Extracting Fingerprint Features using Textures

By

Joshua Mackley, BEng. (Honours)

Submitted in fulfilment of the requirements for the degree of

Master of Engineering

Deakin University, May 2004.
I certify that the thesis entitled:

**Extracting Fingerprint Features using Textures**

submitted for the degree of:

**Master of Engineering**

is the result of my own work and that where reference is made to the work of others, due acknowledgment is given.

I also certify that any material in the thesis which has been accepted for a degree or diploma by any other university or institution is identified in the text.

Full Name............................................................................................................

(Please Print)

Signed ......................................................................................................................

Date.......................................................................................................................
Abstract

Personal identification of individuals is becoming increasingly adopted in society today. Due to the large number of electronic systems that require human identification, faster and more secure identification systems are pursued.

Biometrics is based upon the physical characteristics of individuals; of these the fingerprint is the most common as used within law enforcement. Fingerprint-based systems have been introduced into the society but have not been well received due to relatively high rejection rates and false acceptance rates. This limited acceptance of fingerprint identification systems requires new techniques to be investigated to improve this identification method and the acceptance of the technology within society. Electronic fingerprint identification provides a method of identifying an individual within seconds quickly and easily.

The fingerprint must be captured instantly to allow the system to identify the individual without any technical user interaction to simplify system operation. The performance of the entire system relies heavily on the quality of the original fingerprint image that is captured digitally. A single fingerprint scan for verification makes it easier for users accessing the system as it replaces the need to remember passwords or authorisation codes. The identification system comprises of several components to perform this function, which includes a fingerprint sensor, processor, feature extraction and verification algorithms. A compact texture feature extraction method will be implemented within an embedded microprocessor-based system for security, performance and cost effective production over currently available commercial fingerprint identification systems.
To perform these functions various software packages are available for developing programs for windows-based operating systems but must not constrain to a graphical user interface alone. MATLAB was the software package chosen for this thesis due to its strong mathematical library, data analysis and image analysis libraries and capability. MATLAB enables the complete fingerprint identification system to be developed and implemented within a PC environment and also to be exported at a later date directly to an embedded processing environment.

The nucleus of the fingerprint identification system is the feature extraction approach presented in this thesis that uses global texture information unlike traditional local information in minutiae-based identification methods. Commercial solid-state sensors such as the type selected for use in this thesis have a limited contact area with the fingertip and therefore only sample a limited portion of the fingerprint. This limits the number of minutiae that can be extracted from the fingerprint and as such limits the number of common singular points between two impressions of the same fingerprint. The application of texture feature extraction will be tested using variety of fingerprint images to determine the most appropriate format for use within the embedded system.

This thesis has focused on designing a fingerprint-based identification system that is highly expandable using the MATLAB environment. The main components that are defined within this thesis are the hardware design, image capture, image processing and feature extraction methods. Selection of the final system components for this electronic fingerprint identification system was determined by using specific criteria to yield the highest performance from an embedded processing environment.

These platforms are very cost effective and will allow fingerprint-based identification technology to be implemented in more commercial products that can benefit from the security and simplicity of a fingerprint identification system.
Acknowledgments

I would like to take this opportunity to acknowledge the guidance and support of Professor Saeid Nahavandi, the principle academic supervisor of this thesis.

I also would like to acknowledge Dr. Leon Burgess-Dean, Dr. Abbas Kouzani and Dr. Hong Zheng who helped with various stages of the thesis of which I am most grateful.

I would also like to acknowledge Mr. M. Hunter of Kanowna Technology Ltd. and Mr. D. Daniell of Unique Technology Ltd. for the supply of evaluation equipment.

I personally would like to thank my family and close friends for their support during my studies at Deakin University.
## Contents

1 INTRODUCTION.........................................................................................................................1

1.1 HUMAN IDENTIFICATION .............................................................................................1
1.2 BIOMETRICS ......................................................................................................................1
1.3 FINGERPRINTS ....................................................................................................................3
1.4 FINGERPRINT SENSORS ....................................................................................................5
1.5 FEATURE EXTRACTION ......................................................................................................7
1.6 FINGERPRINT VERIFICATION ........................................................................................9
1.7 THESIS OVERVIEW ..........................................................................................................11

2 FINGERPRINTS FOR IDENTIFICATION.............................................................................12

2.1 REPRESENTATION ..............................................................................................................12
2.2 INFORMATION ..................................................................................................................14
2.3 INDIVIDUALITY ..................................................................................................................15
  2.3.1 Genetic Factors ...........................................................................................................15
  2.3.2 Environmental Factors ...............................................................................................16
2.4 FINGERPRINT CLASSIFICATION ...................................................................................16
  2.4.1 Approaches to Classification .......................................................................................17
2.5 SUMMARY ..........................................................................................................................19

3 IDENTIFICATION SYSTEM ..............................................................................................20

3.1 FINGERPRINT SENSORS ..................................................................................................20
3.2 COMMERCIAL SENSORS ...............................................................................................21
3.3 FINGERPRINT SENSOR SELECTION .............................................................................23
3.4 ID SYSTEM DESIGN ..........................................................................................................24
  3.4.1 Microcontroller ...........................................................................................................26
  3.4.2 Fingerprint Sensor .....................................................................................................27
  3.4.3 External Memory and System Interface ..................................................................31
  3.4.4 Software ....................................................................................................................31
3.5 SUMMARY ..........................................................................................................................32
4 MATLAB INTERFACE .................................................................................................................. 33
  4.1 BENEFITS OF THE MATLAB ENVIRONMENT ................................................. 34
  4.2 USER CONTROL ............................................................................................. 35
  4.3 FEATURE INFORMATION .................................................................................. 36
  4.4 SUMMARY ........................................................................................................... 37

5 FEATURE EXTRACTION & MATCHING ............................................................................. 38
  5.1 IMAGE ANALYSIS ............................................................................................ 38
    5.1.1 Texture Features .................................................................................. 40
    5.1.2 Texture Methods ................................................................................... 40
  5.2 EXTRACTING FEATURES ................................................................................. 42
    5.2.1 Law’s Method ...................................................................................... 43
    5.2.2 Image Partitioning ............................................................................... 43
    5.2.3 Feature Value ...................................................................................... 44
  5.3 FINGERPRINT MATCHING .............................................................................. 44
    5.3.1 Neural Networks .................................................................................. 45
    5.3.2 Simple Neural Network Model .............................................................. 45
    5.3.3 Developing a Suitable Neural Network .................................................. 47
    5.3.4 Matching Template ............................................................................... 49
  5.4 SUMMARY ......................................................................................................... 49

6 EXPERIMENTAL RESULTS ............................................................................................. 51
  6.1 FINGERPRINT IDENTIFICATION HARDWARE ................................................ 51
    6.1.1 Thermal Sensor Testing ........................................................................ 52
    6.1.2 Thermal Sensor Results ........................................................................ 55
    6.1.3 Fingerprint Sensor Selection ................................................................ 57
  6.2 IMAGE FILTERING ............................................................................................. 58
    6.2.1 Filter Response .................................................................................... 58
    6.2.2 Law’s Texture Filter............................................................................... 62
    6.2.3 Filter Selection ..................................................................................... 64
Tables

TABLE 3.1 – VERIDICOM SENSOR AND SPECIFICATIONS [13] ................................................................. 27
TABLE 3.2 – INFINEON SENSOR AND SPECIFICATIONS [33] .............................................................. 28
TABLE 3.3 – ATMEL SENSOR AND SPECIFICATIONS [21] ................................................................. 29
TABLE 3.4 – KSI SENSOR AND SPECIFICATIONS [30] ........................................................................ 30
TABLE 5.1 – THE FIVE 1-D ARRAYS AS IDENTIFIED BY LAWS [53] ......................................................... 42
TABLE 6.1 - SENSOR SELECTION ........................................................................................................... 57
Figures

Figure 1.1 - Sample Fingerprint Image [27] ................................................................. 4
Figure 1.2 - Atmel Thermal Fingerprint Sensor [21] .................................................... 6
Figure 1.3 - Minutiae Located (Square Bifurcation, Circle Ridge Ending) [29] .............. 8
Figure 2.1 - Minutiae Matching [27, 28] ................................................................ 13
Figure 2.2 - Local Features (Minutiae) [22, 46] ............................................................. 14
Figure 2.3 – Five Classes of Fingertips (Henry System) [26, 46] .................................... 17
Figure 2.4 - Fingerprint 'Core' and 'Delta' Locations [46] ......................................... 18
Figure 2.5 - Orientation Field placed over fingerprint image [5] .................................... 18
Figure 3.1 – System Breakdown [12] ........................................................................... 25
Figure 3.2 – Atmel ATmega128 Microcontroller [4, 21] ................................................. 26
Figure 4.1 – MATLAB Interface developed for experiments ....................................... 36
Figure 5.1 – Feature Extraction Flow Chart ................................................................. 39
Figure 5.2 – Simple Neural Network Model [2, 44] ..................................................... 45
Figure 5.3 – Neuron Model [44] ................................................................................ 46
Figure 5.4 - Neural Network Overview [38] ................................................................ 48
Figure 6.1 - Atmel Thermal 'Fingertip' Sensor [19, 21] ................................................. 52
Figure 6.2 - USB Fingerprint Sensor [25] .................................................................... 53
Figure 6.3 - Fingerprint Slices and Re-construction [20] .............................................. 54
Figure 6.4 – User Skew Error [20] ............................................................................. 55
Figure 6.5 - Sensor Surface Temperature Error [20] .................................................... 56
Figure 6.6 - Missing Fingerprint Information [20] ....................................................... 56
Figure 6.7 – Fingerprint Image 20991.bmp [24] ......................................................... 59
Figure 6.8 - Level Texture Filter (20991.bmp) [12] .................................................... 59
Figure 6.9 - Spot Texture Filter (20991.bmp) [12] ..................................................... 60
Figure 6.10 - Wave Texture Filter (20991.bmp) [12] .................................................. 61
Figure 6.11 - Ripple Texture Filter (20991.bmp) [12] .................................................. 61
Figure 6.12 - Edge Texture Filter (20991.bmp) [12] ................................................... 62
Figure 6.13 – Horizontal (left) and Vertical (right) Edge Filter Mask [12] ..................... 63
Figure 6.14 – Noise from Edge Filter (Horizontal Filter Mask) [12] ............................. 63
Figure 6.15 – Noise from Edge Filter (Vertical Filter Mask) [12] ................................. 64
Figure 6.16 - JPG Compression [60] ............................................................................ 65
Figure 6.17 - WSQ Compression [34, 60] .................................................................... 66
Figure 6.18 - Feature Plot (20991.bmp) [12] ............................................................... 68
Figure 6.19 - Plot showing difference between individual fingerprints [12] ................. 70
Figure 6.20 – ANOVA Result for Fingerprint scans 20991-20995 [12] ....................... 71
Figure 6.21 – PCA Feature Plot [12] ........................................................................... 72
Figure 7.1 - Fingerprint System Overview .................................................................... 74
# Acronyms and Abbreviations

<table>
<thead>
<tr>
<th>Acronym</th>
<th>Definition</th>
</tr>
</thead>
<tbody>
<tr>
<td>FAR</td>
<td>False Acceptance Rate of a Fingerprint System</td>
</tr>
<tr>
<td>FRR</td>
<td>False Rejection Rate of a Fingerprint System</td>
</tr>
<tr>
<td>PIN</td>
<td>Personal Identification Number used with bank cards</td>
</tr>
<tr>
<td>ATM</td>
<td>Automatic Transaction Machine</td>
</tr>
<tr>
<td>NIST</td>
<td>National Institute of Standards and Technology</td>
</tr>
<tr>
<td>FBI</td>
<td>Federal Bureau of Investigation</td>
</tr>
<tr>
<td>DPI</td>
<td>Dots Per Inch</td>
</tr>
<tr>
<td>CCD</td>
<td>Charge-Coupled Device</td>
</tr>
<tr>
<td>MIPS</td>
<td>Million Instructions Per Second</td>
</tr>
<tr>
<td>GUI</td>
<td>Graphical User Interface</td>
</tr>
<tr>
<td>FEI</td>
<td>Feature Extraction Interface</td>
</tr>
<tr>
<td>PCA</td>
<td>Principle Component Analysis</td>
</tr>
<tr>
<td>ANOVA</td>
<td>Analysis Of Variance</td>
</tr>
<tr>
<td>DIO</td>
<td>Data Input Output</td>
</tr>
<tr>
<td>USB</td>
<td>Universal Serial Bus</td>
</tr>
<tr>
<td>DSP</td>
<td>Digital Signal Processor</td>
</tr>
<tr>
<td>PC</td>
<td>Personal Computer</td>
</tr>
<tr>
<td>ID</td>
<td>Identification</td>
</tr>
</tbody>
</table>
1 Introduction

1.1 Human Identification

In the world today we are identified by the credit cards we use, cheques we sign and passwords that allow us to sign into a computer system. We cannot remove funds from an account without the correct authorisations e.g. credit card and signature or debit card and PIN. We can benefit from an easier system to determine our identity, so that we don’t have to carry a multitude of bankcards and remember a PIN.

Biometrics promises to provide ease of use for secure systems, as the identification method is present with the individual at all times. This thesis focuses on using biometrics as a way to provide identification of an individual and authorisation to access certain secure systems in use today. Typical systems that can benefit from biometric authorisation are ATM, electronic banking and computer authorisation. This can also be used to replace traditional key-based security methods such as vehicles, houses, garages and buildings using an electronic activated lock validated by an authorised biometric signature.

1.2 Biometrics

Instead of identifying an individual by a bankcard, biometrics identifies an individual on his or her physiological characteristics. Biometrics can be capable of reliable verification and high security that requires the individual to be physically present for the verification process. Biometrics also has the benefit of not requiring the individual to remember numbers or passwords.
The applications of biometric technology are increasing due to the rapid increase in the number of electronic commerce (e-commerce) and banking transactions that are carried out in society. Authentication of the user to allow access within the system has traditionally used identification cards or a token-based method; these applications are currently leaning towards using biometric technology as a more secure and convenient approach for user authentication.

Biometrics covers a wide range of physiological identification techniques (e.g. DNA, facial, fingerprint, retina, voice print, facial thermogram, odour) but there are four major techniques used to identify a person’s identity in society and these are DNA sample, retina scan, facial scans and fingerprint [30]. For day-to-day purposes, a simple identification card such as a credit card and a personal identification number are used as security measures for personal fund transfers. This is very convenient but also very susceptible to abuse by other persons.

To improve security for funds, financial institutions have been researching biometric systems to replace the common personal identification number but these have not been accepted by the market place and are still in early development [30]. Researchers have tested retina scan based identification systems but users find these systems invasive, as the eye must be scanned [30]. Users are very protective of their eyes and find this system uncomfortable even though it does not pose a threat to their vision. Fingerprints are far less invasive but can provide the security required for financial institutions.

Fingerprints are used widely in the military and the police force for determining a persons identification but the majority of identification methods are done manually; with each single finger being copied using a simple piece of paper with ink on the finger then compared using the eye [26]. This is very slow especially when comparison is required at a later date to retrieve these samples and then do the comparison again.
DNA techniques are used in investigations when human blood, tissue or hair samples can be used to extract the DNA information and compare this to samples taken from persons who are believed to be part of the investigation in question. DNA samples are still a specialised identification technique due to the complexity of comparing blood samples and extracting the DNA information. Among all biometrics currently used today, fingerprint based identification is one of the most proven techniques [8].

1.3 Fingerprints

Fingerprints are well known in society for their identification of individuals from criminal identification. This thesis has chosen to use the fingerprint as the method of identification due its non-invasive nature to the individual. The properties of fingerprint formation are well known by forensic scientists and have been used extensively for identification of criminals around the world [51]. The extensive use of fingerprint identification from criminal purposes proves the reliability of an identification system based upon the fingerprint. No two fingerprints have ever been found to have the same ridge detail sequence that is identical between any two individuals [1, 26].

As part of this thesis, research is required to determine what features can be extracted from the fingerprint, then developing a reliable method of electronically finding certain features within the fingerprint and extracting them for identification. The motivation behind this research is to develop an extremely compact algorithm that may be used in a microcontroller based system and that has a mathematical format that executes quickly on a low computational power microcontroller. This will ultimately reduce the cost of fingerprint verification due to lower hardware costs and enable this technology to be implemented widely where identification is required.
The structure of the fingerprint is made from a pattern of ridges and valleys shown in Figure 1.1. Fingerprints contain two main types of features; local features within a restricted region of the fingerprint, and global features that take an overall attribute of the entire fingerprint. Either of these two features can be used for identification of an individual as both features do not change over time.

Figure 1.1 - Sample Fingerprint Image [27]

Fingerprints are formed during foetus development and are a signature that does not change throughout the life of an individual except due to environmental damage to the fingerprint [48]. This unique signature of individuals could be used to replace the many cards that we carry around with us today. With a swipe of our finger we could purchase a book, unlock our car door or even log-on to our computer workstation. This technology depends on the correct electronic interpretation of the fingerprint of each individual within the target population [30]. Such a secure key cannot be reproduced easily and therefore safer than traditional bank cards [26, 30].
Compact solid state fingerprint sensors allow this technology to be integrated into various systems (e.g. laptop computers), and is slowly gaining acceptance in the marketplace. As fingerprint identification has been purely known as a criminal based identification method in the past, society is accepting this technology as a valid method of identification. Advances in fingerprint sensor technology are reducing the cost of fingerprint identification systems while still providing acceptable performance.

Fingerprint based biometric identification also has disadvantages that is holding back this technology from the marketplace. Society finds this technology as an invasion of privacy as the fingerprint is a permanent identification method. The security of the fingerprint must be maintained for society to accept this method of identification. Poor quality fingerprints can be found from approximately 4% of the population due to dry skin, dirty fingers, scratches from manual labour, natural creases and environmental damage to the fingerprint. This thesis is focusing on improving some of these disadvantages to make fingerprint identification more attractive for use in society.

1.4 Fingerprint Sensors

The fingerprint must be captured in an electronic format to use in an automatic fingerprint identification system. There are two popular methods that are used to capture a fingerprint image; using traditional ‘inked’ methods where ink is applied to the finger and rolled or dabbed to a piece of paper and scanned into a computer by hand (offline) or an electronic fingerprint sensor that acquires the fingerprint image directly from the fingerprint in an electronic format (live scan). The offline method is unsuitable for automated systems as humans are still required to process the fingerprint image using a scanner. Using an electronic fingerprint sensor enables the fingerprint image to be acquired directly from the individual without any other human intervention. Figure 1.2 shows a typical fingerprint sensor.
Fingerprint sensors provide an ideal interface between the individual and identification system. Two types of sensors are available, partial sensors that require the individual to roll their finger across the small sensor area to capture the fingerprint, and complete sensors that take a complete fingerprint from a large sensor area. There are different technologies that are used to acquire the fingerprint from the finger, of these the most common and proven are optical, thermal and capacitive sensors. Optical fingerprint sensors use a small camera to take a complete image of the finger pressed against a clear screen. Due to the size of the camera and optical equipment required, the optical sensors are often quite large and produce poor quality images due to the fingerprint being distorted while being pressed against the clear screen. Thermal fingerprint scanners measure the temperature difference between the ridges and valleys of the fingerprint surface. These sensors are usually only a partial fingerprint sensor due to the cost of the thermal technology and require large computational power to compile a complete image from the slices of the image taken while the finger is rolled across the sensor surface. This rolling of the finger also produces errors due to the user not rolling at a constant rate. Capacitive fingerprint sensors measure the pressure differences between the ridges and valleys to cause a capacitance differential in the sensor surface. The capacitive sensors are the most popular used today due to the low cost technology, are very durable and also acquire a complete fingerprint image.
To capture a high quality fingerprint image, the fingerprint sensor must have sufficient resolution to acquire all the information of the fingerprint. The FBI standard for fingerprints is 500dpi (dots per inch) with 8-bits per pixel (256 gray levels) and an image size of 512x512 pixels [51]. Most commercial sensors adhere to the 500dpi and 256 gray level standards set by the FBI, but depending on the sensor size, the majority do not have an image size of 512x512 pixels. For example the Infineon capacitive fingerprint sensor has a sensor area of 224x288 pixels [33] and the DFR-200 from which the images used in this thesis were taken has a sensor area of 360x364 pixels. To adhere to the FBI standard, only images that have been scanned at 500dpi have been used in this thesis.

There are a number of issues that need to be taken into account to improve system accuracy when using live-scan fingerprints. These issues are due to the sensors used to capture the fingerprint image and affect the quality of the fingerprint. Sensing the fingerprint image can add noise to the image such as residue on the sensor surface from a previous fingerprint impression. This may be due the sensor surface not returning to its normal state prior to the next impression. Inconsistent contact of the fingerprint across the sensor surface can distort the fingerprint. As the finger is placed onto the flat sensor surface, the 3-D fingerprint is flattened out and distorted in the process. Different impressions of the same finger can distort and displace various portions of the fingerprint. Non-uniform contact of the finger to the sensor surface due to dryness of the skin, shallow or worn ridges, sweat, dirt and humidity can affect the finger for a non-uniform contact to the sensor surface. These issues taken into account with the system design will reduce errors due to the acquisition process.

1.5 Feature Extraction

The method of feature extraction is the main focus of this thesis. The majority of automatic fingerprint identification systems today use local features known as ‘minutiae’ for identification of individual fingerprints. Figure 1.3 shows the two basic types of minutiae that have been located within the fingerprint; bifurcations and ridge endings. This method is typically used by forensic experts, and as such has been adopted by current identification systems.
There is a limited amount of information available in the minutiae based method; this thesis has represented the fingerprint as a texture as it is more favourable for automated systems.

![Figure 1.3 - Minutiae Located (Square Bifurcation, Circle Ridge Ending)](image)

The quality of the fingerprint can reduce the accuracy of feature extraction and great care must be taken when pre-processing the image (i.e. filtering) so that detailed fingerprint information is not lost in the process and noise is not introduced into the fingerprint. Removing unwanted information from the fingerprint image such as the background will be approached using texture filters. This thesis will be using texture filters to enhance the fingerprint ridges, ultimately improving the quality of the feature information extracted from the fingerprint image.

Using the enhanced fingerprint image, texture features can be extracted by partitioning the fingerprint image into smaller images. This method allows a feature value to be extracted based on the texture component of each partition. These values will be known as the texture features of the fingerprint and will be used by the verification algorithm for individual enrolment to the identification system, and user verification. The number of features that are extracted from the fingerprint affects the verification algorithm and method directly.
Design of the feature extraction algorithm must take into account the affect it has on the verification algorithm for both performance and computational requirements. This thesis focuses on reducing the computational load of the verification algorithm to help with performance on an embedded microprocessor based system. This reduces the number of features that can be used by the verification algorithm and that can be extracted from the fingerprint.

The features that are extracted from the fingerprint are known as the individual’s ‘template’. This template is the signature of the fingerprint, taking only the key features and not the entire fingerprint. To create this template, the individual normally has to enrol onto the identification system by submitting the fingerprint to the system multiple times so that the features can be cross-referenced and check that no features have been missed between scans. Once enrolled this template is used for verification in the future.

1.6 Fingerprint Verification

Verification is the process of matching two fingerprints and determining if they are of sufficient similarity to be the same fingerprint. Verification also defines a metric similarity and threshold to decide whether or not a given pair of fingerprint representations (template) belongs to the same finger i.e. a successful match.

There are two main modes of the verification algorithm that are required by the automatic fingerprint identification system; enrolment to learn a new individual and verification to match a current individual. Once the system has determined that all features available on the fingerprint have been extracted, the individual is enrolled and can be verified at a later date. The enroling process creates a user database where the user details (templates) are stored for reference. Large databases require an indexing scheme to speed up the searching process, and this is usually done with a classification scheme based on global features [51].
Verification is the process of matching a scanned fingerprint (live-scan) with a template stored within the database. Verification matches the template features with the fingerprint features extracted from the sensor. If a successful match can be made between the individual and the database, the individual can be given access or authorised to use the system.

Two main points of interest affect the final performance of the verification algorithm and the entire system; false acceptance rates (FAR) and false rejection rates (FRR) [8, 28]. A false acceptance is defined as the false acceptance of an individual to a system who is not enrolled, and a false rejection is defined as the rejection of an individual who is enrolled on the system. FRR are inversely proportional to FAR, so decreasing the FAR for security purposes, increases the FRR reducing system performance (multiple scans may be required to access the system). Current technology has achieved false acceptance rates of 0.001% [30] although the database of fingerprints used is unknown and performance results can not be compared. Commercial systems available have much higher false acceptance rates of 0.01% to reduce the false rejection rate thereby increasing system performance.

This thesis uses an artificial neural network as the method for using the features extracted from the fingerprint, and enrolling and verification of individuals. Neural networks use simple mathematical functions to perform complex tasks. Neural networks also can be trained for pattern recognition and produce reliable and repeatable answers, ideal for fingerprint pattern recognition [42]. This thesis uses a feed-forward neural network with back propagation for faster training. The size of the network will be kept to a minimum to reduce the memory and processing requirements, keeping in mind that the final system will use a limited embedded microprocessor.
1.7 Thesis Overview

Most of the existing automatic fingerprint identification systems today use forensic expert representations of the fingerprint for matching. This thesis has focused on using a non-minutiae based representation due to the limited information available in minutiae. The performance of the texture feature method used in this thesis will be used for small databases and aimed at integration into a compact embedded microprocessor based system.

Chapter 2 discusses using fingerprints for identification. Chapter 3 is an overview of the fingerprint identification system and the platform for which the algorithms will be developed. Chapter 4 describes the MATLAB environment that was used for software development. Chapter 5 describes the texture feature extraction representation used for the fingerprint and overlooks the possibility to use artificial neural networks for verification of the fingerprint. Chapter 6 presents the results found from conducting experiments in this thesis and Chapter 7 provides discussion and conclusions that have been drawn and future directions of this research.
2 Fingerprints for Identification

Identification of people today is a large part of society. Financial transactions are heavily based upon user identification prior to any funds transfer. The majority of current systems are based on using a bank card and a PIN (personal identification number). This system is open to abuse as transaction can take place using just the card number and expiration date.

Using a fingerprint based identification system can simplify user interaction within a secure environment. A single fingerprint scan for verification against a known template replaces the need to remember passwords or authorisation codes which makes it easier for users accessing the system requiring user authentication [7]. Fingerprint applications require a user interface to capture the image; this image is then compared against a known sample. A verification type system that only compares the individual against a single sample greatly reduces the time required over a database system. This thesis will focus on a verification based system to simplify the overall design.

2.1 Representation

Fingerprint representations can be categorised into two types; global and local. Local representation consists of several components within a restricted region in the fingerprint unlike global representation that takes an overall attribute of the fingerprint [1, 23].
Representations of the fingerprint using local information are based on the ridges or features derived from the ridges. A minutiae based automatic fingerprint identification system locates the minutiae (feature) points within the fingerprint and then matches the relative placement of these points to a stored template [36]. Figure 2.1 shows the matching of minutiae between two separate fingerprint images; the acquired fingerprint image on the left and the stored fingerprint image (template) on the right.

![Figure 2.1 - Minutiae Matching [27, 28]](image)

Fingerprint images of high quality contain 40 to 60 minutiae, but this varies between fingerprints and acquisitions of the same finger. Graph based representation constructs a nearest neighbour graph from the minutiae patterns and uses graph matching techniques [5]. Correlation based techniques use the gray level information within the fingerprint as features to match the global patterns to determine if the ridges align [41].
Global representation is typically determined by examination of the entire fingerprint to obtain a single feature and is typically used for indexing purposes as it does not offer individual discrimination. There are usually a small number of categories (e.g., typically five) that can be effectively identified from the fingerprint [23]. This is one of the most significant global features is known as the fingerprint ‘class’. The class is the overall fingerprint pattern feature that is used for classification of the fingerprint (more detail in section 2.3).

2.2 Information

Major automatic fingerprint identification systems today use local information in the form of minutiae as the features that are used to identify an individual using their fingerprint. Pre-processing is a major element prior to extracting local information, and by removing this need reduces computational requirements. This thesis has focused on working directly with the gray scale fingerprint image taken from an electronic scanner, and extracting the features directly from the fingerprint by using pattern recognition.

Using a particular pattern recognition technique, the texture feature is able to be extracted from the fingerprint that can be used to identify a certain individual.

Figure 2.2 - Local Features (Minutiae) [22, 46]
2.3 Individuality

Fingerprints are fully formed at seven to eight months of foetus development and do not change throughout the life of an individual except due to environmental factors; i.e. damage to the fingerprint [48]. Genes determine the general characteristics of the fingerprint pattern but the finer details are affected by the environment during foetus development. These two varying factors between individuals affect the final fingerprint and make it virtually impossible for two fingerprints to be alike.

Fingerprint identification is based on two fundamental properties: (1) fingerprint details are permanent, and (2) fingerprints of an individual are unique [23]. The individuality problem can be defined as the probability that any two individuals may have similar fingerprints within the target population. In this thesis, the individuality problem is defined as the probability that two individuals have sufficiently similar fingerprints.

Fingerprints are defined as ‘sufficiently’ similar by most human experts and automatic fingerprint identification systems if they originate from the same source. The amount of similarity between fingerprints for a match depends on the variations between multiple impressions of the same finger. This thesis, given a sample fingerprint, will determine the probability of finding a sufficiently similar fingerprint within the target population.

2.3.1 Genetic Factors

Studies into characteristics of a fingerprint that are genetically inherited have shown significant correlation in the fingerprint class and other attributes of the fingerprint; ridge count, ridge width, ridge separation and ridge depth. In dermatoglyphics studies, unrelated persons have very little global similarity in their fingerprints, parent and children have some similarity, siblings have more similarity and identical twins share the most global similarity observed [48].
2.3.2 Environment Factors

Environmental factors have an effect on the fingerprint over the life of an individual. Damage to the original fingerprint (formed during foetal development) can arise due to cuts, bruises and scars. Scars form permanent damage to the fingerprint and in itself creates a new feature within the fingerprint. Damage to the fingerprint in the form of small cuts and bruising does not affect the underlying fingerprint and is only temporary.

2.4 Fingerprint Classification

Traditionally fingerprints are classified into categories based on the information contained within the global patterns of ridges. Classification of fingerprints has been typically used to remove the need to match an input fingerprint to the entire fingerprint database in identification applications. This greatly reduces the time required in searching an electronic database and computations requirements.

One of the most commonly used classification methods is the Henry system that classifies fingerprints into three main classes: loop, whorl and arch. Under this system these main categories are further divided to product a total of twenty classes. The FBI uses the Henry classification system but with a reduced number of classes: radial loop, ulnar loop, double loop, central pocket loop, plain arch, tented arch, plain whorl and accidental.

The five-class Henry classification system will be used in this research as favoured by most research institutions; whorl, left loop, right loop, arch and tented arch. Figure 2.3 shows the five-classes of fingerprints. Research institutes favour the five-class classification system as it is extremely difficult to design an eight-class or higher classifier with high accuracy. The five-class classification system is also preferred by NIST (National Institute of Standards and Technology) [31]. Commercial fingerprint identification systems typically use a five-class to eight-class classifier depending on the security and performance required.
Fingerprint classification systems cannot classify every fingerprint directly into each of these classes. It is a very difficult problem for both human experts and electronic fingerprint classifiers to reach a specific class for every fingerprint. Due to the fact that there exists crossover boundaries between classes, fingerprints can sometimes be placed in different classes depending on the human expert or electronic fingerprint classifier.

### 2.4.1 Approaches to Classification

Classification of fingerprints using a knowledge based approach requires a human expert to use rules for each category and hand-constructing the models. This has reported accuracies up to 85% on the NIST-4 database [22].

Information within the fingerprint that defines the classes is within the central part of the fingerprint called the *pattern area* [26, 51]. One of the biggest challenges for fingerprint classification algorithms is to extract information from low quality images.
Low quality images have limited information to locate the pattern area where the majority of the verification information resides. Knowledge based approaches use both core and delta points for classification, which are the two main locating points about the centre of the fingerprint as shown in Figure 2.4. This information must be present in the image for verification.

![Figure 2.4 - Fingerprint 'Core' and 'Delta' Locations [46]](image)

Another approach is structure-based, that uses the orientation field of the ridges as shown in Figure 2.5. This does not rely on either core or delta points that may not be present in low quality images but the orientation field can also be very difficult to detect with such a low amount of information available.

![Figure 2.5 - Orientation Field placed over fingerprint image [5]](image)
An ideal classification algorithm would use both approaches, depending on the quality of the image and adapt to the quality of the image available. Using a dual-method is out of the scope of this thesis due to the large complexity and will not be used.

2.5 Summary

As this thesis uses an electronic method of individual identification, the fingerprint itself must be captured in a digital format. The fingerprint must be captured in real-time to allow the system to identify the individual without any technical user interaction to simplify system operation. The factors outlined in this chapter that affect the quality of a fingerprint image must be taken into account when determining the ideal method to capture an electronic fingerprint image. The performance of the entire system relies heavily on the quality of the original fingerprint image that is captured in digital form; therefore research has been conducted to determine the most suitable electronic fingerprint sensor to be used for this thesis.
Chapter 3

3 Identification System

Fingerprint identification systems in development today are based on a number of sensor technologies and processing platforms. With more security options required for mobile phones and laptop computing which are constantly shrinking, a compact fingerprint identification system could be integrated into these current technologies. As part of this thesis, research has been conducted into various fingerprint sensors which form the main interface between the user and the technology. This sensor has a large impact on the overall performance of the system and must be selected appropriately so as not to impact the system. This thesis has also focused on implementing within an embedded microcontroller that is very cost effective, small and has very low power consumption; all items that must be addressed for this system to be used with other compact technologies. The selection of a fingerprint sensor for this thesis forms the basis of the hardware that is required and accompanying software that will integrate the final implementation of this fingerprint identification system.

3.1 Fingerprint Sensors

The traditional method of acquiring a fingerprint image is to take a sample using ‘inked’ methods of which there are three types: rolled, dab and latent. These methods are known as an offline process. For the rolled method, ink is applied to the finger and then rolled on paper from one side of the nail to the other to form an impression. In the dab method, ink is applied directly to the finger and then pressed onto paper without rolling. This paper is then scanned to capture the image digitally.
Latent fingerprints are taken directly from surfaces that have been touched by a finger which leave an invisible fingerprint formed from the sweat pores of the skin. The image can be acquired by dying the impression and then scan the fingerprint.

As this inked method is not suited to a verification system where the user must be verified within seconds, the online method is preferred where a live scan of the image is directly taken from the finger. Live scan sensors typically use the dab method where the finger is directly applied to the sensor area, and a complete image is captured electronically. The majority of fingerprint sensors in the market today operate at 500dpi resolution, which is the standard prescribed by the Federal Bureau of Investigation (FBI) [26, 51].

Fingerprint sensor technology is advancing rapidly due to large companies such as NEC, Fujitsu and Sony recognising this identification technique for security purposes. Various sensors have both advantages and disadvantages depending on the environment and requirement of the system. Capturing the image without distorting the fingerprint poses a challenge for fingerprint sensor technology companies, as any loss in fingerprint information has a large effect on the overall system performance.

### 3.2 Commercial Sensors

There are various commercial fingerprint sensors available to capture an electronic form of the fingerprint. Three main technologies are used in fingerprint sensors; thermal, capacitive and optical. Sensors had to be evaluated for their suitability as the input device to the system being designed and developed as part of this thesis. Due to funding constraints, testing was limited to a thermal fingerprint sensor that was provided by Unique Technology.
Thermal sensors measure the temperature of the fingerprint ridges and valleys to acquire the fingerprint pattern. This type of sensor requires the temperature sensor to be at a different temperature than the finger being copied otherwise the image becomes distorted [19]. Thermal sensors that are currently available require the user to swipe their finger across the sensor surface, which can produce image errors when constructing the complete fingerprint image [19].

The capacitive sensor is by far the most used sensor today [30]. This sensor simply requires the user to press their finger against the sensor surface. The sensor detects the pressure difference between ridges and valleys across the fingerprint surface, and this difference is measured as a change in capacitance. By measuring this change in capacitance across the sensor surface a complete fingerprint image can be obtained. These sensors acquire a complete image reducing the processing required by the system and also reduce distortion errors that can be obtained by using sensors that require the user to swipe their finger across the fingerprint surface.

Fingerprint images obtained using capacitive sensors are a better representation than those obtained with a thermal fingerprint sensor. This is due to the thermal sensor requiring the user to swipe their finger across the sensor surface which stretches the finger’s skin, and produces a drawn out fingerprint image. These sensors are extremely thin which makes them suitable for small equipment that requires user authentication. Errors can occur from using this type of system when users do not press their finger hard enough, or only place part of their finger across the sensor surface [20].

Optical sensors illuminate the fingerprint surface using a LED light that is reflected and captured by a CCD array. This type of fingerprint sensor is very large and not suited to portable applications. An image sensor is known as an ‘optical sensor’ as the fingerprint image is not directly obtained by the sensor surface.
These sensors are extremely robust and do not wear during normal operation unlike solid-state sensors. They have the benefit of obtaining a very clear image due to the acquisition process and capture the complete fingerprint image in a single scan similar to the capacitive fingerprint sensor. These sensors are not widely used in commercial applications today due to the physical size of the sensor, which limits the packaging options into portable equipment [30].

### 3.3 Fingerprint Sensor Selection

System performance is greatly affected by the fingerprint sensor used with the system [7]. There are two main sensor profiles available on the commercial market today, partial and full image [30].

A partial image sensor only takes a small sample of the fingerprint area, and requires the user to swipe their finger across the sensor surface to acquire all the fingerprint information. The advantage of this type of fingerprint sensor is smaller cost due to the smaller sensor area and also the device is very compact. As a complete image is required for the system, the image must be constructed from the samples taken. These samples are known as ‘slices’ and are usually the complete width of the finger but only a few pixels in height. This requires more processing time for the system, and can also introduce anomalies into the image if the user does not swipe their finger at a consistent speed or pressure [19].

Full image sensors use a large sensor array that takes a complete fingerprint image once the finger is placed on the sensor surface. By taking the complete fingerprint image, errors are reduced as the user interaction required is also reduced. Unlike partial fingerprint sensors, the user only has to place their finger on the sensor surface instead of swiping their finger across the sensor. By removing the need to swipe the finger, the images are clearer and the processing required by the system is also reduced. The system does not need to construct the image as the full fingerprint image is captured simultaneously [33].
In selecting an appropriate fingerprint sensor, the application and the system that the sensor will be interfacing to must be analysed. This thesis has focused on using an embedded microprocessor that will enable the ID system to be extremely compact. Such an embedded ID system has limited processing power and memory capacity compared to typical personal computers, and therefore the load that the sensor places on the system must be taken into consideration. A partial fingerprint sensor places additional load on the microprocessor as the fingerprint image must be constructed prior to any verification stages of processing. This type of sensor reduces the overall performance of the system, and therefore was not chosen to be used in this thesis. A sensor supplied by Infineon technology that uses a capacitive sensor surface and takes a complete fingerprint image has been selected for use in this thesis. This sensor has been specifically designed to be used with microprocessors, and uses a high speed interface that directly connects with the microcontroller being used in this thesis.

3.4 ID System Design

The ID system had to be designed to be very compact, have enough memory to store the software algorithms and high enough processing power to complete verification within a short amount of time (ideally within a few seconds). The fingerprint sensor also had to be extremely compact, and allow direct interfacing with a microcontroller to simplify the system design. As this ID system was being designed with portability in mind, devices with low-power requirements and standby modes were selected to keep power requirements to a minimum.
The fingerprint identification system has been broken down into four main components that perform the identification task; (i) Microcontroller that contains the processor to perform all the mathematical calculations required by the software, memory to store all data and controls the operation of the fingerprint sensor. (ii) Fingerprint sensor captures the fingerprint image directly from the user accessing the system and also determines if a finger is present on the sensor surface. (iii) External memory to store variables, program data, software and fingerprint features that represent individuals and System Interface to interconnect with other systems and for monitoring the operation of the system (i.e. prototype debugging; opening a door in commercial product). (iv) Software is stored within the microcontroller for security and contains all the instructions of the developed algorithms to control data flow, retrieve a fingerprint image from the sensor and extract features that identify an individual that would use the identification system.
3.4.1 Microcontroller

Selecting a microcontroller was determined by three main factors; the development tools that were available for software design, low cost with integrated interface for the fingerprint sensor and ability to secure the data within the microcontroller memory core. As funding was limited for this thesis, the development tools were supplied for use by Spot-on Technology. The company’s tools supported the entire range of Atmel AT90-series microcontrollers which are an 8-bit family with integrated memory and external interfaces. As performance is a key factor of the fingerprint identification system, the device within this family with the fastest processor and largest memory was selected.

Figure 3.2 – Atmel ATmega128 Microcontroller [4, 21]

The Atmel ATmega128 microcontroller contains a 16MHz 8-bit processor that includes 128k bytes of Flash available for program storage, 4k bytes of data memory available that can be expanded by 64k bytes of external as an option. The processor is able to execute a single instruction per clock cycle, enabling operation up to 16MIPS to improve processing performance.

The memory within the core of the microcontroller device also is able to be hard locked preventing any outside extraction of software algorithms, feature data or individual information. Interfaces within the microcontroller include an SPI (Serial Peripheral Interface) and SCI (Serial Communications Interface) that will enable any fingerprint sensor with either of these interfaces to be directly connected to the microcontroller, increasing throughput between these two devices and simplifying the hardware.
3.4.2 Fingerprint Sensor

Various sensors are available based upon the different technologies used to construct the sensors; capacitive, optical and thermal. As the fingerprint sensor is a critical component (information that is lost or not acquired at this stage can never be recovered) the sensor must have a resolution of 500dpi or higher and able to be directly interfaced with a microcontroller for this embedded system design. Manufacturers that supplied fingerprint sensors to meet these criteria were Veridicom, Infineon, Atmel and KSI.

Veridicom offer two sensors that are manufactured within the required specifications; FPS110 and FPS200. Both sensors are based upon capacitive technology and have a resolution of 500dpi. From these two fingerprint sensor solutions, the FPS200 sensor has some advantages over its younger counterpart; is more compact, has a low-power standby mode and has direct interfacing with microcontrollers as well as USB for testing. This sensor also incorporates a finger detect feature, which enables the microcontroller to be in standby until a finger is detected, and further reduces the power requirements of the system.

<table>
<thead>
<tr>
<th>Veridicom FPS200 Specifications:</th>
</tr>
</thead>
<tbody>
<tr>
<td>Size: 24mm x 24mm x 1.4mm</td>
</tr>
<tr>
<td>Power: 12uA standby, 30mA active</td>
</tr>
<tr>
<td>Image Rate: 30 frames/second</td>
</tr>
<tr>
<td>Image: 256 x 300 pixels</td>
</tr>
<tr>
<td>Resolution: 500dpi</td>
</tr>
</tbody>
</table>

Table 3.1 – Veridicom Sensor and Specifications [13]
Infineon offer a single fingerprint sensor based upon capacitive technology. This sensor is smaller than the Veridicom sensor that uses the same technology and also manages a higher resolution at 513dpi. It is an extremely fast sensor able to capture images 100 frames/second and also has an integrated interface for a microcontroller based system. The sensor has lower power consumption than the Veridicom sensor but does not have a low-power mode.

<table>
<thead>
<tr>
<th>Infineon Fingertip Specifications:</th>
</tr>
</thead>
<tbody>
<tr>
<td>Size: 18mm x 21mm x 1.5mm</td>
</tr>
<tr>
<td>Power: 10mA</td>
</tr>
<tr>
<td>Image Rate: 100 frames/second</td>
</tr>
<tr>
<td>Image: 244 x 288 pixels</td>
</tr>
<tr>
<td>Resolution: 513dpi</td>
</tr>
</tbody>
</table>

**Table 3.2 – Infineon Sensor and Specifications [33]**

Atmel offer an extremely compact fingerprint sensor based upon thermal technology. This sensor takes a partial fingerprint image and is evaluated within this thesis, as it is relatively new technology in this area [19]. This sensor has an interface that is directly compatible with a microcontroller based system [21].
KSI (Kinetic Sciences Inc.) offer an extremely compact optical fingerprint sensor. Similar to the Atmel sensor it operation, this optical sensor takes multiple partial images of the fingerprint image while the user swipes their finger across the sensor surface. This complete fingerprint image must be reconstructed by specific software for the sensor. Due to the limited information on this sensor and that fact the all optical sensors are larger than a similar featured capacitive technology sensor; this type of sensor will not be used for this thesis.

### Table 3.3 – Atmel Sensor and Specifications [21]

<table>
<thead>
<tr>
<th>Specification</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Size</td>
<td>25mm x 8mm x 1.0mm</td>
</tr>
<tr>
<td>Power</td>
<td>10mA</td>
</tr>
<tr>
<td>Image Rate</td>
<td>1780 frames/second</td>
</tr>
<tr>
<td>Image</td>
<td>8 x 280 pixels</td>
</tr>
<tr>
<td>Resolution</td>
<td>500dpi</td>
</tr>
</tbody>
</table>

Atmel FCD4B14 Specifications:
Table 3.4 – KSI Sensor and Specifications [30]

The selection of the fingerprint sensor will be decided after the experiments for the Atmel thermal fingerprint sensor have been conducted. Of all the fingerprint sensors that have been presented in this chapter excluding the Atmel, the Infineon fingerprint sensor offers the highest resolution, compact implementation and low-power and interfacing as expected for use within an embedded fingerprint identification system.
3.4.3 External Memory and System Interface

The microcontroller is self-contained computer that does have limitations that can be expanded upon by connecting external devices. Due to the limited memory available within the microcontroller for storage of information such as templates for individuals and fingerprint image, external memory must be added. This does present a security risk as now the data traffic is able to be monitored from outside the microcontroller core. This must be taken into account when implementing an entire fingerprint identification system to prevent a security breach of valuable information or algorithms.

The fingerprint identification system must also be connected to the outside world to provide a method to control other systems; for example opening a door and granting access to a computer system. This interface would also be required for a prototype that would allow monitoring of system operation and supply of diagnostic information during development. The microcontroller has two inbuilt interfaces that are able to integrate with most systems; both SCI and SPI.

3.4.4 Software

The software that controls the entire operation of the fingerprint identification system has been written specifically as part of this thesis. The software has been broken down into three core components that control the overall system operation; hardware control, image processing/feature extraction and individual verification.
Hardware control is the low-level control of the hardware operation; i.e. getting data from sensor, saving data to memory and prompting user for next step in process. This is a similar environment to an operating system and has to perform the necessary commands to move data and information within and external to the microcontroller. It must also include initialising all the devices connected to the microcontroller, and perform the functions as requested by the higher level components such as feature extraction. As creating an entire fingerprint identification system is out of the scope of this thesis, this component has not been implemented.

Verification is the ability of the system to identify an individual by their own fingerprint. This process requires the matching of an individual with a template store within the identification system. The development of an entire fingerprint identification system is out of the scope of this thesis, but the use of an artificial neural network within the embedded system has been explored and researched as part of Chapter 6. This approach would allow the system to tolerate differences that may be due to inaccuracies of the system as no system can be entirely 100% accurate.

### 3.5 Summary

Image processing and feature extraction is the process of taking a complete fingerprint image and extracting the features or relevant information that can be used to identify an individual. To aid in development of the image processing and feature extraction process outside of the embedded platform, a windows-based environment will be used to evaluate methods and to develop algorithms for use within this identification system.
Chapter 4

4 MATLAB Interface

Various software packages are available for writing programs for a windows based operating system but must not constrain themselves to a graphical user interface alone. The software package chosen for this thesis had to have the capability to develop a graphical user interface (GUI), strong mathematical library, neural network and data analysis libraries, image analysis capability and well proven in the academic field. As this thesis will form the basis of a complete fingerprint identification system, the software had to enable easy integration of the algorithms developed in this thesis to future expansion of this thesis with a neural network or similar verification algorithm. To meet these requirements various programming languages were looked upon such as Java, C, Perl and Visual Basic but none of these met the following requirements for this thesis; highly portable, demonstrated use within the university community, built-in libraries to perform image and statistic analysis and also allow code to be exported to hardware specific solutions more importantly embedded microcontrollers. MATLAB was chosen as the development platform as it is able to meet these requirements and also is not an extremely complex language to learn reducing the development time for the fingerprint identification software. MATLAB uses matrix specifically enhanced functions for efficient handling of large amounts of computational data (MATrix LABoratory). MATLAB has built in libraries or image manipulation, data analysis, neural network development and also enables the programs to be exported directly into an embedded microprocessor based system. MATLAB also has a very strong matrix based computational engine and is used extensively within the research field due to its stability, performance and large libraries which already implement many functions saving development time [40].
There are two main parts to the software required for this thesis and this is the GUI and Texture Feature Extraction and Analysis. These two parts form the basis for the fingerprint identification system; specifically for this thesis is the extraction of feature information from the fingerprint for future verification algorithm developments.

4.1 Benefits of the MATLAB Environment

MATLAB is an interpreted language that is very similar to the original programming language ‘Basic’. The program allows the use of the environment for single line expressions to be evaluated, or entire programs may be written and modified using the integrated development environment with debugger for finding program errors. The language is also very portable in the fact that it is supported for different platforms such as Windows and UNIX. As this is a complete development environment with both editor and debugger, this will enable easy expansion of this system in future research.

MATLAB was chosen as the preferred platform for this thesis as it provided benefits that included:

1. **Simplicity** – The code is easy to understand but more importantly the platform is still extremely powerful especially for complex arithmetic operations.

2. **Code Databases** – MATLAB is extensively used within research organisations of which many publish previous studies and code that can be used by others within their own projects.

3. **Code Libraries** – Built in libraries allow common procedures to be easily implemented using well-proven commercial code within users projects to perform frequent commands.

4. **PC Operation** – The project can be developed within a cheaper PC based environment prior to being used within the expected external hardware specific platform.
5. **Platform Export** – The code can be directly exported to supported hardware directly from MATLAB enabling developed projects to be easily converted to run on the target platform (not PC based).

6. **User Interface** – MATLAB includes an interface for creating custom graphical user interfaces to easily operate the software functionality for development purposes and also to create executable programs that can be directly run on a PC.

This thesis will heavily use the in-built toolboxes (libraries to perform grouped specific functions) for image handling, feature extraction and further analysis on the data obtained from the fingerprint images. This will greatly reduce time spent writing specific software modules and will be used as the building blocks in developing the entire fingerprint identification system.

### 4.2 User Control

The purpose of the GUI is to provide an easy to use interface to gather results and to operate the prototype fingerprint identification system components. The interface will be completely developed within the MATLAB user interface development environment. Implementing this interface will present the user an easy to operate, but also very powerful test program for evaluating the texture feature extraction method, and will be used heavily in this thesis for experiments. The interface is designed to integrate the data input output of the system, and also to send the input to the texture feature extraction algorithm, then return the output of the feature extraction algorithm to a useable output; for example as a data file.

Figure 4.1 shows the MATLAB interface that has been developed for the experiments that need to be conducted in this thesis. The interface enables fingerprint images to be opened from a standard image format (bitmap *.BMP, tagged image format *.TIF) shown visually to the user and stored in MATLAB as a matrix element. The interface enables the user to apply a filter to enhance the fingerprint image and also to save this newly filtered image shown in the right side of Figure 4.1 (the left-side shows the original fingerprint image).
The interface enables information within the fingerprint to be extracted and stored within MATLAB for further processing and also stored for later retrieval if required. Having this data within MATLAB enables the system to directly use this feature information without a secondary program for analysing the data.

4.3 Feature Information

Information used within MATLAB must be controlled in the way it is stored in handled within the MATLAB environment. The data that is handled by the DIO function is:

- Fingerprint Image (Input)
- Filtered Fingerprint Image (Output)
- Feature Data (Output)

For this thesis, the images are in standard windows bitmap (*.BMP) format that must be handled by the DIO function. The fingerprint image must be imported into the MATLAB environment used by the fingerprint identification system.
Once the image is imported into the MATLAB environment, this allows the feature analysis function to manipulate this data and allows a standard matrix format for the fingerprint image. This allows the system to have input from any format of fingerprint image, and store this within the MATLAB environment as a standard data matrix, removing the need for changes to the feature analysis function if a different type of image format is used.

Analysis of the feature information that has been gathered using the texture feature extraction algorithm is used to if this method of feature extraction is both repeatable and reliable, and also to remove unwanted information from within the fingerprint image such as noise.

4.4 Summary

Various methods are to be adopted to determine the most effective way to prove the theories in this thesis for using texture feature extraction. ANOVA (Analysis of Variance) will be used as well as PCA (Principle Component Analysis) for analysing the feature data. Using the MATLAB statistical toolbox, this analysis can be performed directly on the feature data extracted from the fingerprint. This is an important factor that will enable future research to include real-time analysis of the feature data in the MATLAB interface.
5 Feature Extraction & Matching

The nucleus of the fingerprint identification system is the feature extraction approach presented in this thesis that uses global texture information unlike traditional local information in minutiae-based identification methods [1, 18]. Solid-state sensors as selected for use in this thesis have a limited contact area with the fingertip and therefore only sample a limited proportion of the fingerprint. This limits the number of minutiae that can be extracted from the fingerprint and as such limits the number of common singular points between two impressions of the same fingerprint. The application of the global textures method of feature extraction will be tested on a variety of input images; using the original un-modified fingerprint image, filtered image and also the benefits of compressing the fingerprint image for use within the embedded system. This chapter outlines why this approach has been taken to find features within the fingerprint and how this will be used within the fingerprint identification system.

5.1 Image Analysis

Due to budget limitations for this thesis images have been sourced from an online database from Neurotechnologija Ltd. online (www.neurotechnologija.com) which are available for public use. The images were selected as they meet the FBI standard for minimum quality (500dpi 8-bit gray scale images) [26] and have been acquired using a DFR-200 fingerprint scanner. This particular fingerprint sensor captures an image size of 360x364 pixels, and uses optical technology to capture the fingerprint image. The small database contains 40 individuals fingerprints, each of which five successive scans have been taken.
The images are identified by the number (prefix) and the scan number (last digit). For example, a user identified as 2099 has the corresponding fingerprint images 20991, 20992, 20993, 20994, and 20995. The images are standard Windows bitmap format (*.BMP) which enables the MATLAB interface developed to directly import these images in the data matrix. The process that will be used in this thesis is shown in Figure 5.1; this outlines the method that can be used to extract the features from the fingerprint image. The coloured arrows show alternate approaches that may be used if experimentation finds using these paths more efficient when implemented in an embedded system.

Experiments will be conducted to determine which method provides the most efficient approach for extracting features using the texture feature approach. Ideally, the system will be able to extract features from the un-processed fingerprint image directly as acquired without the need for filtering and compression.

Figure 5.1 – Feature Extraction Flow Chart
5.1.1 Texture Features

A fingerprint can be viewed as an oriented texture and for sufficiently complex orientated textures such as fingerprints; invariant texture representations can be extracted [18]. Using this underlying structure within the fingerprint does not rely upon locating the core or delta points that is difficult to detect or not present in low quality images. Texture features that are extracted can be categorised as; statistical, geometrical, structural, model-based and signal processing features [52]. In image analysis there are various methods available to extracting these texture features, and must be understood in selecting the optimal method for system being used.

5.1.2 Texture Methods

Unlike traditional local feature extraction methods (i.e. minutiae extraction) from the fingerprint image, this thesis focused on using a texture feature extraction method. Five popular texture feature extraction methods were selected as candidates for use in this thesis; autocorrelation, edge frequency, primitive-length, law’s method and co-occurrence matrices.

Autocorrelation is a function that evaluates the linear spatial relationship between primitives. The texture character of an image depends on the texture primitives; large primitives give rise to coarse texture and small primitives give fine texture. If the primitives are large the function decreases slowly with increasing distance where if the texture consists of smaller primitives the function decreases rapidly. If the primitives are periodic then the autocorrelation function also increases and decreases periodically with distance.
Co-occurrence matrices texture features use second order statistical methods to model the relationships between pixels within the region specified by constructing Spatial Gray Level Dependency (SGLD) matrices [11]. This matrix is the joint probability occurrence of gray levels $i$ and $j$ for two pixels with a defined spatial relationship in an image.

This spatial relationship is defined in terms of distance $d$ and angle $\theta$. Therefore if the texture is coarse and distance $d$ is small compared to the size of the texture elements, the pairs of points at distance $d$ should have similar gray levels. Also for a fine texture, if distance $d$ is comparable to the texture size, then the gray levels of points separated by distance $d$ should often be quite different [15]. The SGLD matrix represents the texture features that are found within the image analysed.

Edge-frequency based texture features use a number of edge detectors to yield an edge image from an original image. Primitive length texture features represent coarse textures by a large number of neighbouring pixels with the same gray level, whereas fine textures are represented by a small number of neighbouring pixels. Each primitive is defined by its gray level, length and direction that is made up of a continuous set of the maximum number of pixels in the same direction and that have the same gray level. The individual primitives represent the features within the image.

Law’s method is based on a series of pixel impulse response arrays obtained from combinations of 1-D vectors shown in Table 5.1 [47]. These five 1-D filters are capable of producing 25 possible features at each pixel location (using five different filters in both horizontal and vertical directions). The arrays are convolved with other arrays in a combinational manner to generate the masks generally labelled L5L5 for the mask resulting in the convolution of the two L5 arrays. The arrays are based upon the observations of Laws that found certain gradient operators such as Laplacian and Sobel operators accentuated the underlying microstructure of texture within an image.
Level $L_5 = [1 4 6 4 1]$

Edge $E_5 = [-1 -2 0 2 1]$

Spot $S_5 = [-1 0 2 0 -1]$

Wave $W_5 = [-1 2 0 -2 1]$

Ripple $R_5 = [1 -4 6 -4 1]$

| Table 5.1 – The five 1-D arrays as identified by Laws [53] |

Studies that were performed into these methods [15] determined that the co-occurrence matrices and Law’s method perform better than other techniques. Due to the higher computational requirements that co-occurrence matrices require [15], Law’s method was selected for this thesis as the fingerprint identification system is limited in memory and computational power.

5.2 Extracting Features

The performance of fingerprint feature extraction relies heavily on the quality of the input fingerprint images. Poor quality images do not have well defined ridge structures and loss of global information that is required by feature extraction methods. To ensure performance of the feature extraction algorithm, enhancement of the fingerprint image is required to improve the clarity of the fingerprint ridge structures.
5.2.1 Law’s Method

Law’s method is based upon the application of five pixel impulse response arrays to an image which accentuates the underlying microstructure [15]. As there are five texture masks that can be used as outlined by Laws, this thesis will conduct experiments to determine the best mask combination to improve the structure of the fingerprint (ridges and valleys). In this thesis, Law’s method will be applied to the complete fingerprint image to accentuate the fingerprint structure that is weak within the fingerprint database due to the image capture method. Once the texture information within the image has been improved, the next stage in the extraction method will allow feature information to be found within the fingerprint.

5.2.2 Image Partitioning

Using the enhanced fingerprint image, the image is then partitioned into sections and each individual section is normalised to a constant mean based upon the pixel intensity. This partitioning removes the need for a reference point to be extracted from the fingerprint image and provides a method that will enable feature information to be extracted directly from these individual sections. The partition size will be selected to reduce the requirements of the verification algorithm; the more sections will require a more complex algorithm. From previous research, it has been stated that a partition size of less than 20x20 pixels results in limited or loss of texture information [47]. The partitioning function is performed within the MATLAB environment and enables the section size to be selected by the user as depending on the verification algorithm; the number of features available may need to be modified. A partition size of 60x60 pixels has been selected for this thesis as it meets the requirements of [47] and also reduces the number of inputs to the verification algorithm to 36 features. The partition sections remain in gray scale (8-bit) format to retain the texture information prior to extracting the feature information.
### 5.2.3 Feature Value

The section size defined by the partitioning defines the number of features that can be extracted by this system. The sections of the fingerprint image contain 360 pixels with a range of 0-255 (gray scale). Using the pixel intensity value, the feature value calculated as the mean value of 360 individual pixels within the section. This provides a single feature value for each of the 36 sections that have been partitioned from the fingerprint image. As the image is not pre-processed all information is still available within the pixels intensity.

### 5.3 Fingerprint Matching

The purpose of the matching a fingerprint is to determine if the fingerprint identification system is able to match an individual that has been enrolled on the system with a new copy of the fingerprint that is provided by the fingerprint sensor. Although the implementation of a verification algorithm is out of the scope of this thesis, research has been conducted to determine the best possible method that may be used for verification within an embedded processor based system. Due to the complex nature of fingerprint verification, but also the simplicity of the underlying system hardware, neural networks provide the ideal interconnection to provide intelligent verification and matching methods with simple mathematical functions.

The expected requirement for the verification algorithm is to be based on a feed-forward neural network with back-propagation to reduce the training time required. There would be three layers required; input, hidden and output layer that make up the feature detection of the neural network. The input layer would have to contain 36 neurons to cater for the 36 individual partition feature values, a 10 neuron hidden layer and single neuron output layer to make the final decision for either no matching feature found or a valid feature has been matched.
The network would be trained using the available NIST fingerprint image databases, as this would enable the performance of the system to be directly compared against other systems that have been tested using this publicly available fingerprint database.

5.3.1 Neural Networks

Neural networks use simple mathematical functions to perform complex tasks. By using a neural network that has been trained using appropriate data, this allows the programmer to construct near perfect approximations to systems of which limited information is available. Pattern recognition requires complex modelling using traditional ‘rule’ based computing solution, and this is where neural networks are suited because they are far less complex to model compared to traditional computing and they are able to be trained so that the results are consistent; the same input pattern will produce the same answer on the output [2, 55].

5.3.2 Simple Neural Network Model

A simple neural network may consist of one input neuron (input layer), three hidden neurons (hidden layer) and one output neuron (output layer). Between neurons, the connections are ‘weighted’ which describe the effect that each simple function within the neuron will have on the entire neural network [2].

![Simple Neural Network Model](image)

*Figure 5.2 – Simple Neural Network Model [2, 44]*
A neuron is a simple function that sums the weighted inputs together then passes this through the activation function. The activation function determines the output of the neuron and the degree to which the neuron affects the higher neurons.

Each input to the neuron is multiplied by its ‘weight’. The weight for each input is determined by ‘training’ the neural network. Training is the way neural networks learn which is a statistical model of the data that involves using a training set of data, and calculating the error at each output unit and changing the values of the weights which lead to the error [2]. This weight is then changed so that this error would be smaller if the same input was applied again. Once each of the inputs to the neuron has been passed into the summation function where each of the weights is multiplied by the corresponding input, the output of the function is then passed onto the activation function. The activation function is the key to the neural network operation and determines the output of the neurons in the network. This function takes the output from the summation function and calculates the output of the neuron. There are a variety of activation functions that can be used in the neuron, but the following are the most common:

- **Linear**: The linear function is a very simple activation function. It is very fast but due to its simplicity is unable to solve many problems. The output of this function is between –1 to 1.
- **Threshold**: The threshold function is a simple threshold that when the input is below a certain value, the output is minimum, and when the input is above a certain value the output goes to maximum. The output is limited to between 0 and 1.

- **Logistic**: The logistic function is very powerful and very popular. It has two main qualities: it squashes the input into the range from 0 to 1, and the derivative makes very small changes at each end of the range and larger in the middle.

- **Bipolar Logistic**: The bipolar logistic function is also very powerful and popular as like the logistic function, and has the benefit of a range from –1 to 1 which decreases learning time with the ability to output negative values [44].

The logistic function was chosen as the activation function for this project as it is extremely powerful and also executes very quickly in a microprocessor based system.

### 5.3.3 Developing a Suitable Neural Network

Neural networks are extremely powerful when created successfully, and selecting the type of network is very important for overall network performance in solving the problem. The task of the neural network is to use the available feature values provided by the feature extraction algorithm, and to match these with stored templates that identify individuals within the system. The outline of the task definition, design and feasibility of an embedded neural network algorithm has been proposed based the ideas in [2].
The task definition for this particular neural network will be to match the features extracted by the system to stored features that represent an individual enrolled on the system. This task will need to determine if features that match are sufficient enough to determine a valid match for an individual. Once the system has matched valid features to a user, this will enable the system to grant access to the individual to the required output of the system (i.e., open a secure room).

The neural network to be used for this thesis is a feed-forward network with back-propagation for faster training. The network will be a 3-layer network with an input layer, hidden layer and output layer. The network will be used for feature detection and classification. The neural network design will take into account that it will be used within an embedded microcontroller that has limited memory and processing resources.

The feasibility of the system will be determined by the size of the neural network that has been limited by the number of features extracted to reduce the complexity and processing requirements. This can only be determined once the neural network has been developed and evaluated within the MATLAB environment and further testing carried out within a prototype embedded fingerprint identification system.
5.3.4 Matching Template

They key to the system is the ability to store a representation of an individual that can be used at a later time to validate the fingerprint acquired by the sensor. This template embeds the features that have been extracted from the fingerprint image that is stored in the system as a template. This template is created when a user is enrolled into the system; the process of acquiring new person’s features and cross checking these against multiple scans to verify a set of features that will form their template. This is stored within the fingerprint identification system for security and allows this template to be retrieved at a later date to verify the user.

To allow the system to be stand-alone a smartcard would allow the individuals template to be carried with them and not require them to enrol at every single fingerprint identification station and remove the need to connect to a central database containing users templates. This would only require the system to match the features taken from the fingerprint sensor to the features that are available within the smartcard to provide verification of their identification.

5.4 Summary

Using the enhanced fingerprint image, texture features can be extracted by partitioning the fingerprint image into smaller sections. This method allows a feature value to be extracted based on the texture component of each partition. The feature values will be known as the texture features of the fingerprint. The number of features that are extracted from the fingerprint affects the size and complexity of the verification algorithm and must be taken into account when designing the partition size as well as the verification algorithm. For this thesis, the partition size selected is 60x60 pixels which results in 36 individual features that are extracted to represent a fingerprint.
A neural network based verification system will enable the simple embedded processor system to perform complex tasks without the overhead of performing complex floating-point calculations. The network is able to use simple mathematical functions for verification of the fingerprint and its features. Also, various sections of the network may able to be implemented within a multi-processor embedded network, to improve the speed of the identification system by increasing processing power while still using commercially available embedded processors that are cost competitive. As this system has been based upon the MATLAB environment, it is expected that using the inbuilt neural network capability will allow hardware specific code to be directly exported and also to complete development within MATLAB to reduce the cost of commercial development time.
Chapter 6

6 Experimental Results

The fingerprint identification system was based upon various subsections that complete the operations as required by electronic fingerprint identification. The system was identified by the inputs, processing and outputs required in completing the identification process. Research was conducted and identified the following areas that are the primary focus of this thesis:

- The most appropriate hardware for real-time fingerprint identification
- Most efficient algorithms for use within embedded processor platform
- Performance of the texture feature extraction method

The experiments were based upon the focused areas and took into consideration the limitations that are present within the system presented by this thesis. Various approaches that are included within a typical fingerprint identification system have been tested to determine if implementation of these approaches is of benefit to an embedded-based fingerprint identification system. Experiments have been carried out to determine the best hardware and software approaches and have been outlined within this chapter.

6.1 Fingerprint Identification Hardware

As the most critical link in the fingerprint hardware, the fingerprint sensor must acquire the fingerprint image without loss of information or introducing new information that was not part of the original fingerprint image known as noise. Research was conducted to find testing that has already been completed on various fingerprint acquisition technologies but it was found that new thermal fingerprint sensors do not have readily available performance information.
Therefore testing was completed on the newest technology available for fingerprint acquisition; the thermal fingerprint sensor. Thermal technology gives several security advantages against older fingerprint technology (Optical & Capacitive). The thermal acquisition means that the finger must not be a replicated copy as it will not have the same thermal differences between the ridges and valleys of the fingerprint as found on a human finger.

The equipment used for the experiment was an Atmel Thermal Fingerprint Sensor ‘Fingerchip’ which was generously supplied by Unique Technology. Using this sensor allowed testing to be completed that would allow this new technology to be compared directly against the various other fingerprint sensor technologies available. The sensors available in the market were rated against several criteria; resolution, image size, interface, size, type and cost.

### 6.1.1 Thermal Sensor Testing

Commercial interest showed that a very compact fingerprint verification system with high security was required. One of the main concerns of clients was the ability to copy the fingerprint and use this as a template for breaking into a system. This sensor is extremely compact, and uses an array of temperature sensors to acquire the fingerprint. The benefits of this sensor are the temperature component that is required for acquiring the fingerprint image. This increases security as not only must the fingerprint image be re-produced but also a temperature difference must also be obtained for an attempt to re-produce the fingerprint. This sensor is extremely compact due to the small sensor area and is ideal for small equipment such as mobile phones and laptop computers where size is a critical consideration for the final product.

![Figure 6.1 - Atmel Thermal 'Fingerchip' Sensor [19, 21]](image-url)
The Atmel sensor to be used within an embedded system incorporates a direct interface to be used in conjunction with an embedded processor. To complete testing of this thermal fingerprint sensor a USB version that could be directly connected to a PC allowed quick evaluation of the sensor operation and using the accompanying ‘Sweepee’ software package. The sensor operates by taking multiple images in succession as the finger is being swiped across the sensor surface. The software must align all these images known as ‘slices’ together to form the complete fingerprint image prior to user verification. This package allows the acquired fingerprint image to be viewed and also view the individual slices of the fingerprint that are taken during the swipe of the finger.

![USB Fingerprint Sensor](image)

**Figure 6.2 - USB Fingerprint Sensor [25]**

These multiple slices of the fingerprint must be reconstructed to form the fingerprint image and the extra load placed on the embedded system must be taken into consideration due to the large loading required by the processor to reconstruct the fingerprint image. In a microcontroller based fingerprint identification system as to be used in this thesis, using such a sensor will introduce a large overhead in processing requirements before the feature extraction stage of the system.
Another method which could be used to reduce the amount of processing time required would be to analyse the incoming slices in real-time without the need to re-construct the fingerprint image. This does require more processing power which is very limited in a microcontroller based system.

The re-construction test was completed using the Atmel Sweepee software on a PC running windows environment. As can be seen in Figure 6.3, on the right of the figure are the multiple slices that were taken from the fingerprint, and on the left of the figure is the re-constructed fingerprint image.

![Figure 6.3 - Fingerprint slices and re-construction](image)
6.1.2 Thermal Sensor Results

After testing the Atmel thermal fingerprint sensor problems have been identified that can arise from using this type of sensor. The majority of errors that are present in the scans are from the requirement of the user to swipe their finger across the sensor surface in a uniform manner so that the finger is not distorted during the swipe. This can skew the fingerprint image so that the image is no longer in sync and can introduce repetitive errors into the fingerprint as shown in Figure 6.4:

![Figure 6.4 – User Skew Error [20]](image)

After multiple uses in succession, the sensor surface can heat to the same temperature of the finger being swiped and this can lead to parts of the fingerprint image being missed as no temperature difference is measured. This problem cannot be compensated for in software, as the sensor surface is unable to determine the actual fingerprint temperature difference, and can only be compensated for by making sure the temperature of the sensor is lower than the user’s fingerprint or by heating the sensor using the inbuilt heater. This problem is shown in Figure 6.5; the edges and the middle fingerprint are lost due to the fingerprint sensor surface heating to a similar temperature to the fingerprint being scanned.
Capture errors can occur during the acquisition of a fingerprint image where slices of the fingerprint are lost due to user error swiping their finger too fast or at an inconsistent rate. This can also be introduced by the slice matching algorithm not matching the slices correctly (part of the Atmel Sweepee software package). This problem is shown in Figure 6.6 and is unacceptable for user identification as large parts of the fingerprint are lost along with identifying information that may have been present within these slices.
These problems that can occur from user-interaction with the Atmel fingerprint sensor can be reduced by careful design of the fingerprint sensor housing to help guide the finger as it is being swiped across the sensor surface and also to integrate helpful information notes for the user; to let them know if they are swiping their finger too slow or fast and prior education on the use of this fingerprint sensor technology.

These disadvantages of the thermal fingerprint sensor are also accompanied by several advantages that make this sensor a viable option for a fingerprint identification system; simple integration with a microcontroller based system due to the onboard interface, compact size for portable products, high resolution and is very cost competitive with current fingerprint sensor technology. These features will be used in the selection of an appropriate sensor for this thesis.

### 6.1.3 Fingerprint Sensor Selection

The interface between the user and the system is a critical component; any information that is lost or not acquired at this stage can never be recovered. The user-system interface must provide the complete fingerprint image without distortion or introduction of noise. The selection of this sensor was based upon a selection criterion that has an affect the overall system operation and performance.

<table>
<thead>
<tr>
<th>Sensor</th>
<th>Resolution (dots per inch)</th>
<th>Image Size (pixels)</th>
<th>Interface</th>
<th>Type</th>
<th>Power (mA)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Thermal</td>
<td>500</td>
<td>8 x 280</td>
<td>SPI</td>
<td>Partial</td>
<td>20</td>
</tr>
<tr>
<td>Capacitive</td>
<td>500</td>
<td>256 x 300</td>
<td>SPI</td>
<td>Complete</td>
<td>30 (12uA)</td>
</tr>
<tr>
<td>CCD</td>
<td>500</td>
<td></td>
<td></td>
<td>Complete</td>
<td></td>
</tr>
</tbody>
</table>

**Table 6.1 - Sensor Selection**

As all sensors that were review had a resolution of 500dpi (recommended by FBI as the minimum resolution for fingerprint image reconstruction).
The amount of processing requirements of the sensor was a critical component that had to be taken into account in selecting the sensor, as was the size and power requirements. The thermal sensor that was tested in this thesis (due to being new fingerprint sensor technology) was found to have a very high processing requirement due to the method it uses to capture a complete fingerprint.

The CCD camera although with similar specifications to the capacitive sensor, is not very compact and has a higher power requirement and does not have an integrated microcontroller interface. Due to the capacitive sensor being extremely compact also (only 1.4mm thick), very low power requirements and a microcontroller interface, this made the sensor an ideal candidate for use in the fingerprint identification system being designed as a part of this thesis.

6.2 Image Filtering

Filtering the fingerprint image is used to enhance the information within the image that relates particularly to this thesis; i.e. the feature information present within the fingerprint. The aim of filtering the image aims to enhance the ridges and valleys of the fingerprint that aims to improve the features extracted from the fingerprint and therefore provide a more robust feature extraction method. As Law’s texture filter is an algorithm that does not have a large processing requirement and is ideally suited to filtering fingerprint images, this filter has been chosen for experimentation of performance as part of this fingerprint identification system and to determine if implementation within an embedded system is feasible.

6.2.1 Filter Response

To test the theory of the best performance enhancement of the fingerprint image, each filter was tested on a single image from the database (20991.bmp). Figure 6.7 shows the fingerprint image in its original format. This image was filtered using the five separate Law’s filter masks; Level, Edge, Spot, Wave and Ripple in the horizontal direction only to determine the performance of the filter for fingerprint image enhancement.
As the level filter mask is used for smoothing information within an image, this actually removed all fingerprint information from within the image but did locate surrounding irrelevant information that can be removed from the fingerprint image. Figure 6.8 shows the fingerprint image after the Level filter mask has been applied to the image.
This filter will be used in this thesis to remove any irrelevant information surrounding the fingerprint that cannot be used in the feature extraction process. Subtracting this Level texture filtered image from the original image will remove the irrelevant information while preserving the fingerprint information. This process is known as segmentation where surrounding and background information that is not part of the fingerprint is removed from the fingerprint image. This information if not removed may introduce its own features into the fingerprint image creating irrelevant features.

The Spot, Wave and Ripple filter masks were not expected to enhance the ridges as no attributes of the fingerprint are Spot, Wave or Ripple features. As the fingerprint ridges are closest to a Ripple feature, this filter mask provided the largest enhancement out of these three filters.

![Figure 6.9 - Spot texture filter (20991.bmp) [12]](image)
The Edge texture filter mask is used for enhancing edges within an image, and as can be seen from Figure 6.12, the fingerprint ridges have been enhanced and are now more prominent than in the original fingerprint image.
Out of the five Law’s texture filter masks, the Edge filter mask provides the best performance for enhancement of the ridges within the fingerprint image. Filtering the images prior to feature extraction using this filter will allow a rich feature set to be extracted from the fingerprint images.

6.2.2 Law’s Texture Filter

Using Law’s Edge texture filter mask enables the enhancement of the fingerprint ridges within the images, and can be applied both horizontally and vertically to extract separate texture features. From Figure 6.13, the differences can be clearly seen between the two applications of the Edge filter mask. On the left of Figure 6.13 is the horizontal application of the Edge filter mask and it enhances the vertical ridges within the fingerprint image but also introduces noise across the horizontal ridges in the lower part of the fingerprint image. On the right of Figure 6.13 is the vertical application of the Edge filter mask and it enhances the horizontal ridges within the fingerprint image, but also introduces noise across the vertical ridges.
This noise that has been introduced is an unwanted artefact that destroys the feature information within the fingerprint. As can be seen in Figure 6.14, the horizontal application of the Edge filter has destroyed the ridge contours (feature information) that are running horizontally in the fingerprint image.

The vertical application of the Edge filter has also destroyed the ridge contours running vertically in the fingerprint image which is shown Figure 6.15. This reduces the amount of valid feature information within the fingerprint image and also introduces false features into the fingerprint.
6.2.3 Filter Selection

The final filter mask that was selected for this thesis is $F = E_5E_5$. The combination of both Edge filter masks across the fingerprint image produces the highest enhancement of ridges (global information) without the destructive loss that was found to happen when using the filters individually.

6.3 Image Compression

To reduce the amount of information required to be stored within the fingerprint identification system, the images taken of a fingerprint can be compressed so that only relevant data is retained and information that is not of significant relation to the identification of the fingerprint may be discarded. This is known as ‘compressing’ the image as the amount of data that can be used to represent the original fingerprint is reduced. For a microcontroller based system, this is ideal as the storage capacity is limited and can become expensive. This is proven by the FBI which has to store an almost unlimited number of fingerprint images that account for all the criminals which have been identified during their adoption of fingerprint identification for humans.
6.3.1 Wavelet Compression

To determine if fingerprint image compression is valuable to the fingerprint identification system, a wavelet approach of image compression is to be used to evaluate if it will provide an advantage to the overall system performance. Wavelets were chosen as the method for fingerprint compression due to their tried implementation with fingerprint technology in research, and as MATLAB has a wavelet toolbox that enables direct manipulation of the images already acquired [45]. The wavelet method of fingerprint image compression allows higher compression without the introduction of blocky artefacts that can be introduced with simpler and more common compression methods such as the JPG format. Figure 6.16 shows a fingerprint image that has been compressed to a ratio of 15:1 using the JPG format and the resulting artefacts that destroy the feature information within the fingerprint.

![Figure 6.16 - JPG Compression [60]](image)

This same fingerprint image that is compressed using a wavelet-scalar compression algorithm that was developed by the FBI for their criminal database shows that this method of compression does not introduce artefacts into the image as shown in Figure 6.17.
Benefits of using fingerprint image compression is the reduction in amount of data that is required to represent an individual, which in turn reduces the amount of computational data that must be processed by the embedded microprocessor. The wavelet method of image compression allows the algorithm to focus on the relevant information within the fingerprint image that contains the features that are of interest to identification and provide a subset of the fingerprint that contains specific data that can be used for identification with minimal loss of feature data [34]. The amount of compression can be varied using the wavelet method of image compression, but there is a compromise between image compression and before loss of relevant feature information that may affect the system performance [45, 60]. The aim is to determine if a reliable wavelet compression algorithm can be achieved within an embedded microprocessor platform.

**6.3.2 Compression Results**

A compression algorithm was developed using MATLAB and an open-source wavelet toolbox that can be used within MATLAB. Images were compressed and then re-constructed using the algorithms developed which allowed the original and re-constructed images to be compared for deviation and loss of information.
To simplify the problem, the compression rate was reduced to 5:1 and tested across a sample of 100 fingerprint images from the NIST-4 database. Visually, the images had minimal loss and no distortion or blocky artefacts as can be introduced by other compression techniques. The deviation between the compressed and re-constructed images was also measured as an average across the 100 images that were used in this experiment and found that the maximum deviation between images was 15% and average deviation was 11%. This enables the algorithm to reduce the original fingerprint that is taken with the hardware from approximately 150kB down to 30kB. This is a very large reduction for an embedded system and will allow 5 times as much data to be processed by the microprocessor, effectively speeding up the device by five times.

As the compression of the image itself requires processing, this also must be calculated to determine the amount of processing time required to compress the image. The experiment was carried out using a windows based computer with an AMD1800+ processor. The highest time measured to compress and re-construct the image was 2.5 seconds, the average being 1.8 seconds of computational time. The embedded microprocessor selected for this thesis runs at 16MHz, unlike the AMD1800+ processor that runs at 1.53GHz. The embedded microprocessor runs at a 1:95.6 ratio compared to the windows based computer and would take 57.8 seconds to compress a single fingerprint image. This large amount of processing time is an extremely large penalty to pay to compress the fingerprint image.

6.3.3 Compression Discussion

The large amount of computational time required by the wavelet compression algorithm was not expected. Due to the large time required by the compression algorithm, this will not be implemented within the fingerprint identification system. The benefit of reducing the amount of data to be processed by 5:1 does not outweigh the 1:95.6 processing time penalty that is paid by implementing the compression algorithm. As the features are extracted from the fingerprint image as part of the identification system, these features will be used as a representation of the individual and are already a ‘compressed’ or more compact representation of the original fingerprint.
6.4 Feature Extraction

Within the fingerprint image, the features that identify the individual must be extracted reliably and without loss of information. These features can be extracted in a number of methods; one of the most widely used in commercial systems today is minutiae features. This thesis has proposed to use a relatively new method of feature extraction, by extracting texture information from within the fingerprint that will be used as the identification of the individual. The benefit of using texture features is the information is able to be extracted directly from the fingerprint image, without the need for image pre-processing that can add heavily to the system’s computational requirement.

6.4.1 Extracting Texture Features

Using the feature extraction interface (FEI), feature values were extracted for the 36 partitions of the fingerprint. As can be seen from Figure 6.18, a feature value has been extracted for each partition, corresponding to the fingerprint image.

![Figure 6.18 - Feature Plot (20991.bmp) [12]](image)

On initial inspection of the feature plot, there is a row of large feature values at the top of the plot. This corresponds to noise that is present within the top of the fingerprint image. This noise is not part of the fingerprint and introduces erroneous information into the feature extraction process.
On closer inspection of the fingerprint images, this is a shadow that is cast from the fingerprint sensor and therefore requires a better segmentation scheme to remove the fingerprint away from this unwanted information. One method that may be used instead of image pre-processing to remove this unwanted information is to have a threshold for feature values that exceed a certain value. Selecting a suitable threshold for which noise is above, and fingerprint features below, will enable the system to discard unwanted noise features and keep the fingerprint features. As can be seen from Figure 6.18, all features above 50.0 are noise, and below 50.0 are fingerprint features. Using this knowledge, a threshold can be used to remove any unwanted features prior to the verification process.

6.4.2 Statistical Analysis

To determine if the texture extraction method can reliably extract feature from fingerprints, statistical analysis was performed on the data that compiled from the fingerprint image database. Analysis was completed using ANOVA and PCA statistical analysis methods that can be performed within the MATLAB environment.

6.4.3 Texture Performance

The performance and reliability of the texture feature extraction method had to be determined and proven using statistical analysis. Firstly and foremost the fingerprints within the small database that would be present within an embedded were analysed to determine if there was sufficient difference between the feature information to identify an individual. Figure 6.19 shows the analysis of the fingerprints for dissimilarity between the features using PCA.
As can be seen from the plot, there is significant difference between the 22 individual fingerprints. This would allow a verification algorithm to distinguish between individuals enrolled on the system and therefore proves that texture features could be a method for fingerprint identification.

To further determine if this feature extraction method is possible for identification between individuals, the similarity between successive scans of the same fingerprint must also be compared to determine if the feature values are significantly similar between the scans. This was completing using ANOVA to test the variance between the five successive scans of the same fingerprint and is shown in Figure 6.20.

**Figure 6.19 - Plot showing difference between individual fingerprints [12]**
The largest error that was found for this particular fingerprint is 0.02445% between the five successive scans. This shows that the fingerprint features are significantly similar between scans of the same fingerprint. This allows the identification of an individual at different times using the system. The texture feature approach is not very processor intensive as it takes less than 0.1 seconds using the AMD1800+ processor which is ideally suited to an embedded processing identification system. This approach will be further evaluated for performance and reliability as a commercial product.

Figure 6.20 – ANOVA result for fingerprint scans 20991-20995 [12]
6.4.4 Noise Detection

An interesting observation that was found while analysing the texture features directly, was what seemed to be a correlation between the noise in the original fingerprint image and the texture features themselves. Upon closer inspection it was noted (Figure 6.18) that the high value feature (greater than 50 points) were in fact the extreme noise that was present within the fingerprint image. The noise was located at the top of the image that was introduced by the fingerprint sensor. To determine if points within the fingerprint could be located within the image, PCA was used to separate out similar features.

![PCA Feature Plot](image)

*Figure 6.21 – PCA Feature Plot [12]*

The feature plot showed that the features corresponding to the noise within the fingerprint image (Features: 1, 7, 13, 19, 25 and 31) are clearly separated from the rest of the features within the fingerprint.
This could be used as a method for extracting noise from a fingerprint image and to identify the fingerprint features that contain more specific information to identification of an individual; i.e. centred at the core of the fingerprint.

As PCA can be performed within the MATLAB environment, this property of finding noise within the fingerprint could be used to perform pre-processing of the fingerprint image prior to selecting the features that will identify the fingerprint. This process of removing noise from the texture features will improve the performance of the identification system as only the relevant features will be kept and unwanted features can be discarded.

### 6.5 Summary

Due to time constraints within this Masters degree this can not be implemented within the MATLAB interface but will expect that this may be a valuable addition to the fingerprint identification system; being able to remove noise from a fingerprint image with a minimal amount of processing required. The embedded processor would have to calculate the PCA values as a penalty. This increases security of the system due to irrelevant features being disregarding and focusing higher on the individuals fingerprint features. The results that have been presented in this chapter will be discussed further in the final chapter of this thesis.
Chapter 7

7 Conclusion

This thesis has focused on designing a fingerprint-based identification system that is highly expandable by using the MATLAB environment. Further, the focus has been on the core technology of fingerprint feature extraction and design of a cost effective embedded processing environment. The system that has been envisaged that will form the complete fingerprint identification system is shown in Figure 7.1:

![Fingerprint System Overview](image)

This chapter will summarise the main achievement reported in this work. This work provides an insight into the development of an embedded fingerprint identification system that is anticipated to be cost effective bringing secure fingerprint identification to a variety of compact devices available to consumers. There is still a lot of potential to be explored within compact fingerprint identification systems and possible projects that can use embedded identification.
7.1 Achievements

The fingerprint identification system designed in this work consists of four major components as outlined in Chapter 3 and 4:

1. Hardware Design,
2. Image Capture,
3. Image Processing, and
4. Feature Extraction

Various approaches were researched for selection of final components to be used within the fingerprint-based identification system and allowed the software to be designed within the MATLAB environment.

Selection of the final system components for the electronic fingerprint-based identification system was based upon criteria to yield the optimal performance within an embedded processing environment. Although an embedded processing environment is limited in processing capabilities, these cost effective platforms will allow fingerprint-based identification technology to be implemented in more commercial systems that require human identification.

7.1.1 Aspects for Improvement

The techniques developed and system designed in this work can be further improved in order for the identification system to be commercially explored:

1. Introduction of a new method to determine if the finger present on the sensor surface is actually human and not a fake fingerprint representation. If only the impression of the fingerprint is analysed as outline in this thesis, this can be easily represented using latex technology to cover an existing finger with a new fingerprint. Adding a secondary component to detect a human finger would address the need to present the actual finger to the identification system.
2. Improvements can be made to the texture feature extraction method that can improve the features extracted from the fingerprint. Analysis of the features outlined the need to extract more features that would enable the features extracted to link with specific locations of fingerprint. Features located around the core of the fingerprint generally contain more information and are clearer when captured using electronic fingerprint sensors. Additional methods may be used to improve the texture feature extraction such as using not only Laws method but also a secondary method to improve the reliability and security of the system.

3. Further research into the benefits of the analysis methods would improve the type of features that are retained for use within the identification system. Using both principle component analysis and ANOVA together would enable the system to select ideal features in real-time to improve the performance of the verification algorithm. Improvements in this ‘selective’ feature algorithm may reduce the complexity of verification task without the need to use features of low importance or with no importance in the matching stage.

4. Improvements can be made to the MATLAB code platform to allow the code to be directly exported to a developed embedded identification system. This would improve development time for implementing a commercial fingerprint-based identification system that uses embedded processors with hardware specific code.
7.2 Recommendations for Future Work

By considering the achievements of the work reported within this thesis and taking into account the shortcomings, the following work would explore the potential of an embedded microprocessor based fingerprint identification system:

1. There are still a number of challenges in fingerprint verification regarding security. By the use of a fake fingerprint the majority of fingerprint sensors can be fooled into giving the impression of another individual is present. Development of new hardware that is able to determine the ‘liveliness’ of the finger being presented on the sensor surface. Integration of heart pulse measurement would add to the security of the fingerprint sensor, as the user could not have a latex glove with embedded fingerprint of another person to gain unauthorised entry to the system. The system would be able to determine whether an actual finger is present or just an impression. Integrating such devices into the hardware would provide another level of security to the fingerprint identification system.

2. Developing the texture feature extraction method to extract more features from the fingerprint image. This would allow analysis to be completed to select the most relevant features. In addition, using a second texture feature extraction method would enable the system to complete two completely separate analysis of the fingerprint which could be implemented within a multi-processor embedded system to alleviate any performance loss especially since the cost of embedded processors are relatively low compared to dedicated DSP processors.
3. Integration of the variate analysis methods used in this thesis to the identification system would improve the feature extraction. Using the results of the analysis, irrelevant features could be discarded to remove noise that was present in the fingerprint image. This could also be expanded upon to complete further analysis and careful selection of the features to locate the most specific features that have been acquired, reducing the number of inputs that would be required for the verification algorithm.

4. Expanding upon the capability of the MATLAB development environment to directly export the developed algorithms to an embedded system using the inbuilt hardware-specific code generators. The use of this feature would allow a completed system to be evaluated then directly downloaded to the identification system without the need to write hardware specific instructions. This highly portable MATLAB code can then be used for a number of embedded identification products.

In summary, the embedded fingerprint identification system designed in this work and the feature extraction method has the potential to be further researched to be part of a commercially available fingerprint-based identification system using more cost effective embedded technology.
References


42. Maio, D., Neural Network Based Minutiae Filtering In Fingerprints.
43. Denmark, T.U.o., Neural Network Based System Identification Toolbox. 2000, Department of Automation: Denmark.
48. Richards, E.P., Phenotype vs. Genotype: Why Identical Twins Have Different Fingerprints?
55. Fraser, N., Training Neural Networks. 1998.
Publications

These following papers were submitted and have been published containing results and conclusions that have been drawn from work completed during the course of this thesis.


Analysis of extracted fingerprint texture features
Joshua Mackley and Saeid Nahavandi
School of Engineering & Technology, Deakin University, Geelong, VIC. 3217, Australia
{mackley,nahavand}@deakin.edu.au

Abstract
This paper describes an approach for electronic feature extraction from fingerprint images for identification purposes. Statistical analysis of the features extracted from the fingerprint provides the advantage of real-time analysis of the fingerprint information, providing a method to remove unwanted information prior the matching stage required in electronic fingerprint verification. This method has provided detection of noise within the fingerprint image and location of critical data in the fingerprint. In order to demonstrate the effectiveness of this work, experiments have been developed and conducted with the results and analysis provided at the end of this paper for discussion.

1. Introduction

Fingerprints are well known in society for their identification of individuals from criminal identification. This thesis has chosen to use the fingerprint as the method of identification due its non-invasive nature to the individual. The properties of fingerprint formation are well known by forensic scientists and have been used extensively for identification of criminals around the world [1]. The extensive use of fingerprint identification from criminal purposes proves the reliability of an identification system based upon the fingerprint. No two fingerprints have ever been found to have the same ridge detail sequence that is identical between any two individuals [2, 3].

Fingerprint identification is based on two fundamental properties: (1) fingerprint details are permanent, and (2) fingerprints of an individual are unique [4]. The individuality problem can be defined as the probability that any two individuals may have similar fingerprints within the target population. In this thesis, the individuality problem is defined as the probability that two individuals have sufficiently similar fingerprints. Fingerprints are defined as ‘sufficiently’ similar by most human experts and automatic fingerprint identification systems if they originate from the same source. The amount of similarity between fingerprints for a match depends on the variations between multiple impressions of the same finger.

The structure of the fingerprint is made from a pattern of ridges and valleys. Fingerprints contain two main types of features; local features within a restricted region of the fingerprint, and global features that take an overall attribute of the entire fingerprint. Either of these two features can be used for identification of an individual as both features do not change over time.

A fingerprint can be viewed as an oriented texture and for sufficiently complex orientated textures such as fingerprints, invariant texture representations can be extracted [5]. Using this underlying structure within the fingerprint does not rely upon locating the core or delta points that is difficult to detect or not present in low quality images. A texture based representation of the fingerprint will be used combined with a statistical analysis of the texture information to select the most appropriate texture features to use prior to the matching stage of electronic fingerprint identification.

The features that are extracted from the fingerprint are known as the individual’s ‘template’. This template is the signature of the fingerprint, taking only the key features and not the entire fingerprint. To create this template, the individual normally has to enrol onto the identification system by submitting the fingerprint to the system multiple times so that the features can be cross-referenced and check that no features have been missed between scans. Once enrolled this template is used for verification in the future.
2. Texture Features

Texture features that are extracted can be categorised as: statistical, geometrical, structural, model-based and signal processing features [6]. In image analysis, there are various methods available to extracting these texture features, and must be understood in selecting the optimal method for system being used.

The fingerprint image is then partitioned into sections and each individual section is normalised to a constant mean based upon the pixel intensity. This partitioning removes the need for a reference point to be extracted from the fingerprint image and provides a method that will enable feature information to be extracted directly from these individual sections. The partition size will be selected to reduce the requirements of the verification algorithm; the more sections will require a more complex algorithm. From previous research, it has been stated that a partition size of less than 20x20 pixels results in limited or loss of texture information [7].

The partitioning function is performed within the MATLAB environment and enables the section size to be selected by the user as depending on the verification algorithm, the number of features available may need to be modified. A partition size of 60x60 pixels has been selected as it meets the requirements of [7] and also reduces the number of inputs to the verification algorithm to 36 features. The partition sections remain in gray scale (8-bit) format to retain the texture information prior to extracting the feature information.

The section size defined by the partitioning defines the number of features that can be extracted by this system. The sections of the fingerprint image contain 360 pixels with a range of 0-255 (gray scale). Using the pixel intensity value, the feature value calculated as the mean value of 360 individual pixels within the section. This provides a single feature value for each of the 36 sections that have been partitioned from the fingerprint image. As the image is not pre-processed all information is still available within the pixels intensity.

A fingerprint can be viewed as an oriented texture and for sufficiently complex orientated textures such as fingerprints, invariant texture representations can be extracted [6]. Using this underlying structure within the fingerprint does not rely upon locating the core or delta points that is difficult to detect or not present in low quality images. Texture features that are extracted can be categorised as: statistical, geometrical, structural, model-based and signal processing features [16].

Unlike traditional local feature extraction methods (i.e., minutiae extraction) from the fingerprint image, this thesis focused on using a texture feature extraction method. Five popular texture feature extraction methods were selected as candidates for use in this thesis: autocorrelation, edge frequency, primitive-length, law’s method and co-occurrence matrices. Studies that were performed into these methods [8] determined that the co-occurrence matrices and Law’s method perform better than other techniques.

3. Law’s Method

Law’s method is based on a series of pixel impulse response arrays obtained from combinations of 1-D vectors shown in Table 1 [7]. These five 1-D filters are capable of producing 25 possible features at each pixel location (using five different filters in both horizontal and vertical directions). The arrays are convolved with other arrays in a combinational manner to generate the masks generally labelled L5L5 for the mask resulting in the convolution of the two L5 arrays. The arrays are based upon the observations of Law’s that found certain gradient operators such as Laplacian and Sobel operators accentuated the underlying microstructure of texture within an image.

<table>
<thead>
<tr>
<th>Array</th>
<th>1</th>
<th>2</th>
<th>3</th>
<th>4</th>
<th>5</th>
</tr>
</thead>
<tbody>
<tr>
<td>Level</td>
<td>1</td>
<td>4</td>
<td>6</td>
<td>4</td>
<td>1</td>
</tr>
<tr>
<td>Edge</td>
<td>-1</td>
<td>-2</td>
<td>0</td>
<td>2</td>
<td>1</td>
</tr>
<tr>
<td>Spot</td>
<td>-1</td>
<td>0</td>
<td>2</td>
<td>0</td>
<td>-1</td>
</tr>
<tr>
<td>Wave</td>
<td>-1</td>
<td>2</td>
<td>0</td>
<td>-2</td>
<td>1</td>
</tr>
<tr>
<td>Ripple</td>
<td>1</td>
<td>-4</td>
<td>6</td>
<td>-4</td>
<td>1</td>
</tr>
</tbody>
</table>

Table 1 – Five arrays identified by Laws [9]

Using Law’s method enabled features to be extracted from the fingerprint image within the partitioned areas. The features were plotted on a 3d graph to show how the feature values correspond to the matching fingerprint image. These values are scaled between 0 – 255 from the original gray scale image.
Figure 1 – Feature plot: values of features

These 36 individual feature values are used to represent the fingerprint image that has been captured using the electronic fingerprint sensor. Analysis methods will be used to determine if this is a reliable method of feature extraction and also to determine if irrelevant features that have no identifying information can be removed from the feature list known as an individual’s template.

Using the fingerprint image, texture features can be extracted by partitioning the fingerprint image into smaller sections. This method allows a feature value to be extracted based on the texture component of each partition. The feature values will be known as the texture features of the fingerprint. The number of features that are extracted from the fingerprint affects the size and complexity of the verification algorithm and must be taken into account when designing the partition size as well as the verification algorithm. The partition size selected is 60x60 pixels which results in 36 individual features that are extracted to represent a fingerprint.

4. Statistical Analysis

Texture feature information that has been extracted from a fingerprint image may not be relevant information that can be used to represent an individual. Unlike local feature information such as minutiae which are specific points of reference to a particular fingerprint, texture information could refer to noise within the fingerprint image. The purpose of analysing the fingerprint information is to determine a threshold that will enable irrelevant texture information to be removed from the feature set to provide a more robust representation of the individual.

Several analysis methods are available for analysing data, but for these particular experiments the amount of variance between features is to be analysed. Therefore two techniques are available which suit this application of data; Analysis of Variance (ANOVA) and Principle Component Analysis (PCA). Using these two techniques will allow cross-referencing between results to determine if statistical analysis provides a reliable and repeatable method to discard irrelevant feature information that has been extracted from a fingerprint.

Experiments will be used to determine the following from the fingerprint texture features; sufficient difference can be found between fingerprints in database, separation between features can be found and determining noise features within the fingerprint. ANOVA will be primarily used to find sufficient difference between the images in the database as it allows the analysis directly of variance within the sample (texture features). PCA will be used to analyse the difference between fingerprints, between features and for finding noise within the fingerprint.

5. Experimental Results

Using the extracted features analysis was performed to determine if the fingerprints within the database could be separated sufficiently proving that these fingerprints are indeed individual fingerprints. Using ANOVA this allowed the amount of variance between each of the 5 scans of the individual fingerprints within the database to be measured. The results of this experiment found that there was an average 80% similarity between successive scans of the same fingerprint across this small database (40
individuals; total 200 fingerprints). This was also cross-compared using PCA to locate the separation between a sub-set of these fingerprints within the database and the plot shown in Figure 2 depicts the individual fingerprint images.

![Figure 2 – PCA of Individual Fingerprints](image)

This shows that the feature extraction method is able to differentiate between the individual fingerprints within the database. Further to this experiment PCA will be used to directly analyse the features to determine if any key points within the fingerprint itself are able to be extracted using this method.

An interesting observation that was found while analysing the texture features directly, was what seemed to be a correlation between the noise in the original fingerprint image and the texture features themselves. Upon closer inspection it was noted (Figure 3) that the high value feature (greater than 50) were in fact the extreme noise that was present within the fingerprint image. The noise was located at the top of the image that was introduced by the fingerprint sensor. To determine if points within the fingerprint could be located within the image, PCA was used to separate out similar features.

The feature plot showed that the features corresponding to the noise within the fingerprint image (Features: 1, 7, 13, 19, 25 and 31) are clearly separated from the rest of the features within the fingerprint. This could be used as a method for extracting noise from a fingerprint image and to identify the fingerprint features that contain more specific information to identification of an individual; i.e. centred at the core of the fingerprint. Also, the core of the fingerprint is roughly located between features 20, 26 and 32 on the feature plot, enabling the algorithm to detect where the centre point of the fingerprint is located.

As PCA can be performed within the MATLAB environment, this property of finding noise within the fingerprint could be used to perform pre-processing of the fingerprint image prior to selecting the features that will identify the fingerprint. This process of removing noise from the texture features will improve the performance of the identification system as only the relevant features will be kept and unwanted features can be discarded.

Due to time constraints within this Masters degree this can not be implemented within the MATLAB interface but will expect that this may be a valuable addition to the fingerprint identification system; being able to remove noise from a fingerprint image with a minimal amount of processing required. The embedded processor would have to calculate the PCA values as a penalty. This increases security of the system due to irrelevant features being disregarding and focusing higher on the individuals fingerprint features.

6. Conclusion

Presented in this paper is the combination of extracting features from a fingerprint using textures with statistical analysis of these features to determine the most relevant information.

Future improvements can be made to this method to allow more precise location of the feature points within the fingerprint, by using a smaller partition size but this will add to the computational requirements. The concept for using texture feature extraction with statistical analysis is to reduce the processing requirements of an electronic fingerprint identification that may be able to utilise a lower cost and commercially available microcontroller.
Improvements can be made to the texture feature extraction method that can improve the features extracted from the fingerprint. Analysis of the features outlined the need to extract more features that would enable the features extracted to be linked with specific locations of fingerprint. Features located around the core of the fingerprint generally contain more information and are clearer when captured using electronic fingerprint sensors. Additional methods may be used to improve the texture feature extraction such as using not only Laws method but also a secondary method to improve the reliability and security of the system.

Expanding upon the capability of the MATLAB development environment to directly export the developed algorithms to an embedded system using the inbuilt hardware-specific code generators. The use of this feature would allow a completed system to be evaluated then directly downloaded to the identification system without the need to write hardware specific instructions. This highly portable MATLAB code can then be used for a number of embedded identification products.

By providing a method to analyse fingerprint features within the identification system, performance may be increased but security will be increased by reducing the amount of irrelevant information that may be present within the original feature set that represents the individual on the fingerprint identification system.

This has provided a method of detecting and removing noise from a fingerprint image. Improving the feature set that represents an individual increases the reliability of the system by removing irrelevant feature information prior to the matching stage that is required within a fingerprint identification system.

References

Active temperature compensation for an accelerometer based angle measuring device

Joshua Mackley and Saeid Nahavandi
School of Engineering & Technology, Deakin University, Geelong, VIC 3217, Australia
{mackley,nahavand}@deakin.edu.au

Abstract

An angle measuring device using a high performance and very compact accelerometer provides a new and exciting method for producing extremely compact and accurate angle measuring devices. Accelerometers are micro-machined and are able to measure acceleration to a very high accuracy. By using gravity as a reference these compact devices can also be used for measuring angles of rotation. The inherent problem with these devices is that their response characteristic changes with temperature which is detrimental to measurement accuracy. This paper describes an effective method to overcome this problem using an accelerometer sensor and intelligent software to compensate for this drift characteristic. In order to demonstrate the effectiveness of this work, experiments have been developed and conducted with the results and analysis provided at the end of this paper for discussion.

1. Introduction

Measuring angle of rotation using gravity as a reference requires a very precise sensor that ideally has a low response time for high performance angle measurement. For commercial applications, cost analysis of the final product makes a large impact on the selection of the final products used in the system. In large quantity production, small savings can amount to large profits in the long term. There are various sensors available for angle measurement and for accurate angle measurement down to 0.1° for a complete 360° measuring range a dual-axis accelerometer is an ideal sensor. These sensors are very cost competitive, making an ideal selection for a commercial product in this application. Analog Devices manufacture accelerometers which are suitable for angle measurement. One of their suitable products incorporates a dual-axis sensor on the same silicon die which enables high accuracy angle measurement in a very small package.

One of the problems associated with this sensor is that the output drifts with change in temperature. To overcome this problem, a microcontroller based system is used so that software can calibrate the system and automatically compensate for the temperature drift over the operational temperature range. This reduces system cost for more expensive sensors and still achieves the same accuracy by using embedded software within the system to compensate for the temperature error.

The aim of this work is to keep the commercial cost of the system low, while still maintaining high accuracy readout of angle. Using a microcontroller based approach, the system coupled with a temperature sensor enables the accelerometer response against temperature to be stored in memory, and used as an offset during normal operation to remove the inherent temperature error within the accelerometer. The response of the accelerometer must be calibrated for each individual unit, as each accelerometer has a unique temperature vs. output characteristic that must be stored to achieve the required accuracy of ±0.05°.

The system uses an 8-bit microcontroller that measures both the acceleration on both axis, and the current temperature of the accelerometer, and then calculates the angle based on this information. By using a software based approach a linear offset can be stored in memory which is advantageous against discrete circuit options to remove the temperature error. This reduces the error of discrete based approaches to further increase the accuracy of the system which is ideal to achieve the required accuracy of ±0.05° with a repeatability of ±0.10°.

In section 2, the temperature sensor characteristics are described and the different methods that can be used to reduce the errors associated with the temperature drift of the accelerometer; section 3 outlines the microcontroller based approach that was used to reduce the accelerometer temperature drift.
errors; and in section 4 the results are reported and conclusions are drawn on the microcontroller based system.

2. Temperature Effects

Accelerometers produced using micro machined technology suffer from zero bias drift due to temperature change [1, 2]. This property of the sensor affects the overall accuracy of the device which can lead to errors in angle measurement.

*Figure 1* shows a random sample of sensors and the drift characteristics of these sensors. The figure shows how each sensor has its own drift characteristics that require each sensor to be individually calibrated to achieve the highest accuracy without errors in angle measurement due to temperature drift.

![Figure 1 – Analog Devices specification of drift between parts [1]](image)

The sensor was placed within a controlled environment to determine the relationship of temperature vs. acceleration output. The controlled environment was kept within ±2°C and a laser temperature unit was used to measure the sensor temperature within ±0.5°C. The environment was then changed from 20°C to 40°C as this is the expected range that the final product will operate within.

This relatively small drift in acceleration measurement has an adverse effect on the response of the angle measuring device. Based on the samples an error of 0.1mg is common and as our reference for this system is gravity (1g) this has an overall error of 0.1% change per °C. 0.1mg also equates to an error of 0.05° of angle inaccuracy. Over the devices expected operation temperature this is an error of 1.00° for only a 20°C change in environment conditions.

Various methods can be used to compensate for this temperature drift. Discrete dedicated components can be used to offset this drift but accuracy is compromised due to individualistic response of each sensor. Another solution is to use a small processor which is dedicated to correcting this drift using a temperature sensor. The later approach although more accurate introduces more cost due to an additional processor required.

3. Current Architecture Design

The current architecture of the angle measuring device uses an 8-bit microcontroller, dual-axis accelerometer and a temperature sensor. The microcontroller performs all the required calculations while also providing a cost competitive processor for such a product. By adding just the temperature sensor to the existing microcontroller & accelerometer implementation allowed easy integration of temperature compensation while minimising system cost.

![Figure 2 – Block diagram of system](image)

The system uses an 8-bit microcontroller that measures both the acceleration on both axis of the dual-axis accelerometer and the current temperature of the accelerometer by way of the temperature sensor and then calculates the angle based on this information. The force placed on the accelerometer by gravity is used as the reference and also required for the initial calibration to reset the angle measuring device back to 0.0 degrees (as each accelerometer has its own unique zero point).
4. Accelerometer Measurement

For accurate angle measurement a microcontroller based approach is recommended when using the dual-axis accelerometer [3, 4]. This allows the acceleration to be measured digitally, reducing errors due to noise that can be a problem with analogue approaches. With temperature change affecting this acceleration measurement the microcontroller must also compensate for this error to provide a reliable and repeatable angle measurement at all temperatures that the circuit is rated.

Using the examples provided to reduce the accelerometers drift with temperature [5, 6] are effective but do not allow full compensation of the inherit temperature drift. By incorporating the temperature compensation algorithm within the available processor calculating the angle and using an additional temperature sensor this simplifies implementation of a software based solution. Using the microcontroller enables the algorithm to calculate the temperature response of the sensor using a calibration process and store this to allow a constantly updated offset based on the temperature of the accelerometer.

The temperature sensor that was selected has a linear response. This was required to reduce the complexity of the software; with a non-linear sensor, more data points would be required to achieve the same accuracy throughout the temperature range. As the memory capacity is limited in the microcontroller this had to be taken into account when selecting an appropriate temperature sensor.

Integration of the new algorithm into current software to compensate for temperature change would require significant changes to integrate this feature into the following areas:

- Calibration
- Angle Offset
- Memory Allocation

Also there was only a single analogue to digital converter available that had to be shared with the battery voltage monitor function. This required time-sharing of the A/D converter within the microprocessor. Due to the complexity of the existing software there was limited memory and processor capacity available for implementation of such a new function. The new temperature compensation feature had to be both effective but also only include what was strictly required; any additional accuracy or other such benefits had to be removed to give a bare-bones feature that met only requirements as luxury items could not be implemented with such memory limits.

5. Innovative Software Improvements

Dedicated hardware adds complexity to the angle measuring device which is not wanted so another approach that uses only a single additional component was preferred; a highly accurate surface mount temperature sensor.

The approach of the temperature compensation software had to filter the sensor output so that a stable offset would be available, calibrate the sensor characteristic as this varies between units and apply this offset to the angle measurement continually. The microprocessor control loop that processes the angle information is already at 90% of the total processor capacity to keep within the calculation time period required. This processing limitation requires careful selection of the filter to be used and how the final processing offsets are applied.

To reduce processing overhead an 8-cell moving average filter was chosen to provide a smooth temperature reference. This is an extremely efficient filter and as the cells are updated every 100mS the control loop recalculates the temperature reference every 250mS. Figure 2 shows the FILO (First In Last Out) structure that is used to store the last 8 measurements taken from the temperature sensor. These cells are averaged each step to provide the filtered temperature measurement.

![Figure 2 – FILO (First In Last Out) structure](image)

Program Code 1 shows a code extract of measuring the temperature value prior to placing into the filter cell array.

![Figure 3 – 8-cell moving average filter](image)

This provides the processor with a smoothed temperature output that is used to offset the angle measurement. Program Code 1 shows a code extract of measuring the temperature value prior to placing into the filter cell array.
As each accelerometer has an individual response characteristic to temperature change that must be adjusted per angle measuring device as a standard offset in software will not be accurate. This is done by taking two measurements of the angle calculated at 0° at ambient and 10°C different from ambient. The processor then calculates the temperature response (gradient) of the accelerometer and stores this in permanent memory. This formula can then be applied at any temperature (due to the linear response of the accelerometer). By calibrating this temperature response on an individual unit basis this also compensates for other devices in the circuit that affect the accuracy of the angle measuring device as a whole.

Standard angle measuring devices were taken from a sample to determine their response as the temperature was changed from 20°C to 40°C. Figure 3 shows the response (triangle and circle markers) of 2 standard measuring devices and the drift response due to the surrounding temperature change. These two devices drift over 0.8° which is very high when trying to achieve an accuracy of at least ±0.10°.

Figure 4 – Diamond marker shows temperature compensated system

Using an angle measuring device that includes the new temperature sensor and software algorithm to improve the response due to temperature change all three devices were subjected to the same chamber to determine improvements over using an accelerometer that has not been compensated. As can be seen from Figure 3, the diamond marker line shows the low drift characteristic of this new temperature compensated angle measuring device.

This was also tested over an extended period holding each temperature for 1 hour (due to the device being battery powered a power supply was used for this test) and measuring the angle displayed with the device placed at 0.0° (known reference zero). Figure 4 shows the increased accuracy with an angle deviation of 0.10° using the temperature compensated device.

Figure 5 – Extended response test, diamond marker: temperature compensated device.

### 6. Experimental Results

To keep results consistent when testing both current and new approaches to angle measurement a temperature controlled oven was used when changing the environment. Also a calibrated bench to zero degrees (within ±0.05°) to test the repeatability was mounted outside the oven to reliably determine accuracy of both standard and modified devices.

```c
void TempValueMeasure(void)
{
    /* store the previous result for filtering */
    temp_prev= temp_value;
    /* retrieve conversion from ADC*/
    /* must read low byte first */
    temp_value_lowbyte=ADCL;
    temp_value_highbyte=ADCH;
    /* trigger new A/D conversion */
    ADCSR |=ADSC_B;
    /* no need to poll this bit */
    /* temperature filter (8 components) */
    if(offset_cnt==7)
    {
        /* must reset for wrap-around */
        display_offset_cnt=0;
    }
    else
    {
        /* next index position */
        display_offset_cnt++;
    }
    /* calc new filtered value */
    temp_filtd=avg(temp_value,temp_prev);
}
```

Program Code 1 - 8-cell filter example
The accuracy of this device depends on the temperature calibration performed which has a large affect on the final accuracy. This calibration requires an even and accurate temperature of both the accelerometer on the angle measuring device and the temperature sensor otherwise errors will occur. To improve this uniformity within the device both the accelerometer and the temperature sensor were coated to the circuit board with a thermally conductive but insulating compound (and covered with a protective epoxy resin) to keep the temperature between these components uniform. It was found that a drift would occur and a rapidly changing temperature environment due to the thermal mass difference between the accelerometer and its placement on the circuit board compared to the temperature sensor. Using this compound reduced the difference from ±5°C to ±1°C and improves the error from 0.25° to 0.05°.

7. Conclusion

Using a microcontroller based system with an active temperature compensation solution the drift normally associated with accelerometer measurement can be kept to a minimum. This approach also reduces the cost of the system which is a large factor when considering components for high volume production in a commercial application. For the non-temperature compensated system, the actual angle difference over the entire tested temperature range was from 0.39° to 1.35° (0.96°).

For the temperature compensated system, the actual angle difference over the entire temperature range was from 0.16° to 0.03° (0.13°). This is an improvement from an error of 3.5% down to 0.47% error in angle measured. These results show the improvement over a standard non-temperature compensated system and how a low-cost accelerometer can be used to achieve high accuracy angle measurement.

Using the thermally conductive compound and protective epoxy resin to insulate and keep the temperature uniform across the angle measuring device also improved the response of the system and kept errors which are introduced by using the additional temperature sensor to a minimum. The only disadvantage to this approach is the additional cost and added step required in production to add these compounds to the angle measuring device.

Improvements could be made to this method by using a floating-point calculation to characterise the temperature characteristic. This is better solution than the current fixed-point approximation that has been outlined within this paper. This does require more memory and processing capacity that was not available for this solution. With such an approach this would enable the system to achieve an accuracy of ±0.05°. In addition to the hardware requirements, the angle measuring device software would also require to be modified heavily to integrate the floating-point temperature compensation. Using the experiments outlined within this paper would enable a direct comparison of any further improvements of this type of device.

This microcontroller based system with an active temperature compensation solution greatly improves the overall system accuracy to a level that will be accepted in a commercial application and adds relatively little additional cost to the production cost and also minimal impact to the compact size with only the surface mount temperature sensor required.

References

2. Kitchin, C., Compensating for the 0g Offset Drift of the ADXL50 Accelerometer, P. Brokaw, Editor. 1994.