Fingerprint Image Enhancement And It's Feature Extraction For Recognition

Pankaj Bhowmik, Kishore Bhowmik, Mohammad Nurul Azam, Mohammed Wahiduzzaman Rony

Abstract— Fingerprint recognition is one of the most popular and successful methods used for person identification, which takes advantage of the fact that the fingerprint has some unique characteristics called minutiae; which are points where a curve track finishes, intersect with other track or branches off. A critical step in studying the statistics of fingerprint minutiae is to reliably extract minutiae from the fingerprint images. However, fingerprint images are rarely of perfect quality. They may be degraded and corrupted due to variations in skin and impression conditions. Thus, image enhancement techniques are employed prior to minutiae extraction to obtain a more reliable estimation of minutiae locations. The goal of this paper is to represent a complete process of fingerprint feature extraction for minutiae matching.

Index Terms— Fingerprint, Minutiae points, Bifurcation, DFT filter, Histogram Equalization.

————————————◆————————————

1 INTRODUCTION

BIOMETRICS is the automatic identification of an individual based on his or her physiological or behavioral characteristics. The ability to accurately identify or authenticate an individual based on these characteristics has several advantages over traditional means of authentication such as knowledge-based (e.g., password) or token-based (e.g., key) authentication **[1].**Due to its security-related applications and the current world political climate, biometrics has recently become the subject of intense research by both private and academic institutions. There are several human characteristics that can be used as the basis for biometric systems. For example, a person's face, retina, or voice can all be used to identify that individual with a high degree of accuracy. The use of fingerprints has several advantages over the other methods, and therefore is one of the most researched and mature fields of authentication. The uniqueness of fingerprints has been studied and it is well established that the probability of two fingerprints matching is vanishingly small. The probability that two fingerprints are alike is 1 in 1.9×10^{15} **[2].** Furthermore, unlike faces and voice prints, fingerprints are persistent with age and can not be easily disguised. In the biometric process of finger scanning, a **ridge** is a curved line in a finger image. Some ridges are continuous curves, and others terminate specific points called **ridge endings**. Sometimes, two ridges come together at a point called a **bifurcation [3].** Ridge endings and bifurcations are known as **minutiae [4].**

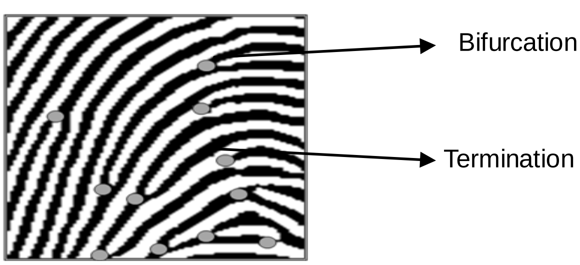


Fig 1.1: Dot representing minutiae

Fingerprints can be categorized based on their global pattern of ridges and valleys. According to Henry there are eight categories are known as "**Henry's Classification**" **[5]** and examples of each are shown in Fig 1.2.

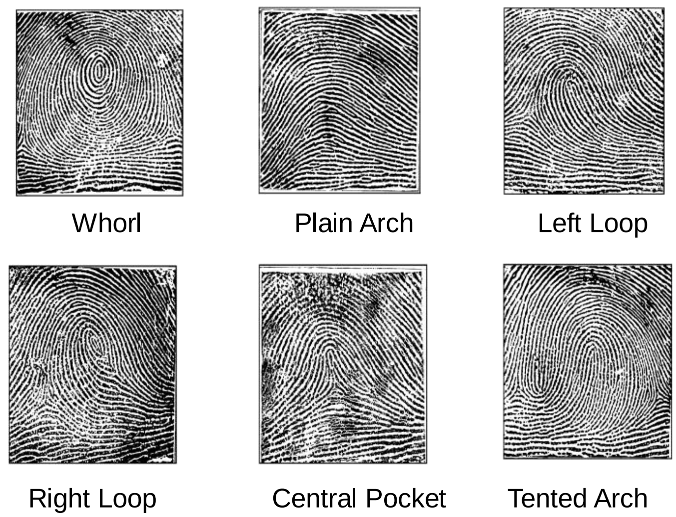


Fig 1.2**:** Henry's fingerprint classes

2. PROPOSED METHODOLOGY

The proposed system firstly acquires the fingerprint image. Next combination of two image enhancement processes (i.e. DFT & Histogram equalization) reconstructs the information of the fingerprint image; the methodologies for these processes viewed as some blocks in **Fig 1.3.** Main steps are (1) Image Acquisition (2) Image enhancement (3) Binarization (4) Thinning and (5) Feature Extraction.

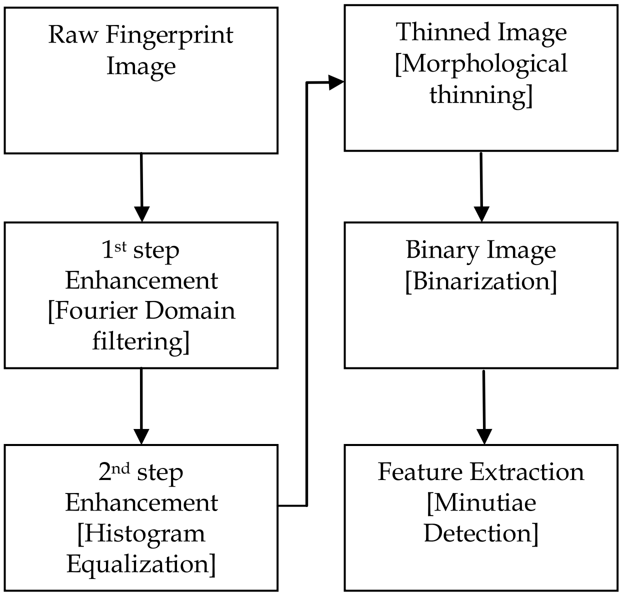


Fig 1.3: Flowchart of the proposed algorithm

2.1 Image Acquisition

Fingerprint Image can be acquiesced by different types of scanners, optical sensor, capacitive sensor or thermal sensor. The images are of poor quality and so the enhancement step is essential.

2.2 Fingerprint Image Enhancement

The performance of minutiae extraction algorithms and other fingerprint recognition techniques heavily relies on the input fingerprint image quality. In an ideal fingerprint image, ridges and valleys alternate and flow in a locally constant direction. The goal of an enhancement algorithm is to improve the clarity of the ridge structures in the recoverable regions and mark the unrecoverable regions as too noisy for further processing.
For image Enhancement we have used Fourier Domain Filtering and histogram equalization techniques.

DFT: The Discrete Fourier Transform is an important image processing tool which is used to decompose an image into its sine and cosine components.The Fourier transform of the block is multiplied by its power spectrum raised to a power of *k*. The basic is splitting image into frequency domain F (u, v) and multiplying this frequency domain with a constant such as **$|F(u, v)|^k$ [6]**. After doing this operation if we take inverse Fourier to this altered frequency domain we will get enhanced image. Flowing equation describe the whole process

$$i_{\text{enh}}(x,y) = F^{-1}\{F(u,v) \times |F(u,v)|^k\} \quad (2.1)$$

Where, ***i(x,y)*** is the original image and F(u,v) in frequency domain. The procedural formula of the conversion of an image of **M x N** to in frequency domain is given by

$$F(u,v) = \sum_{x=0}^{M-1}\sum_{y=0}^{N-1} i(x,y)\exp\left\{ j2\pi\left(\frac{ux}{M}+\frac{vy}{N}\right)\right\} \quad (2.2)$$

Histogram Equalization: This method usually increases the global contrast of many images, especially when the usable data of the image is represented by close contrast values. Through this adjustment, the intensities can be better distributed on the histogram. This allows for areas of lower local contrast to gain a higher contrast without affecting the global contrast. Histogram equalization accomplishes this by effectively spreading out the most frequent intensity values. It is reasonably visible that after performing DFT and histogram equalization fingerprint image quality has increased. By using two stage cascading enhancements process showing much better result. The step by step image are displayed in Fig 2.1

(a) (b) (c)

Fig 2.1: Original Image (a) Enhanced image by DFT transformation (b) Finally Enhanced Image after Histogram Equalization (c)

3. Binarization Of Fingerprint Image:

Binarization is a process that converts a grey level image into a binary image and in a binary image each pixel value is either 0 or 1(255). Most minutiae extraction algorithms operate on binary images where there are only two levels of interest: the black pixels that represent ridges, and the white pixels that represent valleys. This improves the contrast between the ridges and valleys in a fingerprint image, and consequently facilitates the extraction of minutiae. We have implemented adaptive binarization method and in this method there is a threshold value and below this all pixels are 0 and above this all pixels are 255 i.e. 1.

(a) (b)

Fig 3.1: Enhanced Image(a) Binary Image(b)

4. Fingerprint Ridge Thinning:

Thinning is the process applied over binarized image, from previous step, by thinning certain pattern shapes until it is represented by 1-pixel wide lines. Fingerprint thinning is usually implemented via morphological operations such as erosion and dilation to reduce the width of ridges to a single pixel while preserving the extent and connectivity of the original shape **[7]**. The mathematical definition of erosion is explained in the following equation. The erosion of A by element B is denoted by:

$$A \Theta B = \{Z \mid (B)_x \cap A^c \neq \phi \qquad (4.1)$$

Sometimes using erosion might cause some features to be corrupted, so we use another function called dilation, its mathematical definition is shown in the following equation

$$A \oplus B = \{Z \mid (B)_x \cap A^c \neq \phi\} \qquad (4.2)$$

Implementing dilation before erosion stores small gaps before thinning the image. The morphological closing of image A by element B, denoted A · B, is simply dilation of A by B, followed by erosion of the result by B:

$$A.B = \{A \oplus B\} \Theta B \qquad (4.3)$$

After performing these algorithms Fig 4.1(b) was obtained from the Binary image. In the image Fig 4.1(b) the rigid lines are representing a line of width one pixel. After this stage the image is ready for feature extraction.

(a) (b)

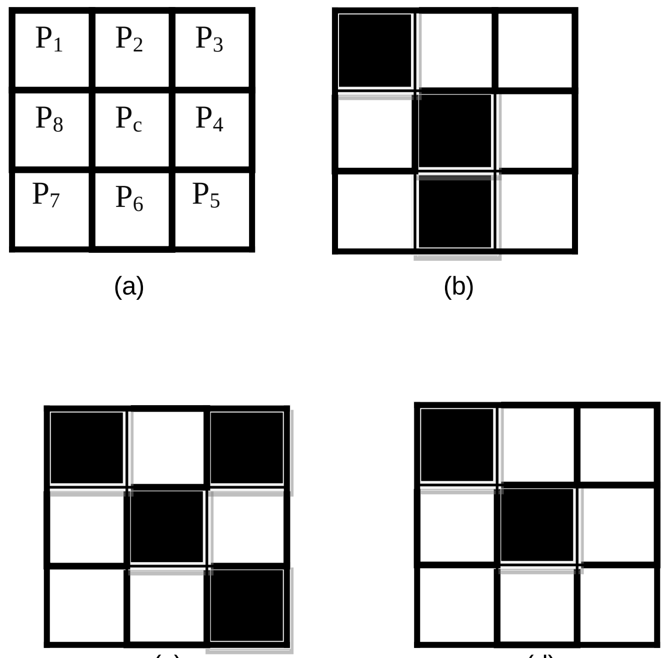
Fig.4.1: Binary Image(a) & thinned Image(b)

5. Minutiae Extraction:

The image obtained after binarization and thinning is ready to extract the features. If all the white pixels are 1 and black pixels are 0 then the width of the rigid lines of fingerprint is represented by only one zero pixel. In the image each pixel is surrounded by eight pixels and called neighbor pixel. In any point of a rigid line the summation of the neighbor pixels must be 6. If the rigid line is terminated then the summation will be changed and that will be 7. On the other hand in a point of bifurcation the summation will be 5 **[6]**. To develop this algorithm we have to consider **3x3** window and the window will scan the whole image where the center pixel is black or 0.

After passing the image through the algorithm we will get a new image where each bifurcation and termination is represented by a dot of a white pixel. The following algorithm and the Fig 5.1 representing the whole process.

$$P_c = \frac{1}{255}\sum_{i=1}^{8} P_i \qquad (5.1)$$



(a) (b)

(c) (d)

Fig 5.1: (a) 3x3 Window for searching minutiae (b) rigid line with P_c=6 (c) bifurcation with P_c=5 (d) termination with P_c= 7

After implementing the algorithm in Visual CPP platform using openCV library the following image is obtained. Each dot in the image represents either a bifurcation or a termination of a rigid line. The image data provide the coordinate of all bifurcating or terminating points of the input fingerprint image. This paved the way for the recognition of the image as these are the required features of an fingerprint image.

(a) (b)

Fig 5.2 (a): thinned image (b): Extracted features

6. VISUAL C++ PROGRAM FOR EXTRACTING THE MINUTIAE POINTS:

All the processes described in above are performed in visual CPP using openCV library. This Program generates an image like **Fig 5.2(b).** The portion of program that implements the algorithm in equation 5.1 and generates the image from Fig 5.2(a) to Fig 5.2(b) is given below.

```
#include <cv.h>                    // includes OpenCV
definitions
#include <highgui.h>               // includes highGUI
definitions
#include <string.h>
#include <iostream>                // includes C standard
input/output definitions
#include <stdio.h>
#include <math.h>
#include <fstream>
#include <iostream>

void filters(IplImage* image, int x,int y, double *xx);
IplImage *image = 0, *dstm = 0, *dstg=0,*src = 0, *srcg=0,
*srcr=0, *dstav=0, *dstsd=0, *dstva=0, *dstmax=0, *dstmin=0,
*dstmxmn=0, *dst2d=0;
char bufg[200], bufm[200], buft[200], buff[105];
char bufa[200], bufs[200], bufv[200], bufx[200], bufn[200],
bufxn[200], bufd[200];

IplImage* result;

// main  programs

int main(int argc, char** argv)
{
        FILE *fpo, *fpg;
        char* filename = argc == 2 ? argv[1] : (char*)"thin.tif";
        //unsigned char av, sd, va, maax, min;
        int w=1;

if( (image = cvLoadImage( filename, 1)) == 0 )
      return -1;

IplImage* img = cvLoadImage( "fp.jpg" );

int v_idx=0;
int pixel=0;

        double cn;

// convert to data or txt file

        if((fpo=fopen("aaa.txt", "w"))==NULL)
        {
                printf("file not open for writing \n");
                exit(0);
        }
        if((fpg=fopen("abb.txt", "w"))==NULL)
        {
                printf("file not open for writing \n");
                exit(0);
        }
        for(int j=0; j<image->height; j=j+1) {

                for(int i=0; i<image->width; i=i+1) {
                        int v_idx = j*image->width + i;
                        unsigned char pixel_value= image-
>imageData[v_idx];

fprintf(fpo,"%d %d %d\n",i,j,pixel_value);

fprintf(fpo, "%d %d %d %d %d\n", x, y, red, grn, blu);

                }
        }

        result =  cvLoadImage( "thin.tif", 0);
        cvZero( result );

        CvScalar s;
        s.val[0]= 255;

        //minutiae extraction

        for(int y=0; y<image->height; y=y+3)
        {
                for(int x=0; x<image->width; x=x+3)
                {
pc=(image->imageData[(j+1)*image->width+ (i+1)]);

                int v_idx = y*image->width + x;

                        filters(image, x, y, &cn);
                if(cn==4||cn==5||cn==7){
                        cvSet2D(result,y+1,x+1,s);
fprintf(fpo, "%d %d %lf\n",x+1,y+1,255.000000);
                }
            }
        }

cvSaveImage( "result.tif", result);

cvShowImage("fp.jpg", image );
cvWaitKey(0);
cvReleaseImage( &image );
cvDestroyWindow("fp.jpg");

return 0;
}
//window generator.

void filters(IplImage* image, int i, int j, double *cn)
{
                int x, y;
                unsigned char c1,c2,c3,c4,c5,c6,c7,c8;

                if (j<0 || j> image->height) {;}
                else if (i<0 || i> image->width) {;}
else {
        c1=(image->imageData[j*image->width+i]);
        c2=(image->imageData[(j+1)*image->width+i]);
        c3=(image->imageData[(j+2)*image->width+i]);
        c4=(image->imageData[(j+2)*image->width+(i+1)]);
        c5=(image->imageData[(j+2)*image->width+(i+2)]);
        c6=(image->imageData[j*image->width+(i+2)]);
```

```
            c7=(image->imageData[j*image->width+(i+2)]);
            c8=(image->imageData[j*image->width+ (i+1)]);
            pc=(image->imageData[(j+1)*image->width+ (i+1)]);

            c1 =abs(image->imageData[(j+1)*image->width+i] -
image->imageData[j*image->width+i]);;
            c2 =abs(image->imageData[(j+2)*image->width+i] -
image->imageData[(j+1)*image->width+i]);
            c3 =abs(image->imageData[(j+2)*image->width+(i+1)]
-  image->imageData[(j+2)*image->width+i]);//255;
            c4 =abs(image->imageData[(j+2)*image->width+(i+2)]
-  image->imageData[(j+2)*image->width+(i+1)]);//255;
            c5 =abs(image->imageData[(j+1)*image->width+(i+2)]
-  image->imageData[(j+2)*image->width+(i+2)]);//255;
            c6 =abs(image->imageData[j*image->width+(i+2)] -
image->imageData[(j+1)*image->width+(i+2)]);//255;
            c7 =abs(image->imageData[j*image->width+ (i+1)] -
image->imageData[j*image->width+(i+2)]);//255;
            c8 =abs(image->imageData[j*image->width+i] -
image->imageData[j*image->width+ (i+1)]);//255;

            *cn = (double)(c1+c2+c3+c4+c5+c6+c7+c8)/255.0;
                        }
                }
}
```

7. CONCLUSION

The key focus of this work is on the enhancement of fingerprint images, and the subsequent extraction of minutiae. By using two enhancement techniques (i.e. Fourier Domain Filtering and histogram equalization) the quality of input fingerprints image greatly increased, which pave the way to extract fingerprint features ie the minutiae points more accurately.

ACKNOWLEDGMENT

Firstly we would like to express our highest gratitude and special thanks to my supervisor, Professor Dr. Mohiuddin Ahamad who helped us at each and every points of our thesis work. His valuable suggestion and guidance which put us on the right track to fulfill the requirement and without his help it would not possible to presenting the thesis work. Support from the teachers of Department of Electrical and Electronic Engineering of Khulna University of Engineering and Technology (KUET) is much cherished us the facility which encouraged us to work more and more. And last but not the least I express my hearty thanks to Creator GOD and all those who supported us directly or indirectly to complete this task.

REFERENCES

[1] Salil Prabhakar “Fingerprint Classification and Matching Using a Filterbank” Ph.D. thesis, Michigan State University, 2001.

[2] Venu Govindaraju, Z. Shi and Schneider "Feature extraction using a chaincoded contour representation" International Conference on Audio and Video Based Biometric Person Authentication, 2003.

[3] EA. Afsar, M. Arif, and M. Hussain, "Fingerprint identification and verification system using minutiae matching" in National Conference on Emerging Technologies, 2004.

[4] Chandan Shanna, "DSP implementation of fingerprintbased biometric system" Tech. Rep., University of Auckland, 2005.

[5] Neil Yager, Adnan Amin “Fingerprint classification: a review” Received: 24 April 2003 / Accepted: 2 February 2004 / Published online: 9 March 2004.Springer-Verlag London Limited 2004.

[6] Gualberto Aguilar, Gabriel Sánchez, Karina To scano, Moisés Salinas, Mariko Nakano, Hector Perez “Fingerprint Recognition” National Polytechnic Institute , Mexico

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