Effectiveness of a Novel Feature and Confidence Level Assignments to Classifiers in Fingerprint Matching

KHURRAM YASIN QURESHI, SHOAB A. KHAN AND JAVED M.Y
Department of Computer engineering
National University of Sciences and Technology
Rawalpindi
PAKISTAN
Khurram_ya@hotmail.com, kshoab@yahoo.com, younus@yahoo.com

Abstract: This is 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. Experimental results show that algorithm is efficient and reliable.

Key-Words: Fingerprint matching, minutiae based approach, fuzzy logic, structural matching, distance from center to ridge intersection and relative angle.

1 Introduction
It is always important to correctly identify an individual. There are many systems to do that but the more reliable and accurate are biometric systems because they do not change with time and nearly remain the same at all stages in life. Some of the biometrics includes face, iris, palm, foot and fingerprints but the most popular among them is recognizing a fingerprint. Automatic fingerprint matching techniques are usually divided in to image based, minutiae based and ridge feature based approaches [2, 3] but we have used minutiae based matching algorithm because of its effectiveness. Several related techniques are also presented to provide background knowledge about this approach to fingerprint matching. In algorithm [4], 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 [1,5] enhance the algorithm proposed by [4] 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 [6], 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. The proposed algorithm comprises two stages, first involves initial matching that is able to filter some of the minutiae that are not able to fulfill initial criteria. After first stage one minutia might match with more than but the second stage would correct all the false matching done by first stage. The technique presented here is a modified form of [1] with an introduction of fuzzy logic and a novel feature in the second stage. The name of this novel feature is 'distance from center to ridge intersection’ and will be explained in detail in section 4. The selection criterion for center point is also adopted in the second stage. This paper is organized as follows: Section 2 includes preprocessing a fingerprint image.
so that it can further be used for matching. Section 3 describes feature extraction process. Section 4 includes the algorithm that is used for matching. Section 5 presents experimental results and the conclusion is drawn in section 6.

2 Preprocessing

Preprocessing a fingerprint is basic for its matching. There are different methods used for it. We have used the method proposed by [7] 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 Fig. 1.

![Figure 1: Different stages of an image during preprocessing](image)

3 Minutiae Extraction

There are many types of minutiae that can be used for comparison but the more prominent among them are endings and bifurcations as they are also supported by FBI. We use a 3x3 window for their detection which is presented below in Fig. 2.

<p>| | | |</p>
<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>P4</td>
<td>P5</td>
<td>P6</td>
</tr>
<tr>
<td>P3</td>
<td>P9</td>
<td>P7</td>
</tr>
<tr>
<td>P2</td>
<td>P1</td>
<td>P8</td>
</tr>
</tbody>
</table>

![Figure 2: window for minutiae detection](image)

The above window is moved across whole image to detect bifurcations and ending. The conditions which are used for their detection are given below.

To confirm that P9 is bifurcation or ending we have used eq-1. If CN/2 is ‘1’ then P9 indicates ending and if it is ‘3’ then this would be an indication of bifurcation. For last iteration (i==8), we consider that P9=P1. ‘Type’ and ‘location’ of each minutia is recorded during minutiae extraction that would further be used in matching process.

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

4 Matching

Matching fingerprints is an important part of this paper. There are a lot of matchers used for minutiae based matching. The job of the matcher is to take input image and compares it with template or stored image. Some matchers are concerned with reference points (core and delta) detection as they are crafted on this theme but bad image quality degrades their performance because in that case reference points may be absent. Our approach is based on structural matching and uses statistical properties of minutiae for matching. This is rotation invariant, reliable and time efficient approach. We have compared the minutia also called a center point based upon its neighbors. There are two stages adopted in this matching process that are explained one by one. The task of first stage is to mark the points that satisfy initial selection criteria. These marked points are included in initial (unconfirmed) list. Majority of the points that are selected in this stage would be a member of final (confirmed) list if both images belong to the same individual. The points that are obtained after second stage will make final (confirmed) list. The structure that is used in the first stage of matching is given in Fig. 3. Some modifications have been made to prepare a structure for second stage matching that will be provided later.

![Figure 3: Local structure used for first stage matching](image)
4.1 Forming initial (unconfirmed list) using first stage matching

An authentication of minutia is assured by using characteristics of five surrounding neighbors. There are two types of feature vectors that are used in first stage matching (FVM1 and FVM2). We have to find the elements mentioned in Fig. 4(a) and Fig. 4(b) to complete our feature vectors ‘FVM1’ and ‘FVM2’ that can later be used for matching. All the neighbors must have their own ‘FVM2’ before matching. The required feature vector ‘FVM2’ consists of type, distance to the central point, relative angle (calculated by using coordinates of neighbor and central point as well as its direction) and ridge count. ‘FVM1’ includes only the type of central minutia. Feature vectors for matching, ‘FVM1’ and ‘FVM2’ are given in Fig. 4 as under.

<table>
<thead>
<tr>
<th>Type of neighbor</th>
<th>Distance to central point</th>
<th>Relative angle</th>
<th>Ridge count</th>
</tr>
</thead>
<tbody>
<tr>
<td>(a)</td>
<td>(b)</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Figure 4: (a) element(s) in FVM1 (b) element(s) in FVM2

All the five neighbors must have their own information mentioned in Fig. 4(b) for matching. Type of central point must also be known. Type of the neighbor and central point can easily be determined by using eq-1 and distance to the central point requires the calculation of Euclidian distance which is not a difficult task because we have already obtained the locations of all the minutiae during minutiae detection phase.

Direction calculation is very important for finding relative angle that is a dominant part of feature vector for matching (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). For the calculation of direction, we have to traverse each ridge some pixels away from the location of detected minutia. After traversing ten 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 formulae that are used for calculating slope (direction) and orientation are given below.

\[
direction = \frac{X' - Xm}{Y' - Ym}
\]  

\[
orientatio n = \arctan(direction)
\]  

The calculation of direction is not very simple and is explained in following steps:

1) First of all, a 21x21 window is mapped on each minutia by keeping it at center.
2) In the case of ending, the concerned ridge is traversed 10 pixels away from the location of detected minutia. After reaching at this point the direction of ending is calculated. 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 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.
3) 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 10th pixel on each leg, all the three locations are stored and labels are assigned to them as 1, 2 and 3. Euclidian distance is calculated between these three locations in this way (Ed (1, 2), Ed (2, 3), Ed (3, 1)). The smallest value is stored and the labels are checked that are involved in calculation of this smallest value. The label which is other than these two labels stores the required location. The graphical representation of this process is given in Fig. 5.

The distance between label ‘1’ and label ‘2’ is smallest so the label other than these two is ‘3’ so...
we will use the location at label 3 ridge to calculate
direction of bifurcation.

4) Now the task is to map the selected location in a
21x21 window 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.

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 equation of a straight line that is given
below.

\[
y = \frac{y_1(x_2 - x) + y_2(x - x_1)}{x_2 - x_1}
\]
\[
x = \frac{x_1(y_2 - y) + x_2(y - y_1)}{y_2 - y_1}
\]

(4)

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’. After
calculating the elements of FVM1 and FVM2, the
first stage matching is ready to be performed. First
of all the type of central minutia in input and
template image is compared. If FVM1 (type of
central minutia) of both the images (input and
template) are same then FVM2 for each of the five
neighbors of central minutia is compared between
two images. It means that there would be five feature
vectors of type FVM2 as there are five neighbors of
one central minutia. There is no need to compare
FVM2 if FVM1 is not satisfied as it is a preliminary
condition for comparing five feature vectors of type
FVM2. Each neighbor has its own FVM2. We
assign a degree of similarity to central minutia on
the bases of the number of neighbors that are matched between input and template images. Table 1
explains the problem effectively.

<table>
<thead>
<tr>
<th>Neighbor number</th>
<th>Vector type</th>
<th>Degree</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>FVM2(input)=FVM2(temp) for 1 only</td>
<td>1</td>
</tr>
<tr>
<td>2</td>
<td>FVM2(input)=FVM2(temp) for 1,2</td>
<td>2</td>
</tr>
<tr>
<td>3</td>
<td>FVM2(input)=FVM2(temp) for 1,2,3</td>
<td>3</td>
</tr>
<tr>
<td>4</td>
<td>FVM2(input)=FVM2(temp) for 1,2,3,4</td>
<td>4</td>
</tr>
<tr>
<td>5</td>
<td>FVM2(input)=FVM2(temp) for 1,2,3,4,5</td>
<td>5</td>
</tr>
</tbody>
</table>

This table is compared only when ‘FVM1’ from
both images are matched. The above table shows
that degree of similarity depends upon the number of
matched neighbors. The minutia has been selected
for second stage matching only if the degree of
similarity is at least one. 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.

4.2 Forming final (confirmed list) using
second stage matching

This stage forms a final list of minutiae that are used
for score calculation. This score tells us about the
percentage of similarity between two images. This
stage is a modified form of second stage in [1]. A
novel feature called “distance from center point to
ridge intersection” is introduced with its graphical
representation in Fig. 6. Fig. 6(a) describes the
second stage features without explaining a novel
feature because it is represented in Fig. 6(b)
effectively.

![Figure 6](image_url)

Figure 6: (a) Structure used for second stage matching
(b) Graphical representation of a novel feature

Fuzzy logic is incorporated with an assignment of a
confidence level to each element of feature vector.
type (FVM3) as described by Fig. 7. In this stage we have to select the center point on the basis of its degree in the first stage. 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. A specific confidence level is attached with each element of FVM3. 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. Fig. 7 explains all the elements in this vector type and the combination of these form final matching criteria. There are multiple ways for representing the performance of an algorithm but we have used equal error rates and ROC curves to depict it. Database (DB1_A) of FVC 2002 [10] was used to evaluate its performance. 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. FMR (false match rate) and FNMR (false Non-match rate) were plotted at different thresholds to calculate EER (equal error rate); it is a point where FMR and FNMR are equal. In [11], the method for calculating these two error rates is available. Fig. 8 shows ROC (receiver operating characteristic) curves of [1] and proposed algorithm. It is observed that performance of our approach is better than [1] with an improvement of 40%. EER obtained by using our technique is 6.85% but if unimproved approach is used it would be 11.42%.

If sum of confidence levels is greater than 70 then decision will be in the favor of minutia and it will be accepted. Where the value in parenthesis represents a confidence level which is attached to each element and the last column tells us about the decision whether the minutia is accepted or rejected. ‘X’ specifies that the value is do not care which means that this value has no credit in final decision.

### 5 Experimental Results

There are multiple ways for representing the performance of an algorithm but we have used equal error rates and ROC curves to depict it. Database (DB1_A) of FVC 2002 [10] was used to evaluate its performance. 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. FMR (false match rate) and FNMR (false Non-match rate) were plotted at different thresholds to calculate EER (equal error rate); it is a point where FMR and FNMR are equal. In [11], the method for calculating these two error rates is available. Fig. 8 shows ROC (receiver operating characteristic) curves of [1] and proposed algorithm. It is observed that performance of our approach is better than [1] with an improvement of 40%. EER obtained by using our technique is 6.85% but if unimproved approach is used it would be 11.42%.

<table>
<thead>
<tr>
<th>Distance to central point</th>
<th>Relative angle</th>
<th>Difference between orientations</th>
<th>Distances from center to ridge intersection</th>
<th>Decision</th>
</tr>
</thead>
<tbody>
<tr>
<td>Y (40)</td>
<td>N (0)</td>
<td>N (0)</td>
<td>N (0)</td>
<td>Y</td>
</tr>
<tr>
<td>Y (40)</td>
<td>N (0)</td>
<td>N (0)</td>
<td>Y (10)</td>
<td>Y</td>
</tr>
<tr>
<td>Y (40)</td>
<td>N (0)</td>
<td>Y (20)</td>
<td>N (0)</td>
<td>N</td>
</tr>
<tr>
<td>Y (40)</td>
<td>N (0)</td>
<td>Y (20)</td>
<td>N (0)</td>
<td>N</td>
</tr>
<tr>
<td>Y (40)</td>
<td>N (0)</td>
<td>Y (20)</td>
<td>Y (10)</td>
<td>Y</td>
</tr>
<tr>
<td>Y (40)</td>
<td>Y (30)</td>
<td>N (0)</td>
<td>N (0)</td>
<td>N</td>
</tr>
<tr>
<td>Y (40)</td>
<td>Y (30)</td>
<td>Y (20)</td>
<td>X</td>
<td>Y</td>
</tr>
<tr>
<td>Y (40)</td>
<td>Y (30)</td>
<td>Y (20)</td>
<td>Y</td>
<td>Y</td>
</tr>
<tr>
<td>N (0)</td>
<td>X</td>
<td>X</td>
<td>X</td>
<td>N</td>
</tr>
</tbody>
</table>

Table 2. Decision about acceptance or rejection of minutia based on fuzzy logic
The comparison of proposed approach with other existing approaches is described in tabular form in Table 3.

**TABLE 3. COMPARISON BETWEEN DIFFERENT FINGERPRINT MATCHING ALGORITHMS**

<table>
<thead>
<tr>
<th>Algorithm</th>
<th>Equal Error Rate (%)</th>
<th>Accuracy (%)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Proposed</td>
<td>6.851</td>
<td>93.15</td>
</tr>
<tr>
<td>Wahab et. al [1]</td>
<td>11.42</td>
<td>88.58</td>
</tr>
<tr>
<td>BioGina</td>
<td>7.95</td>
<td>92.05</td>
</tr>
<tr>
<td>Digital Finger pass Co.</td>
<td>8.27</td>
<td>91.73</td>
</tr>
<tr>
<td>Sagem Inc.</td>
<td>35</td>
<td>65</td>
</tr>
<tr>
<td>PA-16 [10]</td>
<td>16.28</td>
<td>83.72</td>
</tr>
</tbody>
</table>

5 Conclusion

Minutiae based algorithm has been proposed in this paper that incorporates two stages to perform accurate matching between input and template images. A novel feature called “distance from center to ridge intersection” is incorporated in this technique. By using the concept of fuzzy logic, each element of feature vector in the second stage is assigned a confidence level on the basis of experience. The performance of the algorithm has been evaluated by considering the database FVC 2002 (DB1_A). Experimental results show that algorithm is reliable and accurate. Results can further be improved if parameters are further optimized with an incorporation of more classifiers which would help in producing enhanced results.

**References:**


