A Minutiae Based Fingerprint Matching Algorithm Using Fuzzy Logic

By

Khurram Yasin Qureshi

2006-NUST-MS PhD-CSE (E)-10

Submitted to the Department of Computer Engineering
in partial fulfillment of the requirements for the degree of

Master of Science
in
Computer Software Engineering

Advisor:
Dr. Shoab Ahmad Khan

College of Electrical & Mechanical Engineering
National University of Sciences and Technology
2009

ACKNOWLEDGMENTS

I am thankful to Allah Almighty for giving me the ability to finish this research work and allow me to acquire knowledge from it.

I am thankful to my supervisor Dr. Shoab Ahmad Khan for his guidance and help and all the committee members for their time and diligence.

Finally, I want to thank my parents and my friends for the continuous encouragement, understanding and moral support.

I would like to dedicate this thesis to my

teachers, family and friends.

ABSTRACT

The use of biometrics is evolving day by day in our society. Fingerprint recognition is well known for its high acceptability and popularity in the world of biometric systems. Fingerprint recognition has its applications in many fields such as banking, medical and insurance industry, police department, database management systems, identity authentication, police department, border security and many other areas. There are different methods and techniques used for matching fingerprints but the most common and popular approach is minutiae based matching. Our approach is based on structural matching and the matching algorithm presented here is the improved and modified form of [1]. In this method, matching is done on the basis of five closest neighbors of one single minutia that is also called a center minutia. An authentication of minutia is based on these surrounding neighbors. The approach we present here is divided in to two stages, first stage performs initial filtration and the second stage includes special matching criteria that incorporate fuzzy logic as well as a novel feature to select final minutiae for matching score calculation. The method of selecting center point for second stage is also adopted. This algorithm is able to perform well for translated, rotated and stretched fingerprints and does not require any process for alignment before matching. Two error rates (FAR and FRR) are used to represent the performance of an algorithm. Tests have been carried out on the standard database FVC 2002 (DB1_A) using P-IV (1.8MHz) with 512MB of RAM. MALAB 7.0 has been used for the implementation of proposed algorithm. Experimental results show that algorithm is fast, efficient and reliable.

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

Chapter 1

INTRODUCTION

1.1 Biometrics

Biometrics is a science of recognizing individuals based on their physiological (iris, face, fingerprints, retina etc) and behavioral traits (signature, gaits, keystroke dynamics etc) [2]. The division of biometrics on the basis of different traits is represented in Figure 1.1. Since the biometric systems are permanently associated with an individual so they are more reliable as compared to other authentication systems.

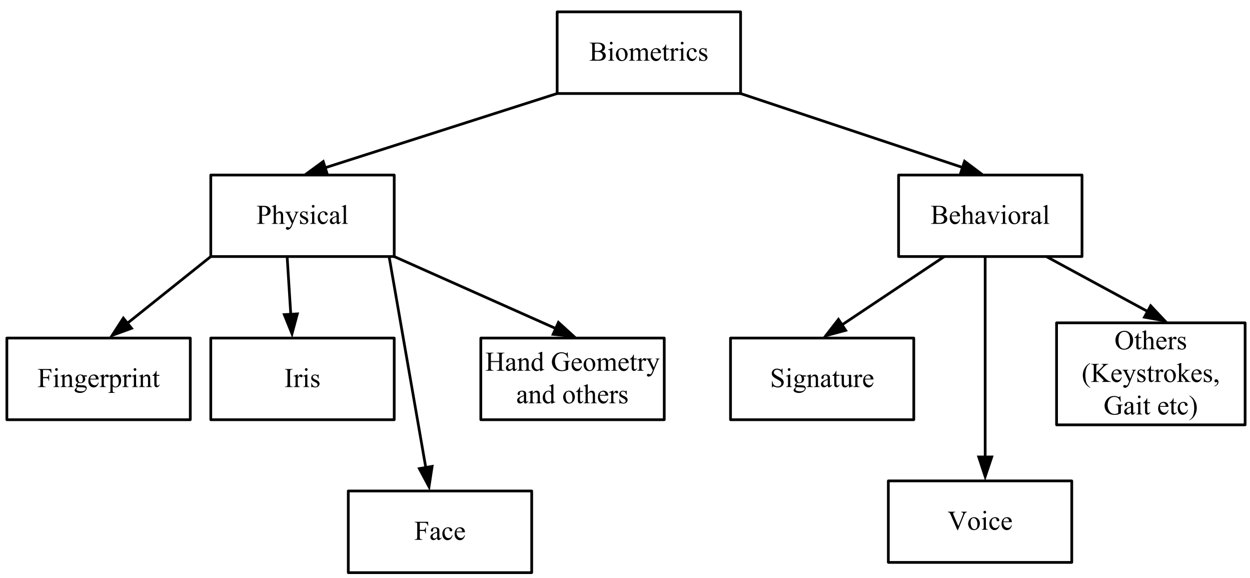


Figure 1.1: Division of biometrics in to different types

The use of biometrics is evolving day by day in our society. Fingerprint readers are available in various laptops, iris scanners are being installed at high risk security places as well as voice recognition software is also helpful for detection in various automobiles. As the security concerns are increasing all over the world, the use of biometrics will remain effective for coming years as well. To become an effective biometric trait, it must possess four qualities. The first one is universality which means that every person in the universe must have this characteristic. Second one is distinctiveness that means that there should be sufficient distinction in the characteristics between two different individuals. Third one is permanence which is concerned with invariability of trait. The feature should be easy to collect and we can compare it quantitatively and this is the fourth quality that is required [3].

1.1.1 Previous authentication systems

Two methodologies were mainly used for verification purpose. These are explained in detail as under.

1.1.1.1 Password based approach

This approach has been used for decades in computer systems and was considered reliable at that time but this system is losing its effectiveness in present era. The problem is that humans are unable to remember long passwords and short passwords are easy to break by current high performing systems. For password to be secure, the human should not be able to get password even after large number of attempts. Unfortunately studies have shown that length of password should be more than seven digits for its security which makes the password difficult to remember [4] whereas humans are able to remember five to nine digits in their short-term memory at one time. The mentioned approach is also called a knowledge based approach because you have knowledge of password which you are using to access different systems [5].

1.1.1.2 Token based approach

The second approach asks for some personal belonging such as smart cards, ID cards and passports etc. These are also called tokens and have been used for a number of years to get access to high security places. The problem with this approach is that any person who has this token becomes authentic so this approach authenticates a token not an individual. If an imposter steals the ID card of an authorized user then there is no mechanism to block that user to enter restricted area.

1.1.2 Operating modes for biometric systems

There are two main operating modes used by biometric systems that are explained below in detail.

1.1.2.1 Verification

In this mode, the person claims an identity through an identification number. The input data is taken up by the system and compared with stored information of that particular person. This previously stored information is also called template information. This type of comparison is actually one to one matching in which the system tries to verify that the person is one who he claims to be. If the input information fails to match with stored one then person will be specified as unauthenticated by the system.

1.1.2.2 Identification

In this mode the system compares the input data to all the templates stored in the system. This type of comparison is same as one to many matching. This mode is usually adopted to match criminal biometrics with the existing database. The person will be granted access to the system if his record matches with any authentic record.

There is no need to claim an initial identity in this mode of authentication. This mechanism is computationally more expensive as compared to verification mode because one record needs to be compared with each record stored in the database [3].

1.1.3 Currently used systems

1.1.3.1 Fingerprint

This is the most popular and oldest biometric system which is recognized for its accuracy worldwide. No two individuals have the same arrangement of ridge patterns and it does not change through out the life. Fingerprint matching is the most widely used and least expensive biometric technology and because of these reasons large numbers of applications are incorporating this technology to increase user security. Some of their examples are PDA's, laptops and personal computers.

1.1.3.2 Iris

Iris scanning is based on visible features like rings, furrows, freckles and corona. The iris is formed in eight months and remains stable through out the life. There are 266 unique spots in iris that are more than any other biometric system. Usually the unique traits in other biometrics are in the range of 13 to 60. Another advantage in using iris is that it is well protected from environmental pollution and remains undamaged. Because of its distinct shape in the face it is easy to isolate iris from other parts of the face [6] (see Figure 1.2).

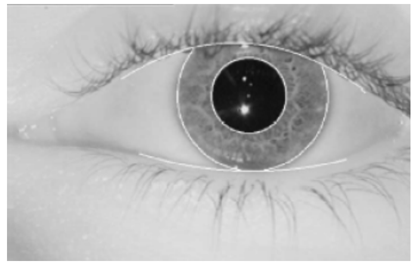


Figure 1.2: Localization of iris [6]

Iris recognition systems are usually fast and can be used for large scale systems. It is very difficult to alter the texture of iris even through surgery so it is very difficult to break the system. As the iris scanners are becoming affordable, the use of this technology will be increased in coming years.

1.1.3.3 Voice

Voice recognition systems are much dynamic as compared to other biometric systems. Variety of methods can be used to implement voice recognition software. It can be understood by considering a system that forces a person to speak a phrase in different ways. At one side, it will be very difficult to deceive a system but on the other side much advanced algorithms will be required to process and authenticate this information [7]. Voice is a combination of behavioral and physiological characteristics. Physiological component remains the same but changes can occur in behavioral component so the difficulty occurs in programming voice recognition software. This biometric technology allows a remote access to person in case of phone based authentication systems as he can authenticate himself through phone line. Different communication parameters i.e. phone speaker, microphone and communication channel also affect the performance of these systems so their quality should be kept at an acceptable level.

1.1.3.4 Face

Face recognition is becoming popular day by day. A lot of research is being carried out in this field and several companies have made accurate face recognition softwares. The focus of current research is on developing recognition systems that can account for changes in lighting, age and expression of face. The problem of dealing with glasses, face hairs and make up is also being carried out in present era [8]. The effect of different parameters on face is shown in Figure 1.3.

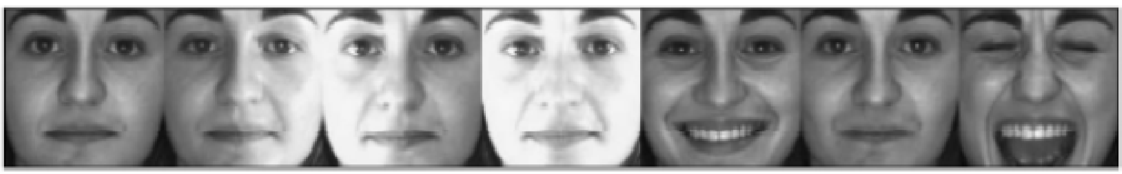


Figure 1.3: Effect of various parameters on face [8]

One of the advantages of this system is that it can operate at a large distance from the subject. The subject does not need to interact with the system. The camera only needs to capture usable image of the face. These types of systems are getting popular and their rate of acceptance is increasing.

1.1.3.5 Hand geometry

Hand geometry is closely related to fingerprinting. Hand Geometry involves the analysis and measurement of the shape of the hand. According to the National Center for State Courts "identifying someone using hand geometry is considered simple and accurate". Unlike fingerprints, the human hand is not distinctive and individual hand features are not descriptive enough for identification. However, hand geometry uses individual features and measurements of fingers to do verification of an individual [9]. To verify geometry of the hand, the person places his hand on the surface of the scanner which has pegs on it. These pegs allow a person to align hand properly so that device can read the attributes of the hand effectively. The device compares the geometry of hand with all previous hand attributes in the database and tells about its decision after comparison. The main drawback with this approach is that it is not possible to detect weather the person is alive or not. Two people may have same hand geometry so it is possible to deceive the system. That is why it is usually combined with some previous authentication systems for verification purpose.

1.2 Fingerprint as a biometric

According to the FBI "Criminal Identification by means of fingerprints is one of the most potent factors in apprehending fugitives who might otherwise escape arrest and continue their criminal activities indefinitely" [10]. Fingerprints were formally accepted as personal identifier in twentieth century and have now become very popular and acceptable in various law enforcement agencies. Fingerprint has a largest share in the market according to the report of international biometric group in 2002. The revenues which are generated by different biometrics are presented in Figure 1.4.

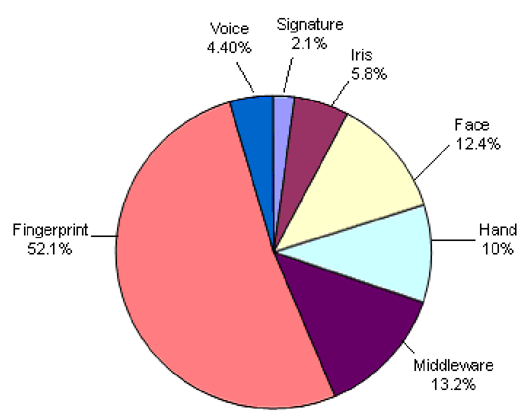


Figure 1.4: Biometric Market Report (International Biometric Group)

FBI now maintains more than 400 million records for identification purpose. This system has several advantages as compared to other biometric systems. Most of them are presented as under.

1. **High universality**

Majority of the people have legible fingerprints and they can be easily authenticated. Many people may not have old authentication requirements (passports, ID cards) but fingerprints are associated with each individual by default.

2. **Uniqueness**

Fingerprints are unique all over the world and no two individuals have same fingerprints. Even twins that share the same DNA structure have different ridge patterns on their fingers. It happens because ridge patterns are not encoded on DNA structure. There are several mathematical models that also satisfy the effective uniqueness of fingerprints [11].

3. **Durability**

Ridge patterns on the fingerprints are very durable and remain stable through out the life. These patterns are only destroyed in case of severe burn and physical injuries.

4. **Easy image acquirement**

The process of acquiring fingerprint image is very easy as compared to other biometrics. Large number of scanners are available that are able to gather an image in a few seconds [12]. No training is required for the user to input his image to the system and he is not reluctant in performing this process. But if we consider iris, we have to train the user and huge cooperation is required from the user for acquiring image.

5. **High familiarity**

Most of the population is familiar with the use of fingerprints for verification and access control. That is why minority of the population is reluctant to give their fingerprints. They know about the forensic databases of law enforcement agencies to verify an individual. This type of biometric is also high projected in media (films, dramas) which makes this biometric system much familiar to people and increases its acceptability worldwide.

1.3 Introduction to fingerprint system

A fingerprint pattern consists of sequence of ridges and valleys. In a fingerprint image, the ridges appear as dark lines whereas the valleys are the light areas between the ridges. A cut or burn to a finger does not affect the underlying ridge structure, and the original pattern will be reproduced when new skin grows. When fingerprint is analyzed at global level, ridge patterns having distinct shape at certain regions are found. These regions are usually characterized by high curvature or frequent ridge endings and are called singularities. There are three basic singularities and these are called loop, data and whorl. Many of the matching algorithms use loop singularity as core to align the two fingerprints for obtaining better matching results but poor image quality and intra class variability of fingerprints makes reliable core detection difficult. The global structures are indicated in Figure 1.5.

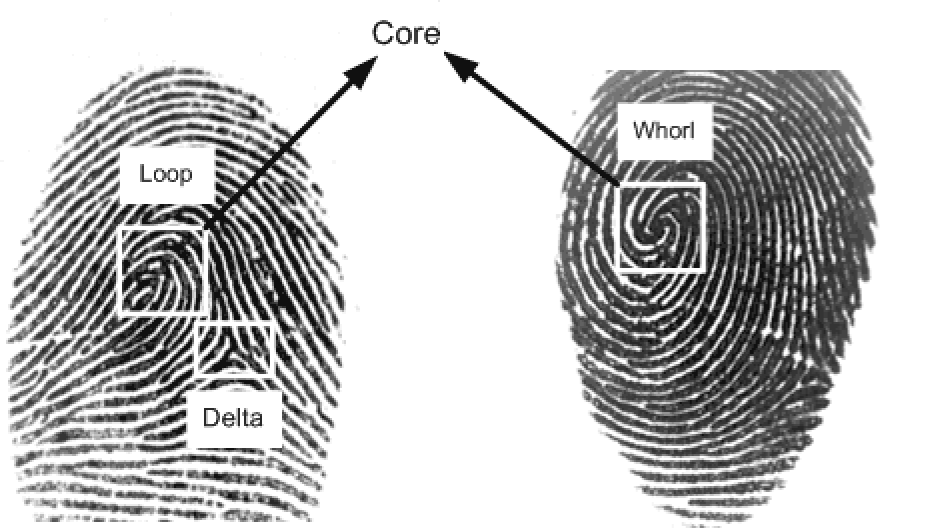


Figure 1.5: Global structures in fingerprints

Analyzing the image at local level provides a lot of significant information. These details can be detected by analyzing the discontinuities in the ridges, also called minutiae points. A lot of minutiae types are in use today [13, 14] but the more distinctive, reliable and permanent among them are endings and bifurcations. Ending is a place on the ridge where it terminates whereas bifurcation is concerned with a location where the ridge splits in to two. FBI is also in favor of using these two minutiae types for comparison purpose [15, 16] because the reliable detection of other minutiae types is very difficult. Analyzing a fingerprint on the local level provides the necessary information to accurately distinguish one fingerprint from the other. Different types of minutiae are presented in Figure 1.6 below.

Figure 1.6: Local structures in fingerprints

Information regarding pores is available in [17]. The various stages of typical fingerprint recognition system are shown in Figure 1.7. First of all the input image is obtained through impression of a finger on paper from ink pad or online scanners. The famous scanners among several are optical, capacitive, ultrasound and thermal. After getting input image, preprocessing is a step that needs to be performed. The main task of algorithms that are used in preprocessing is to remove noise and smoothen down the ridges in fingerprint image. It must be noted that fingerprint has oriented texture with rich structural information. All the enhancement algorithms must incorporate the periodic and directional nature of the ridges during enhancement. After preprocessing, the minutiae must be extracted that are used to match input image with stored or template image. After matching two fingerprints, score is produced which would be helpful in taking decision about authentication of the system.

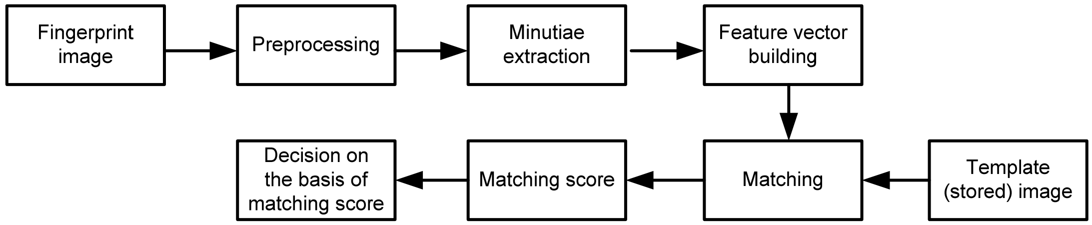


Figure 1.7: General architecture of a fingerprint recognition system

1.4 Problem statement

As the security concerns are increasing all over the world the use of personal identification system becomes vital. National security, e-commerce, and access to computer networks are some examples where these systems are used effectively. To identify criminals and providing authentic access to restricted areas there is a need to develop fingerprint recognition system. The system takes an input image that is compared with template or stored image. After comparison a matching score is produced by using fuzzy logic on the basis of which authenticity of an individual is determined.

1.5 Objective

The primary objective of this thesis is to study and analyze various techniques that are used in fingerprint matching. Various problems in existing systems are analyzed and suitable technique is implemented to rectify most of them. Some areas that need improvement are also identified where their rectification may improve the performance of fingerprint recognition system.

1.6 Thesis organization

Here the organization of the thesis is discussed. Chapter 2 gives the review of different techniques that are used for recognizing a fingerprint. They are very helpful in understanding the basic theme of this research. In Chapter 3, the whole structure (framework) of the proposed approach is presented that provides the solution to the problem in hand. The functionalities which are provided by concerned approach are also described at an abstract level. Chapter 4 covers the detailed design of different algorithms that are used to implement the required functionality. Different methods are adopted to explain the implementation methodology clearly. The sub-modules that may present within large module are also discussed. Chapter 5 presents the required results that are obtained during experimentation and provides a comparison of proposed approach with previous authentication systems. Chapter 6 narrates conclusion and future work that can be used to move this research work forward.

1.7 Summary

This chapter covers the basic knowledge about fingerprints and other biometric systems like face, iris, hand geometry and voice. It also incorporates the information regarding previous authentication systems. Operating modes adopted by different biometric systems as well as issues related to them are also discussed. Problem statement and objective is presented to understand the basic theme of research. Finally, the structure of the thesis is outlined.

Chapter 2

LITERATURE REVIEW

2.1 Introduction

This chapter includes the work done by different researchers in the field of fingerprint recognition. Studies about enhancing fingerprints as well as their matching are reviewed in detail. Different matching techniques are presented and their pros and cons are also discussed to help in selecting better approach for matching.

2.2 Preprocessing

Preprocessing a fingerprint is basic for its matching. Multiple methods are available to perform this process. We have used the method proposed by [18] to make the fingerprint suitable for matching. In this method, grayscale fingerprint is first enhanced by using normalization process in which intended mean and variance is obtained. The orientation image is then calculated that tells us about the orientation of ridges exist in the image. Frequency image tells us about the ridge special frequency within each image block. Gabor filters in x and y directions are then used for enhancing fingerprints that also smoothen down the ridges. After this process the image is converted in to binary which is pure black and white image. Next step involves thinning down the image to one pixel thick so that minutiae can be extracted easily that are used for matching. The figures that are produced at each step of preprocessing are shown in Figure 2.1.

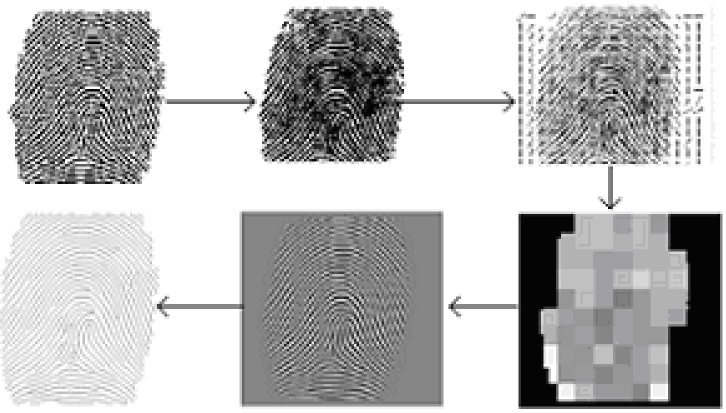


Figure 2.1: Different stages of an image during preprocessing

2.3 Matching techniques

Matching fingerprints is extremely difficult due to large variability in different impressions of the same finger. The main factors that incorporate problems are displacement, rotation, partial overlap, pressure, skin condition and stretching. There are different techniques available for matching fingerprints. All of them use different methods and parameters to match fingerprint images. There are different merits and demerits associated with each approach so they must be analyzed in detail to develop better approach that tries to minimize demerits and increases merits. Fingerprint matching techniques are broadly classified in to three approaches which are explained as under [12].

2.3.1 Correlation based matching

The fingerprint images are superimposed on each other and the correlation between the corresponding pixel intensities is computed for different alignments [12]. The correlator matches by finding the peak magnitude value in the correlation image. The location of the peak indicates the translation between the images and the strength of the peak indicates the similarity between the images. Correlation may also be implemented through various optical techniques [19, 20, 21]. Since the correlation does not remain stable with rotation or translation of image so the performance of correlation based algorithm is degraded in case of rotated and translated images. The accuracy of correlation based techniques degrades with nonlinear distortion of fingerprints. To avoid this problem to some extent, local correlation based algorithms are incorporated [22, 23]. In [23] important regions are selected to perform this correlation. The important regions according to him are regions having high curvature, regions around the minutiae points and regions that include core and delta.

2.3.2 Minutiae based matching

Minutiae based matching is the most familiar and widely used method for fingerprint matching. It is the approach that is most commonly adopted by different law enforcement agencies to authenticate an individual. For understanding minutiae based matching lets suppose the minutiae from two different images which are represented as under.

$$I1 = \{m1, m2, m3, ..., mn\} \tag{2.1}$$

$$I2 = \{m1', m2', m3', ..., mk'\} \tag{2.2}$$

In some matching approaches first minutia of image I_1 is matched with all the remaining minutiae of the image I_2. There are different characteristics associated with each minutia which depends upon the used approach that must match to put the minutia from image I_1 and I_2 in common minutiae list. Specific threshold is associated with each feature or characteristic and only that minutia will be added to common points list if all these features are satisfied by that concerned point. Some minutiae based approaches are in favor of direct comparison between related minutiae but they are unordered set of data and in case of rotated image different transformations must be performed to align two prints. These transformations are rigid transformation [24], affine transformation and non linear transformation.

In algorithm [25], eight-dimensional feature vector is attached with one minutia. Different types of minutiae which are used in this technique are dots, islands, spurs, crossovers, endings, bifurcations and two others. The occurrence of each type of minutia in the neighborhood of center point is recorded. The problem in reliably discriminating different minutiae types automatically makes this approach practically impossible to implement.

The techniques presented in [1,26] enhance the algorithm proposed by [25] through incorporating multiple features that are distance to center point, relative angle between orientation of central minutia and direction of line connecting neighbor minutia to center point, ridge count and direction of each minutia with respect to center point. The mentioned features are then used to compare input and template fingerprints.

In [27], a minutiae based approach is specified that uses ratios of relational distances as a function for comparison. This technique also uses the five closest neighbors of a central point and calculates the ratios of relative distances and angles between neighbors. A tree is drawn by using common points between two images in a bottom up manner. The matching score is then calculated which tells about the similarity between two trees of images.

In [28], a new approach called K-plet is introduced in which ‘K’ either refers to the number of nearest neighbors or all the neighbors in the circular radius. These K-plets are used to match local regions between two images (input and template). The adjacency graph is drawn for each of the K-plets and the matching score is calculated by traversing the nodes of a graph using coupled breadth first search algorithm.

2.3.3 Ridge feature based matching

In this approach, we use the characteristics of ridges to match two fingerprints. Since the pixel intensities and locations of minutiae are the features of ridge pattern so they can also be considered as subcategory of ridge pattern. This technique is very helpful for matching when the quality of the fingerprints is bad because in that case extracting the minutiae reliably is very difficult. Minutiae based algorithms are rather slow for implementing them in embedded systems. The most common characteristics of ridges which are very helpful in matching are local ridge orientation, ridge frequency, ridge shape, ridge texture information, ridge path deviation, breaks, creases, scars and pores. All the properties of ridges which are mentioned above are also called Level 3 information. Level 3 features are also immutable, permanent and unique [29, 30]. In fingerprint matching, the use of level 3 features was first introduced by [29]. They focused on pore based level 3 matching using fingerprint segments but the alignment of images must be done manually or it should be predetermined. This process of alignment is automated in [17] where the algorithm is responsible for doing this. Usually this representation has less discriminating power and it is usually combined with minutiae based representation for better matching results [17]. Pore based matching usually requires images having the resolution more than 500dpi so their availability is not easy and the individual working on this technique usually needs to develop a database himself by using high resolution scanners which are expensive in the present era.

In [31] a filter bank based approach is presented in which the fingerprint is considered as a system of oriented texture. Forensic experts analyze this rich structural and textural information during matching. Most of the textured images have limited range of spatial frequency so they are usually discriminated by using dominant frequency content. Many images have the same range of spatial frequency which reduces their discriminating power. In [31] both global and local texture descriptors are used for matching purpose. Eight oriented Gabor filters are used to calculate texture descriptors and then average deviation in pixel values is computed for each cell. These average deviations form feature vectors that are later used for comparison. The matching is performed by calculating the Euclidean distance between the feature vectors of two different fingerprints. Five different samples of the same finger that are rotated in different directions are stored in the database to make algorithm rotation invariant.

2.4 Summary

This chapter discusses various techniques and methodologies that are helpful in understanding the basic theme of research. The method used for enhancing the fingerprint as well as different stages of an image during preprocessing are also discussed briefly. Different matching techniques with their advantages and disadvantages are also reviewed to help us in selecting the better approach for matching. Since the selected approach is minutiae based so the methods belong to this class are explained more thoroughly as compared to others.

Chapter 3

METHODOLOGY

3.1 Introduction

This chapter incorporates the basic knowledge about the concerned research. The scope of the project is explained that tells us about the functionality of the project as well as its limitations. The conditions and assumptions which are used in this project are also discussed that will be very helpful in understanding the situation in which the research is done. To make the problem easy to grasp, it is decomposed in to several modules. The theoretical background used to perform the required functionality of the module as well as its position within the hierarchy is also presented.

3.2 Scope

As the security concerns are increasing, the need for effective authentication system is increasing. Biometric systems are very effective for authentication of an individual because of their permanence through out the life and among them fingerprint recognition is most popular. The conducted research is crafted on minutiae based fingerprint matching. Our approach is based on structural matching and the matching algorithm presented here is the improved and modified form of [1]. There are several limitation associated with [1] which are analyzed and removed effectively in the proposed technique. In this method, matching is done on the basis of five closest neighbors of one single minutia that is also called a center minutia. An authentication of minutia is based on these surrounding neighbors. The approach we present here is divided in to two stages, first stage performs initial filtration and the second stage includes special matching criteria that incorporate fuzzy logic as well as a novel feature to select final minutiae for matching score calculation. The method of selecting center point for second stage is also adopted. This algorithm is able to perform well for translated, rotated and stretched fingerprints. Many of the matching algorithms use reference points (core and delta) for alignment of fingerprint before matching. The performance of this approach is badly affected in case of noisy or degraded images because in these type of images reference point may be absent or not detected and in case of its absence the matching process totally fails. On the other hand, the algorithm that we present here does not require any process for alignment before matching that is why the problems associated with detection of reference point are not encountered.

3.3 Conditions and assumptions

In order to confine the scope of work, certain conditions and assumptions are made. These conditions and assumptions are described in Table 3.1.

Table 3.1: Conditions and Assumptions

Conditions and Assumptions
Scanner is not connected to acquire images directly
FVC 2002 (DB1_A) database is used for performance evaluation

3.4 Problem decomposition in to modules

Big thing always appears as big but if it is broken down in to smaller pieces then it no more remains big. In this section, the approach presented by a famous Muslim scientist Al-Khawarzimi is adopted. According to him, the big problem can easily be solved if small problems that are part of this big one are solved. The problem we present here is divided in to several modules. The primary input of the system is the fingerprint image that is processed to make it suitable for matching. The modules forming the entire system include; 'Minutiae extraction', 'Building feature vectors for matching', 'first stage matching' and 'second stage matching'. The graphical representation of decomposition of big problem in to smaller modules is presented in Figure 3.1 where the flow of information between different modules can also be viewed.

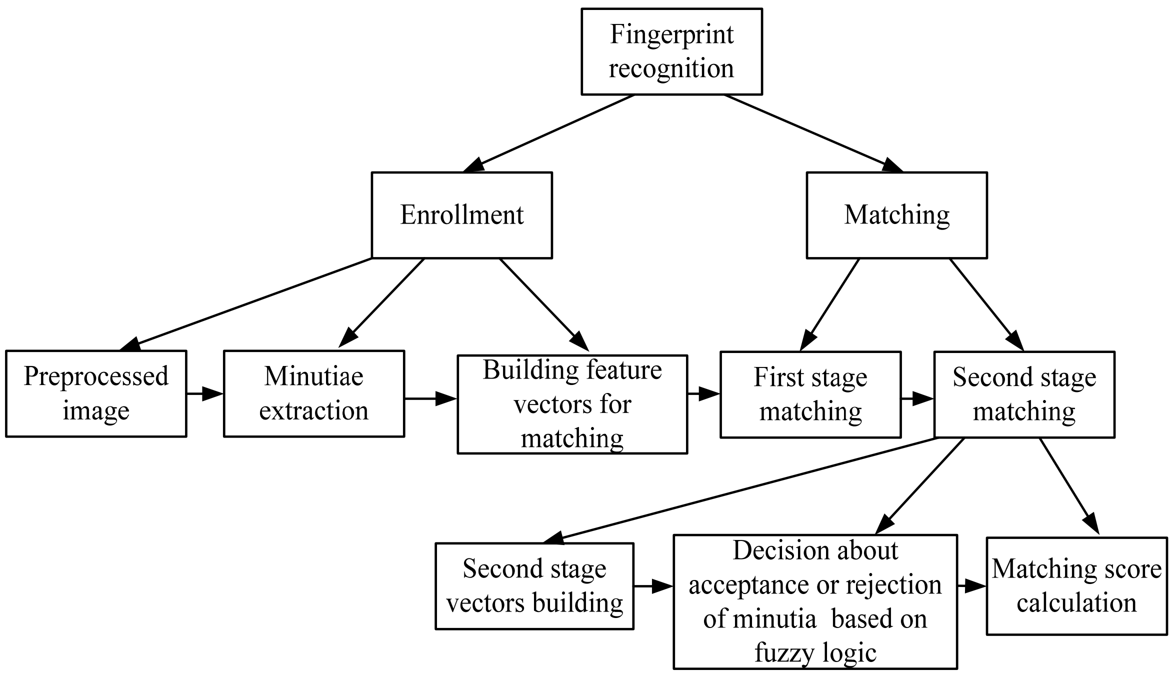


Figure 3.1: System Process Diagram

The process for extracting minutiae is very important for effective matching procedure. There are several methods which are popular for extracting them. For extracting minutiae, the binary image of the fingerprint is passed to the ridge thinning process. This process is responsible for reducing the width of the ridge to one pixel thick. Some minutiae extraction approaches work directly on the grayscale images to detect them. According to their authors, a significant amount of information is lost during binarization process as well as binarization and thinning are time consuming processes. But the process to extract the minutiae from binarization and thinning is easy to understand and implement as compared to mentioned technique. The presented technique uses thinned image to extract two types of minutiae i.e. endings and bifurcation as they are more informative, discriminative and permanent. There are more than 150 different local structures but FBI supported these two discontinuities [11, 14, 15, 16]. A 3x3 window is placed on each pixel by keeping it in the center and all the pixels in the eight-neighborhood of this pixel are analyzed to determine its type. The values of adjacent pixels at the window boundary are subtracted and stored which are later added to determine whether the center pixel belongs to ending or bifurcation. The pixels at the boundary of the window are analyzed where the value of center pixel is not encountered in required calculation. This window is moved across whole image for identifying all the minutiae exist in the image. Their locations are saved so that they can be referred later for matching process. The detail of this process is explained in the next chapter where the implementation of the method is discussed using flow chart and algorithm.

Feature vectors building is very important part of this approach. These feature vectors can be categorized with respect to different stages in which they are compared. Each minutia has its own feature vectors that are calculated by using characteristics of surrounding neighbors. Five closest neighbors are used to characterize one single minutia. A central minutia along with these five neighbors is called a local structure which has strong discriminating power. If number of neighbors is increased from five then there will be a significant effect of this act on processing time. Two types of feature vectors which are compared in first stage matching are ‘FVM1’ and ‘FVM2’.

‘FVM1’ includes the type (ending or bifurcation) of the central minutia whereas ‘FVM2’ includes four features or classifiers which are very helpful for matching process. ‘FVM1’ is a preliminary condition for ‘FVM2’. If the criterion mentioned in ‘FVM1’ is not satisfied then there is no need to check ‘FVM2’. Each neighbor of the central minutia has its own ‘FVM2’ type vector for matching. The elements of ‘FVM2’ are type of the neighbor, distance from center point to concerned neighbor, relative angle and ridge count. The graphical representations of these two feature vector types are given in Figure 3.2. This figure helps in understanding both feature vectors efficiently. The elements of each feature vector as well their implementation will be explained in detail later.

Type of central minutia

(a)

Type of neighbor
Distance to central point
Relative angle
Ridge count

(b)

Figure 3.2: (a) Element(s) in FVM1 (b) Element(s) in FVM2

After building both feature vector types ‘FVM1’ and ‘FVM2’, first stage matching is ready to be performed. This stage of matching uses the classifiers mentioned in ‘FVM1’ and ‘FVM2’ to perform initial filtration of minutiae. Any point that satisfies all the features of both vector types would be selected as common minutia. All the common minutiae between input and template image are stored and forms a list called initial or unconfirmed list. A specific threshold is associated with each element of feature vector type ‘FVM2’ that is set after running the algorithm multiple times. Because the matching performed in our approach is statistical that is why selection of thresholds is important. The result of first stage is the degree of similarity that is assigned to each minutia in the image. If one of the neighbors of minutia in the input image is matched with one neighbor of minutia in template image then the degree 'one' will be assigned to the concerned minutia. In this way, the assigned degree to the minutia depends upon the number of neighbors that are matched. If no neighbor of the minutia from input image is able to match with corresponding minutia in the template image then the degree 'zero' will be assigned to concerned point (minutia). The first stage of matching would perform good filtration and clears a lot of irrelevant or false minutiae but some of them are still included that are removed during second stage of matching.

After completion of first stage, second stage matching needs to be performed. First stage matching provides us with initial or unconfirmed list that must be finalized. This matching stage is more stringent as it has very powerful features with an incorporation of fuzzy logic that helps to increase matching score effectively. Second stage matching is decomposed in to 'second stage vectors building', 'fuzzy logic based decision' and 'matching score calculation'. Before explaining the functionalities of these modules, understanding the concept of fuzzy logic is vital. Fuzzy logic is a superset of conventional or Boolean logic that has been extended to handle the concept of partial truth. It is a truth that lies between completely true and completely false. It is a logic that is usually used to show approximation rather than exactness. The significance of fuzzy logic originates from the fact that usually the modes of human reasoning and especially common sense reasoning are approximate in nature. The vital properties of fuzzy logic as founded by Zader Lotfi are given below [32].

1. In fuzzy logic, exact reasoning is viewed as a limiting case of approximate reasoning.
2. In fuzzy logic, every thing is a matter of degree.
3. Any logical system can be fuzzified.
4. Inference is viewed as a process of propagation of elastic constraints.

Fuzzy sets can be a complex mathematical expression in multi-valued logic. To implement fuzzy logic we must have a way to assign some number to humans' intuitive assessments of fuzzy sets. We must transform human fuzziness to numbers that can be used by a computer. We can do this by assigning fuzzy sub-set conditions a value from 0 to 1.0 or 0 to 100. We can assign different confidence levels or numbers to different conditions depending on the situation that is being faced [33].

After understanding the advantages and flexibilities associated with fuzzy logic, it is incorporated in feature vector of type 'FVM3'. The elements of FVM3 are 'distance from central point to concerned neighbor', 'relative angle', 'angle between orientations' and a novel feature called 'distance from center point to ridge intersection'. Each of these features is assigned a confidence level based upon experience as well as by considering its importance. This assignment of confidence levels is used to take decision about the acceptance or rejection of minutia. Different numerical values are assigned to different features or classifiers on the basis of their effect in the final matching score calculation. A confidence value of '40' is assigned to 'distance from center point to concerned neighbor'. Relative angle is able to get a value of ‘30’. Similarly ‘20’ and ‘10’ are the values which are associated with ‘angle between orientations’ and ‘distance from center point to ridge intersection’. The use of these values in final decision making (acceptance or rejection of minutia) will be explained in detail in coming literature.

Matching score is the value that measures how similar the two fingerprints are when compared to each other, and its generation is important to an automatic fingerprint recognition system. It is used to make the final decision about authentication of an individual in the identification process and is usually the last stage of the system. Human experts make the final decision according to forensic guidelines and experiences which suggest that a minimum of 12 matched minutiae is sufficient to conclude that two fingerprints are from same finger [12]. However, a minutiae based automatic fingerprint recognition system can not make a decision like human experts. Unlike human experts who have access to all the information that a fingerprint contains like overall ridge flow pattern, location and configuration of singular points (core and delta), location of pores and ridge counts between pairs of minutiae, the minutia based automatic systems have access to only the information from the minutiae representation of fingerprints.

Fuzzy logic based decision making helps us to gather more authentic minutiae which are used for matching score calculation. In matching score calculation module, all the common minutiae that fulfill initial matching criteria in first stage as well as fuzzy logic based final matching criteria in the second stage would be considered in final score calculation. In this procedure, the square of total number of common minutiae will be taken that is divided by the number which is calculated by multiplying the minutiae in input image and minutiae in template or stored image. The calculated value is used to determine the degree of similarity between two images. The threshold which is associated with matching score depends upon the application in which the system (fingerprint recognition) is embedded. For critical applications, a tight threshold is used that can be relaxed in case of less critical applications. For critical applications, large value of matching score is required to declare an individual as authentic whereas for less critical applications, this value may not be very large.

3.5 Summary

Chapter 3 sets up the basis to understand the proposed approach effectively. It narrows down the vastness of the topic to the conditions and assumptions under which this work has been done. The process diagram is given which lists the steps involved in the work. The functionality of each module mentioned in the diagram is explained. The flow of information between different modules is also discussed to give a broader understanding of the proposed system.

Chapter 4

DESIGNING AND IMPLEMENTING THE FRAMEWORK

4.1 Introduction

This chapter discusses various algorithms and techniques in detail which are used to implement the modules given in system process diagram. The concerned modules are minutiae extraction, building feature vectors, first stage matching, second stage vectors building, fuzzy logic based decision about acceptance or rejection of minutia and final matching score calculation. For better understanding of complex modules, the flow charts of these modules are also presented. After completion of this chapter, the reader would be able to understand the proposed approach in detail.

4.2 Minutiae extraction

The process for extracting minutiae is critical part of this approach as it provides the base on which the algorithm runs. Multiple techniques are available which can extract minutiae effectively. Some of the techniques are able to extract them directly from the gray scale image without performing a thinning process but our approach requires thinned image for their extraction. A 3x3 window is used that is placed on each pixel to determine its type. The preferred types are endings and bifurcations because they are permanent, unique and also supported by FBI. To determine the type of center pixel, the concerned window is given in Figure 4.1.

P4	P5	P6
P3	P	P7
P2	P1	P8

Figure 4.1: Window for minutiae extraction

The values of pixels at the boundary of the window are taken and differences between adjacent pixels are calculated. These subtracted values are added later to get a single value. We call this value CN (crossing number) that is divided by '2' to determine the type of center pixel. If CN/2 is '1' then 'P' indicates ending and if it is '3' then this would be an indication of bifurcation. The equation which is used to perform this task is given below.

$$CN = \sum_{i=1}^{8} (Xor(P(i),(P(i+1))) \qquad \textbf{(4.1)}$$

For last iteration (i=8), we consider that P9=P1. Thin image is a collection of black and white pixels so we can calculate the difference between adjacent pixels by using exclusive OR (XOR). To elaborate this concept, a practical implementation of this whole process is also discussed in this section. Let's suppose a 3x3 window that is mapped on a pixel in the image. To determine the type of the pixel, the below mentioned algorithm will be followed.

Algorithm

Begin

1. *Make a 3x3 window and place it on a pixel in the image.*
2. *Calculate the values P12, P23, P34, P45, P56, P67, P78 and P81 for a center pixel.*

P12=XOR (P1, P2)

P23=XOR (P2, P3)

P34=XOR (P3, P4)

P45=XOR (P4, P5)

P56=XOR (P5, P6)

P67=XOR (P6, P7)

P78=XOR (P7, P8)

P81=XOR (P8, P1)

3. *Add all the above mentioned values.*

   CN=P12+ P23+P34+ P45+ P56+ P67+ P78+ P81

4. *Divide 'CN' by 2 to determine the type of concerned pixel.*

   If CN/2 is '3' then

   'P' indicates bifurcation

   Else if CN/2 is '1' then

   'P' indicates ending

   Else

   'P' indicates some other local structure.

5. *Save the type and coordinates of 'P' in case of ending and bifurcation and move the window to the next pixel.*

6. *Repeat the whole process from step 2 till step 5 until all the pixels in the image are encountered.*

End

Practical implementation of this whole process is given below in Figure 4.2.

1	1	0
0	0	1
1	1	0

Figure 4.2: Bifurcation's view in the form of pixels

By using Figure 4.2 the required values are calculated that are presented below.

P12=0, P23=1, P34=1, P45=0, P56=1, P67=1, P78=1 and P81=1. By adding these values, CN is obtained which is '6' in this case. After dividing CN by '2', we get '3' which is an indication of bifurcation. The calculated result is correct which is also supported by Figure 4.2.

If all the pixels having value '0' are joined then the shape of bifurcation will be obtained. The flow chart of this whole process is also given in Figure 4.3 to understand the concept more clearly.

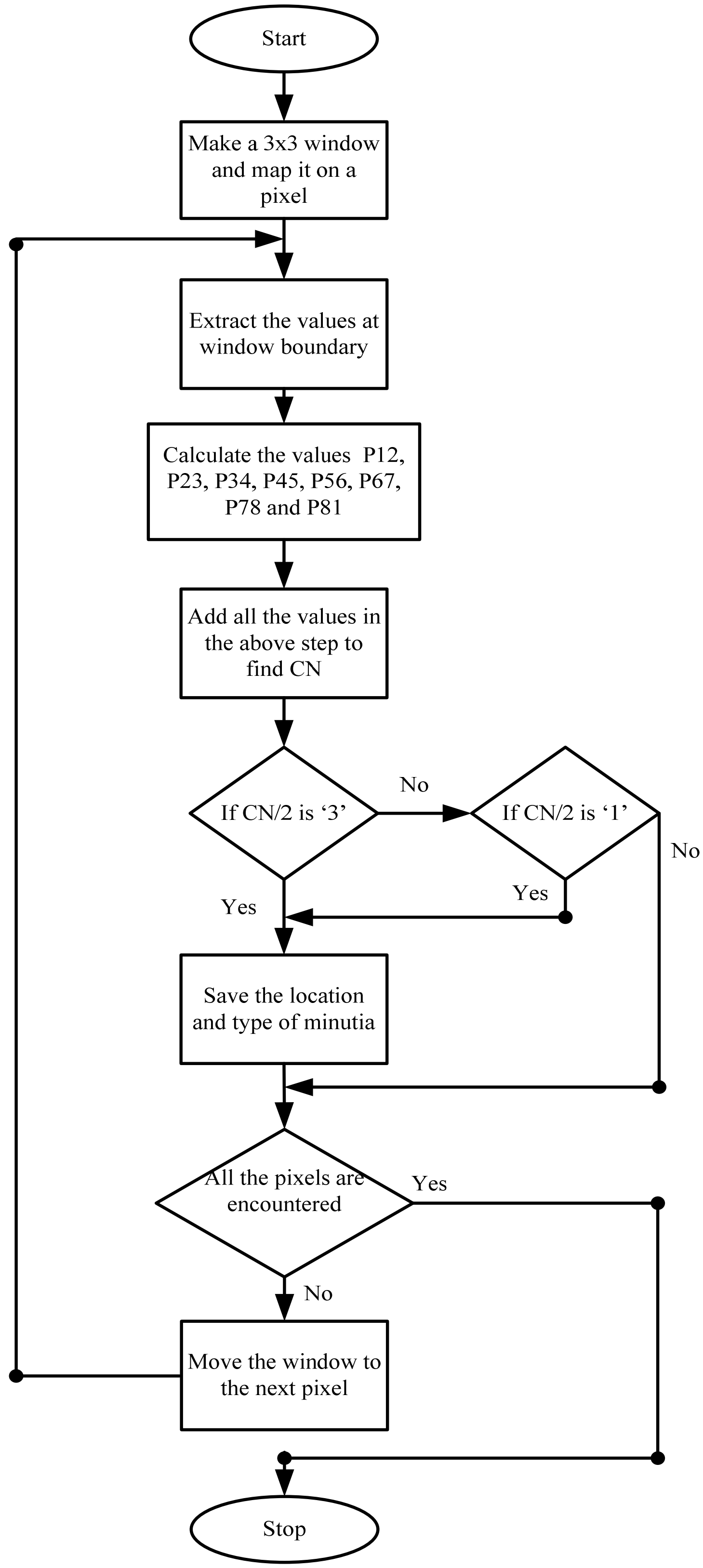


Figure 4.3: Flow chart of minutiae extraction algorithm

4.3 Building feature vectors for matching

Feature vectors for matching are considered as part of enrollment because they must be calculated before performing matching process. Two types of feature vectors which are core of this module are ‘FVM1’ and ‘FVM2’. There are different classifiers or elements which are associated with these vector types. In the first stage matching, these two feature vectors need to be compared. In the proposed matching procedure, authentication of minutia is assured by considering the characteristics of its five closest neighbors. A collection of minutia with these five neighbors is called a local structure.

The process of forming a local structure is not a trivial task. To make a local structure, Euclidian distances are calculated between a center minutia and all the other minutiae in the image. These distances are saved and then organized to make a decision about the selection of five closest neighbors of center minutia. The distance of minutia with itself is not required because the interest lies in finding the five closest neighbors of minutia not the minutia itself so the Euclidean distance having value ‘zero’ is simply rejected because it would be that distance which is not required. Lets suppose that there are ‘10’ minutiae in the image other than center point which are labeled as 1, 2, 3…, 10. Their distances with center point are represented as under in eq-4.2.

$$D = \{dc1, dc2, dc3, ..., dc10\} \qquad \textbf{(4.2)}$$

All of these distances are stored in an array ‘D’. The above mentioned distances are sorted in ascending order and top five distances along with their locations are taken as five closest neighbors. These neighbors are labeled as N1, N2, N3, N4 and N5 where the first neighbor N1 is closer to the center as compared to other neighbors. In the same way as the label number increases, the distance of that neighbor to the center also increases. To grasp this concept, the graphical representation of local structure is given below in Figure 4.4.

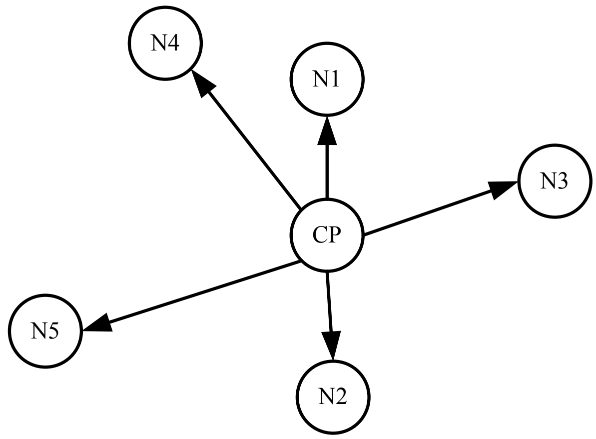


Figure 4.4: Graphical representation of local structure

To compare a local structure in the input image with template image the two feature vectors 'FVM1' and 'FVM2' must be calculated. Only one feature vector of type 'FVM1' is associated with one local structure. But on the other hand, each surrounding neighbor of central minutia must have its own feature vector of type 'FVM2' to perform matching. It means that there are five feature vectors of type 'FVM2' and one feature vector of type 'FVM1'. 'FVM1' includes the type of central minutia whereas 'FVM2' incorporates type of the neighbor, distance from center point to concerned neighbor, relative angle and ridge count. Each of these elements has different strengths and weaknesses that are taken in to consideration to increase the accuracy in matching process. It is not difficult to calculate 'FVM1' because the method to determine the type of the minutia is already discussed in detail in minutiae extraction process. To make this task simpler, the location and type of each minutia must be stored during minutiae extraction phase. However, the method to calculate the elements associated with 'FVM2' is not as simple as 'FVM1' so their computation mechanism is explained below.

4.3.1 Type of minutia

Type of minutia for 'FVM2' will be calculated in the same way as element of 'FVM1' by using a 3x3 window. This feature is very helpful for comparison purpose in the first stage matching.

4.3.2 Distance from center to corresponding neighbor

In this feature, Euclidian distance is calculated from center point to concerned neighbor. Each neighbor has to find this distance to complete its feature vector of type 'FVM2'.As the locations of neighbors are known including center point so Euclidian distances can be calculated by using the following formula.

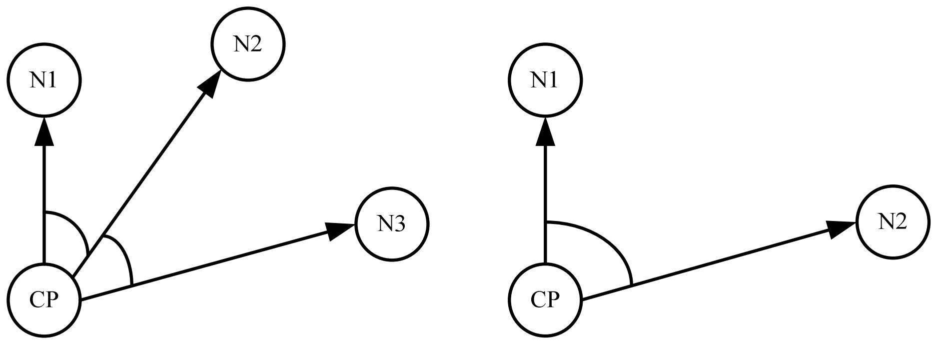
$$Distance = \sqrt{(x_2 - x_1)^2 + (y_2 - y_1)^2} \quad \textbf{(4.3)}$$

Where (x_1, y_1) and (x_2, y_2) are the coordinates of centre point and corresponding neighbor. The formula which is used for forming local structure is same as given in eq-4.3.

4.3.3 Calculating relative angle

Relative angle calculation is very important in increasing accuracy of our algorithm. For its calculation, it is necessary to understand the steps which are involved in computation of direction. Relative angle must also be computed for each neighbor of central minutia because it is also a part of 'FVM2'.

The direction of central point is used as a base or origin (reference point) and all the five angles for each neighbor are calculated by using it as reference (fixed point). But if the angles are calculated between neighbors and not between base and neighbor, different problems may arise that can be understood by using Figure 4.5.



(a) (b)

Figure 4.5: Pictorial representation of relative angles between neighbors

Let us consider that the image on the left hand side is a situation which occurs in input image whereas right hand side image shows a situation in template image. The above figure shows a problem where a minutia in one image may not present in the other. The minutia N2 in left image is not present in the right one but the minutia N3 exists in it with name N2. The angle calculated between neighbours N1 and N2 in the input image may not match with any angle in the template image and because of this true point N1 will be discarded.

Another problem is the existence of false minutia that may come between two true minutiae. This situation can also be understood by considering Figure 4.5. If N2 is considered as false minutia in left image then it will be responsible for introduction of fake angle that would not be available in template image and because of this a true minutia will not be matched between two images. As the angle between two neighbours is stored in the low labelled neighbour, (angle between N1 and N2 is stored in N1) the angle stored by N1 would not match with any other angle in template image which results in disposition of N1 that is a true point.

To eliminate the problem of loss of true neighbour, it may be possible to consider one neighbour as fixed. All the five angles corresponding to different neighbours are calculated by considering it as reference point (base). But if this fixed point in input image is replaced by some other neighbour or not present in template image then all the relative angles will be disturbed which results in matching process failure.

To resolve all the above mentioned issues, incorporating direction calculation in relative angle computation is very critical. Direction serves as base or reference and angle for a neighbour is calculated between this base and the neighbour itself and not between two neighbours. For the calculation of direction, we have to traverse each ridge some pixels away from the location of detected minutia. After traversing some pixels this new location will be stored and the slope of a line is calculated by using location of a minutia and this new location. . Let us suppose that location of detected minutia is X_m, Y_m and the coordinates of this new location are X', Y'. The formula that is used for calculating slope (direction) is given in eq-4.4.

$$direction = \frac{X' - Xm}{Y' - Ym} \tag{4.4}$$

The procedure for calculating slope is not very simple and is explained below in detail.

First of all, a 21x21 neighborhood is mapped on each minutia by keeping it at center. The values of the pixels in this neighborhood are analyzed for direction calculation with out disturbing the pixel values in the original image. A 3x3 window within this big neighborhood is mapped on the center pixel. To determine the direction of traversal, all the pixels at the boundary of this window are analyzed. A pixel with value ‘0’ is replaced by ‘1’ and the center of this window is moved to the location of that pixel which is most recently replaced by value ‘1’. The process to be followed for ending is different from bifurcation. For execution of the proposed process, ridges in the image should be one pixel thick and the possible values of pixels in the image must be 0 or 1.

In the case of ending, the concerned ridge is traversed ‘10’ pixels away from the location of detected minutia. For effective traversing of a ridge, all the ‘0’ valued pixels are followed where the pixels having value ‘1’ are simply neglected. After moving ‘10’ pixels the required location is obtained which is used to calculate the direction of ending. To make the traversal easy, we put each pixel with value ‘0’ to ‘1’ after it is traversed. By doing this there is no confusion about the path where we have to move further. If we do not put ‘0’ for traversed pixel then a situation is reached where there are multiple pixels with value ‘0’ in different directions. In that case we can not decide about suitable path and accurate direction can not be calculated. If the ridge ends before reaching 10^{th} pixel then the location where the ridge ends would be selected as our final location for direction calculation. Another minutia may come in the path when ‘10’ pixels are being traversed away from concerned minutia. In this situation, the location of minutia which occurs in the path would be selected as our final location. The direction of traversal depends upon the ridge shape which can not be predetermined. Only two types of local structures (endings and bifurcations) are considered for direction calculation because they are more prominent, informative and permanent as compared to others. FBI also supported these two structures for effective fingerprint recognition.

For bifurcation, the procedure that is adopted in above situation is repeated three times. It has to be done because there are three legs associated with each bifurcation. To calculate the direction of bifurcation, a 21x21 window is mapped on the bifurcation in the same way as above but we have to traverse each of the three legs one by one. To traverse each leg, we have assigned '1' to the location where bifurcation is detected. By doing this, all the three legs would be separated and handled in the same way as ending. After reaching at 10^{th} pixel on each leg, all the three locations are stored and labels are assigned to them as 1, 2 and 3. These labels are assigned on the basis of order in which the concerned legs are traversed. Euclidian distance is calculated between three locations that are associated with labels in this way (Ed (1, 2), Ed (2, 3), Ed (3, 1)) where 'Ed' specifies Euclidian distance between locations. The formula used to calculate Euclidian distance is same as mentioned in eq-4.3. All these values are stored and then sorted in ascending order. The smallest value is taken and the labels are checked which are involved in calculation of this value. These two labels are ignored and the label which is not involved in calculation of this smallest value would be selected which stores the required location. This would be that location which lies on the tail of bifurcation and is farther from two other locations. The graphical representation of this whole process is given in Figure 4.6.

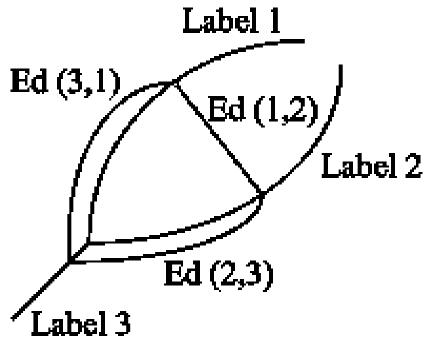


Figure 4.6: Selecting appropriate location for computing bifurcation's direction

It is clear from above figure that distance between locations that are associated with label 1 and label 2 is smallest. The label other than these two labels is label 3 so the location which is associated with label 3 would be required one. Now the locations of label 3 and the bifurcation will be used for direction calculation later. The locations which are obtained at this time are intermediate final locations for direction computation because these are obtained within a 21x21 window. For finding final locations, the coordinates which are found in this 21x21 neighborhood must be mapped on the original image. We use a center location (11, 11) of this window and call this location as IR (initial row) and IC (initial column) respectively. After traversing '10' pixels within this window, algorithm stops and stores the location (X', Y') automatically. To keep the track of the ridge movement, difference between the corresponding locations (IR, IC) and (X', Y') will be calculated. A name (IR', IC') is given to this value and it tells us about the movement of the ridge in x and y directions. The location of the minutia in the original image (X, Y) and recently calculated coordinates (IR', IC') are subtracted to form (FX, FY) where (FX, FY) is the final location in the actual image that is obtained after traversing '10' pixels on the ridge. Three locations (X, Y), (FX, FY) and (NX, NY) are required to calculate relative angle between central minutia and concerned neighbor where the locations (X, Y) and (FX, FY) are associated with central point. (NX, NY) specifies the location of concerned neighbor. All of the locations are available so relative angle can be calculated easily (see Figure 4.7).

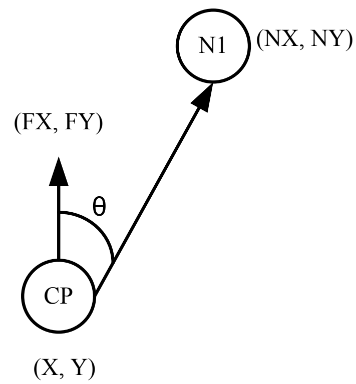


Figure 4.7: Associations of different locations in relative angle calculation

For computation of relative angle between center point (CP) and concerned neighbor (N1), two directions (slopes) need to be calculated by using the formula presented in eq-4.3. The locations (FX, FY) and (NX, NY) are analyzed and put in different quadrants with respect to center point (CP). The process for determining relative angle depends upon the quadrants in which the two locations (FX, FY) and (NX, NY) exist with respect to center point. All the cases that may occur during angle calculation are presented in Figure 4.8.

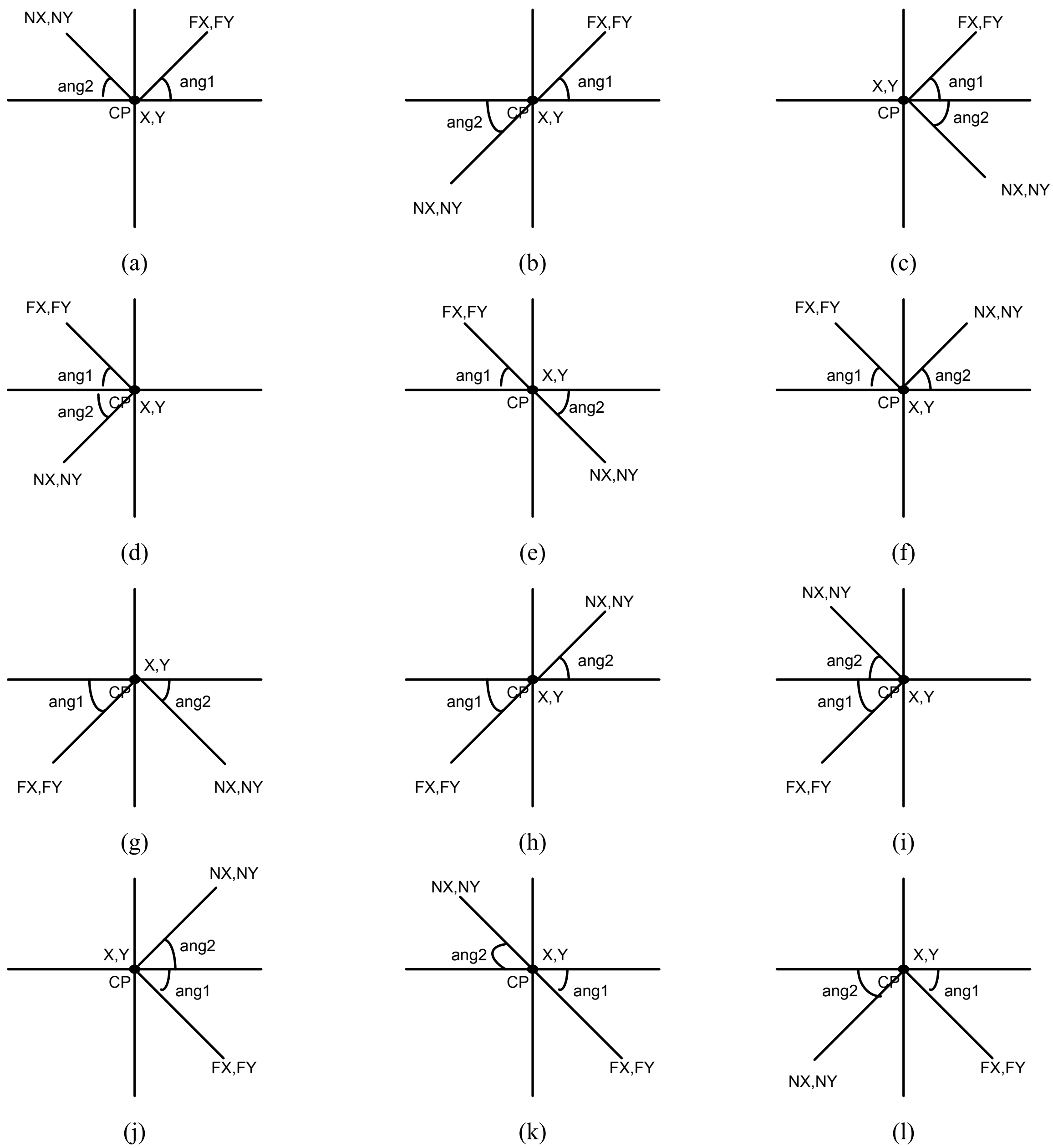


Figure 4.8: Different possible cases in relative angle calculation

Figure 4.8 shows all possible cases that may occur during angle calculation. The solutions to the above mentioned ‘12’ cases are presented below.

1. final angle=180-(ang1+ang2);
2. angtemp=90-ang2;

final angle=270-(ang1+angtemp);

3. final angle=ang1+ang2;
4. final angle=ang1+ang2;
5. angtemp=90-ang1;

   final angle=270-(angtemp+ang2);
6. final angle=180-(ang1+ang2);
7. final angle=180-(ang1+ang2);
8. angtemp=90-ang1;

   final angle=270-(angtemp+ang2);
9. final angle=ang1+ang2;
10. final angle=ang1+ang2;
11. angtemp=90-ang2;

    final angle=270-(ang1+angtemp);
12. final angle=180-(ang1+ang2);

The algorithm for relative angle computation is very important to present for better understanding of this whole procedure. Algorithmic form of the process is very helpful for implementation purpose. It is language independent which makes it understandable to the individual who does not have any programming language background. The algorithm used for finding relative angle is explained in the following steps.

Algorithm

Begin

1. *Take a 21x21 window and map it on the center pixel. Use the values of pixel in this window without disturbing actual image.*
2. *If the concerned minutia is ending*

   If no minutia comes in traversal path and ridge does not end before traversing ten pixels

   Traverse '10' pixels to find required location (X', Y')

   Else

   Save the location where ridge ends or minutia comes
3. *If the concerned minutia is bifurcation*

   3.1 replace the value '0' with '1' at the location of bifurcation detection.

   3.2 Traverse all the three legs of bifurcation one by one and follow the same

   rules of traversing as mentioned in step 2.

   3.3 Assign labels to all three locations and calculate Euclidian distances between locations which are associated with these labels.

   3.4 Sort these distances in ascending order and take the smallest distance

   3.5 Store the location associated with that label which is not involved in calculation of smallest Euclidian distance and call it (X', Y').
4. *Take a center of big neighborhood (11x11) and call this location as (IR, IC).*
5. *Subtract (IR, IC) from (X', Y') and name it as (IR', IC').*

6. *Subtract (IR', IC') from location of minutia in actual image (X, Y) and call it (FX, FY).*
7. *Use the locations (X, Y) and (FX, FY) to calculate first direction.*
8. *Use the location (X, Y) and the neighbor's location (NX, NY) to calculate second direction.*
9. *Use directions in step 7 and 8 to calculate relative angle between center point and concerned neighbor.*
10. *Repeat the steps 7 to 9 until all the five relative angles for each neighbor are calculated.*

End

The flow chart of relative angle computation is also presented here in Figure 4.9 to understand the concerned calculation more easily.

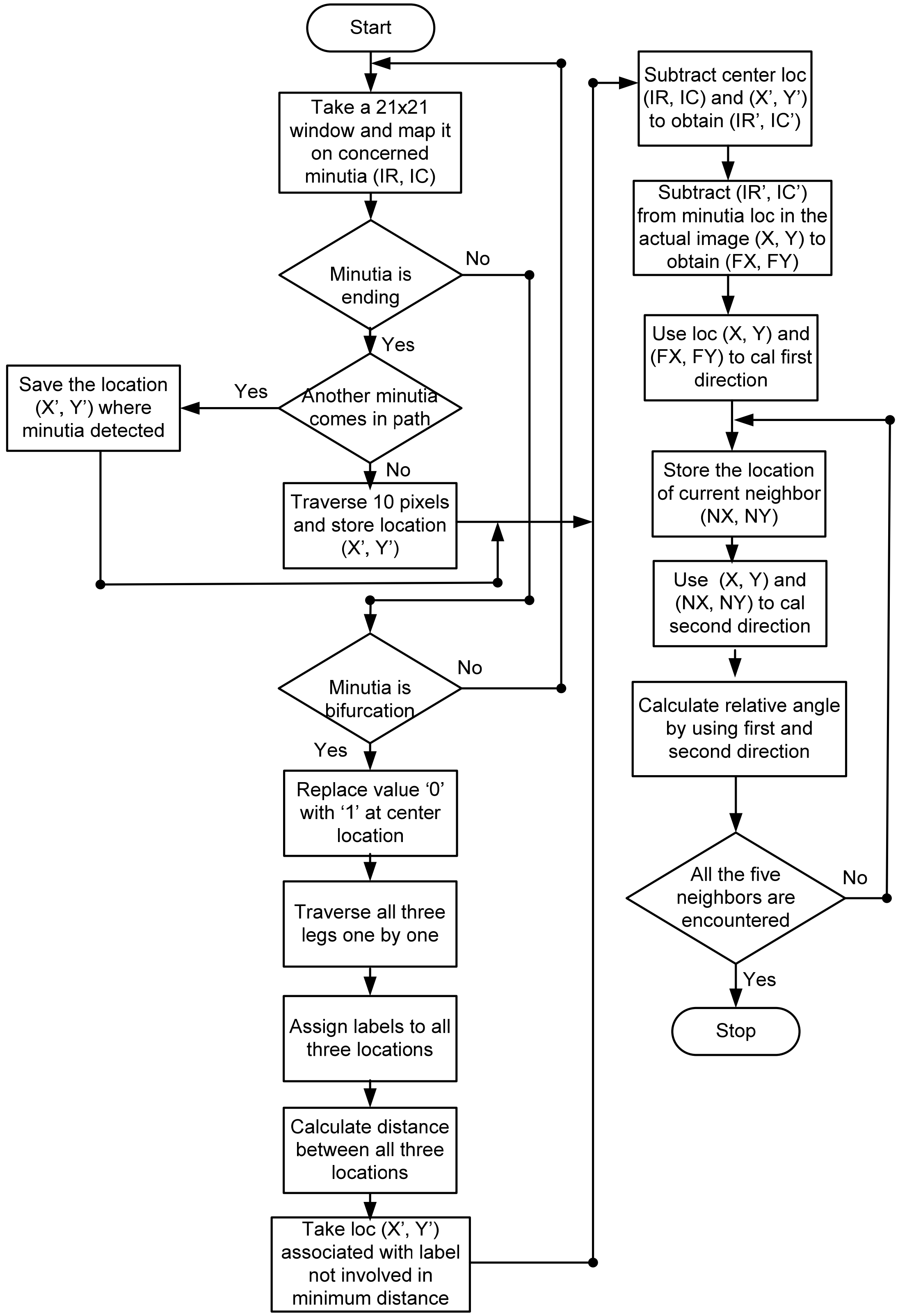


Figure 4.9: Flow chart of relative angle computation

4.3.4 Ridge count computation

Ridge count is a feature that is obtained by counting the ridges between central point and the concerned neighbor. We have calculated this feature by using equations of a straight line that are presented below as eq-4.5.

$$y = \frac{y1(x2 - x) + y2(x - x1)}{x2 - x1} \qquad \textbf{(4.5)}$$

$$x = \frac{x1(y2 - y) + x2(y - y1)}{y2 - y1}$$

Where (x1, y1) and (x2, y2) represent endpoints of a line at which different values of x and y will be calculated. In above equation, we put one point (x or y) and the corresponding point (y or x) is obtained on this straight line. To use this equation properly we must know endpoints of a straight line. Let's suppose that we put 'y' that is not beyond the range of endpoints in above equation, point 'x' would be obtained and a pair (x, y) would lie on this straight line. In this way all the points on this straight line is obtained one by one by incrementing one point and obtaining corresponding next point on the straight line. Ridges are counted by counting the pixels that have value '0' along a straight line as in thin images ridges are made up of pixels with value '0' and all the pixels other than ridges have value '1'.

4.4 First stage matching

After calculating the elements of 'FVM1' and 'FVM2', the first stage matching is ready to be performed. The elements of 'FVM1' and 'FVM2' are compared in this stage by considering the thresholds associated with each of these classifiers. A central minutia with its five neighbors constitutes a structure that is called a local structure. Each local structure from input image is compared with all the local structures in the template or stored image. A specific degree is assigned to each minutia depending on the number of neighbors that are matched between input and template minutia (centre point). If one neighbor is matched between two minutiae then the assigned degree would be '1'. One minutia may match with multiple minutiae in the template image but this problem would be removed by second stage matching. The assignment of degree to each minutia depends on the classifier presented in 'FVM1'. If 'type' of two minutiae is not matched then remaining features mentioned in 'FVM2' would be neglected and a mismatch will be declared simply. After satisfying 'FVM1' criteria, classifiers of 'FVM2' are checked and degree is assigned to them on the basis of number of neighbors that are matched. Each neighbor has its own feature vector of type 'FVM2' which means that there would be five feature vectors of type 'FVM2'. On matching one 'FVM2', '1' would be assigned as degree. The minutia having '1' or greater degree of similarity is marked and its locations in input and template images are stored that would be used in second stage. The concept of degree assignment is mentioned in Table 4.1 effectively.

Table 4.1: Assignment of degree to the minutia on the basis of number of matched feature vectors that are belonged to different neighbors

Neighbor number	**Vector type**	**Degree**
1	FVM2(input)=FVM2(temp) for 1 only	1
2	FVM2(input)=FVM2(temp) for 1,2	2
3	FVM2(input)=FVM2(temp) for1,2,3	3
4	FVM2(input)=FVM2(temp) for 1,2,3,4	4
5	FVM2(input)=FVM2(temp)for 1,2,3,4,5	5

The algorithm for first stage matching is given below for proper understanding of this process.

Algorithm

Begin

1. do

do

If the 'types' of minutiae from input and template images are same

do

do

If the types of two neighbors are same

If difference between distances satisfies a threshold

If difference between relative angles satisfies a threshold

If difference between ridge counts satisfies a threshold

Add one to the variable 'degree'

Until all the neighbors from template image are checked

Until all the neighbors from input image are checked

Until all the minutiae from template image are checked

If degree of a minutia is greater than '1'

Add the minutia with its degree to unconfirmed common points list.

Until all the minutiae from input image are checked

end

The first stage matching would perform good filtration and clears a lot of irrelevant or false minutiae but some of them are still included that are removed in the second stage of matching. The job of first stage matching is to form initial (unconfirmed) common points list.

4.5 Second stage matching

After the completion of first stage, the second stage must be performed in order to eliminate false minutiae. In the first stage, one minutia in input image is matched with multiple minutiae from template image which results in the formation of illegal or false pairs. In this collection of pairs, only one pair is correct that would be determined by second stage matching. This stage forms a final list of minutiae that are used for score calculation. This score will tell us about the percentage of similarity between two images.

The feature vector of type 'FVM3' is used in this stage. Certain modifications are made to increase the accuracy of the proposed approach. The elements which are included in 'FVM3' are 'distance', 'relative angle', 'angle between orientations' and 'distance from center point to ridge intersection'. The detail of these classifiers as well as their implementation method is discussed in coming literature. In this stage, fuzzy logic is incorporated with an assignment of a confidence level to each element of feature vector type (FVM3). The selection of center point in this stage is an important task because it is used to align two fingerprint images. The center point is selected on the basis of degree which is assigned in the first stage matching. The point with the maximum degree would be selected as center point but there may be multiple points with the same maximum degree. To eliminate this confusion, all the points with the maximum degree will be selected as center one by one as well as one more point having a degree one less than maximum will also be considered to increase accuracy. Sometimes only one point with the maximum degree is detected and we can not depend only on this point our whole matching process because it may produce bad results for true images (images from same individual). So we will take one more point as center having MAX-1 degree in order to eliminate our dependence on only one point. We have one feature vector type called 'FVM3' that will be compared in this stage and the points which fulfill the criteria mentioned in this feature vector type will make the final list. The number of feature vectors of type (FVM3) is not fixed as they were in first stage matching and this number depends upon the neighbors that fulfill first stage criteria.

4.5.1 Building feature vectors for second stage matching

The existence of false minutiae in initial (unconfirmed) list is a problem that must be solved. To eliminate this problem, the need for strong classifiers is now apparent which can draw a line between true and false minutiae. The feature vector (FVM3) includes these strong features that must be incorporated. The graphical representation of this feature vector type is given in Figure 4.10.

Distance to central point
Relative angle
Angle between orientations
Distance from center to ridge intersection

Figure 4.10: Elements of FVM3

The methods which are used to calculate the above mentioned classifiers are given below.

4.5.1.1 Distance computation

In this feature, the Euclidian distance is calculated between center point and concerned neighbor. The formula which is used for its computation is same as specified by eq-4.2. Unlike first stage where center point is considered as a base for local structure, in second stage it is assumed as a base or reference point for whole image.

4.5.1.2 Relative angle computation

The method of computation of this feature is same as previously done in section 4.3. The difference between these two calculations is the scope of center point which is used as a base in relative angle calculation. At this stage, the central minutia is assumed as an origin for whole image on the basis of which all the computations are performed.

4.5.1.3 Angle between orientations

To calculate this feature, the orientations (directions) of both the center point and concerned neighbor are required. The method to calculate these orientations is same as specified in section 4.3.3. The Figure 4.11 presented below is very helpful for grasping this concept effectively.

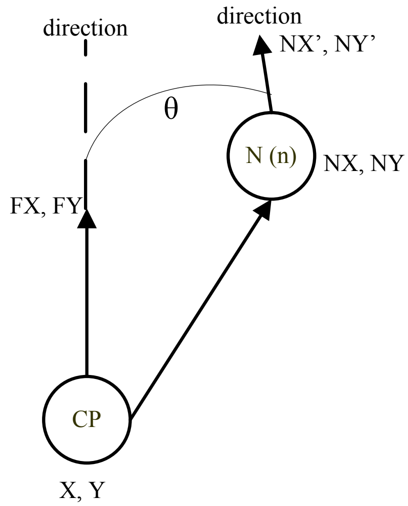


Figure 4.11: Angle between orientations

At this stage, the locations (X, Y) and (FX, FY) are used to calculate first direction whereas (X, Y) and (NX', NY') are used for computation of second direction. The locations used in the calculation of first direction are same as in section 4.3.3 but in the case of second direction, new location (NX', NY') is introduced instead of (NX, NY). The location (NX', NY') is obtained by traversing the ridge ten pixels away from the location (NX, NY). By using these two directions, angle between orientations (θ) can be calculated easily. Different cases as mentioned in Figure 4.8 are handled in the same way as previously discussed. A confidence level of '20' is attached with this classifier on the basis of experience which is earned by running the algorithm multiple times.

4.5.1.4 Distance from center to ridge intersection

To increase the accuracy of algorithm, a novel feature called 'distance from center to ridge intersection' is introduced. To calculate this feature, the program traces a straight line from center point to concerned neighbor. The ridges of the image on which this process is being followed should be one pixel thick. There should be only two pixel values in the image (0 or 1) which means that it must follow the characteristics of pure binary image. A lot of ridges may come in the path from central minutia to concerned neighbor. A pixel is determined as a ridge when a transition from '1' to '0' occurs in pixel values while tracing a straight line by algorithm. The location of intersection of straight line with ridge is saved and the Euclidian distance is calculated between the location of the center minutia and this new location. Similarly, all the locations of intersection between straight line and ridges that come in the path from center point to concerned neighbor are saved and the Euclidian distances are computed between center point and all these new locations. This graphical representation of this whole procedure is given in Figure 4.12 where bifurcation is considered as center point.

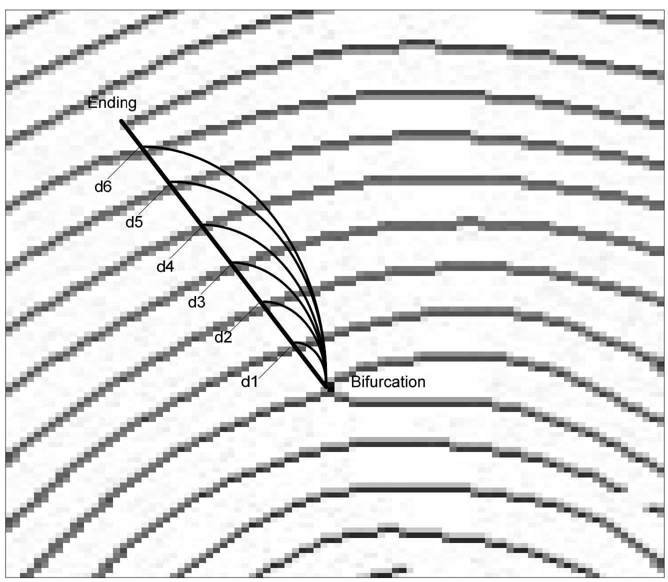


Figure 4.12: Pictorial representation of distance from center to ridge intersection

The number of distances in the straight path depends upon the number of ridges that are detected. Let us consider that six ridges are detected in the straight path as shown in above figure.

$$Idis = \{d1, d2, d3, ..., d6\} \quad \textbf{(4.6)}$$

$$Tdis = \{d1', d2', d3', ..., d6'\} \quad \textbf{(4.7)}$$

where 'I_{dis}' and 'T_{dis}' are arrays of distances from input and template images. These two arrays are compared and on matching a confidence level of '10' is attached with central minutia. If this feature is not able to match between two images and the sum of confidence levels reaches a value greater than '70' then center point will be accepted. The detail of this process will be explained later. Because of the stares effect in the ridges, detection of some ridges may miss. To eliminate this problem, we check the 8-connected neighbors of every pixel on the straight line obtained by using eq-4.5 in order to avoid the ridge missing. The problem of detection of multiple locations on the same ridge is avoided by taking only those locations that belong to different ridges. It is assured by taking only those locations which are not 8-connected. When a location is detected on the ridge, the point that we increment manually is increased by '3' in order to speed up the task of location detection on different ridges.

4.5.2 Fuzzy logic based decision

For increasing the accuracy of algorithm and to include human thinking and behavior in an automated fingerprint recognition algorithm, fuzzy logic based decision for acceptance or rejection of each minutia is incorporated. Introduction of fuzzy logic helps in saving true minutiae that may be deleted from initial (unconfirmed) list otherwise. By considering the importance of each classifier, a confidence level is attached with them. The decision of assigning specific confidence level to each feature depends upon experience that is obtained by running the algorithm multiple times. Confidence levels which are assigned to 'distance', 'relative angle', 'angle between orientations' and 'distance from center to ridge intersection' are '40', '30', '20' and '10'. The decision about acceptance or rejection of minutia which is taken on the basis of these confidence levels is specified in Table 4.2.

Table 4.2: Decision about acceptance or rejection of minutia based on fuzzy logic

Distance to central point	**Relative angle**	**Angle between orientations**	**Distances from center to ridge intersection**	**Decision**
Y (40)	N (0)	N (0)	N (0)	N
Y (40)	N (0)	N (0)	Y (10)	N
Y (40)	N (0)	Y (20)	N (0)	N
Y (40)	N (0)	Y (20)	Y (10)	N
Y (40)	Y (30)	N (0)	N (0)	N
Y (40)	Y (30)	N (0)	Y (10)	Y
Y (40)	Y (30)	Y (20)	X	Y
N (0)	X	X	X	N

If the minutia fails to satisfy first feature in the Table 4.2 (distance to central point) then all other classifiers or features become do not care as specified by 'X' in last row. 'X' means that values (confidence levels) attached to these features have no effect in final decision about acceptance or rejection of minutia. On satisfying each classifier, corresponding value (confidence level) is added in a variable (confidence). If the sum of confidence levels becomes greater than 70% (0.7) then that minutia will be added to final (confirmed) list which is used for matching score calculation. To understand the working of fuzzy logic based module, the flow of control in this module is presented in the form of flow chart in Figure 4.13.

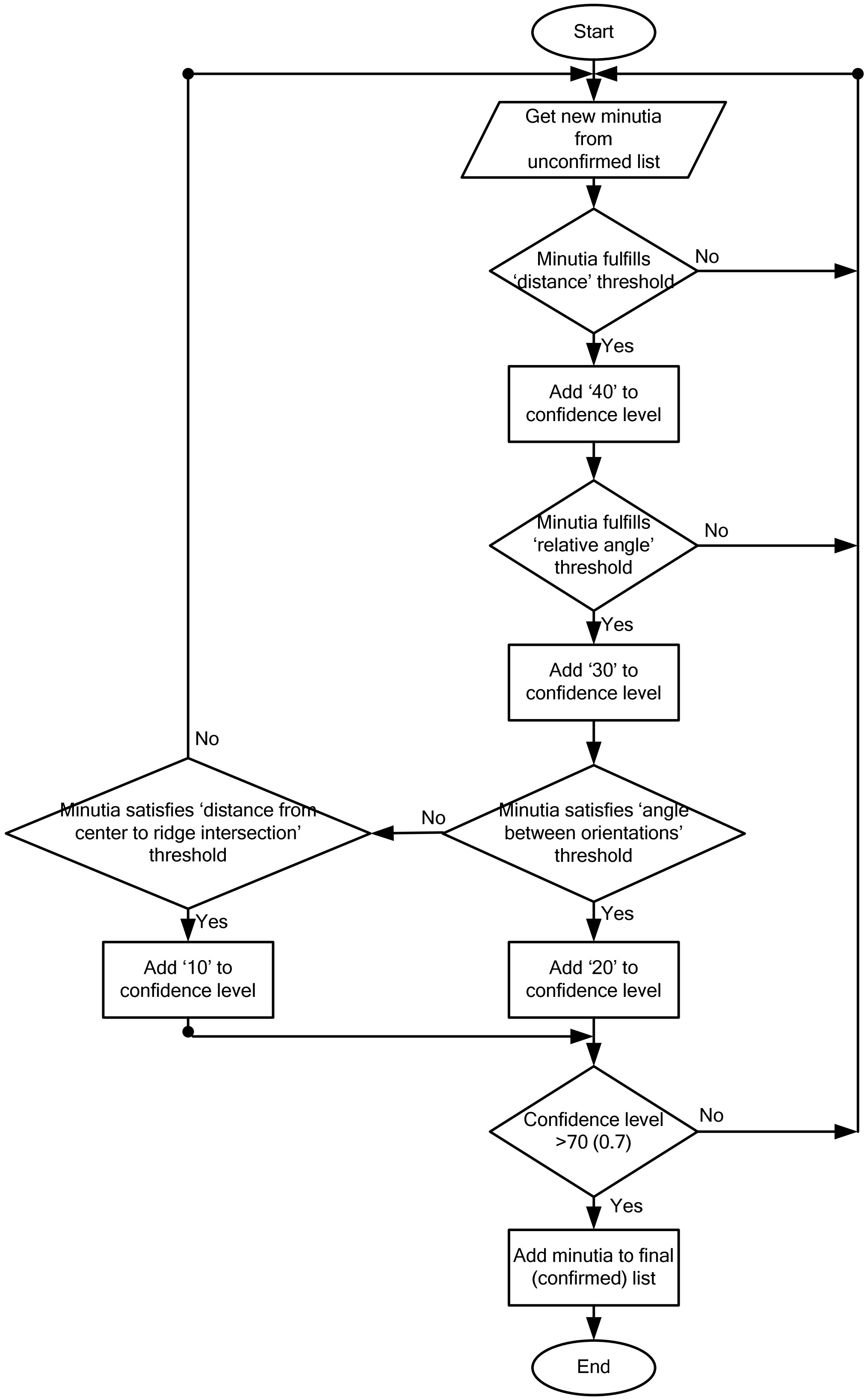


Figure 4.13: Flow chart of fuzzy logic based decision

4.5.3 Matching score calculation

All the minutiae that are included in final (confirmed) list after satisfying the criteria mentioned in above section are used in final matching score calculation. Matching score specifies the degree of similarity between two images. This score is usually higher in case of similar images (images from same individual) and lesser for different images (images from different individuals) otherwise some modifications are required to correct the behavior of concerned algorithm. But some algorithms use these matching scores in different ways. Many different methods are available for its calculation but we have used the most common and popular approach [34]. The formula which is used for its calculation is specified by eq-4.8.

$$score = \frac{MP * MP}{NI * NT} \tag{4.8}$$

Where 'MP' is the number of points matched between two images. 'NI' and 'NT' are the number of points (minutiae) in input and template fingerprints. If one of the images (input or template) is incomplete then this method is very helpful in handling this situation. The number of matched minutiae would be reduced because of incompletion of one image that may affect the matching score. This problem has no influence on matching score because the total points in corresponding image (NI or NT) would also be reduced that eliminates the bad effect of this problem.

4.6 Summary

This chapter provides the implementation details of this thesis work and describes the adopted approaches for materializing the earlier studied theoretical baseline. The methods of implementation of decomposed modules mentioned in chapter 3 are discussed in detail with their effect on accuracy improvement. For better understanding, the required concepts and techniques are presented in different (algorithmic, flow charts and theoretical) ways.

Chapter 5

RESULTS AND ANALYSIS

5.1 Introduction

This chapter contains detailed description of the experiments conducted along with their results. The characteristics of database which is used for performance evaluation of the algorithm are presented. The performance parameters or metrics which are required to check the accuracy of software are discussed. The comparisons between different algorithms are also made to analyze the performance of proposed algorithm.

5.2 Performance measures

The performance parameters which are discussed in this section with their implementation methodology are FAR (false accept rate), FRR (false reject rate) and ROC (receiver operating characteristic) curve. Before going in to details of these metrics, the properties of database must be known on which the algorithm runs that is analyzed by using above mentioned evaluation parameters. Database (DB1_A) of FVC 2002 [35] was used to evaluate the performance of proposed algorithm. There are 100 fingerprints in this database belonging to different individuals with 8 impressions per finger (100 subjects x 8 fingers per subject) so in total 800 fingerprints were used for performance evaluation. The images are limited to 256 gray levels with TIF format and 500dpi resolution. Images are uncompressed, with orientation of image in the range of [-15°, +15°] is used with respect to vertical orientation.

5.2.1 False accept rate Vs False reject rate

To calculate these two error rates, the understanding of procedure to compute genuine and imposter scores is critical. Each sample of a fingerprint is matched against the remaining samples of same finger and this process is repeated for all individuals to compute genuine score distribution. The first sample of each finger of an individual is matched against the first sample of remaining individuals and this process is repeated for all individuals to calculate imposter matching scores.

The genuine matching score distribution and imposter matching score distribution are computed that show how the algorithm separates the two classes. In fingerprinting, higher scores are associated with more closely matched images. Every input image (fingerprint) that is provided to the matcher has one of the two possible results, “match” or “non-match”. As a result, there are four possible scenarios:

- a genuine individual is accepted
- a genuine individual is rejected
- an imposter individual is accepted
- an imposter individual is rejected

An ideal fingerprint authentication system may generate only the first and fourth outcome. Because of image quality and other intra class variations in the fingerprint capture devices, and the limitations of fingerprint image analysis systems, enhancement methods, feature detection algorithms and matching algorithms, a genuine individual could be mistakenly recognized as an imposter. This scenario is called “false reject” and the corresponding error rate is called the false reject rate (FRR). An imposter individual could be also incorrectly recognized as genuine. This scenario is called "false accept" and the corresponding error rate is called the false accept rate (FAR). These two error rates can be defined in the following way.

"Given a threshold (t), FAR denotes the percentage of 'ims' (imposter matching scores) which are greater than or equal to t where 't' specifies a threshold at which the value of FAR is computed". To grasp this concept, related formula is presented below. FAR is also called FMR (false match rate).

$$FAR(t) = \frac{card\{ims \mid ims \geq t\}}{NIRA} \tag{5.1}$$

Where 'NIRA' is the number of imposter recognition attempts. FRR can be defined in the following way.

"Given a threshold (t), FRR denotes the percentage of 'gms' (genuine matching scores) which are less than t where 't' specifies a threshold at which the value of FRR is computed". To grasp this concept, related formula is presented below. FRR is also called FNMR (false non-match rate).

$$FRR(t) = \frac{card\{gms \mid gms < t\}}{NGRA} \tag{5.2}$$

Where 'NGRA' is the number of genuine recognition attempts.

The above mentioned equations are presented in [36, 37]. The distributions of the similarity score of genuine attempts and imposter attempts cannot be separated completely by a single carefully chosen threshold. Both FAR and FRR are actually functions of a threshold (t). If the value of threshold (t) increases then system will become more secure and FAR will be reduced. But on decreasing the value of threshold (t), system will be more flexible to intra class variation and noise. FRR will decrease but FAR will increase in that case which makes the system insecure. The selection of a threshold depends upon the application in which recognition software will be embedded. The pictorial representation of this concept is given in Figure 5.1.

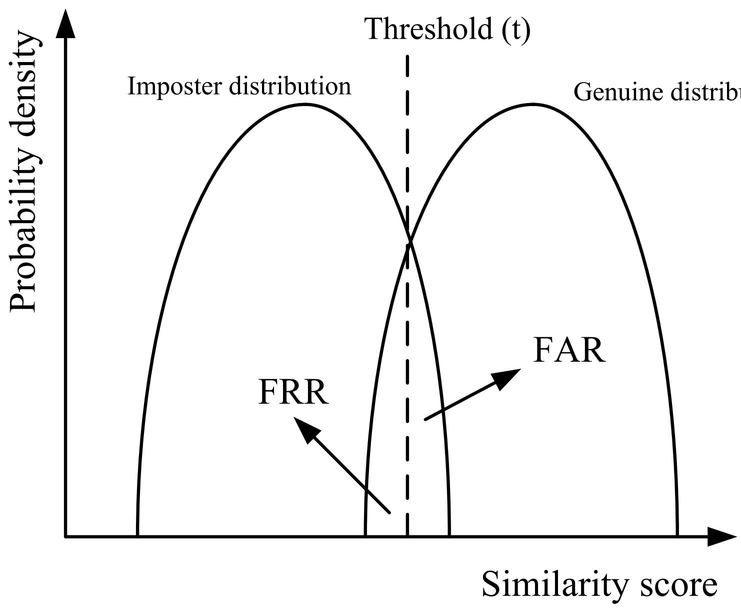


Figure 5.1: Example of genuine and imposter distributions

5.2.2 Receiver operating characteristic (ROC) curve

ROC is obtained by plotting FAR (false accept rate) against FRR (false reject rate) at all thresholds. The threshold (t) of authentication system can be tuned to meet the requirements of a particular application.

5.3 Experimental results

There are different methods that can be used to analyze the performance of an algorithm. Here ROC (receiver operating characteristic) curve is presented in which FAR (false accept rate) is plotted against FRR (false reject rate) at different thresholds. FRR is plotted as a function of FAR. The graph is plotted in log-log scale for better comprehension. The performance of approach adopted by [1] is depicted in Figure 5.2.

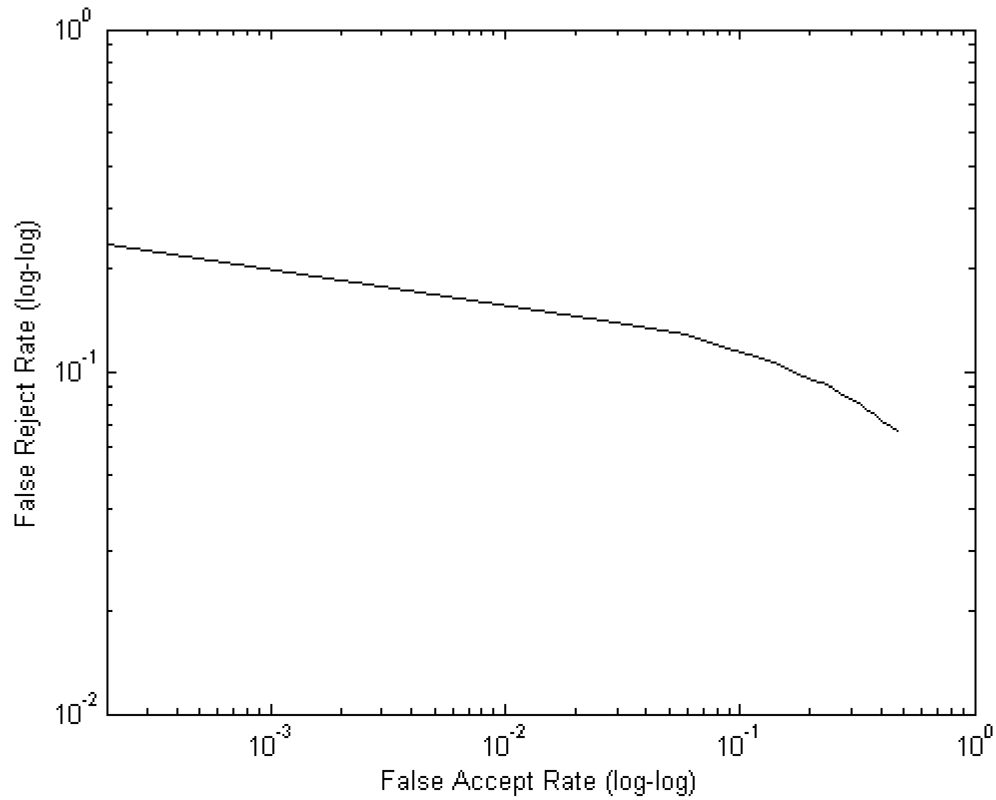


Figure 5.2: ROC curve for previous approach

The above mentioned results in pictorial form are demonstrated in tabular form in Table 5.1 for better understanding.

Table 5.1: Tabular representation of FAR Vs FRR for previous approach

Serial #	False Accept Rate	False Reject Rate
1	0.0002	0.2386
2	0.0294	0.1393
3	0.0587	0.1279
4	0.0880	0.1161
5	0.1174	0.1111
6	0.1467	0.1050
7	0.1760	0.0985

The above table shows FAR versus FRR. The selection of one error against other depends upon the application in which the recognition software is going to use. For very critical system, FAR must be kept as minimum as possible by increasing threshold value. On reducing FAR too much FRR becomes unacceptable so this problem must be kept in mind before the selection of two error rates. For less critical applications, FAR is allowed to increase to some extent for decreasing FRR. This can be done easily by reducing threshold value (t). Several modifications have been made in the technique presented by [1] to increase the accuracy of this approach. To see the effect of these modifications, receiver operating characteristic (ROC) curve of the proposed approach is given in Figure 5.3.

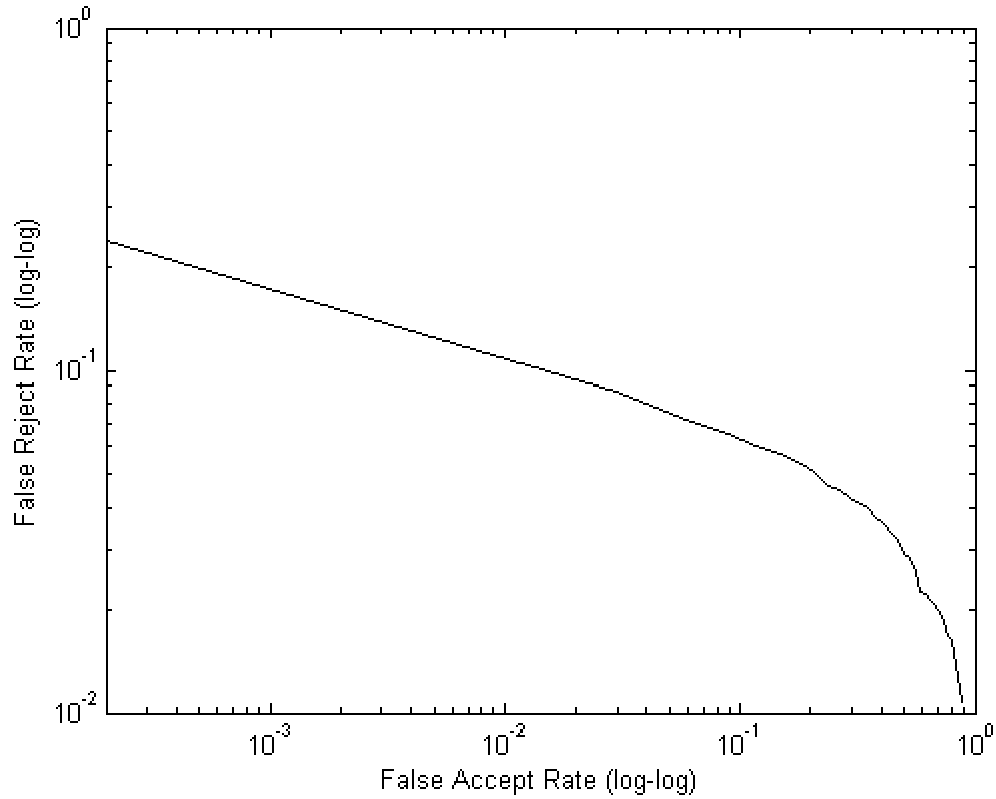


Figure 5.3: ROC curve for proposed approach

The tabular form of above figure is given in Table 5.2 for grasping the results of proposed approach effectively.

Table 5.2: Tabular representation of FAR Vs FRR for proposed approach

Serial #	False Accept Rate	False Reject Rate
1	0.0002	0.2386
2	0.0294	0.0864
3	0.0587	0.0714
4	0.0880	0.0657
5	0.1174	0.0607
6	0.1467	0.0575
7	0.1760	0.0542

For better understanding, it is necessary to compare the approach presented by [1] and proposed approach. ROC curves for both of these approaches are depicted in Figure 5.4 in log-log scale for better comprehension.

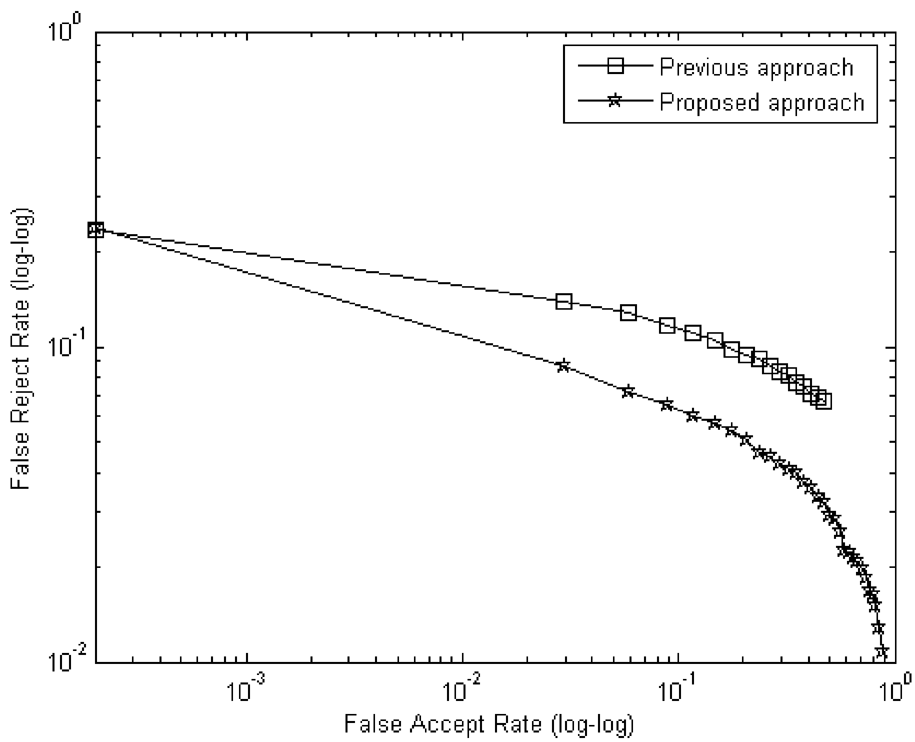


Figure 5.4: ROC curves for previous and proposed approach

The comparison between [1] and proposed approach in tabular form is presented in Table 5.3.

Table 5.3: Comparison between two approaches

Serial #	Previous Approach		Proposed Approach	
	False Accept Rate	**False Reject Rate**	**False Accept Rate**	**False Reject Rate**
1	0.0002	0.2386	0.0002	0.2386
2	0.0294	0.1393	0.0294	0.0864
3	0.0587	0.1279	0.0587	0.0714
4	0.0880	0.1161	0.0880	0.0657
5	0.1174	0.1111	0.1174	0.0607
6	0.1467	0.1050	0.1467	0.0575
7	0.1760	0.0985	0.1760	0.0542

To increase the accuracy of [1], some additional classifiers (features) were required to filter false or irrelevant minutiae. By adding these features, the average matching was slightly increased that is depicted in Table 5.4. The benefit which is achieved by adding these additional features is more as compared to cost in terms of time.

Table 5.4: Average times taken by algorithms

Algorithm	Matching time (sec)	Enrolment time (sec)
Wahab et al.'s method	0.5806	5.7
Proposed method	0.7200	

The improvement in accuracy that is achieved at the cost of increased time is given in Table 5.5.

Table 5.5: Improvement in accuracy

Algorithm	Equal Error Rate (EER)	Improvement (%)
Wahab et al.'s method	0.0685	40
Proposed method	0.1142	

Equal error rate is an error rate where FAR and FRR are equal. It is another method to depict the performance of an algorithm. The accuracy of an algorithm can be computed by subtracting equal error rate from 1 or 100 (when EER is represented in percentage).

EER is used as a parameter to compare the performance of our algorithm with various academic and industrial fingerprint recognition algorithms. The comparison between them is depicted in Table 5.6.

Table 5.6: Comparison between different fingerprint matching algorithms

Serial #	**Algorithm**	**Equal Error Rate (%)**	**Accuracy (%)**
1	**Proposed**	6.851	93.15
2	**BioGina**	7.95	92.05
3	**Digital Finger pass Co.**	8.27	91.73
4	**Satish et al's method [38]**	9	91
5	**Wahab et. al [1]**	11.42	88.58
6	**Algo-16 [35]**	16.28	83.72
7	**Algo-22 [35]**	17.34	82.66
8	**Algo-25[35]**	35	65
9	**Algo-03 [35]**	50	50

The entries from 6 to 9 in the above table are taken from [35]. The EERs and accuracies of each algorithm are presented in Table 5.6 to analyze the performance of algorithms. The accuracy of proposed algorithm is 93.15% which means that in 6.85% of the cases genuine users will be rejected or wrong users will be accepted.

5.4 Summary

This chapter provides the results of proposed technique when it is executed on standard database. The obtained results clearly show that adopted approach has been proved to be highly successful in fingerprint recognition. The proposed technique is compared with previous approach and various other approaches on the basis of accuracy. To demonstrate the accuracy of an algorithm, receiver operating characteristic (ROC) curve and equal error rate (EER) are used that are independent of computer platforms. The improvement that is achieved by introducing new classifier and modifications are also discussed.

Chapter 6

CONCLUSION

6.1 Overview

Because of the security concerns, the use of biometrics is increasing rapidly in our society. Among several biometric systems, fingerprint recognition is most popular because of low cost sensors, accuracy, time efficiency and availability of literature related to this field. There are several matching algorithms used nowadays which can be classified in to image based, minutiae based and ridge feature based. Several difficulties are associated with image and ridge feature based approaches so the method which is adopted in this research is minutiae based that eliminates most of the problems associated with two mentioned approaches.

The approach which has been presented here is based on structural matching that is not concerned with detection of reference points (core or delta) to perform its matching process. There is no pre-alignment stage in this technique which is used to align the corresponding minutiae of two images before matching. The matching algorithms which require reference points for alignment before matching produce bad results in case of bad quality images because the reference point may not present in those images. This problem is totally eliminated by proposed technique as reference points are not required for matching at all. In the proposed method, matching is done on the basis of five closest neighbors of one single minutia that is also called center point. An authentication of minutia is based on these surrounding neighbors. Increasing the number of neighbors for center point from five to six may rise processing time that is why this number is kept at five. Five neighbors are suitable for getting good performance and processing time. The method used to calculate the direction of bifurcation is adopted. The location where the bifurcation is detected and the location on the farther leg (leg of bifurcation which is far from other two legs) are used for direction calculation.

The proposed approach is divided in two stages. First stage performs initial matching and makes a list called unconfirmed common points list. Degree is assigned to each point on the basis of neighbors that are matched. After first stage, a suitable center point is selected on the basis of which second stage is performed. All the points with maximum degree as well as one point with max-1 degree are selected as center one by one. The matching score is calculated by considering all these points and the maximum score among them would be used as final matching score. Fuzzy logic is incorporated in second stage on the basis of which the decision about acceptance or rejection of each minutia has been made. A novel feature 'distance from center to ridge intersection' is introduced to increase the performance of proposed approach.

6.2 Future work

A lot of classifiers have been added to improve the performance of proposed approach but the accuracy which is achieved can be improved by incorporating more classifiers. The threshold associated with each classifier can be improved so that they can classify images more effectively. A lot of research is being carried out on Level 3 features. Level 3 features include pores and ridges. Pores are small dots that can be used for matching purpose.

In the future, these level 3 features can be merged with proposed approach for improving accuracy but these level 3 features can be detected effectively in the images having equal to or more than 1000 dpi resolution so high performance sensors are required to do the practical implementation of this hybrid approach. Along with two structures (endings and bifurcations), other structures (islands, bridges, dots) may be incorporated in different matching techniques for improving results.

BIBLIOGRAPHY

[1] Wahab A., Chin S.H., and Tan E.C., "Novel Approach to Automated Fingerprint Recognition," IEE Proceedings Vision Image and Signal Processing, vol. 145, no. 3, pp.160-166, 1998.

[2] U.S. Department of Defense, Biometric Management Office (BMO). Biometrics 101. http://www.biometrics.dod.mil/bio101/3.aspx, 2006.

[3] Jain, A.K., Ross, A., & Prabhakar, S., "An introduction to biometric recognition," IEEE Transactions on Circuits and Systems for Video Technology, vol.14, pp. 4-20, 2004.

[4] Frykholm N., "Passwords: Beyond the Terminal Interaction Model", University of Umea, 2000.

[5] Miller G. A., "The magical number seven, plus or minus two: some limits on our capacity for processing information," Psychological Review, vol. 63, pp. 81-97, 1956.

[6] Daugman, J., "How iris recognition works," IEEE Transactions on Circuits and Systems for Video Technology, vol.14, pp.21-30, 2004.

[7] Faundez-Zanuy, M., & Monte Moreno, E., "State-of-the-art in speaker recognition," IEEE Aerospace and Electronic Systems Magazine, vol.20, pp.7-12, 2005.

[8] Pentland, A., & Choudhury, T., "Face recognition for smart environments," Computer, vol.33, pp.50-55, 2000.

[9] BioAPI Consortium, "BioAPI Specification ver 1.1", http://www.bioapi.org/, 2001.

[10] Federal Bureau of Investigation (FBI) Homepage, "Fingerprint Identification Overview.," http://www.fbi.gov/hq/cjisd/iafis.htm, April 17, 2006.

[11] S. Pankanti, S. Prabhakar, and A. K. Jain., "On the individuality of fingerprints," Transactions on Pattern Analysis and Machine Intelleigence (PAMI), vol.24, no.8, pp.1010–1025, 2002.

[12] D. Maio, D. Maltoni, A. K. Jain, and S. Prabhakar., Handbook of Fingerprint Recognition, Springer Verlag, 2003.

[13] A.K.Jain, L. Hong and S.Pankanti and R. Bolle. "An Identity authentication system using fingerprints," In the proceeding of IEEE, vol. 85, no.9, pp. 1365-1388, 1997.

[14] H.C.L.ee and R.Gaenssleh, Advances in Fingerprint Technology, Elsevier, New York, 1991.

[15] Federal Bureau of Investigation, The Science of Fingerprints: Classification and Uses, U. S. Government Printing Office, Washington D.C., 1984.

[16] Stosz, J. D. and Alyea, L. A., "Automated System for Fingerprint Authentication Using Pores and Ridge Structure," In Proceedings of SPIE Conference on Automatic Systems for the Identification and Inspection of Humans, vol. 2277, San Diego, USA, pp. 210–223, 1994.

[17] Anil Jain, Yi Chen, and Meltem Demirkus, "Pores and Ridges: Fingerprint Matching Using Level 3 Features," IEEE transactions on pattern analysis and machine intelligence, vol. 4, pp 477–480, 2006.

[18] Hong, L., Wan, Y., and Jain, A. K. "Fingerprint image enhancement: Algorithm and performance evaluation," IEEE Transactions on Pattern Analysis and Machine Intelligence, vol. 20, no.8, pp 777-789, 1998.

[19] A. M. Choudhary and A. A. S. Awwal., "Optical pattern recognition of fingerprints using distortion-invariant phase-only filter," In SPIE, Photonic devices and algorithms for computer, vol. 3805, pp. 162–170, October 1999.

[20] S. H. Lee, S. Y. Yi, and E. S. Kim., "Fingerprint identification by use of volume holographic optical correlator," In SPIE Optical pattern recognition, vol. 3715, pp. 321–325, March 1999.

[21] D. Roberge, C. Soutar, and B. V. K. Kumar, "Optimal trade-off filter for the correlation of fingerprints," Optical Engineering, vol. 38, pp.108–113, 1999.

[22] K. Nandakumar and A. K. Jain., "Local correlation-based fingerprint matching," In Indian Conference on Computer Vision, Graphics and Image Processing, pp. 503–508, 2004.

[23] A. M. Baze, G. T. B Verwaaijen, S. H. Garez, L. P. J. Veelunturf, and B. J. van der Zwaag, "A correlation-based fingerprint verification system," In ProRISC2000 Workshops on Circuits, Systems and Signal Processing, Nov 2000.

[24] Gonzalez, Woods, and Eddins, Digital Image Processing, Prentice Hall, 2004.

[25] Hrechak A. and McHugh J., “Automated Fingerprint Recognition Using Structural Matching,” Pattern Recognition, vol. 23, no. 8, pp. 893-904, 1990.

[26] Chen Z. and Kuo C.H.,“A Topology-Based Matching Algorithm for Fingerprint Authentication,” In Proc. of 25th Int. Carnahan Conf. on Security Technology, pp. 84-87, 1991.

[27] Abinandhan Chandrasekaran, Bhavani Thuraisingham, “Fingerprint Matching Algorithm Based on Tree Comparison using Ratios of Relational Distances,” The Second International Conference on Availability, Reliability and Security(ARES), pp. 273-280, 2007.

[28] S.Chikkerur, A. N. Cartwright and V. Govindaraju, “Kplet and CBFS: A Graph based Fingerprint Representation and Matching Algorithm,” ICB, Hong Kong, 2006.

[29] Thornton, J., “Latent Fingerprints, Setting Standards in the Comparison and Identification,” In 84th Annual Training Conference of the California State Division of IAI, Nevada, USA, May 2000.

[30] D. R. Ashbaugh, Quantitative-Qualitative Friction Ridge Analysis: An Introduction to Basic and Advanced Ridgeology, CRC Press, 1999.

[31] Anil Jain, Salil Prabhakar, Lin Hong, and Sharath Pankanti, “Filterbank-based fingerprint matching,” In Transactions on Image Processing, vol. 9, pp. 846–859, May 2000.

[32] http://www.doc.ic.ac.uk/nd/surprise96/journal/vol4/sbaa/report.fuzzysets.html

[33] http://www.fuzzy-logic.com/Ch1.htm

[34] Asker M. Bazen and Sabih H. Gerez, “Fingerprint matching by thin-plate spline modeling of elastic deformations,” Pattern Recognition, vol.36, no.8, pp.1859 1867, 2003.

[35] http://Bias.Csr.Unibo.It/Fvc2002/, the Fingerprint Verification Competition.

[36] http://Bias.Csr.Unibo.It/Fvc2000/, the Fingerprint Verification Competition.

[37] Weiping Chen and Yongsheng Gao, “A Minutiae-based Fingerprint Matching Algorithm Using Phase Correlation,” Digital Image Computing Techniques and Applications, pp. 233-238, 2007.

[38] Satishgunnam, “Fingerprint recognition and Analysis System,” Indiana State University, Department of Electronics and Computer technology, 2004.