American Journal of Electrical and Computer Engineering
2018; 2(2): 16-26
http://www.sciencepublishinggroup.com/j/ajece
doi: 10.11648/j.ajece.20180202.12
ISSN: 2640-0480 (Print); ISSN: 2640-0502 (Online)

Fingerprint Verification System Using Combined Minutiae and Cross Correlation Based Matching

Akinleye Okedola Akinyele[1], Adegbola Jamiu Sarumi[1], Badmus Abdulsamad[2], Olawole Olakunle Green[1]

[1]Department of Computer Engineering, Lagos State Polytechnic, Lagos, Nigeria
[2]Department of Electrical/Electronics, University of Lagos, Lagos, Nigeria

Email address:
leye_akinyele@yahoo.com (A. O. Akinyele), sarrosoft@gmail.com (A. J. Sarumi), toyin_ng@yahoo.com (B. Abdulsamad), greenoo@mocped.edu.ng (O. O. Green)

To cite this article:
Akinleye Okedola Akinyele, Adegbola Jamiu Sarumi, Badmus Abdulsamad, Olawole Olakunle Green. Fingerprint Verification System Using Combined Minutiae and Cross Correlation Based Matching. *American Journal of Electrical and Computer Engineering*. Vol. 2, No. 2, 2018, pp. 16-26. doi: 10.11648/j.ajece.20180202.12

Received: May 22, 2018; **Accepted**: June 20, 2018; **Published**: November 27, 2018

Abstract: An effective system to verify human identity is a major challenge in most traditional access control systems in developing countries, as it can easily be compromised. From the vulnerability of banking transactions to the multiple registration in most civic identification projects like voters' registration, Payroll System and Pension Scheme underscores the urgent need for an effective system that provide immediate technological solution. This research work presents an Automated Fingerprint Verification System simulating both Minutiae Based Matching and Cross Correlation Coefficient Matching that provides an effective and efficient means of verifying human identity which significantly decreases the possibility of fraud in access control. The method used MATLAB simulation to align the minutiae of the two-fingerprint image (query template) and stored templates (reference template) inputted to find the total number of minutiae matched. After alignment, two minutiae are considered for matching when spatial distance and direction difference between them are not up to a given tolerance. Finally, the templates were further verified with cross correlation algorithm to improve the result accuracy. This approach has better performance as compared to individual matching technique. The result obtained after testing several fingerprints for identification proved to be efficient by verifying correctly the identities of the persons enrolled and achieving a matching score above 80% threshold for Matching Pair and below 80% threshold for Non-Matching Pair.

Keywords: Access Control System, Fingerprint Verification System, Normalized Cross Correlation, Minutiae Score, Fingerprint Matching

1. Introduction

Fingerprint Verification System happened to be among the most effective and well-known form of biometrics generally used to uniquely verify and authenticate the identity of a person through automated system. This is achieved when two human fingerprints features are compared to obtain a match. It helps in the verification of a claimed identity by an individual and it is much related to techniques used in applications such as access control systems.

Scientific Research has shown that fingerprints are amazingly unique, there are no two persons in the whole world with the same ridge details, not even the identical twins. Fingerprints are permanent, inseparable from a person, except if damaged through scaring, fingerprints stay the same from birth to death. This unique fingerprint feature used in identification of persons helps to provides an immediate biometric solution to existing problems of traditional security access control system of lock and keys which can be stolen or forged, PIN code or passwords which can be forgotten or overheard and RFID card which can be used by an intruder without system detection of fraud.

In the fingerprints analysis for the purpose of matching, it requires generally the comparison of some important features of the fingerprint pattern. This include fingerprint patterns like arch, loop and whorl which are the combined characteristics of ridges and minutia points. Scientist has discovered that every family member often possesses the same fingerprints patterns. This actually is the basis of the common belief that patterns might be inherited. [1]

2. Methods of Fingerprint Recognition System

2.1. Categories of Fingerprint Recognition

The different fingerprint verification system techniques that exist today can be summarized into three major categories namely.

2.1.1. Pattern Based Matching

The algorithm for pattern-based matching is implemented by comparing the four basic fingerprint type patterns i.e. arch, whorl, left loop, and right loop between reference template and the user fingerprint. This enable the two images to be positioned using the same orientation. The algorithm then locates and centers on a central point in the image. This technique considers the fingerprint orientation, type and size of the pattern with the fingerprints pair examined. The user fingerprint image is graphically compared with the stored image to obtain a match.

2.1.2. Cross Correlation Based Matching

In this method, the algorithm two fingerprint images (query template and stored template) are super imposed whereby cross correlation coefficient for the two pixels are calculated for different rotations and displacements. It adopts a system of template matching that uses a high-level machine vision technique which identifies the parts on an image that match a predefined template.

2.1.3. Minutiae Based Matching

This system uses an algorithm to extract minutiae from the two fingerprints that is being compared and save it a group of Cartesian coordinates. It further checks the alignment of the two templates to know the minutia pairing with highest number.

2.2. Image Preprocessing

The next stage in the Fingerprint Verification System is the fingerprint image pre-processing, the digital fingerprint image is further processed by the following three stages grouped below.

1. Enhancements.
2. Binarization.
3. Image Segmentation.

MATLAB software implement both *Fourier Transform* and *histogram equalization* to achieve image enhancement and then a local adaptive technique was used to achieve fingerprint binarization.

The fingerprint image is segmented in three stages: the first is *block direction estimation,* while the second stage is *segmentation using direction intensity* and the third is *Region of Interest* extraction by Morphological operations. The algorithm used in the pre-processing stage was developed by other researchers. [3]

Enhancement.

This is a very vital stage in image processing. It enhances the fingerprint image by making it clearer for further processing operations. It is generally observed that fingerprints obtained from the scanner sensors usually come in low quality. This makes enhancing the fingerprint images very necessary by linking the broken points of the fingerprint ridges through contrast improvement. This actually helps in reducing the error rates in the Fingerprint Verification. Histogram Equalization and Fourier Transform were used to achieve image enhancement.

Histogram Equalization: It is a method to vary the intensities of the image to improve the contrast.

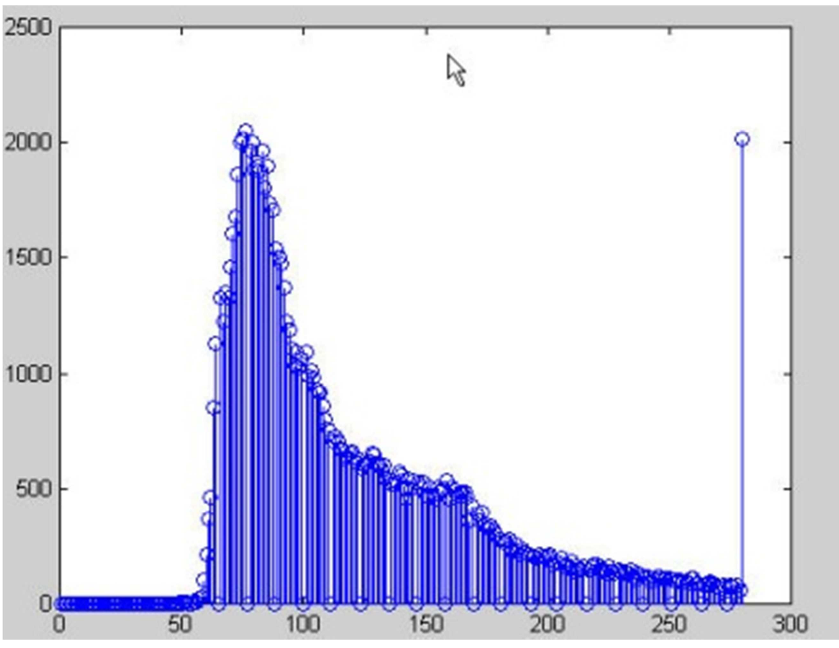

Figure 1. *Input fingerprint image graph.*

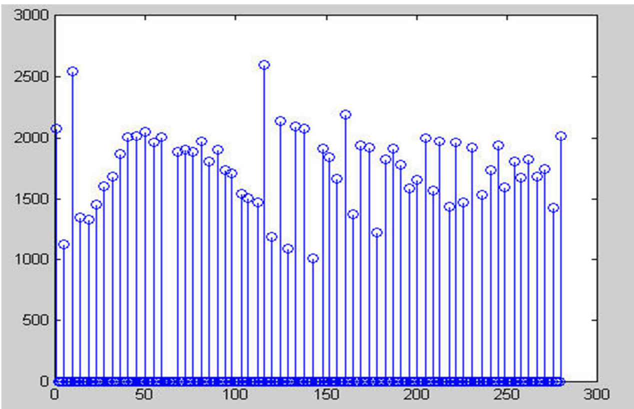

Figure 2. *Output Fingerprint Image after enhancement.*

The figure 1 is the input image graph while the figure 4 is the output after the histogram equalization. This reveals that the fingerprint image was enhanced using contrast for easy image processing.

Image Binarization.

This process implements a locally adaptive technique to convert the original fingerprint image (8-bit gray scale) to a binary image(1-bit) whereby the ridges and furrows are coded in 0-value and 1-value respectively. Once this is done, both the ridges and furrows of the fingerprint are displayed with black and white colors respectively.

Image Segmentation.

It is worth mentioning that only a Region of Interest (ROI) is useful in Fingerprint recognition. The other sections of the fingerprint image are of no value because it contains only the background information.

In order to eliminate the confusion of combining the spurious minutiae with the genuine ones, a boundary is sketched out for the effective area of interest.

Minutiae Extraction.

Minutia feature extraction stage involves both image thinning and minutiae marking operations. For minutia extraction stage, most studies have revealed that morphological thinning operation usually produce high efficiency and a very good thinning quality as compared with other techniques of thinning.

Fingerprint Ridge Thinning.

This stage used parallel iterative thinning algorithm to remove every ridge with redundant pixels until it becomes one pixel wide. In every image scanned, the algorithm eliminates redundant pixels by marking them down in a 3X3 pixels window. At the end, it removes the marked down pixels through series of scanning.

During testing stage, the parallel iterative algorithm for thinning had low efficiency. This can be improved after several scans. The thin ridge is extracted using one in all method. It thereafter finds through traces of fingerprint ridges with the highest intensity value in gray scale using enforced binarization.

Minutia Marking.

Once the fingerprint ridge thinning is completed, the next vital step is marking minutiae points, it is important to note that, the number of minutiae detected the more the probability of obtaining accurate result increases. The biometric concept of crossing numbers is used for extracting the minutiae points within the MATLAB block code.

2.3. Matching

The Fingerprint Verification System minutia matcher require three steps namely:

1. Correlation of the ridges to determine minutiae pair referenced.
2. Alignment Stage.
3. Match Stage.

The minutia matcher selects any two of the genuine minutiae to be the referenced pair then match their ridges associated first. After matching the ridges well, thereafter, two fingerprints are aligned and the remaining minutia goes through the same process for matching. The minutiae matcher algorithm finds whether the two sets of minutiae belong to the same finger or not.

Alignment stage: At this stage, the algorithm receives two fingerprints that would be matched, then select one minutia from each image, compute the similarities of the ridges when compared with the two minutia points referenced. Then, if the distance of the similarity is longer than a given threshold, transform the minutiae set to a new cartesian coordinate with reference point origin.

Match stage: once the two minutia points have been aligned and transformed to a new coordinates system. The match algorithm then counts the total number of matched minutiae by considering the two points of minutiae with almost the same value for direction and position with the parameters (x, y,θ).

The total match ratio computed for the two fingerprints is the number of total matched pair over the number of minutiae of the template fingerprint. It further adds the matching score for the result obtained during cross correlation based matching and average matching score is computed. The score is 100*matching ratio which ranges around (0 to 100). If the score is larger than a given specified threshold (i.e. 80%), the two fingerprints are similar else they are from different fingers.

2.4. Cross Correlation Verification

This stage further verifies the identity of the enrollee by processing both the queried and stored fingerprints to obtain a match. This is achieved by the algorithm for normalized cross correction (NCC) template matching which made use of the pixel value of the fingerprint images. The algorithm obtains information from the raw fingerprint images on a grey scale. It first chooses the fingerprint templates, chooses its pixel value and then correlate it with other pixel vales of the remaining database number of images and then finds the maximum value of correlated data usually more than a given threshold. At the end, a matching score is produced for two fingerprint templates compared. The result is given in addition to the matching score of minutiae-based matching to determine average matching score.

2.5. Implemented Algorithm

The following is the algorithm implemented for the minutia based matching and cross correlation coefficient matching technique of Fingerprint Recognition System.

Algorithm: To Compare Two Fingerprint Images and Determine Whether They match using Minutiae Based Matching.

Step 1: Read the User Fingerprints
a = Imread ('image 1')
Step2: Load the User's Fingerprints on the Software
Disp('image 1')
Step 3: Read the Unknown Fingerprint from the scanner
b = Imread('image 2')
Step 4: Load the Unknown Fingerprint on the software
Disp ('image 2')
Step 5: Compute the cross correlation coefficient of the two fingerprints
Step 6: Store the result in the memory and display.
Step 7: Implement Histogram Equalization of Image 1 and Image 2
Histeq('image 1')
Histeq('image 2')
Step 8: Compute Fast Fourier Transform of the Fingerprint for Image1 and Image 2
fft('image 1')
fft(image 2')
Step 9: Implement Binarization and Direction for the two Fingerprints Image
image1=adaptiveThres(double(image1)
image2=adaptiveThres(double(image2)
image1=im2double(bwmorph(o2,"thin",Inf))
image2=im2double(bwmorph(o2,"thin",Inf))
Step 10: Estimate Region of Interest on the two Fingerprints
a = drawROI(image1,o1Bound,o1Area)
b = drawROI(image2,o1Bound,o1Area)
Step 11: Extract Real Minutiae Points Both fingerprint images
mark_minutia(o1,o1Bound,o1Area,w)show_minutia(o1,end_list1,branch_list1)
Step 12: Store Minutiae Points Coordinate Positions for both Images
save(W,"real_end1","pathMap1","-ASCII");'])
Step 13: Compare whether the minutiae points are equal['finger1=fingerTemplateRead;'…
'finger2=fingerTemplateRead;'…'percent_match=match_end(finger1,finger2,10);']);
h =percent_match + max_cc
Step 14:
IF Yes Goto 14
Step 15:
IF No Goto 15
Step 16: Print the two fingerprints Match
fprint('The two fingerprints match')
Step 17: Print the two fingerprint does not match.
Fprint('The two fingerprints does not match')
Step 18: End of the Algorithm
Stop

2.6. Flowchart

This is the diagrammatic representation of step by step modules of how the system software is implemented

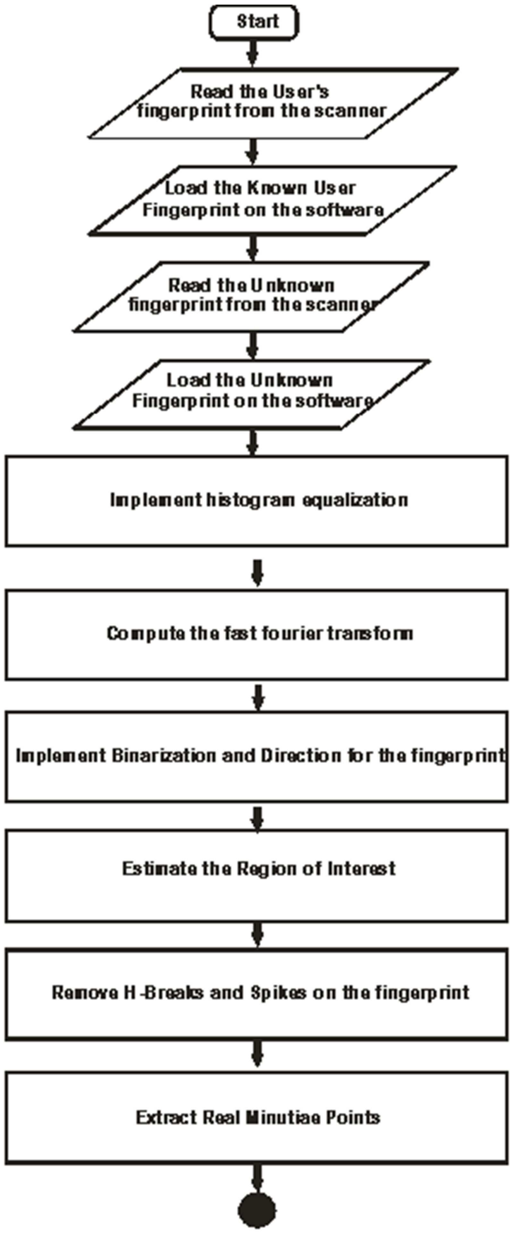

Figure 3. *Flow Chart.*

3. Results

Fingerprint Verification System simulation tested twelve live scan fingerprints pairs. The experiment was able to differentiate spurious minutia pairs from genuine minutia pairs and could verify easily and more efficiently the identity of a person based on the template data of the fingerprints stored. The Simulations produced the following results and the steps taken to verify an unknown fingerprint.

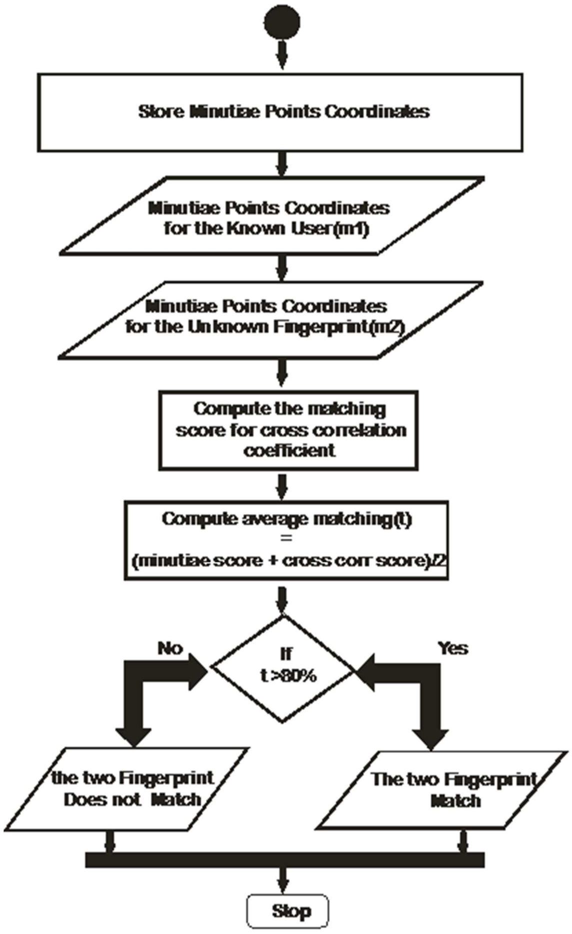

Figure 4. *Flow chart.*

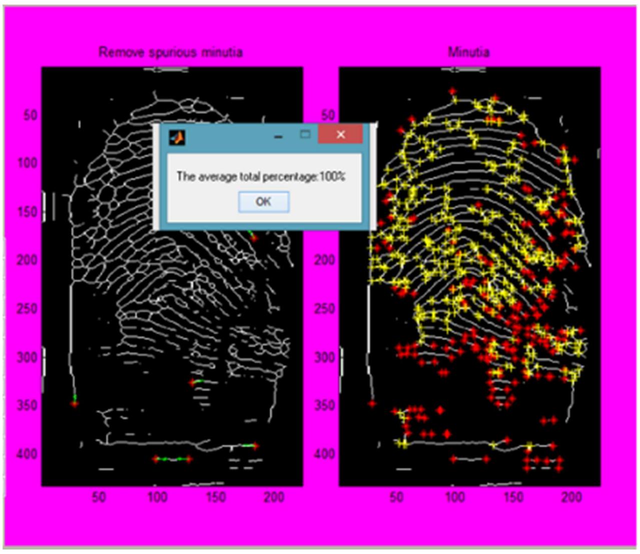

Figure 5. *Fingerprint Verification System Output using MATLAB Simulation.*

The Simulation is launched in MATLAB application to load an unknown image from the user and verify its identity based on the stored template. The output result is shown in the figure 7 below.

3.1. Performance Evaluation for False Acceptance and False Rejection

Various forms of error rates are used as measurement for evaluating the performance of fingerprint authentication systems generally. The output results of the comparison of the minutiae feature matcher within a fingerprint image recognition system is known as the matching score. It actually records and measures the similarity between the questioned fingerprint image and the stored fingerprint template based on the numbers of minutiae points and geometric directions. It would be interesting to know that the closer the fingerprint matching score approaches the unity value of 1, i.e. Range [0 1], the more likely it is that both fingerprints originate from the same finger. However, if matching score is near zero, it will be quite probable that both fingerprints are from different fingers. The decision of the system is determined by threshold T, i.e. if matching score passed the threshold, the fingerprints are regarded as being of the same finger (Matching Pair). If the matching score is below the threshold, the fingerprints are regarded as being different (Non-Matching Pair). In connection with this, two erroneous decisions, i.e. two kinds of mistakes, can be made by biometric systems during testing in which they are:

False Match- It happens in a situation whereby two fingerprint images of different fingers are categorized as being identical

False Non-Match-It happens in a case whereby two fingerprint images of the same finger are categorized as being different.

These two mistakes are often referred to as *False Acceptance* and *False Rejection.*

3.2. Types of Errors

False Acceptance Rate (FAR)
Definition:

$$\text{FAR} = \frac{\text{Number of comparison of different fingers}}{\text{Total number of comparisons of different fingers}}$$

False Rejection Rate (FRR)

$$\text{FRR} = \frac{\text{Number of comparisons of the same fingers}}{\text{Total number of comparison of the same fingers}}$$

The two indexes above are well accepted to determine the performance of a fingerprint recognition system: FRR (false rejection rate) and the other is FAR (false acceptance rate). In the case of image database, one template is matched against the other templates of the same finger to determine the False Rejection Rate. If aagainst bmatchingis performed, the symmetric one (i.e., b against a) is not evaluated to prevent correlation. [9]

The results obtained from the experiment on a robust fingerprint databaseare shown below.

Table 1. *FAR and FRR for incorrect distribution.*

A	B	C	D	E
Fingerprint Pairs	**False Acceptance Ratio**	**False Rejection Ratio (FRR)**	**Matching Score (FAR) %**	**Matching Score (FRR)**
1	0.9	0.2	0.2	20.11
2	0.9	0.21	10.11	29.99
3	0.8	0.21	18.21	40.12
4	0.3	0.25	19.12	49.98
5	0.2	0.9	20.14	60.12
6	0.15	0.93	50.43	79.98
7	0.1	0.97	60.11	90.12
8	0.09	0.98	70.31	90.13
9	0.07	0.99	80.21	91.98
10	0.07	0.99	80.21	91.98
11	0.05	0.99	90.42	92
12	0.45	0.99	90.45	92.01

Table 2. *Showing the matching score results for correct matching for both minutiae-based correlation based matching technique and the average total score.*

S/N	PAIR	A	B	C	E
1	D	30.00	36.72	30.36	NON-MATCH
2	D	22.58	34.41	28.50	NON-MATCH
3	S	100.00	99.99	99.95	MATCH
4	D	33.11	21.22	27.17	NON –MATCH
5	S	100.00	100	100	MATCH
6	S	98.99	97.53	98.26	MATCH
7	S	80.99	79.89	80.44	MATCH
8	D	28.76	31.20	29.98	NON-MATCH
9	S	80.11	83.22	81.67	MATCH
10	D	56.11	33.22	44.67	NON- MATCH
11	D	19.92	30.11	25.01	NON-MATCH
12	S	90.54	97.77	94.16	MATCH

Key: A= A minutiae matching score (%)
B = Cross correlation matching score (%)
C = Total average matching score (%)
F= Matching status
D= Different Pair
S = Same Pair

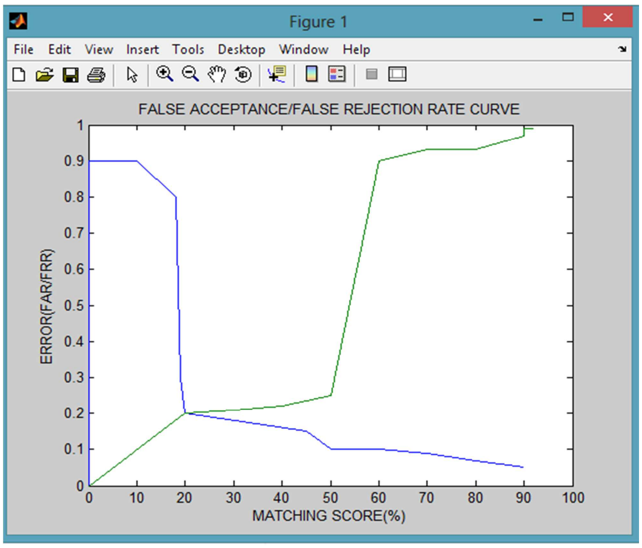

Figure 6. *Graph of Error (FAR/FRR) against Matching Score.*

4. Discussion of Results

The Table 1 above showed the matching scores for different fingerprints samples tested with the simulation program on MATLAB. The fingerprints pair that match achieved a matching score of 80% and above as expected which showed they are from the same person, while fingerprints pairs that do not match achieved a low matching score below 80% threshold. The performance of fingerprint feature extraction and matching algorithm was discovered to depend heavily on the quality of input fingerprint image. The method used offers an algorithm for image enhancement by improving the contrast of the fingerprint images before enrolment.

The above diagram in figure 8 shows the FRR and FAR curves. At the equal error rate 20%, the separating score 20% will falsely reject 20% genuine minutia pairs and falsely accept 20% spurious minutia pairs and has 80% verification rate.

5. Conclusion

This above implementation was an effort to understand how Fingerprint Verification System is used as a form of biometric to uniquely verify identities of human beings. The software simulation experimented twelve live scan fingerprint pairs for comparison to make a matching decision. The fingerprint verification system using minutiae based and cross correlation coefficient matching technique were simulated using MATLAB software and it showed an improved verification results compared to individual methods. The result obtained for all the correct matching achieved matching scores of 80% and above.The algorithm is also useful for mass checking of fingerprints and manufacturing of industrial products for biometric verification systems.

Appendix

MATLAB SOURCE CODE

```
clear
FigWin = figure('Position', [50 -50 750 600],...
'Name','Fingerprint Verification System Using combined Minutiae and Cross Correllation Based Algrorithm',...
'NumberTitle','off',...
'Color', [1 0 1]);
% 'Color', [0.827450980392157 0.815686274509804 0.776470588235294 ]);
AxesHandle1 = axes('Position', [0.2 0.15 0.35 0.7],...
'Box','on');
AxesHandle2 = axes('Position', [0.6 0.15 0.35 0.7],...
'Box','on');

BackColor = get(gcf,'Color');
%[0.827450980392157 0.815686274509804 0.776470588235294 ]

%[0.741176470588235 0.725490196078431 0.658823529411765 ]

FrameBox = uicontrol(FigWin,...
'Units','normalized', ...
'Style','frame',...
'BackgroundColor', [0.741176470588235 1 0.658823529411765 ],...
'ForegroundColor', [0.741176470588235 0.725490196078431 0.658823529411765 ],...
'Position', [0 0 0.15 1]);

%create static text.
Text2 = uicontrol(FigWin,...
'Style','text',...
'Units','normalized', ...
'Position', [0 0.95 1 0.05],...
'FontSize',15,...
'BackgroundColor', [0.741176470588235 0.725490196078431 0.658823529411765 ],...
'HorizontalAlignment','center', ...
'String','Fingerprint Verification System Using Minutiae and Cross Correlation Based Matching ');


Text2 = uicontrol(FigWin,...
'Style','text',...
'Units','normalized', ...
'Position', [0 0 1 0.05],...
'FontSize',15,...
'BackgroundColor', [0.741176470588235 0.725490196078431 0.658823529411765 ],...
'HorizontalAlignment','left', ...
'String','MSC PROJECT ');
%..................................................................
%CROSS CORRELLATION COEFFICIENT MATCHING
a = imread('01.tif');
[row1, col1] = size(a);
%imshow(a);
sprintf('The size of image1 %dx%d', row1, col1)
b = imread('01.tif');
[row2, col2] = size(b);
%imshow(b);

sprintf('The size of image2 %dx%d', row2, col2)
cc= normxcorr2(a,b);
[max_cc, imax]= max(abs(cc(:)));
sprintf('The highest correlation matrix is %f', max_cc)

if(max_cc>=0.8)

disp('The two fingerprints match');
else

disp('The two fingerprint does not match');

end

%CROSS CORRELATION COEFFICIENT MATCHING

%.....................................................................
w=16;
textLoad='Load Fingerprint Image';
h=uicontrol(FigWin,...
'Style','pushbutton',...
'Position', [0,320,115,20],...
'String','Load',...
'Callback',...
  ['image1=loadimage;'...
'subplot(AxesHandle1);'...
'imagesc(image1);'...
'title(textLoad);'...
'colormap(gray);']);

text_filterArea='Orientation Flow Estimate';
h=uicontrol(FigWin,...
'Style','pushbutton',...
'Position', [0,240,115,20],...
'String','Direction',...
'Callback',...

['subplot(AxesHandle2);[o1Bound,o1Area]=direction(image1,16);title(text_filterArea);']);

text_ROI='Region Of Interest(ROI)';
h=uicontrol(FigWin,...
```

```
    'Style','pushbutton',...
    'Position', [0,220,115,20],...
    'String','ROIArea',...
    'Callback',...

['subplot(AxesHandle2);[o2,o1Bound,o1Area]=drawROI(im
age1,o1Bound,o1Area);title(text_ROI);']);

    text_eq='Enhancement by histogram Equalization';
    h=uicontrol(FigWin,...
    'Style','pushbutton',...
    'Position', [0,300,115,20],...
    'String','his-Equalization',...
    'Callback',...

['subplot(AxesHandle2);image1=histeq(uint8(image1));imag
esc(image1);title(text_eq);']);

    text21='Adaptive Binarization after FFT';
    h=uicontrol(FigWin,...
    'Style','pushbutton',...
    'Position', [0,260,115,20],...
    'String','Binarization',...
    'Callback',...
      [%'W=inputdlg(text);W=str2num(char(W));'...
    'subplot(AxesHandle1);'...
    'image1=adaptiveThres(double(image1),32);title(text21);'])
;

    text='Please input the FFT factor(0~1)';
    text_fft='Enhancement by FFT';
    h=uicontrol(FigWin,...
    'Style','pushbutton',...
    'Position', [0,280,115,20],...
    'String','fft',...
    'Callback',...
      ['W=inputdlg(text);W=str2double(char(W));'...
    'subplot(AxesHandle1);image1=fftenhance(image1,W);im
agesc(image1);title(text_fft);']);

    text31='Thinned-ridge map';
    h=uicontrol(FigWin,...
    'Style','pushbutton',...
    'Position', [0,200,115,20],...
    'String','Thining',...
    'Callback',...

['subplot(AxesHandle2);o1=im2double(bwmorph(o2,"thin",I
nf));imagesc(o1, [0,1]);title(text31);']);

    text41='Remove H breaks';
    h=uicontrol(FigWin,...
    'Style','pushbutton',...
    'Position', [0,180,115,20],...
    'String','remove H breaks',...
    'Callback',...
['subplot(AxesHandle2);o1=im2double(bwmorph(o1,"clean")
);o1=im2double(bwmorph(o1,"hbreak"));imagesc(o1,
[0,1]);title(text41);']);

    textn1='remove spike';
    h=uicontrol(FigWin,...
    'Style','pushbutton',...
    'Position', [0,160,115,20],...
    'String','Removingspike',...
    'Callback',...

['subplot(AxesHandle2);o1=im2double(bwmorph(o1,"spur"))
;imagesc(o1, [0,1]);title(textn1);']);

    %% locate minutia and show all those minutia
    text51='Minutia';
    h=uicontrol(FigWin,...
    'Style','pushbutton',...
    'Position', [0,140,115,20],...
    'String','Extract',...
    'Callback',...

['[end_list1,branch_list1,ridgeMap1,edgeWidth]=mark_minu
tia(o1,o1Bound,o1Area,w);'...
    'subplot(AxesHandle2);show_minutia(o1,end_list1,branch
_list1);title(text51);']);

    %Process for removing spurious minutia
    text61='Remove spurious minutia';
    h=uicontrol(FigWin,...
    'Style','pushbutton',...
    'Position', [0,120,115,20],...
    'String','RealMinutiae',...
    'Callback',...

['[pathMap1,real_end1,real_branch1]=remove_spurious_Min
utia(o1,end_list1,branch_list1,o1Area,ridgeMap1,edgeWidth)
;'...
    'subplot(AxesHandle1);show_minutia(o1,real_end1,real_b
ranch1);title(text61);']);

    %save template file, including the minutia
position,direction,and ridge information
    textSaveName='file name';
    h=uicontrol(FigWin,...
    'Style','pushbutton',...
    'Position', [0,100,115,20],...
    'String','save',...
    'Callback',...
      ['W=inputdlg(textSaveName);W=char(W);'...
    'save(W,"real_end1","pathMap1","-ASCII");']);

    %invoke template file loader and do matching
    h=uicontrol('Style','pushbutton',...
    'String','Match',...
    'Position', [0,80,115,20],...
```

```
'Callback',...
  ['finger1=fingerTemplateRead;'...
'finger2=fingerTemplateRead;'...
'percent_match=match_end(finger1,finger2,10);'...
'total_match = final_match(percent_match, max_cc);']);


%Total matching score
%t = (((percent_match)/100 + max_cc)/2 )*100;
%sprintf('The total matching score is %f', t);

%text=strcat('The max matching percentage is ',num2str(t),'%');
%msgbox(text);
function [pathMap, final_end,final_branch] =remove_spurious_Minutia(in,end_list,branch_list,inArea,ridgeOrderMap,edgeWidth)

%Honors Project 2001~2002
%wuzhili 99050056
%comp sci HKBU
%last update 19/April/2002


[w,h] = size(in);

final_end = [];
final_branch=[];
direct = [];
pathMap = [];

end_list(:,3) = 0;
branch_list(:,3) = 1;

minutiaeList = [end_list;branch_list];
finalList = minutiaeList;
[numberOfMinutia,dummy] = size(minutiaeList);
suspectMinList = [];

fori= 1:numberOfMinutia-1
for j = i+1:numberOfMinutia
    d =( (minutiaeList(i,1) - minutiaeList(j,1))^2 + (minutiaeList(i,2)-minutiaeList(j,2))^2)^0.5;

if d <edgeWidth
suspectMinList =[suspectMinList;[i,j]];
end;
end;
end;

[totalSuspectMin,dummy] = size(suspectMinList);
%totalSuspectMin

for k = 1:totalSuspectMin
typesum = minutiaeList(suspectMinList(k,1),3) + minutiaeList(suspectMinList(k,2),3);

iftypesum == 1
% branch - end pair
if ridgeOrderMap(minutiaeList(suspectMinList(k,1),1),minutiaeList(suspectMinList(k,1),2) ) == ridgeOrderMap(minutiaeList(suspectMinList(k,2),1),minutiaeList(suspectMinList(k,2),2) )
finalList(suspectMinList(k,1),1:2) = [-1,-1];
finalList(suspectMinList(k,2),1:2) = [-1,-1];
end;

elseiftypesum == 2
% branch - branch pair
if ridgeOrderMap(minutiaeList(suspectMinList(k,1),1),minutiaeList(suspectMinList(k,1),2) ) == ridgeOrderMap(minutiaeList(suspectMinList(k,2),1),minutiaeList(suspectMinList(k,2),2) )
finalList(suspectMinList(k,1),1:2) = [-1,-1];
finalList(suspectMinList(k,2),1:2) = [-1,-1];
end;

elseiftypesum == 0
% end - end pair
    a = minutiaeList(suspectMinList(k,1),1:3);
    b = minutiaeList(suspectMinList(k,2),1:3);

ifridgeOrderMap(a(1),a(2)) ~= ridgeOrderMap(b(1),b(2))

        [thetaA,pathA,dd,mm] = getLocalTheta(in,a,edgeWidth);
        [thetaB,pathB,dd,mm] = getLocalTheta(in,b,edgeWidth);

%the connected line between the two points

thetaC = atan2( (pathA(1,1)-pathB(1,1)), (pathA(1,2) - pathB(1,2)) );


angleAB = abs(thetaA-thetaB);
angleAC = abs(thetaA-thetaC);


if ( (or(angleAB< pi/3, abs(angleAB -pi)<pi/3 )) & (or(angleAC< pi/3, abs(angleAC - pi) < pi/3)) )
finalList(suspectMinList(k,1),1:2) = [-1,-1];
finalList(suspectMinList(k,2),1:2) = [-1,-1];
end;

%remove short ridge later
elseifridgeOrderMap(a(1),a(2)) == ridgeOrderMap(b(1),b(2))
finalList(suspectMinList(k,1),1:2) = [-1,-1];
finalList(suspectMinList(k,2),1:2) = [-1,-1];

end;
```

```
end;
end;

for k =1:numberOfMinutia
iffinalList(k,1:2) ~= [-1,-1]
iffinalList(k,3) == 0
        [thetak,pathk,dd,mm] =
    getLocalTheta(in,finalList(k,:),edgeWidth);
if size(pathk,1) >= edgeWidth
final_end=[final_end;[finalList(k,1:2),thetak]];
                [id,dummy] = size(final_end);
pathk(:,3) = id;
pathMap = [pathMap;pathk];
end;
else

final_branch=[final_branch;finalList(k,1:2)];

        [thetak,path1,path2,path3] =
    getLocalTheta(in,finalList(k,:),edgeWidth);

if                          size(path1,1)>=edgeWidth&
size(path2,1)>=edgeWidth& size(path3,1)>=edgeWidth

final_end=[final_end;[path1(1,1:2),thetak(1)]];
        [id,dummy] = size(final_end);
path1(:,3) = id;
pathMap = [pathMap;path1];

final_end=[final_end;[path2(1,1:2),thetak(2)]];
path2(:,3) = id+1;
pathMap = [pathMap;path2];

final_end=[final_end;[path3(1,1:2),thetak(3)]];
path3(:,3) = id+2;
pathMap = [pathMap;path3];

end;

end;
end;
end;

%final_end
%pathMap
%edgeWidth

%script file for batched match
%Honors Project 2001~2002
%wuzhili 99050056
%comp sci HKBU
%last update 19/April/2002

percent_match = [];
fname=[];

fori=101:110
for j=1:3
tname = sprintf('d:\\419\\image\\%d_%d.tif',i,j);
fname = [fname;tname];
end;
end;

for i=1:3:12
for j=i+3:3:12

    t=cputime;
    fname1 = fname(i,:);
    fname2 = fname(j,:);
    template1=load(char(fname1));
    template2=load(char(fname2));
num = match_end(template1,template2,10,0);
deltaT=cputime-t
i
j
tmp = [i,j,deltaT,num];
percent_match = [percent_match;tmp];
end;
end;

fname = sprintf('d:\\419\\image\\interclassTest.dat');

save(fname,'percent_match','-ASCII');

%percent_match
```

References

[1] Langenburg Gkenn, "Are one's fingerprints similar to those of his or her parents in any discernable way?" Scientist American, Springer Nature, January 24, 2005. Available: "http://www.scientificamerican.com/article.cfm?id=are-ones-fingerprints-sim". [Retrieved: 30 November 2016].

[2] Lawrence O' Gorman, "Fingerprint Verification", Veridicom Incorporation, Springer International Publishing, October 2006 Available:"https://www.google.com/url?sa=t&rct=j&q=&esrc=s&source=web&cd=2&cad=rja&uact=8&ved=0ahUKEwiene7fr9DQAhVkL8AKHfmQDVcQFggiMAE&url=http%3A%2F%2Flink.springer.com%2Fchapter%2F10.1007%252F0-306-47044-6_2&usg=AFQjCNFKayQNVhAilWHio4OBW_O_fgthng&sig2=gaKqdd-BLPhdjjGv7-Vqng". [Accessed: 29th November 2016].

[3] Sangram Bana, et al, "Fingerprint Recognition using Image Segmentation", Research Gate, (IJAEST) International Journal of Advanced Engineering Sciences and Technologies, Volume No 5, Issue No 1, 012-023. [October 13 2016]. Available:https://www.researchgate.net/publication/228422763_Fingerprint_Recognition_using_Image_Segmentation. [Accessed: November 30, 2016].

[4] Justice MuhammedLawalGarba, et al, "Full Text of the Judgment of Aregbeshola versus oyinlola", Nigerian muse, Court of Appeal, Ibadan Judiciary Division, [26th November 2010], Avaliable: "www.nigerianmuse.com/20101201041725zg/sections/general-articles/full-text-of-the-judgment-of-appeal-of-aregbesola-vs-oyinlola-delivered-nov-26-2010-by-justice-ogunbiyi-et-al/". [Accessed: 28th November, 2016].

[5] Kenneth R. Moses, et al, “Automated Fingerprint Identification System”, United States office of the Justice Programs, National Criminal Justice Reference Service, [October 2014]. Available: “https://www.ncjrs.gov/pdffiles1/nij/225326.pdf”. [Accessed: 30th November 2016]

[6] Setlak, Dale, “Advances in Biometric Fingerprint Technology and Driving Rapid Adoption in consumer marketplace”, Wikipedia, AuthenTec, [Retrieved: 4 November, 2010]. Available:”https://docs.google.com/viewer?url=www.authentec.com%2Fdocs%2Fwhite%2520paper%2520for%2520ziffdavis.doc”.[Accessed: 28th November, 2016]

[7] J. L Wayman et al, “Technical Testing and Evaluation of Biometric Devices”, 360 biometrics, [September 2011], Available:” http://www.360biometrics.com/fingerprint-scanners/hamsterplus.php”.[Accessed: 30th October, 2016]

[8] M. J Stephen, “Removal of false minutia with modified fuzzy rules”, Research Gate, Welfare Institute of Science Technology and Management, [July 2013], Available: “https://www.researchgate.net/publication/256470785_Removal_of_False_Minutiae_with_Modified_Fuzzy_Rules”. [Accessed: 30th November, 2016]

[9] Dario Maio et al, ” Performance Evaluation of Fingerprint Verification System”, Research Gate, Department of Computer Science and Engineering, University of Bologna, [February, 2006], Available: https://www.researchgate.net/publication/7368583_Performance_evaluation_of_fingerprint_verification_systems. [Accessed: 29th November, 2016]

[10] Gabriel Iwasokun et al, “An Investigation into Impact of False Minutia Points on Fingerprint Matching”, Research Gate, Federal University of Akure, Department of Computer Science, [June 2014], Available: https://www.researchgate.net/publication/284631114_An_Investigation_into_the_Impact_of_False_Minutiae_Points_on_Fingerprint_Matching. [Accessed: 30th November, 2016]

[11] 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.