BIOMETRIC ATTENDANCE SYSTEM

By

ANKUR KATHAYAT (071080)

ANKUR SHARMA (071117)

Under the guidance of

Ms. Neeru Sharma

May, 2011

A thesis submitted in partial fulfilment of the requirements for
the degree of Bachelor of Technology
in Electronics and Communication Engineering

Department of Electronics and Communication Engineering

Jaypee University of Information Technology

CERTIFICATE

This is to certify that the thesis entitled, "BIOMETRIC ATTENDANCE SYSTEM" submitted by Ankur Kathayat and Ankur Sharma in partial fulfilments for the requirements for the award of Bachelor of Technology Degree in Electronics and Communication Engineering at JAYPEE UNIVERSITY OF INFORMATION TECHNOLOGY, Waknaghat is an authentic work carried out by them under my supervision and guidance.

Date : 18.05.2011

Place : Waknaghat

NSharma

Ms. Neeru Sharma

Department of Electronics and Communication Engineering

Jaypee University of Information Technology

It is certified that this work has not been submitted partially or fully to any other University or Institute for the award of this or any other degree or diploma.

Ankur Kathayat

071080

Ankur Sharma

071117

ACKNOWLEDGEMENT

We would like to express my sincere thanks to Ms. Neeru Sharma, Department of Electronics and Communication Engineering, for her constant encouragement and guidance during this project. Her continuous involvement in the progress of our project was of great help to us. She has been a constant source of inspiration and has helped us at each and every step of the project. We would like to thank her for her valuable suggestions in the project which improved the usefulness of the system as a whole.

The Zeal to accomplish the task of formulating the project could not have been realized without the support and cooperation of the faculty members of ECE Department. We Sincerely thank Prof.Dr. T. S. Lamba (Dean) and Prof.Dr. Sunil V. Bhooshan (HOD, ECE) for their consistent support throughout the project work.

Date: 23.05.2011

Ankur Kathayat
Ankur Sharma

ABSTRACT

Our project aims at introducing biometric capable technology for use in automating the entire attendance system for the students pursuing courses at an educational institute. The goal can be disintegrated into finer sub-targets; fingerprint capture, fingerprint image processing and matching. For each sub-task, various methods from literature are analyzed. From the study of the entire process, an integrated approach is proposed.

Biometrics based technologies are supposed to be very efficient personal identifiers as they can keep track of characteristics believed to be unique to each person. Among these technologies, Fingerprint recognition is universally applied. It extracts minutia- based features from scanned images of fingerprints made by the different ridges on the fingertips. The student attendance system is very relevant in an institute like ours since it aims at eliminating all the hassles of roll calling and malpractice and promises a full-proof as well as reliable technique of keeping records of student's attendance.

Accuracy and reliability are the two most important parameters when it comes to biometric applications. Fingerprint verification is one of the oldest known biometric techniques known but still is the most widely used because of its simplicity and good levels of accuracy. It's a well known fact that every human being is born with a different pattern on the fingers and this feature is exploited to identify and differentiate between two different persons.

TABLE OF CONTENTS

LIST OF FIGURES

CHAPTER 1 : INTRODUCTION

1.1 Why biometrics?

The human body has the privilege of having features that are unique and exclusive to each individual. This exclusivity and unique characteristic has led to the field of biometrics and its application in ensuring security in various fields. Biometrics has gained popularity and has proved itself to be a reliable mode of ensuring privacy, maintaining security and identifying individuals. It has wide acceptance throughout the globe and now is being used at places like airports, hospitals, schools, colleges, corporate offices etc.

Biometrics is the very study of identifying a person by his/her physical traits that are inherent and unique to only the person concerned. Biometric measurement and assessment include fingerprint verification, iris recognition, palm geometry, face recognition etc. The above mentioned techniques work with different levels of functionality and accuracy.

Accuracy and reliability are the two most important parameters when it comes to biometric applications. Fingerprint verification is one of the oldest known biometric techniques known but still is the most widely used because of its simplicity and good levels of accuracy. It's a well known fact that every human being is born with a different pattern on the fingers and this feature is exploited to identify and differentiate between two different persons.

1.2 Why this project?

The application in an educational institute is worth noting because of the benefits it brings along with it. The fingerprint recognition and verification technique can easily replace an attendance sheet and save time wasted on calling out roll numbers in the class. A fingerprint detecting device needs to be placed in each classroom and students would be made to swipe their finger over the sensor so as to mark their presence in the class. The database would contain all the fingerprints beforehand. So, the moment a finger would be swiped, a check would be carried out with the existing database and the corresponding student would get a present mark on his attendance record maintained in a server.

The transfer of the fingerprint from the device to the computer can be carried out using the software development kit provided with the fingerprint reader. For further security of the entire system and to detect illegal activities, a security camera can be installed to keep track of the enrollments made in the classroom.

1.3 System Design

The design of fingerprint based attendance system can be divided into the 4 different modules. They are :

- Fingerprint Capture
- Image Enhancement
- Fingerprint Matching
- Database Management

Fingerprint capturing is done using an optical fingerprint reader, whereas the image processing and matching are done on a PC using MATLAB.

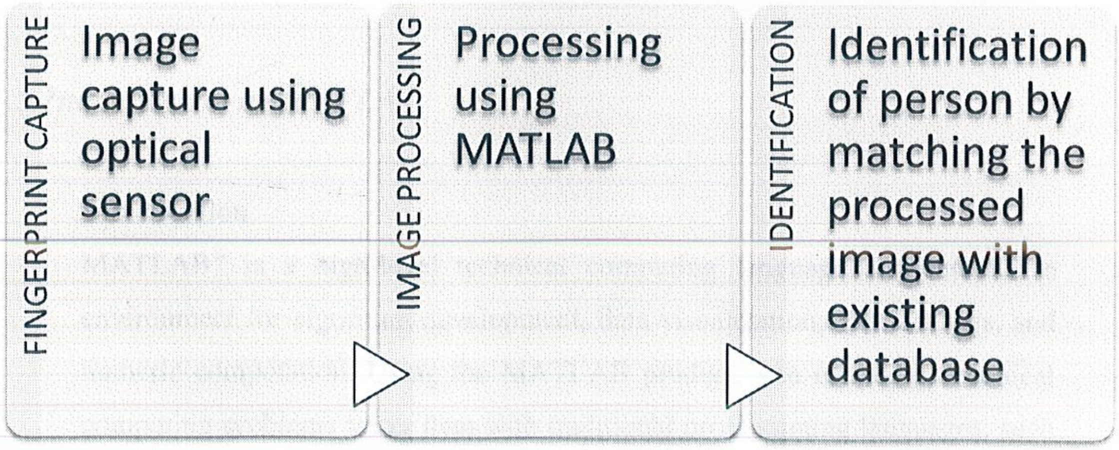


Figure 1.1 *A block diagram depicting the working of the project*

The module-wise approach to the design of the system helps in better understanding of the individual function levels. Also, a parallel approach to the system helps in distributing the effort on a multi-level range and helps in identifying the best features and available products in the market that suit the design requirements. This has been done in the following chapters.

1.4 Plan of action

The following table shows our plan of action of the implementation of project.

MONTH	YEAR	WORK DONE
July	2010	Analysed literature
August	2010	research work
September	2010	optical sensors research
October	2010	introduction to matlab
November	2010	image enhancement techniques
January	2011	studied literature for matching technique
February	2011	implementation of gabor filters
March	2011	fingCode technique implementation
April	2011	database management
May	2011	finalization of project and reports

1.5 Introduction to MATLAB

Introduction

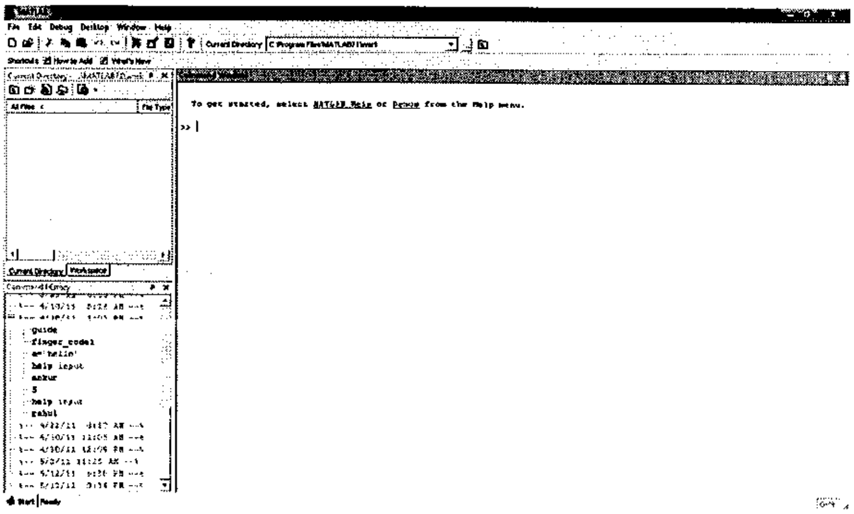
MATLAB® is a high-level technical computing language and interactive environment for algorithm development, data visualization, data analysis, and numeric computation. Using the MATLAB product, you can solve technical computing problems faster than with traditional programming languages, such as C, C++, and Fortran.

You can use MATLAB in a wide range of applications, including signal and image processing, communications, control design, test and measurement, financial modeling and analysis, and computational biology. Add-on toolboxes (collections of special-purpose MATLAB functions, available separately) extend the MATLAB environment to solve particular classes of problems in these application areas.

MATLAB provides a number of features for documenting and sharing your work. You can integrate your MATLAB code with other languages and applications, and distribute your MATLAB algorithms and applications.

Key Features

- High-level language for technical computing
- Development environment for managing code, files, and data
- Interactive tools for iterative exploration, design, and problem solving
- Mathematical functions for linear algebra, statistics,. Fourier analysis, filtering, optimization, and numerical integration
- 2-D and 3-D graphics functions for visualizing data
- Tools for building custom graphical user interfaces
- Functions for integrating MATLAB based algorithms with external applications and languages, such as C, C++, Fortran, Java, COM, and Microsoft Excel



```
To get started, select MATLAB Help or Demos from the Help menu.
>> |
```

```
guide
finger_code1
a='hello'
help input
ankur
5
help input
rahul
```

Figure 1.2 MATLAB command window

CHAPTER 2 : FINGERPRINT

2.1 What is a fingerprint?

A fingerprint, as the name suggests is the print or the impression made by our finger because of the patterns formed on the skin of our palms and fingers since birth. With age, these marks get prominent but the pattern and the structures present in those fine lines do not undergo any change. For their permanence and unique nature, they have been used since long in criminal and forensic cases.

Shown below, is a fingerprint pattern obtained from an optical sensor. The figure shows faint and dark lines emerging from a particular point and spiraling around it all over the finger.

Figure 2.1.1 *A fingerprint image acquired by an optical sensor*

Every fingerprint consists of ridges and furrows. These ridges and furrows are known to show good similarities but when it comes to identifying a person or distinguishing between two different prints, these do not prove efficient enough. Research shows that fingerprints are not distinguished by ridges and furrows but by Minutia. Minutia refers to some abnormalities in a ridge, which shall be discussed in detail in the following pages.

As already mentioned, Minutia are abnormal points in a ridge. There can be various such Minutia but the two most important and useful minutia types are Termination and Bifurcation. Termination refers to the abrupt ending of a ridge, as shown in fig.2.1.2. Bifurcation on the other hand refers to the point on the ridge where branching occurs, as shown in fig.2.1.3

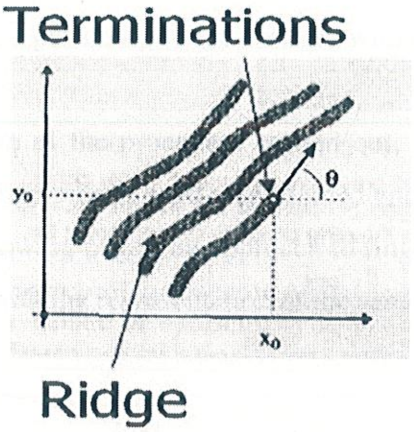


Figure 2.1.2 *Termination minutia*

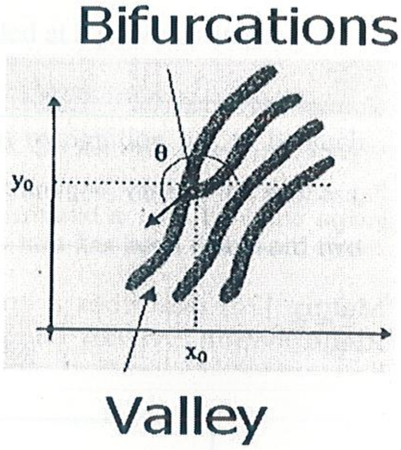


Figure 2.1.3 *Bifurcation minutia(Furrow, also known as valley*

2.2 *Fingerprint Recognition*

Once the fingerprint is captured, the next step is the recognition procedure. The recognition procedure can be broadly sub grouped into

a. Fingerprint identification
b. Fingerprint verification

Fingerprint identification refers to specifying one's identity based on his fingerprints. The fingerprints are captured without any information about the identity of the person. It is then matched across a database containing numerous fingerprints. The identity is only retrieved when a match is found with one existing in the database. So, this is a case of one-to-n matching where one capture is compared to several others. This is widely used for criminal cases.

Fingerprint verification is different from identification in a way that the person's identity is stored along with the fingerprint in a database. On enrolling the fingerprint, the real time capture will retrieve back the identity of the person. This is however a one-to-one matching. This is used in offices like passport offices etc. where the identity of a person has to be checked with the one provided at a previous stage.

Irrespective of the procedure carried out, the fingerprint recognition has to be such that the fingerprint is well- represented and retains its uniqueness during the process. In the following pages, an approach to fingerprint recognition has been discussed that will deal with the representation of the same.

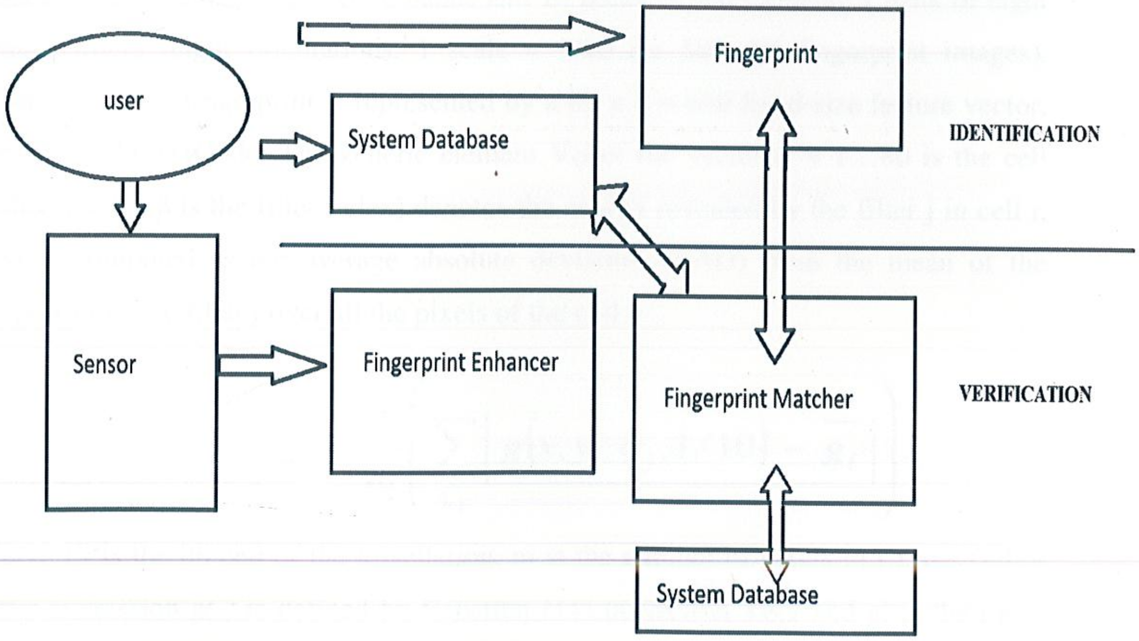


Fig 2.2.1: Verification Vs Identification

2.3 Approach to fingerprint recognition

The most popular technique to match fingerprints based on texture information remains the FingerCode approach by Jain et al. (2000). The fingerprint area of interest is tessellated with respect to the core. A feature vector is composed of an ordered enumeration of the features extracted from the local information contained in each sector specified by the tessellation. Thus the feature elements capture the local texture information and the ordered enumeration of the tessellation captures the global relationship among the local contributions.

The local texture information in each sector is decomposed into separate channels by using a Gabor filterbank; in fact, the Gabor filterbank is a well-known technique for capturing useful texture information in specific bandpass channels as well as decomposing this information into biorthogonal components in terms of spatial frequencies.

In their experimentation, Jain et al. (2000) obtained good results by tessellating the area of interest into 80 cells (five bands and 16 sectors), and by using a bank of eight Gabor filters (eight orientations, 1 scale = 1/10 for 500 dpi fingerprint images). Therefore, each fingerprint is represented by a 80 × 8 = 640 fixed-size feature vector, called the FingerCode. The generic element V*ij* of the vector (*i* = 1…80 is the cell index, *j* = 1…8 is the filter index) denotes the energy revealed by the filter j in cell i, and is computed as the average absolute deviation (AAD) from the mean of the responses of the filter *j* over all the pixels of the cell *i*:

$$V_{ij} = \frac{1}{n_i}\left(\sum_{C_i}\left| g\left(x, y : \theta_j, 1/10\right) - \overline{g_i} \right|\right).$$

where *Ci* is the ith cell of the tessellation, ni is the number of pixels in *Ci*, the Gabor filter expression *g( )* is defined by Equation (11) in Section 3.6.2 and gi is the mean value of *g* over the cell *Ci*. Matching two fingerprints is then translated into matching their respective Finger- Codes, which is simply performed by computing the Euclidean distance between two Finger-Codes.

CHAPTER 3 : FINGERPRINT CAPTURING DEVICES

3.1 Optical Sensors

- *Frustrated Total Internal Reflection (FTIR):*

This is the oldest and most commonly ridges are in optical contact with the prism surface, but the valleys remain at a certain distance (see Figure 2.6). The left side of the prism is typically illuminated through a diffused light (a bank of light-emitting diodes [LEDs] or a film of planar light). The light entering the prism is reflected at the valleys, and randomly scattered (absorbed) at the ridges. The lack of reflection allows the ridges (which appear dark in the image) to be discriminated from the valleys (appearing bright). The light rays exit from the right side of the prism and are focused through a lens onto a CCD or CMOS image sensor. Because FTIR devices sense a three-dimensional finger surface, they cannot be easily deceived by presentation of a photograph or printed image of a fingerprint.

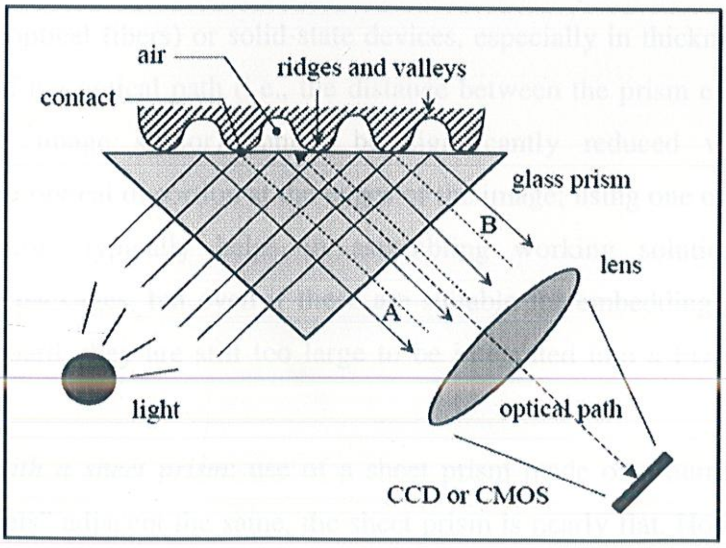


Figure 3.1.1 *FTIR-based fingerprint sensor operation.*

The FTIR-based sensor shown in Figure 3.1.1 often introduces certain geometrical distortions. The most evident one is known as trapezoidal or keystone distortion and is produced by the perspective view of the imaging surface. Since the fingerprint plane is not parallel to the CCD plane, rays A and B in Figure 2.6 have different lengths, resulting in a stretching or compression of the image regions which is a function of their distance from the optical axis. Compensation for this distortion may be opticsbased or software-based (i.e., calibration techniques). Some optics-based techniques make use of ad hoc pre-moulded plastic lenses or holograms as proposed by Seigo, Shin, and Takashi (1989), Igaki et al. (1992), Drake, Lidd, and Fiddy (1996) and Bahuguna and Corboline (1996) or of a correcting wedge prism as proposed by Rao (2008).

When a finger is very dry, it does not make uniform and consistent contact with the FTIR imaging surface. To improve the formation of fingerprints from dry fingers, whose ridges do not contain sufficient sweat particles, some scanner producers use conformal coating (typically made of silicone), which improves the optical contact of the skin with the prism. With the aim of reducing the cost of the optical devices, plastic is often used instead of glass for prisms and lenses, and CMOS cameras are used instead of the more expensive CCD cameras. In spite of generally superior image quality and potentially larger sensing areas, FTIR-based fingerprint devices cannot be miniaturized unlike other optical techniques (e.g., optical fibers) or solid-state devices, especially in thickness. In fact, the length of the optical path (i.e., the distance between the prism external surface and the image sensor) cannot be significantly reduced without introducing severe optical distortion at the edges of the image; using one or more intermediate mirrors typically helps in assembling working solutions in reasonably small packages, but even if these are suitable for embedding into a mouse or a keyboard, they are still too large to be integrated into a PDA or a mobile phone.

- *FTIR with a sheet prism*: use of a sheet prism made of a number of "prismlets" adjacent the same, the sheet prism is nearly flat. However, the quality of the acquired images is generally lower than the traditional FTIR techniques that use glass/plastic prisms.

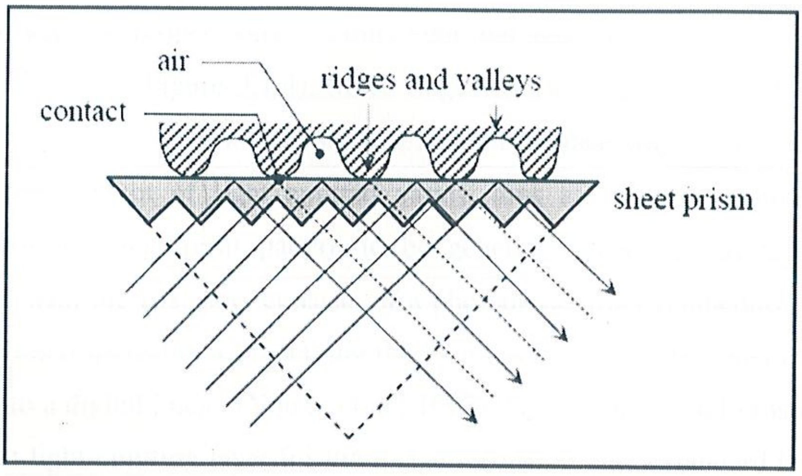


Figure 3.1.2 *The use of a sheet prism in FTIR fingerprint acquisition.*

- *Optical fibers*: a significant reduction of the packaging size can be achieved by substituting prism and lens with a fiber–optic platen (Fujieda, Ono, and Sugama (1995); Dowling and Knowlton (1988)). The finger is in direct contact with the upper side of the platen; on the opposite side, a CCD or CMOS, tightly coupled with the platen, receives the finger residual light conveyed through the glass fibers (see Figure 3.1.3). Unlike the FTIR devices, here the CCD/CMOS is in direct contact with the platen (without any intermediate lens), and therefore its size has to cover the whole sensing area. This may result in a high cost for producing large area sensors. A similar mi- crolens-based sensor has been proposed by Shogenji et al. (2004) based on compound-eye imaging principle.

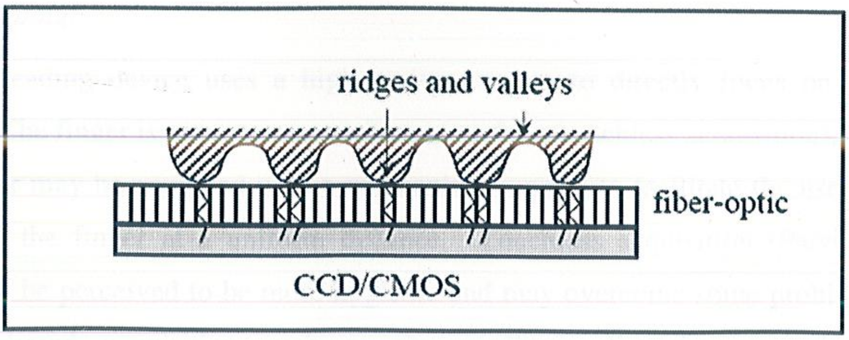


Figure 3.1.3 A sensor based on optical fibers. Residual light emitted by the finger is conveyed through micro-optical guides to the array of pixels that constitute the CCD/CMOS.

•*Electro-optical:*

These devices consist of two main layers; the first layer contains a polymer that, when polarized with the proper voltage, emits light that depends on the potential applied on one side (see Figure 3.1.4). Since ridges touch the polymer and the valleys do not, the potential is not the same across the surface when a finger is placed on it; the amount of light emitted varies, thus allowing a luminous representation of the fingerprint pattern to be generated. The second layer, strictly coupled with the first one, consists of a photodiode array (embedded in the glass) which is responsible for receiving the light emitted by the polymer and converting it into a digital image (Young et al., 1997). Some commercial sensors use just the first light-emitting layer for the image formation and a standard lens and CMOS for the image acquisition and digitization. In spite of substantial miniaturization, images produced by commercial scanners based on this technology are not comparable with the FTIR images in terms of quality.

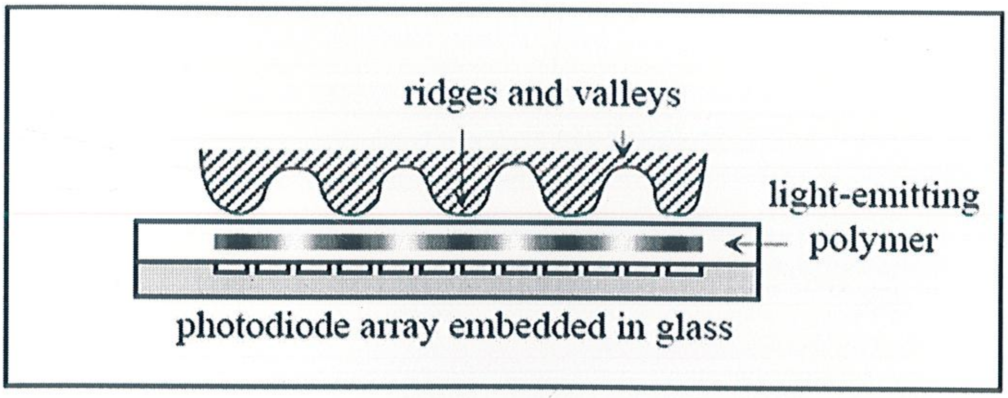


Figure 3.1.4 *Electro-optical fingerprint sensor.*

•*Direct reading:*

A direct reading device uses a high-quality camera to directly focus on the fingertip. The finger is not in contact with any surface (touchless acquisition), but the scanner may be equipped with a mechanical support to facilitate the user in presenting the finger at a uniform distance. Touchless acquisition (Parziale, 2007) may be perceived to be more hygienic and may overcome some problems of touch- based acquisition such as the nonlinear distortion caused by pressing the finger against the sensor platen and the need of periodically cleaning the sensor surface; however, obtaining well-focused and high-contrast images is still quite challenging with the touchless methods.

•*Multispectral imaging*:

Multispectral sensors capture multiple images of the same finger using different wavelengths of light, different illumination orientations, and different polarization conditions (Rowe, Nixon, and Butler, 2007). The resulting data can be processed to generate a single composite fingerprint image. Multispectral imaging is considered more robust than other acquisition techniques when fingerprint images are captured under adverse influences such as suboptimal skin condition and bright ambient light. Furthermore, features extracted from multispectral images are better suited in discriminating between real and fake fingers. On the other hand, multispectral imaging devices are more complex and expensive than conventional optical scanners and their adoption is still limited.

3.2 Solid-State Sensors

Although solid-state sensors (also known as silicon sensors) have been proposed in patent lit- erature since the 1980s, it was not until the middle 1990s that they became commercially vi- able (Xia and O'Gorman, 2003). Solid-state sensors were designed to overcome the size and cost problems which, at the time seemed to be a barrier against the wide spread deployment of fingerprint recognition systems in various consumer applications. All silicon-based sensors consist of an array of pixels, each pixel being a tiny sensor itself. The user directly touches the surface of the silicon: neither optical components nor external CCD/CMOS image sensors are needed. Four main technologies have been proposed to convert the fingerprint pattern into electrical signals: capacitive, thermal, electric field, and piezoelectric.

- *Capacitive:*

  This is the most common method used today within the silicon-based sensor arena. A capacitive sensor is a two-dimensional array of micro-capacitor plates embedded in a chip (see Figure 3.2.1).

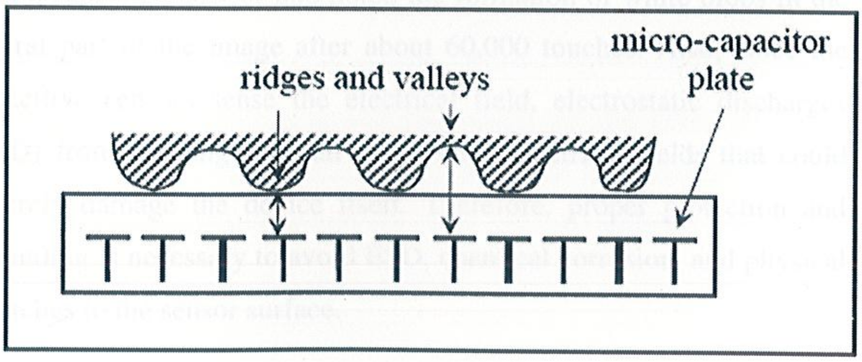


Figure 3.2.1 *Capacitive sensing.*

The other plate of each micro-capacitor is the finger skin itself. Small electrical charges are created between the surface of the finger and each of the silicon plates when a finger is placed on the chip. The magnitude of these electrical charges de- pends on the distance between the fingerprint surface and the capacitance plates (Tartagni and Guerrieri, 1998). Thus fingerprint ridges and valleys result in different capacitance patterns across the plates. An accurate capacitance measurement is quite difficult to make and adjust, and each vendor has its own method to get enough sensi- tivity to make a difference between the ridges and the valleys. The capacitive sensors, like the optical ones, cannot be easily deceived by presentation of a flat photograph or printed image of a fingerprint since they measure the distances and therefore only a three-dimensional surface can be sensed.

A critical component of capacitive sensors is the surface coating: the silicon chip needs to be protected from chemical substances (e.g., sodium) that are present in fin- ger perspiration. But a coating that is too thick increases the distance between the pixels and the finger, lowering the ability to discriminate between a ridge and a val- ley, especially for poor quality fingers, where the depth of a valley is in the range of a micron. As a result, the coating must be as thin as possible (a few microns), but not too thin, as it will not be resistant to mechanical abrasions. Yau, Chen, and Morguet (2004) performed stability tests with a capacitive sensor and noted the formation of white blobs in the central part of the image after about 60,000 touches. Also, since the capacitive sensors sense the electrical field, electrostatic discharges (ESD) from the fingertip can cause large electrical fields that could severely damage the device itself. Therefore, proper protection and grounding is necessary to avoid ESD, chemical corrosion, and physical scratches to the sensor surface.

An interesting property of capacitive sensors is the possibility of adjusting some electrical parameters to deal with non-ideal skin conditions (wet and dry fingers); a drawback is the need for frequently cleaning the surface to prevent the grease and dirt from compromising image quality.

- *Thermal*:

  These sensors are made of pyro-electric material that generates current based on temperature differentials. The fingerprint ridges, being in contact with the sensor surface, produce a different temperature differential than the valleys, which are at a distance from the sensor surface. The sensors are typically maintained at a high temperature by electrically heating them up, to increase the temperature difference between the sensor surface and the finger ridges. The temperature differential produces an image when contact occurs, but this image soon disappears because the thermal equilibrium is quickly reached and the pixel temperature is stabilized. Hence a sweeping may be necessary to acquire a stable fingerprint image. On the other hand, thermal sensing has some advantages: it is not sensitive to ESD and it can accept a thick protective coating (10–20 μm) because the thermal information (heat flow) can easily propagate through the coating.

- *Electric field*:

  In this arrangement (also known as RF imaging), the sensor consists of a drive ring that generates an RF (radio frequency) sinusoidal signal and a matrix of active antennas that receives a very small amplitude signal transmitted by the drive ring and modulated by the derma structure (subsurface of the finger skin). The finger must be simultaneously in contact with both the sensor and the drive ring. To image a fingerprint, the analog response of each (row, column) element in the sensor matrix is amplified, integrated, and digitized.

- *Piezoelectric*:

  Pressure-sensitive sensors have been designed that produce an electrical signal when mechanical stress is applied to them. The sensor surface is made of a non-conducting dielectric material which, on encountering pressure from the finger, generates a small amount of electric current (this effect is called the piezoelectric effect). The strength of the generated current depends on the pressure applied by the finger on the sensor surface. Since ridges and valleys are present at different distances from the sensor surface, they result in different amounts of current. Unfortunately, these materials are typically not sensitive enough to detect the difference and, moreover, the protective coating blurs the resulting image. An alternative solution is to use micro-mechanical switches (a cantilever made of silicon). Coating is still a problem and, in addition, this device delivers a binary image, leading to minimal information about the fingerprint pattern.

3.3 Ultrasound Sensors

Ultrasound sensing may be viewed as a kind of echography. It is based on sending acoustic signals toward the fingertip and capturing the echo signal (see Figure 2.11). The echo signal is used to compute the range (depth) image of the fingerprint and, subsequently, the ridge structure itself.

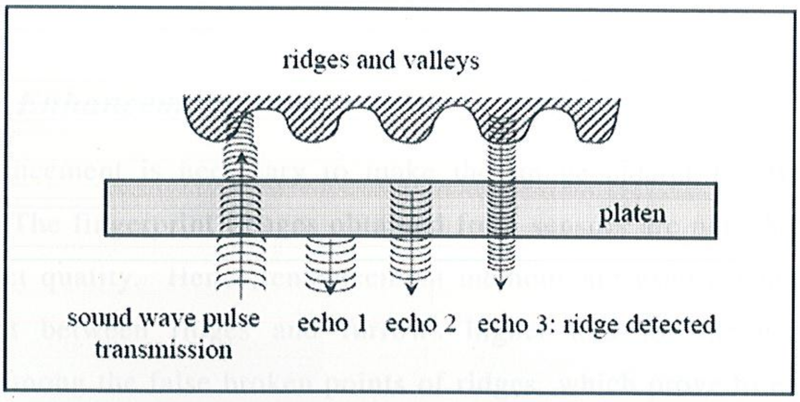


Figure 3.3.1 *The basic principle of the ultrasound technique. A characteristic of sound waves is their ability to penetrate materials, giving a partial echo at each impedance change.*

The ultrasound sensor has two main components: a transmitter, which generates short acoustic pulses, and a receiver, which detects the responses obtained when these pulses bounce off the fingerprint. This method images the subsurface of the finger skin (even through thin gloves); therefore, it is resilient to dirt and oil accumulations on the finger. While good quality images may be obtained by this technology, current ultrasound scanners are bulky with mechanical parts and quite expensive (several hundred dollars). Moreover, it takes a few seconds to acquire an im- age. Hence, this technology is not yet mature enough for large-scale deployment.

CHAPTER 4 : FINGERPRINT IMAGE PROCESSING

The fingerprint image is processed through a three step procedure. The image undergoes pre-processing, enhancement and post-processing. The three stages involve different steps and procedures which need to be discussed in detail.

4.1 Image Enhancement

Image enhancement is necessary to make the image clearer for further operations. The fingerprint images obtained from sensors are not likely to be of perfect quality. Hence, enhancement methods are used for making the contrast between ridges and furrows higher and for maintaining continuity among the false broken points of ridges, which prove to ensure a higher accuracy for recognition of fingerprint.

Generally two types of procedures are adopted for image enhancement:

1) Histogram Equalization
2) Fourier Transform

4.1.1 Histogram Equalization

Histogram equalization is responsible for expanding the pixel distribution of an image in order to increase perceptional improvement. The pictorial description is given below. The fingerprint initially has a bimodal type histogram as shown in fig 4.1.1.1. After histogram equalization is carried out, the image occupies the entire range from zero to 255, enhancing the visualization effect in the process.

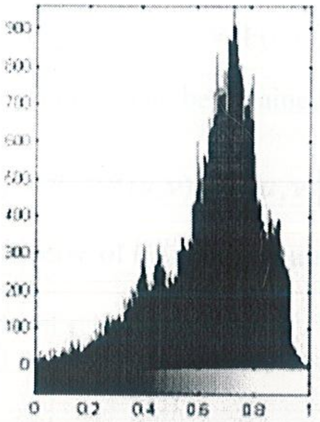


Figure 4.1.1.1 Fingerprint with original Histogram

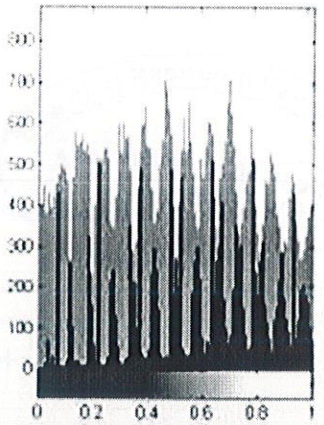
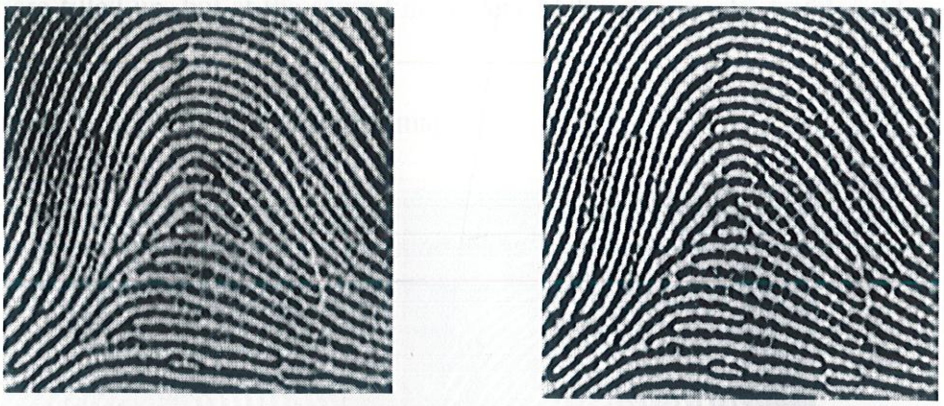


Figure 4.1.1.2 After histogram equalization



Original Image

Enhanced Image

Figure 4.1.1.3 *Effect of Histogram equalization*

4.1.2 Using Fourier Transform

In this process of enhancement the image is divided into small processing blocks (32 x 32 pixels) and Fourier transform is performed.

The function is as follows:

$$F(u,v) = \sum_{x=0}^{M-1}\sum_{y=0}^{N-1} f(x,y) \times \exp\left\{-j2\pi \times \left(\frac{ux}{M} + \frac{vy}{N}\right)\right\}$$

For $u = 0,1,2, \ldots ,31$ $v = 0,1,2, \ldots ,31$

For enhancing a particular block by its dominant frequencies, the FFT os the block is multiplied by its magnitude a few times. Where the magnitude of the FFT is given by abs $F(u,v) = |F(u,v)|$.

The enhanced block can be obtained as per

$$g(x,y) = F^{-1}\left\{F(u,v) \times |F(u,v)|^{k}\right\} \quad (2),$$

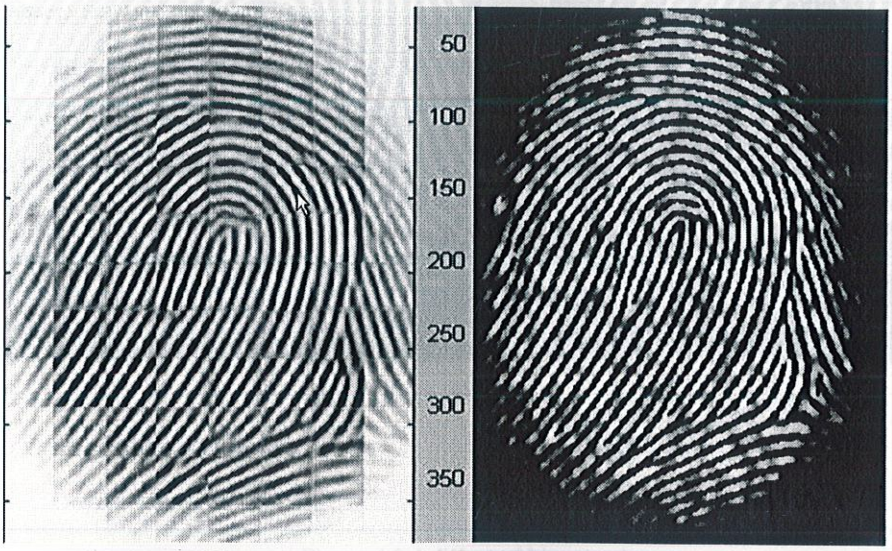
where the inverse of $(F(u,v))$ is found by:

$$f(x,y) = \frac{1}{MN}\sum_{x=0}^{M-1}\sum_{y=0}^{N-1} F(u,v) \times \exp\left\{ j2\pi \times \left(\frac{ux}{M} + \frac{vy}{N}\right)\right\} \quad (3)$$

for $x = 0, 1, 2, ..., 31$ & $y = 0, 1, 2, ..., 31$.

The k is a constant whose value has been experimentally found.Here,k is chosen as 0.45. When k is higher, the ridges appear improved, since the holes in the ridges are filled up, but at the same time a very high value results in false ridge joining.

Figure 4.1.2.1 depicts FFT enhanced image.



Before Enhancement After Enhancement

Figure 4.1.2.1 *FFT enhanced fingerprint image (Source: Davide Maltoni, Dario Maio, Anil K. Jain, Salil Prabhakar, Handbook of Fingerprint recognition*

The image after enhancement connects falsely broken points on the ridges and removes spurious connections in between the ridges.

4.2 Image Binarization

The original image is a 8-bit grayscale image. This process transforms the original image into a 1-bit image that assigns values 0 for ridges and 1 for furrows. After binarization, the ridges appear black while the furrows appear white.

Binarization changes the pixel value to 1 if the value is found to exceed the mean intensity of the current block to which it belongs.

The figure clearly depicts the effect of binarization on a normal grayscale image that has been only enhanced.

Binarized Image **Gray image**

Figure 4.2.1 *Effect of binarization*

CHAPTER 5 : FINGERPRINT MATCHING

5.1 Introduction

A fingerprint matching algorithm compares two given fingerprints and returns either a degree of similarity (without loss of generality, a score between 0 and 1) or a binary decision (mated/non-mated). Only a few matching algorithms operate directly on grayscale fingerprint images; most of them require that an intermediate fingerprint representation be derived through a feature extraction stage. Without loss of generality, hereafter we denote the representation of the fingerprint acquired during enrollment as the *template* (**T**) and the representation of the fingerprint to be matched as the *input* (**I**). In case no feature extraction is performed, the fingerprint representation coincides with the grayscale fingerprint image itself; hence, throughout this chapter, we denote both raw fingerprint images and fingerprint feature vectors (e.g., minutiae) with **T** and **I**.

The fingerprint feature extraction and matching algorithms are usually quite similar for both fingerprint verification and identification problems. This is because the fingerprint identification problem (i.e., searching for an input fingerprint in a database of *N* fingerprints) can be implemented as a sequential execution of *N* one-to-one comparisons (verifications) between pairs of fingerprints. The fingerprint classification and indexing techniques are usually exploited to speed up the search in fingerprint identification problems.

Matching fingerprint images is a very difficult problem, mainly due to the large variability in different impressions of the same finger (i.e., large *intra-class* variations). The main factors responsible for intra-class variations are summarized below.

- *Displacement*: the same finger may be placed at different locations on a touch sensor during different acquisitions resulting in a (global) translation of the fingerprint area. A finger displacement of just 2 mm (imperceptible to the user) results in a translation of about 40 pixels in a fingerprint image scanned at a resolution of 500 dpi.

- *Rotation*: the same finger may be rotated at different angles with respect to the sensor surface during different acquisitions. In spite of the finger "guide" mounted in certain commercial scanners, involuntary finger rotations of up to ±20° with respect to vertical orientation can be observed in practice.

- *Partial overlap*: finger displacement and rotation often cause part of the fingerprint area to fall outside the sensor's "field of view," resulting in a smaller overlap between the foreground areas of the template and the input fingerprints. This problem is particularly serious for small-area touch sensors.

- *Non-linear distortion*: the act of sensing maps the three-dimensional shape of a finger onto the two-dimensional surface of the sensor. This mapping results in a non-linear distortion in successive acquisitions of the same finger due to skin plasticity. Often, fingerprint matching algorithms disregard the characteristic of such a mapping, and consider a fingerprint image as non-distorted by assuming that it was produced by a correct finger placement; a finger placement is correct when: (i) the trajectory of the finger approaching the sensor is orthogonal to the sensor surface; (ii) once the finger touches the sensor surface, the user does not apply traction or torsion. However, due to skin plasticity, the components of the force that are non-orthogonal to the sensor surface produce non-linear distortions (compression or stretching) in the acquired fingerprints. Distortion results in the inability to match fingerprints as rigid patterns.

- *Pressure and skin condition*: the ridge structure of a finger would be accurately captured if ridges of the part of the finger being imaged were in uniform contact with the sensor surface. However, finger pressure, dryness of the skin, skin disease, sweat, dirt, grease, and humidity in the air all confound the situation, resulting in a non- uniform contact. As a consequence, the acquired fingerprint images are very noisy and the noise strongly varies in successive acquisitions of the same finger depending on the magnitude of the above cited causes.

- *Noise*: it is mainly introduced by the fingerprint sensing system; for example, residues are left over on the glass platen from the previous fingerprint capture.

- *Feature extraction errors*: the feature extraction algorithms are imperfect and often introduce measurement errors. Errors may be made during any of the feature extraction stages (e.g., estimation of orientation and frequency images, detection of the number, type, and position of the singularities, segmentation of the fingerprint area from the background, etc.). Aggressive enhancement algorithms may introduce in- consistent biases that perturb the location and orientation of the reported minutiae from their gray-scale counterparts. In low-quality fingerprint images, the minutiae ex- traction process may introduce a large number of spurious minutiae and may not be able to detect all the true minutiae.

5.2 Matching Techniques

Approaches to fingerprint matching can be coarsely classified into three families.

- *Correlation-based matching*: two fingerprint images are superimposed and the correlation between the corresponding pixels is computed for different alignments (e.g., various displacements and rotations).

- *Minutiae-based matching*: this is the most popular and widely used technique, being the basis of the fingerprint comparison made by fingerprint examiners. Minutiae are extracted from the two fingerprints and stored as sets of points in the two- dimensional plane. Minutiae-based matching essentially consists of finding the alignment between the template and the input minutiae feature sets that result in the maximum number of minutiae pairings.

- *Non-Minutiae feature-based matching*: minutiae extraction is difficult in extremely low-quality fingerprint images. While some other features of the fingerprint ridge pattern (e.g., local orientation and frequency, ridge shape, texture information) may be extracted more reliably than minutiae, their distinctiveness as well as persistence is generally lower. The approaches belonging to this family compare fingerprints in term of features extracted from the ridge pattern. In principle, correlation-based matching could be conceived of as a subfamily of non-minutiae feature-based match- ing, inasmuch as the pixel intensity are themselves features of the finger pattern.

5.3 Matching Technique used in the project

The widely used minutiae-based representation does not utilize a significant component of the rich discriminatory information available in the fingerprints. Local ridge structures cannot be completely characterized by minutiae. Further, minutiae-based matching has difficulty in quickly matching two fingerprint images containing different number of unregistered minutiae points.

The proposed filter-based algorithm uses a bank of Gabor filters to capture both local and global details in a fingerprint as a compact fixed length FingerCode. The fingerprint matching is based on the Euclidean distance between the two corresponding FingerCodes and hence is extremely fast. We will be able to achieve a verification accuracy which is only marginally inferior to the best results of minutiae-based algorithms published in the open literature.

5.4 Matching Algorithm

- The most popular technique to match fingerprints based on texture information remains the FingerCode approach.

- The fingerprint area of interest is tessellated (close arrangement of polygons, specially in a repeated pattern) with respect to the core point. A feature vector is composed of an ordered enumeration of the features extracted from the local information contained in each sector specified by the tessellation.

- The local texture information in each sector is decomposed into separate channels by using a Gabor filterbank; in fact, the Gabor filterbank is a well-known technique for capturing useful texture information in specific bandpass channels as well as decomposing this information into biorthogonal components in terms of spatial frequencies .

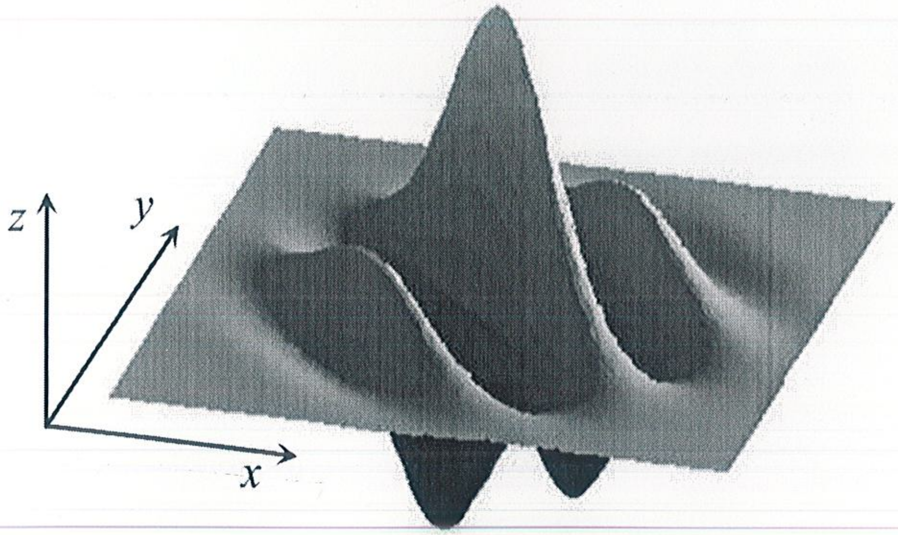


Figure 5.2.1 *Gabor Filter Visualisation*

- We obtain the results by tessellating the area of interest into 64 cells (four bands and 16 sectors), and by using a bank of eight Gabor filters.

- Therefore, each fingerprint is represented by a 64 × 8 = 512 fixed-size feature vector, called the *FingerCode.*

FINGCODE APPROACH BLOCK DIAGRAM

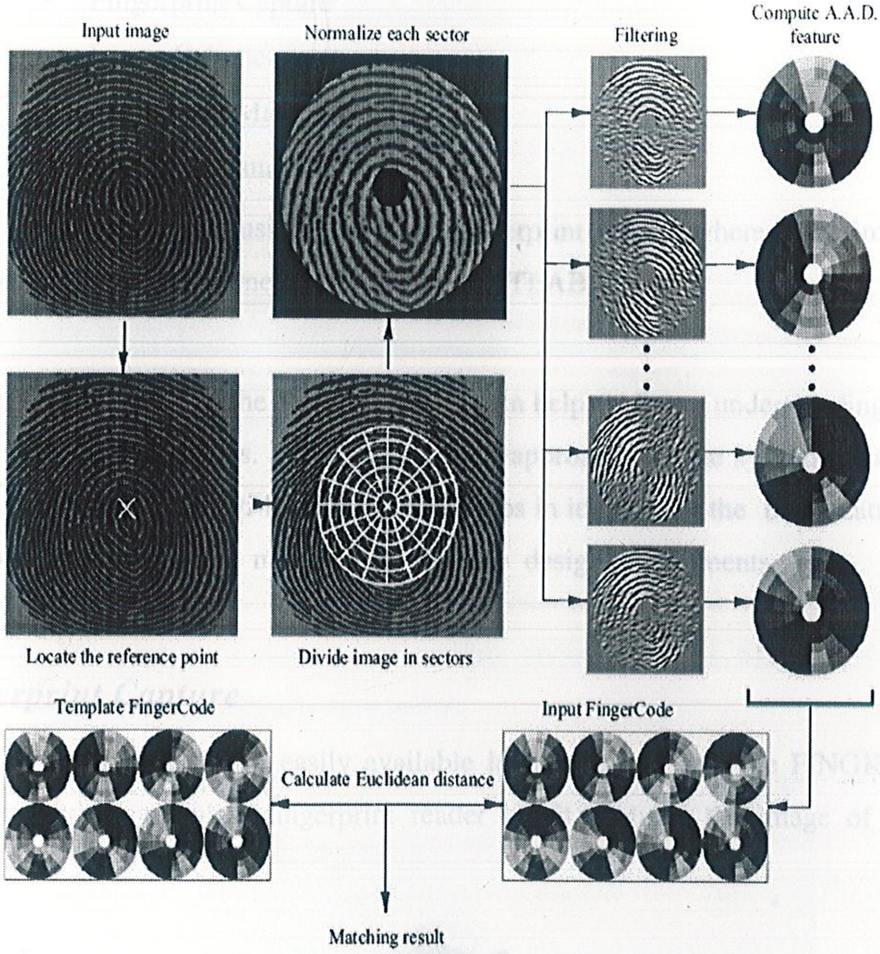


Figure 5.2.2 *FingerCode approach by Jain*

CHAPTER 6: SYSTEM DESIGN

The design of fingerprint based attendance system can be divided into the 4 different modules. They are –

- Fingerprint Capture
- Image Enhancement
- Fingerprint Matching
- Database Management

Fingerprint capturing is done using an optical fingerprint reader, whereas the image processing and matching are done on a PC using MATLAB.

The module-wise approach to the design of the system helps in better understanding of the individual function levels. Also, a parallel approach to the system helps in distributing the effort on a multi-level range and helps in identifying the best features and available products in the market that suit the design requirements.

6.1 Fingerprint Capture

The fingerprint capturing device easily available in the country was the FINGKEY HAMSTER. This is an opical fingerprint reader i.e. it captures the image of the fingerprint.

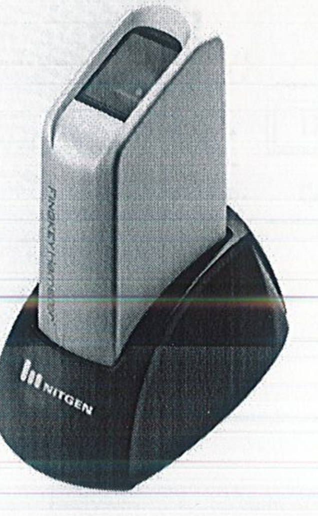
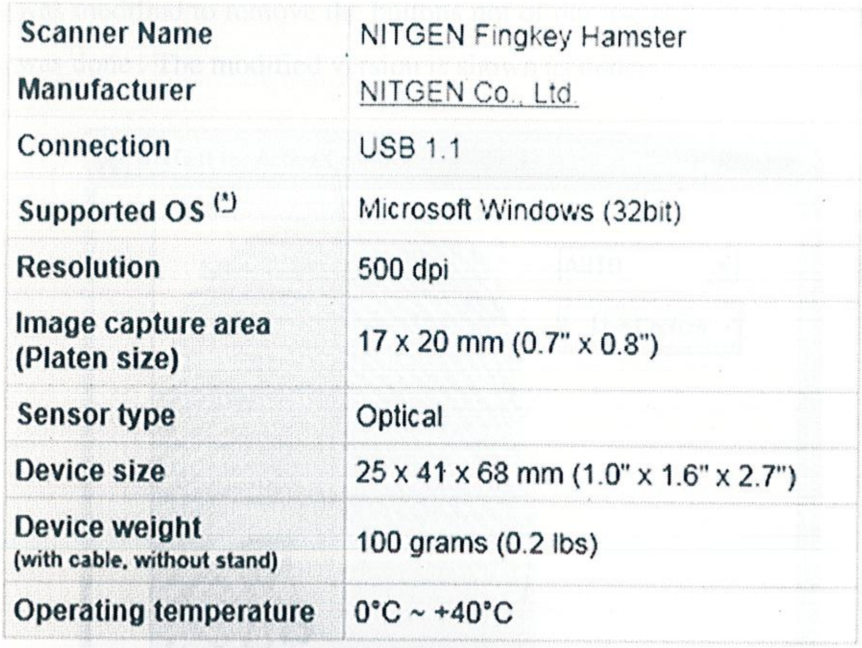


Figure 6.1.1 *The NITGEN Fingkey Hamster DX optical reader to capture fingerprints*

The specifications of the device are as follows :



Scanner Name	NITGEN Fingkey Hamster
Manufacturer	NITGEN Co., Ltd.
Connection	USB 1.1
Supported OS (*)	Microsoft Windows (32bit)
Resolution	500 dpi
Image capture area (Platen size)	17 x 20 mm (0.7" x 0.8")
Sensor type	Optical
Device size	25 x 41 x 68 mm (1.0" x 1.6" x 2.7")
Device weight (with cable, without stand)	100 grams (0.2 lbs)
Operating temperature	0°C ~ +40°C

Figure 6.1.2 *The NITGEN Fingkey Hamster DX specifications*

The module came with an SDK (software development kit) based on visual basic programming. We plan to integrate a modified version of this SDK with our MATLAB coding. The original SDK is as under:

```
0) SecuAPIERROR_NONE
```

6.1.3 fingerprint live capture

This SDK was modified to remove the buttons not of our use and auto initialization of the device was done . The modified version is shown as under:

0) SecuAPIERROR_NONE

6.1.4 modified fingerprint live capture

6.2 Image Enhancement

- The image captured by the reader is then enhance using various techniques such as fast Fourier transform and Image binarization.

- The final enhanced image appears as under

- This has been discussed in chapter 4 in detail.

6.3 Fingerprint Matching

The fingerprint matching is done on a one to one matching basis. This type of matching involves matching the captured image with the database of stored images.

This is done by the matlab functions *whichsector.m, sector_norm.m, gabor2d_sub.m, and conv2fft.m* which perform the function of matching the fingerprint using the algorithm explained in chapter 5.

6.4 Database Management

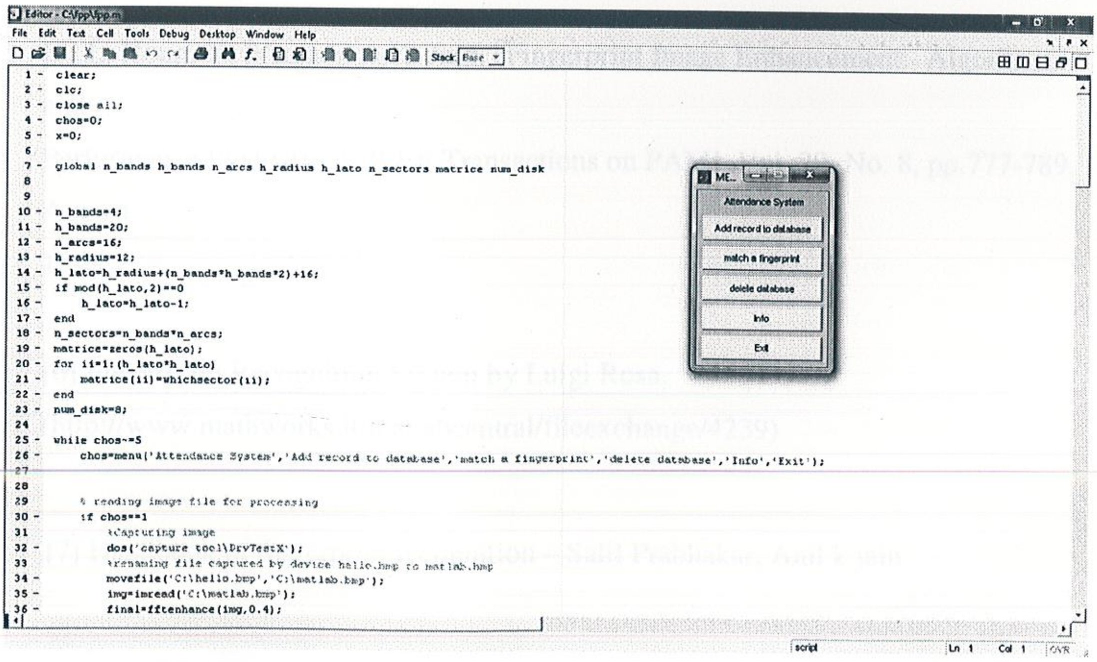
- The database is created by using the **save** command.
- Changes are made to the file, only after loading it to the workspace by using the **load** command
- `save('fp_database.dat','data','fp_number','-append');`
- `load('fp_database.dat','-mat');`
- The database is deleted using the **delete** command
- `delete('fp_database.dat');`

CHAPTER 7 : CONCLUSION

Initial progress is mentioned below:

- **i.** The optical fingerprint reader was tested for proper functioning. The device was found to work properly
.
- **ii.** A demo software was run on the fingerprint module and its operation was analyzed. It was observed to be an Enroll-Once-Verify-Once software. The threshold for content matching was very low and flexibility for different orientations of the finger was not present.
- **iii.** Established connection between the fingerprint capturing device and the PC.

The main objective of the project then was to enroll fingerprints of different students and add them to the database which would be referred at the time of verification. This was successfully achieved and is shown by the screenshot below.



```
clear;
clc;
close all;
chos=0;
x=0;

global n_bands h_bands n_arcs h_radius h_lato n_sectors matrice num_disk


n_bands=4;
h_bands=20;
n_arcs=16;
h_radius=12;
h_lato=h_radius+(n_bands*h_bands*2)+16;
if mod(h_lato,2)==0
    h_lato=h_lato-1;
end
n_sectors=n_bands*n_arcs;
matrice=zeros(h_lato);
for ii=1:(h_lato*h_lato)
    matrice(ii)=whichsector(ii);
end
num_disk=8;

while chos~=5
    chos=menu('Attendance System','Add record to database','match a fingerprint','delete database','Info','Exit');


    % reading image file for processing
    if chos==1
        %Capturing image
        dos('capture tool\DrvTestX');
        %renaming file captured by device hello.bmp to matlab.bmp
        movefile('C:\hello.bmp','C:\matlab.bmp');
        img=imread('C:\matlab.bmp');
        final=fftenhance(img,0.4);
```

7.1 running MATLAB program

CHAPTER 8 : REFERENCES

[1] *Younhee Gil,* Access Control System with high level security using fingerprints,IEEE the 32nd Applied Imagery Pattern Recognition Workshop (AIPR '03)

[2] Jain, A.K., Hong, L., and Bolle, R.(1997), "On-Line Fingerprint Verification," IEEE Trans. On Pattern Anal and Machine Intell, 19(4), pp. 302-314.

[3] D.Maio and D. Maltoni. Direct gray-scale minutiae detection in fingerprints. IEEE Trans. Pattern Anal. And Machine Intell., 19(1):27-40, 1997.

[4] Lee, C.J., and Wang, S.D.: Fingerprint feature extration using Gabor filters, Electron. Lett., 1999, 35, (4), pp.288-290.

[5] L. Hong, Y. Wan and A.K. Jain, "Fingerprint Image Enhancement: Algorithms and
Performance Evaluation", IEEE Transactions on PAMI ,Vol. 20, No. 8, pp.777-789, August
1998.

[6] Fingerprint Recognition System by Luigi Rosa,
(http://www.mathworks.it/matlabcentral/fileexchange/4239)

[7] Handbook of fingerprint recognition – Salil Prabhakar, Anil k jain

SOURCE CODE

Main m-file fpp.m

```
clear;
clc;
close all;
chos=0;
x=0;


global n_bands h_bands n_arcs h_radius h_lato n_sectors matrice num_disk


n_bands=4;
h_bands=20;
n_arcs=16;
h_radius=12;
h_lato=h_radius+(n_bands*h_bands*2)+16;
if mod(h_lato,2)==0
   h_lato=h_lato-1;
end
n_sectors=n_bands*n_arcs;
matrice=zeros(h_lato);
for ii=1:(h_lato*h_lato)
   matrice(ii)=whichsector(ii);
end


num_disk=8;


while chos~=5
```

```
chos=menu('Attendance System','Add record to database','match a fingerprint','delete
database','Info','Exit');


 % reading image file for processing
  if chos==1
     %Capturing image
     dos('capture tool\DrvTestX');
     %renaming file captured by device hello.bmp to matlab.bmp
     movefile('C:\hello.bmp','C:\matlab.bmp');
     img=imread('C:\matlab.bmp');
     final=fftenhance(img,0.4);
     new=binarization(final,8);
     [NormalizedPrint,vector]=sector_norm(new,0);
     for (angle=0:1:num_disk-1)
        gabor=gabor2d_sub(angle,num_disk);
        ComponentPrint=conv2fft(new,gabor,'same');
        [disk,vector]=sector_norm(ComponentPrint,1);
        finger_code1{angle+1}=vector(1:n_sectors);
     end


     img=imrotate(img,180/(num_disk*2));
     fingerprint=double(img);

      final=fftenhance(img,0.4);
      new=binarization(final,8);
      [NormalizedPrint,vector]=sector_norm(new,0);

     for (angle=0:1:num_disk-1)
        gabor=gabor2d_sub(angle,num_disk);
        ComponentPrint=conv2fft(NormalizedPrint,gabor,'same');
        [disk,vector]=sector_norm(ComponentPrint,1);
        finger_code2{angle+1}=vector(1:n_sectors);
```

```
  end

  % FingerCode added to database
  if (exist('fp_database.dat')==2)
    load('fp_database.dat','-mat');
    fp_number=fp_number+1;
    data{fp_number,1}=finger_code1;
    data{fp_number,2}=finger_code2;
    save('fp_database.dat','data','fp_number','-append');
  else
    fp_number=1;
    data{fp_number,1}=finger_code1;
    data{fp_number,2}=finger_code2;
    save('fp_database.dat','data','fp_number');
  end
  clc;
  close all;
  message=strcat('FingerCode was succesfully added to database.
     Fingerprint no.',num2str(fp_number));
  msgbox(message,'FingerCode DataBase','help');

end

  if chos==2
  clc;
  close all;
  %Capturing image
  dos('capture tool\DrvTestX');

  %renaming file captured by device hello.bmp to matlab.bmp
  movefile('C:\hello.bmp','C:\matlab.bmp');
  img2=imread('C:\matlab.bmp');
  final=fftenhance(img2,0.4);
  new=binarization(final,8);
```

```
[NormalizedPrint,vector]=sector_norm(new,0);

N=h_lato;

vettore_in=zeros(num_disk*n_sectors,1);
for (angle=0:1:num_disk-1)
   gabor=gabor2d_sub(angle,num_disk);
     ComponentPrint=conv2fft(new,gabor,'same');
   [disk,vector]=sector_norm(ComponentPrint,1);
   finger_code{angle+1}=vector(1:n_sectors);
   vettore_in(angle*n_sectors+1:(angle+1)*n_sectors)=finger_code{angle+1};
end



% FingerCode of input fingerprint has just been calculated.
% Checking with DataBase
if (exist('fp_database.dat')==2)
   load('fp_database.dat','-mat');
   vettore_a=zeros(num_disk*n_sectors,1);
   vettore_b=zeros(num_disk*n_sectors,1);
   best_matching=zeros(fp_number,1);
   valori_rotazione=zeros(n_arcs,1);
   % start checking ---------------------------------------
   for scanning=1:fp_number
      fcode1=data{scanning,1};
      fcode2=data{scanning,2};
      for rotazione=0:(n_arcs-1)
         p1=fcode1;
         p2=fcode2;
         % ruoto i valori dentro disco
         for conta_disco=1:num_disk
            disco1=p1{conta_disco};
            disco2=p2{conta_disco};
```

```
   for old_pos=1:n_arcs
      new_pos=mod(old_pos+rotazione,n_arcs);
      if new_pos==0
         new_pos=n_arcs;
      end
      for conta_bande=0:1:(n_bands-1)

   disco1r(new_pos+conta_bande*n_arcs)=disco1(old_pos+conta_bande*
   n_arcs);

   disco2r(new_pos+conta_bande*n_arcs)=disco2(old_pos+conta_bande*
   n_arcs);
   end
   end
   p1{conta_disco}=disco1r;
   p2{conta_disco}=disco2r;
end
% ruoto i dischi circolarmente
for old_disk=1:num_disk
   new_disk=mod(old_disk+rotazione,num_disk);
   if new_disk==0
      new_disk=num_disk;
   end
   pos=old_disk-1;
   vettore_a(pos*n_sectors+1:(pos+1)*n_sectors)=p1{new_disk};
   vettore_b(pos*n_sectors+1:(pos+1)*n_sectors)=p2{new_disk};
end
d1=norm(vettore_a-vettore_in);
d2=norm(vettore_b-vettore_in);
if d1<d2
   val_minimo=d1;
else
   val_minimo=d2;
end
```

```
            valori_rotazione(rotazione+1)=val_minimo;
          end
          [minimo,posizione_minimo]=min(valori_rotazione);
          best_matching(scanning)=minimo;
        end
        [distanza_minima,posizione_minimo]=min(best_matching);
        beep;
        message=strcat('The nearest fingerprint present in DataBase which matches
         input fingerprint is  : ',num2str(posizione_minimo),...
            ' with a distance of : ',num2str(distanza_minima));
        msgbox(message,'DataBase Info','help');

    else
        message='DataBase is empty. No check is possible.';
        msgbox(message,'FingerCode DataBase Error','warn');
    end

end

if chos==3
    clc;
    close all;
    if (exist('fp_database.dat')==2)
        button = questdlg('Do you really want to remove the Database?');
        if strcmp(button,'Yes')
            delete('fp_database.dat');
            msgbox('Database was succesfully removed from the current
            directory.','Database removed','help');
        end
    else
        warndlg('Database is empty.',' Warning ')
    end
end
```

```
% info

  if chos==4
    clc;
    close all;
    msgbox('********    Fingerprint Based Attendance system    ********
Developed by:                                          Ankur Sharma (071117)  Ankur
Kathayat (071080)','DEVELOPERS','help');
  end
end
```

binarization.m

```
function [o] = binarization(a,W);

[w,h] = size(a);
o = zeros(w,h);

%seperate it to W block
%step to w with step length W

for i=1:W:w
for j=1:W:h
mean_thres = 0;

%white is ridge -> large

if i+W-1 <= w & j+W-1 <= h
  mean_thres = mean2(a(i:i+W-1,j:j+W-1));
  %threshold value is choosed
   mean_thres = 0.8*mean_thres;
   %before binarization
   %ridges are black, small intensity value -> 1 (white ridge)
   %the background and valleys are white, large intensity value -> 0(black)
   o(i:i+W-1,j:j+W-1) = a(i:i+W-1,j:j+W-1) < mean_thres;
end;

end;
end;
```

centralizaing.m snippet

```
 posizioni=zeros(size(temp2));
  posizioni(max(1,x0-dx):min(size(temp2,1),x0+dx),max(1,y0-
dy):min(size(temp2,2),y0+dy))=1;
  temp2=temp2.*posizioni;

  [massimo_vettore,posizione_vettore]=max(temp2);
  [massimo,posizione]=max(massimo_vettore);
  y_max=posizione;
  x_max=posizione_vettore(posizione);
  massimo;

  x0=2*x_max;
  y0=2*y_max;
  dx=10;
  dy=10;
  posizioni=zeros(size(temp1));
  posizioni(max(1,x0-dx):min(size(temp1,1),x0+dx),max
   (1,y0dy):min(size(temp1,2),y0+dy))=1;
  temp1=temp1.*posizioni;
  [massimo_vettore,posizione_vettore]=max(temp1);
  [massimo,posizione]=max(massimo_vettore);
  y_max=posizione;
  x_max=posizione_vettore(posizione);
  massimo;.

  x0=2*x_max;
  y0=2*y_max;

  dx=5;
  dy=5;
```

```
posizioni=zeros(size(temp0));
posizioni(max(1,x0-dx):min(size(temp0,1),x0+dx),max
      (1,y0-dy):min(size(temp0,2),y0+dy))=1;
temp0=temp0.*posizioni;

[massimo_vettore,posizione_vettore]=max(temp0);
[massimo,posizione]=max(massimo_vettore);
y_max=posizione;
x_max=posizione_vettore(posizione);
massimo;

disp('Coordinate x y');
disp(x_max);
disp(y_max);
XofCenter=y_max;
YofCenter=x_max;
Outputprint=zeros(50);
```

Fftenhance.m

```
function [final]=fftenhance(image,f)
I = 255-double(image);

[w,h] = size(I);



w1=floor(w/32)*32;
h1=floor(h/32)*32;

inner = zeros(w1,h1);

for i=1:32:w1
  for j=1:32:h1
    a=i+31;
    b=j+31;
    F=fft2( I(i:a,j:b) );
    factor=abs(F).^f;
    block = abs(ifft2(F.*factor));

    larv=max(block(:));
    if larv==0
      larv=1;
    end;

    block= block./larv;
    inner(i:a,j:b) = block;
  end;
end;

final=inner*255;
final=histeq(uint8(final));
```

whichsector.m

```
function [sector_num] = whichsector(index)


global immagine n_bands h_bands n_arcs h_radius h_lato n_sectors matrice


length = h_lato;

x = rem( index , length );

y = floor(index / length);


x = x - floor(length / 2);

y = y - floor(length / 2);


rad = (x*x) + (y*y);

if rad < (h_radius*h_radius)                    % innerest radius = 12 (144=12*12)

   sector_num = (n_sectors-1)+1;

   sector_num;

   return

end


if rad >= (h_bands*n_bands+h_radius)^2         % outtest radius = 72 (5184=72*72)

   sector_num = (n_sectors-1)+2;

   sector_num;

   return

end


if x ~= 0

   theta = atan( y / x );
```

```
else
   if y > 0
      theta = pi/2;
   else
      theta = -pi/2;
   end
end

if x < 0
   theta = theta + pi;
else
   if theta < 0
      theta = theta + 2*pi;
   end
end

if theta < 0
   theta = theta + 2*pi;
end
r = floor(rad ^ 0.5);
ring = floor(( r-h_radius )/h_bands);
arc = floor(theta /(2*pi/n_arcs));
 sector_num = ring * n_arcs + arc;
```