnecessary. If additional features were implemented, it would be advisable to decorrelate the features because it is more probable that correlations will exist when many features are employed. Furthermore, as the purpose of these investigations was to serve as a proof of concept, feature decorrelation falls outside the scope of this research. Another expansion should be to incorporate multi-dimensional mapping. The main problem with the map-based methodology is that erroneous mapping does not produce errors of equal magnitude. This is because of the 1-dimensional mapped-space. By positioning virtual modes in multi-dimensional space, where the distance between virtual modes is equal, erroneous mapping will result in limited deviations along certain dimensions. The final recommended expansion should be to include multi-dimensional feature vectors as this is a severe limitation of the current map-based method.

7.4. Conclusion

The investigations performed within this chapter highlight two vastly different approaches to biometric encryption. Both methodologies have their advantages and disadvantages. The key release system provides a mechanism to release a key, whereby a query sample does not have to be entirely identical to the initial samples used to construct this mechanism. The key itself is completely independent from the biometric sample which means, provided the key generation process produces keys which are strong, that this approach fulfils the criteria of highly stable and strong keys. Furthermore, if the key was ever compromised it is very simple to obtain a new key without rendering the biometric useless. However, the main disadvantage is that the mechanism is stored either centrally or non-centrally. This means the user must always have access to the stored data through a token or a communications channel. However, direct key generation from biometric

samples does not require constant access to the stored data. Provided the key generation software is installed on a device, the key can be generated anywhere.

This is one of the most appealing factors of direct key generation. However, as witnessed throughout the experiments conducted, the consistent reproduction of strong keys is a very complex task. The performance required for a practical implementation was not achieved. It is vital that the direct key generation method performs well. This is because weak keys can easily be compromised and, as the key is generated from a biometric modality, the modality is rendered useless as the key is bound to it. Revocable biometrics may offer a viable solution to this problem, although this falls out of the scope of this research.

Direct key generation from biometric samples is a relatively new and immature field, which, in theory, should improve with the advancement of feature selection and extraction techniques. However, both the biometric encryption approach investigated can easily be adapted to work with multiple biometric modalities. Therefore, the release mechanism can be made stronger and direct key generation should improve performance considerably as the number of fingerprint features is currently limited. Although, the results of direct key generation suggest that a practical system maybe feasible, currently the most suitable approach lies with key management systems, which release the key.

Chapter 8. Conclusion

The purpose of the research carried out was to investigate and assess candidate fingerprint features for implementation in a template-free biometric encryption scheme. To fulfil this purpose, many aspects of a traditional image-based pattern recognition system was researched and implemented. Feature selection is an important component of pattern recognition systems and therefore, candidate features needed to be identified. Through thorough research, a subset of features was identified to be investigated further to obtain performance metrics. These performance metrics were subsequently used to assess the suitability of each feature for deployment in a template-free implementation. However, initial investigations resulted in poor performance characteristics. Therefore, further research was conducted, not just into alternative fingerprint features but also, into improving the performance of pre-existing features by means of pre-processing.

The fingerprint modality is regarded as one of the most variable. Factors such as inconsistent rotation, translation, pressure and region acquisition, coupled with dirty prints or lens, cause huge deviations from the norm of a fingerprint. In an effort to minimise the variations attributed to a combination of these factors, pre-processing and alternative ROI schemes were researched. Pre-processing methods such as segmentation/quality maps, fingerprint rotation, normalisation and enhancement were all implemented to create a range of input images from which the features can be extracted from. In addition to the varying pre-processed images, numerous ROI schemes were used to determine which one consistently extracted the same relative region. From the many combinations of pre-processing and ROI extraction methodologies, the features were extracted from the fingerprints contained within two databases. In order to assess the suitability of each feature independently, performance metrics, obtained from statistical tests, were employed. These statistical tests were the F-Test, for uni-dimensional features, and Wilks' lambda, in combination with the Mahalanobis distances, were used for multi-dimensional features.

For feature selection, no strict rule was applied. Simply the best performing variant of each feature was compared and these optimal features were ranked accordingly. A subset of these optimal feature vectors were then subsequently employed in various template-free biometric encryption schemes in order to determine whether a practical implementation is viable.

Biometric encryption is an ambiguous title because no encryption is actually performed. It is an umbrella term referring to the mechanisms of key creation and management. Template-free biometric encryption is simply a method of generating keys directly from biometric modality's feature data, without storing private information of an individual. Template-free does not necessarily mean the abolition of storing any data. In the methods explored, helper data, which does not contain individual's biometric or private information, is stored to help generate the encryption key. As with all key generation methodologies, the performance of such a scheme was

measured by the entropy and stability of the generated keys. If both these performance metrics are high, then the scheme is judged suitable for product development.

8.1. Findings

The vast proportion of the research conducted was concentrated on the fingerprint modality and the experiments carried out to achieve the optimal performance of each feature. An attempt to improve pre-existing methods, in association with developing alternative features, was conducted because of the limited number of fingerprint features available. Although fingerprints are one of the most mature biometric modalities with a plethora of research, the majority of this research is centred on minutiae. A significant reason why there is an abundance of minutiae-based fingerprint systems available is because there isn't that much to a fingerprint. A fingerprint mostly comprises ridges and furrows, which flow in patterns dictated by singularities. Occasionally these ridges end and bifurcate to create minutiae.

Other artefacts, such as creases and scars can be located but these are often overlooked because not every fingerprint possesses them. Poroscopy is a relatively new approach to fingerprint systems, whereby the pores are extracted. However, poroscopy requires high definition acquisitions and these scanners are not, as yet, universally implemented. Besides these features, the fingerprint is often viewed as an oriented, periodic, textured pattern, so pattern descriptors, which are heavily influenced by frequency and orientation, are also popular. Other than the outlined features, alternatives are few and far between. In many cases, the feature is simply an alternative representation of a pre-existing feature, which offers slightly improved performance. The feature catalogue compiled during this research is an example of implementing a pre-existing feature where alterations have been made in an effort to squeeze every drop of performance from it.

These alterations have been implemented by changing the amount of pre-processing performed on the input image and the ROI schemes employed. These adaptations also involved examining the extent of performance deterioration as a result of employing current automated methods of reference point location and orientation estimation. Other notable modifications implemented were the directional sum of the orientation-based feature vectors and the sum of the inter-minutiae distances. All variations were designed in order to determine reproducibility and robustness to differences in rotation and translation of each feature. A highly reproducible feature vector is essential if the subsequently generated encryption key is to possess high stability. In this research, key stability takes slight precedence over key entropy as it presumed that future implementation of additional features will improve key strength.

8.1.1. Fingerprint Features

From the experiments conducted, general consensus suggests superior performance can be obtained by rotating the image and enhancing the ROI prior to feature extraction for image-based feature vectors (i.e. FingerCode, wavelets, etc). However, this was provided that both the reference point location and fingerprint orientation was determined manually. If automated methods were to be employed, which in a real-world scenario is essential, the results suggest that the image orientation should be left as it is. This is because the algorithms employed in the experiments can exacerbate the error in problem prints, such as the arch and whorl-type. If the reference point position cannot be consistently acquired, simply enhancing the ROI produces improved results than rotating without enhancement. However, as the results for a manually estimated reference point and fingerprint orientation outperforms automated estimation, it suggests that if consistent automated methods were developed, the performance of the features will improve.

Currently, fingerprint orientation is severely underrepresented in the literature as alignment normally occurs at a matching stage. As "template-free" suggests no matching takes place. Therefore rotation robustness needs to be managed at the feature extraction stage by using invariant features, or suppressing the rotation by orientating the fingerprint to a base orientation at the pre-processing stage. It may seem that by rotating the image and enhancing the ROI improves feature vector stability is stating the obvious. However, the extent of this improvement has, until now, been investigated. It is accepted that many new and improved methods of reference point estimation are available, although the algorithm implemented was deemed the most popular at the time researched commenced. Even with a better reference point estimation algorithm, the most stable features will be those that are invariant and extracted from a single region about the reference point. As soon as sub-ROIs (i.e. small ROIs positioned at different locations within the image) are employed, it does not matter if the feature is invariant if the region it is being extracted from differs each time. This was witnessed with the implementation of invariant moments. Therefore, it is vital for ROI schemes, which employ sub-ROIs to, that the rotation variance is handled before extraction by means of orientating the fingerprint.

It was also determined from the experiments that the ROI scheme employed influences the performance of the feature vector. Although all ROI schemes perform best when the fingerprint is rotated and a consistent reference point has been located, it was found that non-grid based ROI schemes were superior. Single, large circular ROIs performed best when invariant features, such as invariant moments, wavelets (to some extent) and the inter-minutiae sum, were implemented. However, for features that are extracted from a local neighbourhood, the DISK or RAD ROI schemes extracted the same relative region with improved consistency, resulting in features of higher effective dimensionality. As a result of numerous experiments, with many different permutations, 4 out of 6 feature categories investigated were judged suitable for application in a biometric encryption system.

Invariant moments and frequency based feature vectors were rejected due to a very high rate of overlapping within the dimension-space. This was initially the case with orientation-based feature vectors until the directional sum was implemented, which expanded the feature-space, reducing the amount of overlapping distributions. In a real-world implementation, the number of users should be unlimited, which means that the dimension-space and, consequently, the feature-space can become over-populated. This will lead to many overlapping distributions where minimal discriminatory power can be obtained.

In features where the dimension-space is limited such as orientations where there is high variance within a small range (i.e. π), an alternative representation maybe required to reduce population density. By calculating the directional sum of the orientation-based feature vectors, the performance was improved but, importantly, removed the cyclic-nature of the feature values. This facilitated smoother implementation of the feature vector into a biometric encryption scheme. In addition to directional sum of OFRAD, the FingerCode also produced encouraging results, whereby the average Mahalanobis distance between means of different classes was significantly greater than the average intra-sample Mahalanobis distance. This suggests good group separation. However, the FingerCode's dimensionality and fingerprint coverage needed to be reduced as too many fingerprints were rejected as the ROI exceeded the boundaries of the print, causing the Wilks' Lambda statistical test to fail. This was because the dimensionality was significantly greater than the number of samples. It was found that reducing the DISK ROI scheme to 3 tracks, of width 20 pixels, instead of 5 was superior than reducing the track-width and/or the number of sectors.

Prior to the fingerprint feature investigation, it was already decided that some minutiae-based descriptor was to be implemented in the biometric encryption scheme in some form. This was because minutiae are one of the most popular fingerprint features, offering a high level of individuality. Unfortunately, without the aid of matching, it was inevitable that uniqueness would suffer. In an attempt to incorporate minutiae, only the inter-minutiae distances were used because different subsets of minutiae are detected in different acquisitions. A benefit of this approach is that the feature value is rotationally invariant, although the results do reflect the deterioration in individuality. However, stability was improved by taking the mean value from numerous samples. Regrettably, for a practical system, user-acceptability restricts the number of samples from which to obtain the mean. Although performance exhibited improvement as the number of samples increased, general consensus from a small survey suggested users would be willing to provide up to 3 samples. Therefore, the proof-of-concept, biometric encryption scheme was limited to 3 samples.

The final feature that was deemed suitable was a feature based on the energy ratio of Daubechies-12 wavelet. Many different scales of a variety of wavelets were investigated. From the features

obtained from each detail, it was determined that the energy ratio was the best. Furthermore, it was discovered that the statistical tests peaked when 3 scales was used.

8.1.2. Biometric Encryption

Although the statistical tests employed provided an estimation of the discriminatory information possessed by each feature, the "proof is in the pudding". The test values, when compared between feature vectors, may seem good, but how well do these features translate to a practical system? There are two schools of thought when it comes to biometric encryption, key release and key generation.

A rudimentary key release mechanism was initially investigated whereby a fuzzy vault was constructed and unlocked using the OFRAD feature vector. By employing the key release methodology, the key generation is unrelated to the biometric feature. Therefore, the entropy of the key is the responsibility of the method of key generation employed. Whereas the unlocking of the vault (i.e. release of the key) influences the reproduction of the key, which becomes a factor of the reproducibility of the feature vector that unlocks the vault. As the vault can be unlocked with an inexact feature vector, the key reproduction rate improves, although the probability of multiple users unlocking the vault increases. The experiments conducted showed that a genuine acceptance rate greater than 90%, with a false acceptance rate in the region of 20%, can be achieved provided the reference point and the fingerprint orientation is estimated manually. With the automated methods implemented in the experiments, reasonable performance can still be achieved provided no fingerprint rotation is applied. When the fingerprint is rotated to a base orientation using an automated method, the performance was found to deteriorate severely. However, even when the reference point is located algorithmically, without rotation, only a slight deterioration was exhibited.

The main point of consternation regarding the fuzzy vault method proposed and investigated is that each dimension of the feature vector is used to extract a key share independently. This presents a problem similar to that of concatenating individual dimensions to construct a key. If one of the dimensions is completely wrong, it does not matter whether all the others are correct, the released key will be wrong. By using multiple samples at the vault construction stage, the variability of each dimension is used to encode the vault. It was found that this problem occurred with reduced frequency as the number of samples used to construct the vault increased. Although this approach to biometric encryption easily meets the desired requirements of a high entropy, high stability key, there are concerns about its resistance to attack and the false acceptance rate. The false acceptance rate can be reduced by incorporating multiple features, such as minutiae, a vault can be constructed similarly for most features.

The key release system offers many benefits although this methodology is essentially a key management system whereby each user will have their own, personalised mechanism stored somewhere. Therefore in order to release the key, the user must be able to access the mechanism, which suggests each user carries a smartcard (i.e. token) or a communication channel to a centralised repository is available. Although, for most applications, this is acceptable, for some applications this is not appropriate. By generating a key directly from the biometric samples, the user does not need to access anything, except a device with a method of biometric acquisition and the key generation software installed.

A novel approach and an improvement to a pre-existing method to direct key generation were proposed. The pre-existing method was an adapted version of the ICMetric approach [217, 218], which used debug features from embedded systems to generate keys. However, this approach did not perform well on fingerprint features due to the high degree of overlapping exhibited between user distributions. An enhancement was proposed whereby overlapping distributions would be mapped to the same sub-space. This was found to improve the reproducibility of the key, although at a cost to key uniqueness. The ICMetric approach, and its subsequent improvement, both employ feature-maps which transform feature values to mapped values, from which an encryption key can be constructed. It is the intention of all key generation methods investigated that further features can be added and previously unseen users can generate stable encryption keys. Therefore, in a multi-feature application, a suitable approach to combining the features to generate a single key is required. Two methods were implemented and compared, where one approach simply summed the mapped values of each feature together to create a single value to create a single value, and the other concatenated together keys generated from individual features. Alternatively, as the feature vectors are normalized, they can all be summed together to create a single value for all features, so only one map is required.

From the mapped values, a key can be generated in one of two ways. The first approach converted a range of values centred on the mapped value to Gray Codes, whereby the location and value of the stable bits were used to construct the key. The alternative approach was simply to drop lower order bits from the Gray Code representation of the mapped value. The range conversion method was found to produce keys with a greater number of effective bits, although in order to obtain an acceptable reproduction rate, the range may need to be very large. This method of key generation was judged to be more secure because only the size of the range is required, whereas the number of bits to discard is required by the alternative technique. If an attacker was to know the number of bits to drop, it will reduce the sub-space significantly. However, the range conversion method will only perform well in this scenario if the size of the range was huge. When this occurs, the number of iterations required converting so many values to Gray Codes, and the subsequent comparisons, would be time and memory consuming. Coupled with the time required to extract the features from the biometric samples and the mapping, the overall elapsed time may not be acceptable.

It was discovered from the experiments conducted that the concatenation method generally resulted in inferior reproduction performance as one incorrect feature would prevent correct key reproduction. This was also evident when the mapped values were summed together for the same reason, although larger ranges will improve this approach significantly. As more features are added, the probability of a feature being incorrect increases. If the purpose of the system was to obtain optimal key reproduction, without considering entropy, the best performing combined method was the summation of the normalized feature vectors prior to mapping. This reduces the error due to erroneous mapping because only a single mapping process is performed, rather than a mapping process per feature. By employing this method of combining the feature vectors, the most optimal setup reproduced a key, with 11 effective bits, nearly 90% of the time, with 37 unique keys out of a possible 94. However, if each component feature vector produces highly reproducible keys individually, then this significantly increases the reproducibility of the post-mapping combination approaches. As these approaches typically offer improved entropy, a suitable balance between stability and entropy can be determined. For instance, by concatenating the keys of the component features together, with an overlapping factor of 0.5, a key, with 21 effective bits, was reproduced almost 80% of the time, with 52 unique keys. This is a significant achievement considering the high intra-sample variance of most fingerprint features investigated.

The results obtained suggest that this map-based key generator, under its current evolution, is inappropriate for a practical implementation using the fingerprint biometric modality. A significant hindrance of the map-based approach investigated is that the features are mapped independently to a linear scale. The errors incurred by erroneous mapping are significant. Therefore, a method that generates keys from multi-dimensional space was proposed.

There is a considerable limitation of the current map-based approaches, which is that they are unable to handle multi-dimensional features. In order for multi-dimensional features to be permitted, they have to undergo dimensional reduction, which may significantly increase overlapping between distributions. Therefore, an alternative approach to mapping was presented which endeavoured to create stable keys from the multi-dimensional feature-space, by using the mean vectors of each user who submitted samples for calibration. During normal operation, the means serve as points of reference in multi-dimensional space, whereby the position of a sample is determined from the positions of the means. Ideally, the key should be based on the direction as well as the distance from each mean, although for a simple proof of concept the distances were decided to be sufficient. Using the sum of the Euclidean distances between a sample and each mean vector, a basis number is generated from which a key can be derived using the range conversion method or by discarding lower order bits.

Similarly to the map-based methods, the performance of the system using the range conversion method of isolating stable bits to construct the key resulted in poorer key reproducibility. However, the size of the range had to be restricted as the time taken, to generate a single key for the key concatenation approach, exceeded 5 seconds on account of the number of iterations required per feature. This execution time would not be acceptable in a practical system. Nonetheless, by discarding the lower order bits, a reproduction rate of >80% was obtained with a key of 11 effective bits. To achieve this performance, a key for each feature vector, with the lower order bits discarded, were concatenated together to create a longer key. This method of combining feature vectors consistently produced keys with more effective bits, and a higher percentage of unique keys, than the alternative method of concatenating the normalised feature vectors into one large multi-dimensional vector. Unfortunately, this comes at a cost to the key reproducibility because, when one component of the key is incorrect, the full key cannot be reproduced.

It is clear from the results obtained that the performance of the direct key generation methods investigated are far from acceptable for use in a practical system. The reproduction rate is far too low and so is the number of effective bits, although this could be somewhat attributed to the inclusion of a high degree of unseen users. At this stage, an attacker could easily crack the encryption key, although the key can be padded in an effort to confuse the attacker. However, the results are encouraging and it is envisaged that, with more suitable features, this performance can be significantly improved. In addition, the key reproduction rate can be further improved if the range, or number of bits to discard, was feature specific rather than constant. This ensures that each feature performs optimally.

8.2. Contributions

The primary objective of the research presented was to investigate candidate fingerprint features for implementation in a template-free biometric encryption scheme. The secondary objective was to produce a proof of concept prototype employing the fingerprint feature investigated. The thesis attempts to integrate a multitude of algorithms, ranging from pre-processing to feature extraction and from feature representations to biometric key generation, into a working biometric encryption scheme. Since fingerprint features are highly variable, the identification of suitably stable features, and the subsequent generation of a stable key from variable data, presented many challenges. In order to address these challenges, many methods from different fields were investigated. As a result, numerous contributions have been made. The following summarises the contributions of this thesis.

- The main contribution of this thesis was the exhaustive, in-depth analysis of fingerprint features from recognised benchmarking databases and their subsequent compilation into a

feature catalogue for feature selection. By performing this task, further contributions were made. These include:

- Assessing performance based on statistical tests of individual features rather than the performance of a complete system. By using the performance metrics of a complete system, the rates obtained include the matching stage, which provides a mechanism for handling variation. These system performance rates provide scant detail about the suitability of the feature vector when matching cannot be performed, as in the desired implementation. The statistical tests of the features presented in this thesis can be used by the research community for feature selection without the need of implementing them into a system first.
- The investigation into the deterioration exhibited as a result of using automatic reference point detection and fingerprint rotation. Previously, little research had been conducted that attempts to find the extent to which the performance drops when automatic estimation is used.
- The investigation into improving existing features by implementing alternative ROI schemes and the rejection of samples based on a rudimentary screening algorithm. This included the novel ROI schemes RAD and GRIDC (see Chapter 3).
- A novel representation of orientation-based feature vectors by using the directional sum. In addition to improved statistical tests, removing the cyclic values permits simpler integration in a multi-feature system as the feature can be manipulated without consideration for the wraparound values.
- The investigation of using Zernike moments as an alternative to Hu moments.
- A novel representation of minutiae by summing the inter-minutiae distances, within a radius about the reference point, and the subsequent normalisation of the total distance by the average inter-ridge distance. This representation attempts to reduce the impact of rotation and scale variance.
- A novel method of detecting core singularities by tracing the fingerprint orthogonally to the ridge direction and the subsequent performance metric of consistent placement. This metric was size of the ellipse which encapsulated all the reference point estimations based on polar coordinates using the ground truth reference point as the origin and the ground truth fingerprint orientation as the normal.

- An improved map-based key generation method, whereby an existing technique was investigated and enhanced with the introduction of mapping overlapping distributions in the feature space to a common sub-space in the mapped domain.
- A novel approach to key generation using the sum of the Euclidean distances between a sample and the means of the training sets.

- A novel key release mechanism (i.e. fuzzy vault) using an orientation-based feature vector. This approach can be adapted for other features and is suitable for multiple features.
- 2 novel, yet simple, fingerprint segmentation algorithms.
- A novel approach to peak/trough detection, whereby a simple model based on rudimentary Newtonian mechanics was employed to find local maximums and minimums similarly to line-searching algorithms used in neural network.

8.3. Future Work

Although fingerprints are probably the most mature biometric modality, the results from the experiments carried out suggest that the fingerprint features examined are not ideal for a template-free implementation. Therefore, continual development of new features, and improvement of existing features, is a necessity if strong and stable keys are to be generated directly from the feature data. Furthermore, as highlighted by the findings presented, the performance of some of the existing features exhibits significant deterioration when automatic reference point location and fingerprint orientation is used. Reference point detection is currently a popular research topic and consistency is claimed to be improving, although arch-type fingerprints are still troublesome. However, research into fingerprint orientation is scarce. As the results indicate, significant improvement was observed when the orientation was estimated manually. If further improvements to fingerprint orientation estimation algorithms can be made so that the estimation was as good as, or better than, human estimation, rotational variance should be sufficiently suppressed. This will in turn improve the performance of existing features that are not invariant to rotation.

Suitable features are imperative for generating strong and stable encryption keys. However, the poor reproduction rate and entropy of the generated keys cannot be attributed to just the features. The key generation methods employed are not appropriate for fingerprints as the feature distributions overlap considerably. As direct key generation from feature data is not currently well-researched, there is scope for significant improvement. For instance, the key generation methods investigated lack feature decorrelation. Some features maybe highly correlated that potentially can be exploited to reduce the search-space from which an attacker could lead to key compromisation. A potential approach to decorrelation is to generate maps that transform feature values to normalised, decorrelated values. Although a map-based method was investigated, it is not recommended to adapt this approach for decorrelation as it will not improve the reproduction rate and key entropy. An alternative mapping methodology would need to be developed. This is because the current mapping approach does not map deviations in multiple dimensions as each dimension is treated independently. A single erroneous mapping of one dimension prevents key reproduction. It is suggested that a multi-dimensional mapping function, or a suitable approach to dimension reduction, would be more appropriate. This is so deviations in individual dimensions do not severely change the base number from which a key can be generated.

Implementing the suggested improvements outlined should result in a superior biometric encryption scheme. This could be further improved by employing multiple biometric modalities as the method of key generation should be completely independent of the feature data used. However, generating a key directly from biometric data binds the key to the biometric. If the key was compromised, the biometric modality it was generated from is rendered useless. If the system was based solely on fingerprints (or something similar where the number of possible samples is restricted), then only a maximum of 10 different keys can be generated. If all of these keys were compromised then there would be no way of creating a new key. Therefore, combination of multiple modalities can be used to generate a new key. At which point a challenge, which will need to be addressed arises. If, for example, 3 out of 4 modalities used produce values of high confidence that these samples belong to a certain user, but the final modality is ambiguous, what does the system do? Ask for a new sample, which may irritate the user if this occurs frequently, or implement a mechanism which permits this flexibility? However, if the intended system is limited to a single modality, such as fingerprints, it would be ideal if the key was cancellable as the number of possible keys is limited. Hence, further research can be performed into permitting template-free biometric encryption to cancel and issue new keys with the same biometric data. This field of research is called revocable biometrics.

Finally, as the direct key generation methods presented in this thesis were used only as a proof of concept, further improvement and testing is required. Once an acceptable key strength and reproduction rate has been attained, attack analysis should be performed and an appropriate policy for managing outliers is recommended.

8.4. Closing Remarks

An in-depth investigation into fingerprint features, and their implementation into a template-free biometric encryption scheme, has been presented in this thesis. Fingerprint features are highly variable, which has produced many challenges. Different methods, which attempt to address these challenges, have been analysed and it has been determined that a key release mechanism is a more feasible approach for template-free biometric encryption at this time. Direct key generation is still in its infancy with many outstanding problems. However, even though the performance obtained from the approaches investigated was poor, it is anticipated that, with further research and development, a user-acceptable and practical system can be achieved.

References

[1] Sharat Chikkerur, *et al.*, "A Systematic Approach For Feature Extraction In Fingerprint Images," *Lecture Notes in Computer Science,* vol. 3072/2004, pp. 344-350, 2004.

[2] H. Lin, *et al.*, "Fingerprint image enhancement: algorithm and performance evaluation," *Pattern Analysis and Machine Intelligence, IEEE Transactions on,* vol. 20, pp. 777-789, 1998.

[3] N. Otsu, "A Threshold Selection Method From Gray-Level Histograms," *IEEE Transactions On Sys, Man & Cyb,* vol. 9, pp. 62-66, 1979 1979.

[4] T. R. Weiss, "Weak Passwords Really Do Help Hackers," *http://www.computerworld.com/s/article/9010540/Study_Weak_passwords_really_i_do_i_help_hackers?taxonomyName=security*, 06/02/2007 2007.

[5] R. Lemos, "Unpatched PCs Compromised in 20 Minutes," *http://news.zdnet.com/2100-1009_22-137900.html*, 17/08/2004 2004.

[6] D. Maltoni, *et al.*, *Handbook of Fingerprint Recognition*: Springer Verlag, 2003.

[7] CNN, "FBI wants palm prints, eye scans and tattoo mapping," *CNN.com/technology,* 04/02/2008 2008.

[8] A. Jain, *et al.*, "Pores and Ridges: Fingerprint Matching Using Level 3 Features," *International Conference on Pattern Recognition, ICPR2006,* 2006.

[9] P. Wasserman, "Level 3 Feature Detection Using Support Vector Machines," *http://www.itl.nist.gov/iad/894.03/latent/workshop/proc/P20_Wassm_DetectingLevel3Features.pdf*.

[10] J. Xudong and Y. Wei-Yun, "Fingerprint minutiae matching based on the local and global structures," 2000, pp. 1038-1041 vol.2.

[11] M. Faundez-Zanuy, "On the vulnerability of biometric security systems," *Aerospace and Electronic Systems Magazine, IEEE,* vol. 19, pp. 3-8, 2004.

[12] R. Wanderhoof, "Smart cards, biometrics and privacy," *Card Technology Today,* 2003.

[13] C. Soutar, *et al.*, "Biometric Encryption," *ICSA Guide to Cryptography,* 1999.

[14] A. Juels and M. Sudan, "A Fuzzy Vault Scheme," *Design, Codes and Cryptography,* vol. 38, pp. 237 - 257, 2005.

[15] Y. Shenglin and I. Verbauwhede, "Automatic secure fingerprint verification system based on fuzzy vault scheme," 2005, pp. v/609-v/612 Vol. 5.

[16] U. Uludag, *et al.*, "Fuzzy Vault for Fingerprints," *Lecture Notes in Computer Science,* vol. 3546, pp. 310 - 319, 2005.

[17] F. Monrose, *et al.*, "Cryptographic Key Generation From Voice," *IEEE Symposium on Security and Privacy,* May 2001 2001.

[18] F. Monrose, *et al.*, "Password Hardening Based on Keystroke Dynamics," *ACM Conference on Computer and Communications Security,* 1999.

[19] T. C. Clancy, *et al.*, "Secure Smartcard-Based Fingerprint Authentication," *Proc. ACM SIGMM 2003 Multim., Biom. Met. & App,* pp. 45 - 52, 2003.

[20] Y. Chang, *et al.*, "Biometric-based Cryptographic Key Generation," *Proc. IEEE Conf. Multimedia and Expo,* pp. 2203 - 2206, 2004.

[21] M. T. John, "Region-of-interest detection for fingerprint classification," 1994, pp. 48-59.

[22] R. Krishnan, "Fingerprint Capture Challenges and Opportunities," *Presentation for US Department for Homeland Security,* 2006.

[23] N. K. Ratha and R. M. Bolle, "Effect of Controlled Acquisition on Fingerprint Matching," *Proceedings of the International Conference on Pattern Recogntion,* vol. 2, pp. 1659 - 1661, 1998.

[24] A. K. Jain, *et al.*, "Filterbank-based fingerprint matching," *Image Processing, IEEE Transactions on,* vol. 9, pp. 846-859, 2000.

[25] L. G. Shapiro and G. C. Stockman, *Computer Vision*: Prentice Hall, 2001.

[26] E.-K. Yun and S.-B. Cho, "Adaptive fingerprint image enhancement with fingerprint image quality analysis," *Image and Vision Computing,* vol. 24, pp. 101-110, 2006.

[27] X. Wang, *et al.*, "Adpative Fingerprint Enhancement by Combination of Quality Factor and Quantitative Filters," *Lecture Notes in Computer Science,* vol. 3781, pp. 111 - 118, 2005.

[28] D. Zhao, *et al.*, "A Hierarchical Fingerprint Matching Method Based on Rotation Invariant Features," in *Lecture Notes in Computer Science*, ed, 2004, pp. 498-505.

[29] M. Horton, *et al.*, "The costs and benefits of using complex 2-D Gabor filters in a filter-based fingerprint-matching system," 2002, pp. 171-175.

[30] F. Alonso-Fernandez, *et al.*, "An enhanced Gabor filter-based segmentation algorithm for fingerprint recognition systems," in *Image and Signal Processing and Analysis, 2005. ISPA 2005*, ed, 2005, pp. 239-244.

[31] V. Areekul, *et al.*, "Fast Separable Gabor Filter For Fingerprint Enhancement," *Lecture Notes in Computer Science,* vol. 3072/2004, pp. 403-409, 2004.

[32] W. Jang, *et al.*, *Fingerprint Image Enhancement Based on a Half Gabor Filter*, 2005.

[33] j. Xudong, "A study of fingerprint image filtering," 2001, pp. 238-241 vol.3.

[34] E. Zhu, *et al.*, *Fingerprint Enhancement Using Circular Gabor Filter*, 2004.

[35] M. A. Khan and M. K. Khan, "Improved fingerprint identification using directional filter banks," ed, 2003, pp. 49-54.

[36] S. K. Oh, *et al.*, "New Fingerprint Image Enhancement using Directional Filter Bank," *Journal of WSCG,* 2003.

[37] T. R. Randolph and M. J. T. Smith, "Fingerprint image enhancement using a binary angular representation," 2001, pp. 1561-1564 vol.3.

[38] W. Chaohong, *et al.*, "Fingerprint image enhancement method using directional median filter," 2004, pp. 66-75.

[39] S. Greenberg and D. Kogan, "Structure-adaptive anisotropic filter applied to fingerprints " *Optical Engineering* Dec 2005 2005

[40] S. Greenberg, *et al.*, "Fingerprint image enhancement using filtering techniques," 2000, pp. 322-325 vol.3.

[41] J. Cheng, *et al.*, *Fingerprint Enhancement Using Oriented Diffusion Filter*, 2003.

[42] B. G. Sherlock, *et al.*, "Fingerprint enhancement by directional Fourier filtering," *Vision, Image and Signal Processing, IEE Proceedings-*, vol. 141, pp. 87-94, 1994.

[43] T. Kamei and M. Mizoguchi, "Image filter design for fingerprint enhancement," 1995, pp. 109-114.

[44] G. Zhu and C. Zhang, *A Top-Down Fingerprint Image Enhancement Method Based on Fourier Analysis*, 2004.

[45] T. Ko, "Fingerprint enhancement by spectral analysis techniques," 2002, pp. 133-139.

[46] K. Najarian and R. Splinter, *Biomedical Signal and Image Processing*: CRC Press, 2005.

[47] C.-T. Hsieh, *et al.*, "An effective algorithm for fingerprint image enhancement based on wavelet transform," *Pattern Recognition,* vol. 36, pp. 303-312, 2003.

[48] Z. Wei-Peng, *et al.*, "A wavelet-based method for fingerprint image enhancement," 2002, pp. 1973-1977 vol.4.

[49] S. Y. Lee, *et al.*, "Recent Fingerprint Image Enhancement Using Wavelet Transforms," *2002 Applications of Computer Algebra Conference (ACA 2002),* 2002.

[50] K. Larsson, "Scale-Space Methods as a Means of Fingerprint Image Enhancement," 2004.

[51] A. Almansa and T. Lindeberg, "Fingerprint Enhancement by Shape Adaptation of Scale-Space Operators with Automatic Scale Selection," *IEEE Transaction in Image Processing,* vol. 9, pp. 2027 - 2042, 2000.

[52] J. Cheng and J. Tian, "Fingerprint Enhancement with Dyadic Scale-Space," *Pattern Recognition Letters,* vol. 25, pp. 1273 - 1284, 2004.

[53] V. Espinosa-Duro, "Fingerprints thinning algorithm," *Aerospace and Electronic Systems Magazine, IEEE,* vol. 18, pp. 28-30, 2003.

[54] S. Greenberg, *et al.*, "Fingerprint Image Enhancement using Filtering Techniques," *Real-Time Imaging,* vol. 8, pp. 227-236, 2002.

[55] X. Luo and J. Tian, "Knowledge based Fingerprint Image Enhancement," *Proceedings of 15th International Conference on Pattern Recognition,* vol. 4, pp. 783 - 786, 2000.

[56] N. K. Ratha, *et al.*, "Adaptive flow orientation-based feature extraction in fingerprint images," *Pattern Recognition,* vol. 28, pp. 1657-1672, 1995.

[57] N. Ikeda, *et al.*, "Fingerprint image enhancement by pixel-parallel processing," 2002, pp. 752-755 vol.3.

[58] X. Liang and T. Asano, "A Linear Time Algorithm for Binary Fingerprint Image Denoising Using Distance Transform," *IEICE Transactions on Informations and Systems,* vol. E89-D, pp. 1534-1542, April 2006 2006.

[59] Neurotechnologija. (2006, *VeriFinger (www.neurotechnologija.com/verifinger.html).*

[60] Q. Huang, *et al.*, "Thresholding Technique with Adaptive Window Selection for Uneven Lighting Image," *Pattern Recognition Letters,* vol. 26, pp. 801 - 808, 2005.

[61] J. Sauvola and M. Pietikainen, "Adaptive Document Image Binarization," *Pattern Recognition,* vol. 33, pp. 225 - 236, 2000.

[62] I. Emiroglu and M. B. Akhan, "Pre-processing of fingerprint images," 1997, pp. 147-151.

[63] W. Bieniecki and S. Grabowski, "Multi-pass Approach to Adaptive Thresholding based Image Segmentation," *Proceedings of CADSM'2005,* 2005.

[64] G. Leedham, *et al.*, "Separating Text and Background in Degraded Document Images," *Proceedings of the 8th International Workshop on Frontiers in Handwriting Recognition,* pp. 244 - 249, 2002.

[65] M. Sezgin and B. Sankur, "Survey over Image Thresholding Techniques and Quantitative Performance Evaluation," *Journal of Electronic Imaging,* vol. 13, pp. 146 - 165, 2004.

[66] A. M. Bazen and S. H. Gerez, "Segmentation Of Fingerprint Images," *Workshop On Circuits, Systems, and Signal Processing - ProRISC 2001,* 2001.

[67] B. M. Mehtre, *et al.*, "Segmentation of fingerprint images using the directional image," in *Pattern Recognition* vol. 20, ed, 1987, pp. 429-435.

[68] B. M. Mehtre and B. Chatterjee, "Segmentation of fingerprint images -- A composite method," in *Pattern Recognition* vol. 22, ed, 1989, pp. 381-385.

[69] C. I. Watson, *et al.*, "User's Guide to NIST Biometric Image Software," *National Insititute Of Standards And Technology.*

[70] S. Klein, *et al.*, "Fingerprint Segmentation Based On Hidden Markov Models," *13th Annual Workshop on Circuits, Systems and Signal Processing ProRISC 2002,* pp. 310 - 318, 2002.

[71] X. Chen, "Segmentation Of Fingerprint Images Using Linear Classifier," *Journal Of Applied Signal Processing,* vol. 4, pp. 480 - 494, 2004 2004.

[72] Y. Yin, *et al.*, "Fingerprint Image Segmentation Based on Quadric Surface Model," in *Lecture Notes in Computer Science* vol. 3546/2005, ed, 2005, pp. 647-655.

[73] E. Zhu and e. al, "A Systematic Method For Fingerprint Ridge Orientation Estimate & Image Segmentation," *Pattern Recognition,* pp. 1452 - 1472, 2006 2006.

[74] T. S. Ong, *et al.*, "Fingerprint Images Segmentation Using Two Stages Coarse to Fine Discrimination Technique," in *Lecture Notes in Computer Science*, ed, 2003, pp. 624-633.

[75] A. C. Pais Barreto Marques and A. C. Gay Thome, "A neural network fingerprint segmentation method," ed, 2005, p. 6 pp.

[76] Wang. Sen, *et al.*, "New Features Extraction And Application In Fingerprint Segmentation," in *ACTA AUTOMATICA SINICA* vol. 29, ed, 2003, pp. 622-627.

[77] S. Sato and T. Umezaki, "A fingerprint segmentation method using a recurrent neural network," in *Proceedings of the 2002 12th IEEE Workshop on Neural Networks for Signal Processing*, ed, 2002, pp. 345-354.

[78] S. Bernard, "Fingerprint Segmentation Using The Phase Of Multiscale Gabor Wavelets," *The 5th Asian Conference On Computer Vision,* p. 23, 25 January 2002 2002.

[79] R. Qun, *et al.*, "Automatic Segmentation of Fingerprint Images," *3rd Workshop on Automatic Identification Advanced Technologies,* 2002.

[80] C. Wu, *et al.*, "Robust Point-Based Feature Fingerprint Segmentation Algorithm," *IEEE International Conference on Biometrics (ICB07),* 2007.

[81] A. Baig, *et al.*, "A Corner Strength Based Fingerprint Segmentation Algorithm with Dynamic Thresholding," *IEEE International Conference on Pattern Recognition (ICPR08),* 2008.

[82] M. Y.-S. Yao, *et al.*, *Automatic Fingerprint Recognition Systems - Chapter 3*: Spinger New York, 2004.

[83] B. Bavarian, "Operational Elements of Fingerprint Image Quality," *http://www.itl.nist.gov/iad/894.03/quality/workshop/proc/bavarianimagequalitypresnistworkshop08march06.pdf,* 2006.

[84] E. Lim, *et al.*, "Fingerprint image quality analysis," 2004, pp. 1241-1244 Vol.2.

[85] J. Qi, *et al.*, "Measuring fingerprint image quality using gradient " *Proc. SPIE Vol. 5779, p. 455-459, Biometric Technology for Human Identification II;,* vol. 5779, pp. 455-459, Mar 2005 2005.

[86] T. P. Chen, *et al.*, "Fingerprint image quality analysis," 2004, pp. 1253-1256 Vol.2.

[87] S. Lee, *et al.*, "Model-Based Quality Estimation of Fingerprint Image," *Lecture Notes in Computer Science,* vol. 3832, pp. 229 - 235, 2005.

[88] Y. Chen, *et al.*, "Fingerprint Quality Indices for Predicting Authentication Performance," *Lecture Notes in Computer Science,* vol. 3546, pp. 160 - 170, 2005.

[89] L. Manhua, *et al.*, "Fingerprint Reference-Point Detection," *EURASIP Journal on Applied Signal Processing,* vol. 4, pp. 498 - 509, 2004.

[90] E. Zhu, *et al.*, *Quality Estimation of Fingerprint Image Based on Neural Network*, 2005.

[91] H. Fronthaler, *et al.*, "Automatic Image Quality Assessment with Application in Biometrics," *Proc. Workshop on Biometrics, Assoc. CVPR 2006,* pp. 30-35, 2006.

[92] M. Yao, *et al.*, "Quanitfying Quality: A Case Study in Fingerprints," *Proceedings of IEEE Conference on Automatic Identification Advanced Technologies (AutoID),* pp. 126 - 131, 2002.

[93] Z. Shi, *et al.*, "A New Segmentation Algorithm for Low Quality Fingerprint Images," *Proceedings of the 3rd International Conference on Image and Graphics (ICIG'04),* 2004.

[94] E. Lim, *et al.*, "Fingerprint Quality and Validity Analysis," *Proceedings of IEEE Conference on Image Processing,* vol. 1, pp. 469 - 472, 2002.

[95] H. Kyungtae, "Statistical quality assessment of a fingerprint," 2004, pp. 448-452.

[96] M. D. Garris and C. I. Watson, "Users Guide To NIST Fingerprint Image Software," *National Insititute Of Standards And Technology,* pp. 22-25.

[97] R. Bolle, "System and Methods for Determing the Quality of Fingerprint Images," *United States Patent Number US596356,* 1999.

[98] J. Fierrez-Aguilar, *et al.*, "Incorporating Image Quality in Multi-Algorithm Fingerprint Verification," *Lecture Notes in Computer Science,* vol. 3832, pp. 213-220, 2005.

[99] F. Alonso-Fernandez, *et al.*, "A Review of Schemes for Fingerprint Image Quality Computation," *COST 275 - Biometrics-Based Recognition of People over the Internet,* 2005.

[100] D. Simon-Zorita, *et al.*, "Image quality and position variability assessment in minutiae-based fingerprint verification," *Vision, Image and Signal Processing, IEE Proceedings-,* vol. 150, pp. 402-408, 2003.

[101] P.-S. Liao, *et al.*, "A Fast Algorithm For Multilevel Thresholding," *Journal of Information Science and Engineering,* vol. 17, pp. 713-727, 2001.

[102] K. Harmer, *et al.*, "A Novel Level-Independent Fingerprint Segmentation Algorithm based on Fourier Energy," *Proceedings of the 2007 International Conference on Image Processing, Computer Vision, & Pattern Recognition, IPCV 2007,* 2007.

[103] K. Harmer, *et al.*, "A Computationally Efficient Fingerprint Segmentation Algorithm Using Digital Closed Curves," *Proceedings of the 2007 International Conference on Image Processing, Computer Vision, & Pattern Recognition, IPCV 2007,* 2007.

[104] G. M. Nijm, *et al.*, "Comparison of Signal Peak Detection Algorithms for Self-Gated Cardiac Cine MRI," *Computers in Cardiology 2007,* 2007.

[105] R. B. Fisher and D. K. Naidu, "A Comparison of Algorithms for Subpixel Peak Detection," *Image Technology - Springer-Verlag, Heidelberg,* 1996.

[106] M. Ma, *et al.*, "Developing and Implementing Peak Detection for Real-Time Image Registration," *Proceedings of the 16th Annual Workshop on Circuits, Systems & Signal Processing (ProRISC2005),* pp. 641-652, November 2005 2005.

[107] B. S. Todd, "An Algorithm for the Detection of Peaks and Troughs in Physiological Signals," *Oxford University Computing Laboratory Technical Report,* vol. TR-31-97, September 1997 1997.

[108] K. L. Priddy and P. E. Keller, *Artificial Neural Networks: An Introduction*: SPIE Society of Photo-Optical Instrumentation Eng, 2005.

[109] T. Xuejun and B. Bhanu, "Fingerprint verification using genetic algorithms," 2002, pp. 79-83.

[110] M. M. A. Allah, "Artificial Neural Networks Based Fingerprint Authentication With Clusters Algorithm," *Informatica,* vol. 29, pp. 303 - 307, 2005.

[111] I. Dagher, *et al.*, "Fingerprint recognition using fuzzy artmap neural network architecture," 2002, pp. 157-160.

[112] A. L. H. Jin, *et al.*, "Fingerprint identification and recognition using backpropagation neural network," 2002, pp. 98-101.

[113] A. Murat Ozbayoglu and C. H. Degli, "Unsupervised hierarchical fingerprint matching," 1997, pp. 1439-1442 vol.3.

[114] W. Sheng, *et al.*, "Consensus Fingerprint Matching with Genetically Optimised Approach," *Pattern Recognition,* vol. 42, pp. 1399 - 1407, 2009.

[115] W. Sheng, *et al.*, "A Genetic Algorithm for Fingerprint Matching Based on an Integrated Measure," *IEEE Congress on Evolutionary Computation,* pp. 183 - 191, 2008.

[116] W. Sheng, *et al.*, "Fingerprint Matching with an Evolutionary Approach," *ICB 2007,* pp. 484 - 492, 2007.

[117] G. Qun and G. S. Moschytz, "Fingerprint feature matching using CNNs," 2004, pp. III-73-6 Vol.3.

[118] R. Kohavi and G. H. John, "Wrappers for Feature Subset Selection," *Artificial Intelligence,* vol. 97, pp. 273 - 324, 1997.

[119] G. K. Kanji, *100 Statistical Tests*: Sage Publications Ltd, 1999.

[120] G. A. Marcoulides and S. L. Hershberger, *Multivariate Statistical Methods: A First Course*: Psychology Press, 1997.

[121] B. S. Yandell, *Practical Data Analysis for Designed Experiments*: Chapman & Hall, 1997.

[122] A. M. Bazen and S. H. Gerez, "Systematic methods for the computation of the directional fields and singular points of fingerprints," *Pattern Analysis and Machine Intelligence, IEEE Transactions on,* vol. 24, pp. 905-919, 2002.

[123] M. Ballan, *et al.*, "A fingerprint classification technique using directional images," vol. 1, ed, 1997, pp. 101-104 vol.1.

[124] C. Byoung-Ho, *et al.*, "Core-based fingerprint image classification," 2000, pp. 859-862 vol.2.

[125] R. Cappelli, *et al.*, "Fingerprint classification by directional image partitioning," *Pattern Analysis and Machine Intelligence, IEEE Transactions on,* vol. 21, pp. 402-421, 1999.

[126] S. C. Dass and A. K. Jain, "Fingerprint Classification Using Orientation Field Flow Curves."

[127] Z. Han and C.-P. Liu, *Fingerprint Classification Based on Statistical Features and Singular Point Information*, 2005.

[128] A. K. Jain and S. Minut, "Hierarchical kernel fitting for fingerprint classification and alignment," vol. 2, ed, 2002, pp. 469-473 vol.2.

[129] M. Kawagoe and A. Tojo, "Fingerprint pattern classification," *Pattern Recognition,* vol. 17, pp. 295-303, 1984.

[130] T.-L. Huang, *et al.*, "An Improved Algorithm for Fingerprint Image Enhancement," *Proceedings of Signal and Image Processing,* 2005.

[131] Z. Shi and V. Govindaraju, "A chaincode based scheme for fingerprint feature extraction," *Pattern Recognition Letters,* vol. 27, pp. 462 - 468, 2006.

[132] A. M. Bazen and S. H. Gerez, "Directional Field Computation for Fingerprint based on the Principal Component Analysis of Local Gradients," *Workshop On Circuits, Systems, and Signal Processing - ProRISC 2000,* 2000.

[133] C. H. Park and H. Park, "Fingerprint classification using fast Fourier transform and nonlinear discriminant analysis," *Pattern Recognition,* vol. 38, pp. 495-503, 2005.

[134] A. Jain and H. Lin, "On-line fingerprint verification," 1996, pp. 596-600 vol.3.

[135] G. Jinwei and J. Zhou, "Analysis of Singular Points in Fingerprints based on Topological Structure and Orientation Field," *Technical Report,* 2005.

[136] T. Tang, *et al.*, "An Improved Fingerprint Singular Point Detection Algorithm based on Continuous Orientation Field," *2008 International Symposium on Computer Science and Computational Technology (ISCST'08),* 2008.

[137] J. Gu, *et al.*, "A combination model for orientation field of fingerprints," *Pattern Recognition,* vol. 37, pp. 543-553, 2004.

[138] G. Jinwei and Z. Jie, "A novel model for orientation field of fingerprints," 2003, pp. II-493-8 vol.2.

[139] W. Sen, *et al.*, "Fingerprint classification by directional fields," 2002, pp. 395-399.

[140] J. V. Kulkarni, *et al.*, "Orientation Feature for Fingerprint Matching," *Elsevier - Pattern Recognition,* vol. 39, pp. 1551 - 1554, 2006.

[141] S. Shah and P. S. Sastry, "Fingerprint classification using a feedback-based line detector," *Systems, Man and Cybernetics, Part B, IEEE Transactions on,* vol. 34, pp. 85-94, 2004.

[142] W. Sheng-De and L. Chih-Jen, "Fingerprint recognition using directional micropattern histograms and LVQ networks," 1999, pp. 300-303.

[143] D. Maio and D. Maltoni, "A structural approach to fingerprint classification," 1996, pp. 578-585 vol.3.

[144] Gian Luca Marcialis, *et al.*, "Fingerprint Classification by Combination of Flat and Structural Approaches," *Lecture Notes in Computer Science,* vol. 2091, pp. 241 - 246, 2001.

[145] Michel Neuhaus and H. Bunke, "An Error-tolerant Approximate Matching Algorithm for Attributed Planar Graphs and its Application to Fingerprint Classication."

[146] S. Chikkerur and N. Ratha, "Impact of Singular Point Detection on Fingerprint Matching Performance," 2005, pp. 207-212.

[147] ISO/IEC and ANSI, "Biometric Data Interchange Formats — Part 2: Finger Minutiae Data," *ISO/IEC JTC 1/SC 37,* 04-05-2004 2004.

[148] S. C. Dass, "Markov random field models for directional field and singularity extraction in fingerprint images," *Image Processing, IEEE Transactions on,* vol. 13, pp. 1358-1367, 2004.

[149] M. R. Rahimi, *et al.*, "An Adaptive Approach to Singular Point Detection in Fingerprint Images," *AEU - International Journal of Electronics and Communications,* vol. 58, pp. 367-370, 2004.

[150] M. Rahmati and A. Jannatpour, "Fingerprint classification using singular points and Fourier image " *Proc. SPIE Vol. 3808, p. 547-555, Applications of Digital Image Processing XXII,* vol. 3808, pp. 547-555, Oct 1999 1999.

[151] T. Ohtsuka and A. Kondo, "Improvement of the fingerprint core detection using extended relation graph," 2005, p. 20.

[152] T. Liu, *et al.*, "Fingerprint Reference Point Detection Based on Local Axial Symmetry," *Proceedings of the 18th International Conference on Pattern Recognition (ICPR'06),* 2006.

[153] K. Nilsson and J. Bigun, "Localization of corresponding points in fingerprints by complex filtering," *Pattern Recognition Letters,* vol. 24, pp. 2135-2144, 2003.

[154] L. Wang, *et al.*, "Fingerprint Image Segmentation by Energy of Gaussian-Hermite Moments," in *Lecture Notes in Computer Science*, ed, 2004, pp. 414-423.

[155] C.-H. Park, *et al.*, "Singular point detection by shape analysis of directional fields in fingerprints," in *Pattern Recognition* vol. 39, ed, 2006, pp. 839 - 855.

[156] X. Wang and M. Xie, "Fingerprint Classification: An Approach Based on Singularities and Analysis of Fingerprint Structure," in *Lecture Notes in Computer Science*, ed, 2004, pp. 324-329.

[157] H. O. Nyongesa, *et al.*, "Fast Robust Fingerprint Feature Extraction and Classification," *Journal of Intelligent and Robotic Systems,* vol. 40, pp. 103-112, 2004.

[158] X. Zhan, *et al.*, *Fingerprint Ridge Distance Estimation: Algorithms and the Performance*, 2005.

[159] Z. Jie and G. Jinwei, "A model-based method for the computation of fingerprints' orientation field," *Image Processing, IEEE Transactions on,* vol. 13, pp. 821-835, 2004.

[160] J. Zhou and J. Gu, "Modeling orientation fields of fingerprints with rational complex functions," *Pattern Recognition,* vol. 37, pp. 389-391, 2004.

[161] K. Rerkrai and V. Areekul, "A new reference point for fingerprint recognition," 2000, pp. 499-502 vol.2.

[162] D. Maio and D. Maltoni, "Direct gray-scale minutiae detection in fingerprints," *Pattern Analysis and Machine Intelligence, IEEE Transactions on,* vol. 19, pp. 27-40, 1997.

[163] G. Bebis, *et al.*, "Fingerprint identification using Delaunay triangulation," ed, 1999, pp. 452-459.

[164] G. Parziale and A. Niel, *A Fingerprint Matching Using Minutiae Triangulation*, 2004.

[165] K. Mieloch, *et al.*, "Dynamic threshold using polynomial surface regression with application to the binarization of fingerprints " *Proc. SPIE Vol. 5779, p. 94-104, Biometric Technology for Human Identification* vol. 5779, pp. 94-104, Mar 2005

2005.

[166] H. Fronthaler, *et al.*, "Local Feature Extraction in Fingerprints by Complex Filtering," *Lecture Notes in Computer Science,* vol. 3781, pp. 77 - 84, 2005.

[167] C. J. Lee and T. N. Yang, "Direct Minutiae Detection in Raw Fingerprint Images," *Signal and Image Processing,* 2004.

[168] J.-H. Shin, *et al.*, *Minutiae Extraction from Fingerprint Images Using Run-Length Code*, 2003.

[169] A. K. Jain, *Fundamentals of Digital Image Processing*: Prentice Hall, 1988.

[170] N. Yager and A. Amin, "Fingerprint verification based on minutiae features: a review," *Pattern Analysis & Applications,* vol. 7, pp. 94-113, 2004.

[171] D. C. D. Hung, *et al.*, "A Fuzzy Technique for Flow Oriented Image Direction Computing," *Proceedings of ANNIE 96,* 1996.

[172] B. Lee, *et al.*, "A novel measure of fingerprint image quality using the Fourier spectrum " *Proc. SPIE Vol. 5779, p. 105-112, Biometric Technology for Human Identification II;,* vol. 5779, pp. 105-112, Mar 2005 2005.

[173] Z. M. Kovacs-Vajna, *et al.*, "Fingerprint ridge distance computation methodologies," *Pattern Recognition,* vol. 33, pp. 69-80, 2000.

[174] V. A. Sujan and M. P. Mulqueen, "Fingerprint identification using space invariant transforms," *Pattern Recognition Letters,* vol. 23, pp. 609-619, 2002.

[175] S. B. Nikam and S. Agarwal, "Level 2 Features and Wavelet Analysis based Hybrid Fingerprint Matcher," *Proceedings of the 1st Annual Bangalore Compute Conference,* 2008.

[176] M. Tico, *et al.*, "Fingerprint recognition using wavelet features," 2001, pp. 21-24 vol. 2.

[177] H. Ke and S. Aviyente, "Combining generalized Gaussian density and energy distribution in wavelet analysis for texture classification," 2004, pp. 2094-2098 Vol.2.

[178] S. Ribaric, "A Biometric Identification System Based on Eigenpalm and Eigenfinger Features," *IEEE Transactions on Pattern Analysis and Machine Intelligence,* vol. 27, pp. 1698 - 1709, 2005.

[179] T. Kamei and M. Mizoguchi, "Fingerprint preselection using eigenfeatures," *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition,* pp. 918-923, 1998.

[180] S. B. Nikam and S. Agarwal, "Gabor Filter-Based Fingerprint Anti-Spoofing," *Lecture Notes in Computer Science,* vol. 5259, pp. 1103 - 1114, 2008.

[181] Aparecido Nilceu Marana and A. K. Jain, "Ridge-Based Fingerprint Matching Using Hough Transform," *18th Brazilian Symposium on Computer Graphics and Image Processing,* pp. 112 - 119, 2005.

[182] J. C. Yang and D. S. Park, "A Fingerprint Verification Algorithm using Tessellated Invariant Moment Features," *Neurocomputing,* vol. 71, pp. 1939 - 1946, 2008.

[183] J. Daugman, "How Iris Recognition Works," *IEEE Transactions in Circuits and Systems for Video Technology,* vol. 14, pp. 21-30, 2004.

[184] A. Ross, *et al.*, "A hybrid fingerprint matcher," 2002, pp. 795-798 vol.3.

[185] L. Chih-Jen and W. Sheng-De, "Fingerprint feature extraction using Gabor filters," *Electronics Letters,* vol. 35, pp. 288-290, 1999.

[186] S. Lifeng, *et al.*, "Improved fingercode for filterbank-based fingerprint matching," 2003, pp. II-895-8 vol.3.

[187] A. Lumini and L. Nanni, "Two-class fingerprint matcher," *Pattern Recognition,* vol. In Press, Corrected Proof.

[188] A. Jain, *et al.*, "Fingerprint matching using minutiae and texture features," in *Image Processing, 2001. Proceedings. 2001 International Conference on*, 2001, pp. 282-285 vol.3.

[189] C.-J. Lee and S.-D. Wang, "Fingerprint feature extraction using Gabor filters," *Electronics Letters,* vol. 35, pp. 288-290, 1999.

[190] M. K. Hu, "Visual Pattern Recognition by Moment Invariants," *IRE Transactions in Information Theory,* vol. IT-8, pp. 179 - 187, 1962.

[191] A. Khotanzad and Y. H. Hong, "Invariant Image Recognition by Zernike Moments," *IEEE Transactions on Pattern Analysis and Machine Intelligence,* vol. 12, pp. 489 - 497, 1990.

[192] M. Nixon and A. S. Aguardo, *Feature Extraction and Image Processing*: Academic Press, 2008.

[193] J. C. Yang and D. S. Park, "A Fingerprint Verification Algorithm using Tessellated Invariant Moment Features," *Elsevier - Neurocomputing,* vol. 71, pp. 1939 - 1946, 2008.

[194] S. G. Mallat, "A Wavelet Tour of Signal Processing," *Academic Press,* p. 577, 1998.

[195] A. Adler, "Biometric System Security," in *Handbook of Biometrics*, ed: Springer, 2007.

[196] A. K. Jain, *et al.*, "Biometric Template Security," *EURASIP Journal on Advances in Signal Processing (Special Issue on Biometrics),* 2008.

[197] N. Ratha, *et al.*, "Privacy Enhancements for Inexact Biometric Templates," in *Security with Noisy Data*, ed: Springer London, 2007.

[198] A. T. B. Jin and J. Kim, "FuzzyHash: A Secure Biometric Template Protection Technique," *2007 Frontiers in the Convergence of Bioscience and Information Technologies,* 2007.

[199] Y. Sutcu, *et al.*, "A Secure Biometric Authentication Scheme Based on Robust Hashing," *ACM Multimedia and Security Workshop,* 2005.

[200] A. Shamir, "How to Share a Secret," *Communications of the ACM,* vol. 22, pp. 612-613, November 1979 1979.

[201] S. Yang and I. M. Verbauwhede, "Secure fuzzy vault based fingerprint verification system," 2004, pp. 577-581 Vol.1.

[202] B. Chen and V. Chandran, "Biometric Based Cryptographic Key Generation from Faces," *9th Biennial Conference of the Australian Pattern Recognition Society on Digital Image Computing Techniques and Applications,* pp. 394 - 401, 2007.

[203] W. J. Scheirer and T. E. Boult, "Cracking Fuzzy Vaults and Biometric Encryption," *Biometrics Symposium 2007,* pp. 1 - 6, Sept 2007 2007.

[204] A. Adler, "Vulnerabilities in Biometric Encryption System," *Audio- and Video-based Biometric Person Authentication,* vol. LNCS3546, 2005.

[205] A. Bodo, "Method for Producing a Digital Signature with Aid of a Biometric Feature," *German Patent DE 4243908A1,* 1994.

[206] P. K. Janbandhu and M. Y. Siyal, "Novel Biometric Digital Signatures for Internet-based Applications," *Information Management and Computer Security,* vol. 9, pp. 205 - 212, 2001.

[207] F. Hao and C. W. Chan, "Private Key Generation from On-Line Handwritten Signatures," *Information Management and Computer Security,* vol. 10, pp. 159 - 164, 2002.

[208] Y. Yamazaki and N. Komatsu, "A Secure Communication System using Biometric Identity Verification," *IEICE Transactions on Informations and Systems,* vol. E84-D(7), pp. 879 - 884, 2001.

[209] S. Hoque, *et al.*, "Feasability of Generating Biometric Encryption Keys," *Electronics Letters,* vol. 41, 17 March 2005 2005.

[210] W. Sheng, *et al.*, "Template-Free Biometric Key Generation by Means of Fuzzy Genetic Clustering," *IEEE Transactions in Information Forensics and Security,* vol. 3, pp. 183 - 191, 2008.

[211] A. K. Jain, *et al.*, *Handbook of Biometrics*: Springer, 2007.

[212] N. K. Ratha, *et al.*, "Cancelable Biometrics: A Case Study in Fingerprints," *Proceedings of 18th International Conference on Pattern Recognition,* vol. 4, pp. 370 - 373, 2006.

[213] W. C. Ku, *et al.*, "Further cryptanalysis of fingerprint-based remote user authentication scheme using smartcards," *Electronics Letters,* vol. 41, pp. 240-241, 2005.

[214] E.-J. Yoon and K.-Y. Yoo, *Secure Fingerprint-Based Remote User Authentication Scheme Using Smartcards*, 2005.

[215] Y. Wu, Dong (Kent Ridge Digital Labs), "Method of Using Biometric Information for Secret Generation (Patent No: WO 02/078249 A1)," 2002.

[216] W. G. J. Howells, *et al.*, "A Method and Apparatus for the Generation of Code from Pattern Features (Patent No: WO/2008/003945)," 2006.

[217] E. Papoutsis, *et al.*, "Integrating Multimodal Circuit Features within an Efficient Encryption System," *Journal of Information Assurance and Security,* vol. 2, pp. 117 - 126, 2007.

[218] E. Papoutsis, *et al.*, "Key Generation for Secure Inter-Satellite Communication," *NASA / ESA Conference on Adaptive Hardware and Systems (AHS - 2007),* 2007.

[219] E. Papoutsis, *et al.*, "Effects of Feature Trimming on Encryption Key Stability for an ICmetric System," *2008 ECSIS Symposium on Bio-Inspired, Learning and Intelligent Systems for Security,* 2008.

[220] F. Gray, "Pulse Code Communication (Patent No: 2,632,058)," 1953.

[221] U. Korte and R. Plaga, "Cryptographic Protection of Biometric Templates: Chance, Challenges and Applications," *Lecture Notes in Informatics,* vol. 108, 2007.