Proceedings of the 2009 IEEE International Conference on Systems, Man, and Cybernetics
San Antonio, TX, USA - October 2009

Effectiveness of Assigning Confidence Levels to Classifiers and a Novel Feature in Fingerprint Matching

Khurram Yasin Qureshi
Department of Computer Engineering, College of Electrical and Mechanical Engineering
National University of Sciences and Technology
Rawalpindi, Pakistan
khurram_ya@hotmail.com

Shoab A. Khan
Department of Computer Engineering, College of Electrical and Mechanical Engineering
National University of Sciences and Technology
Rawalpindi, Pakistan
kshoab@yahoo.com

***Abstract*—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 adapted. 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.**

***Keywords*—Fingerprint matching, Fuzzy logic, Structural matching, Distance from center to ridge intersection.**

I. 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. Biometric systems include face, iris, palm, foot, fingerprints and some others 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 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, bridges and short ridges. 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 difficult 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 trees of two images.

The proposed algorithm comprises two stages. First stage performs initial matching that is able to filter some of the minutiae that are unable to fulfill required 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 IV. The criterion for selecting center point in the second stage is also adapted. This paper is organized as follows: Section II includes preprocessing a fingerprint image so that it can further be used for matching. Section III describes minutiae extraction process. Section IV includes the details of matching algorithm. Section V presents experimental results and the conclusion is drawn in section VI.

II. Preprocessing

Preprocessing a fingerprint is basic for its matching. There are different methods used for it but 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.

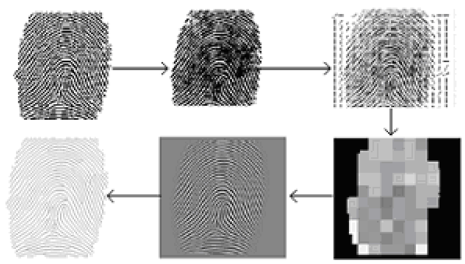


Figure 1. Different stages of an image during preprocessing

III. Minutiae Extraction

There are many types of minutiae that can be used for matching 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.

P4	P5	P6
P3	P	P7
P2	P1	P8

Figure 2. Window for minutiae detection

The above window is moved across whole image to detect bifurcations and endings. The conditions which are used for their detection are specified in this section effectively. To detect the type (ending or bifurcation) of ‘P’, the formula specified in (1) is used. If CN/2 is ‘1’ then ‘P’ 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 the matching process.

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

IV. 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 not be detected correctly. 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 on its neighbors. Every detected minutia will act as a center point on its term. 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 first stage of matching is given in Fig. 3. In this figure, ‘cp’ is a center point whereas N1, N2, N3, N4 and N5 are the five closest neighbors of centre point. Some modifications have been made to prepare a structure for second stage matching that will be provided later.

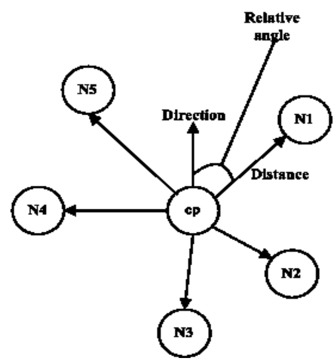


Figure 3. Local structure used for first stage matching

A. *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’ type vector before matching. The required feature vector ‘FVM2’ consists of type, distance to the central point, relative angle (calculated by using coordinates of central point, coordinates of neighbor and direction of central point) 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.

Type

(a)

Type of neighbor
Distance to central point
Relative angle
Ridge count

(b)

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 (1) whereas 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 is 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 the new location are X', Y'. The formulae that are used for calculating slope (direction) and orientation are given in (2) and (3).

$$direction = \frac{Y' - Ym}{X' - Xm} \quad (2)$$

$$orientation = \arctan(direction) \quad (3)$$

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 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.

3) For bifurcation, the procedure that is adopted in above situation is repeated three times. It has to be done because three legs are 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 label 1, label 2 and label 3. Euclidian distance is calculated between these three locations in this way (Ed (label 1, label 2), Ed (label 2, label 3), Ed (label 3, label 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.

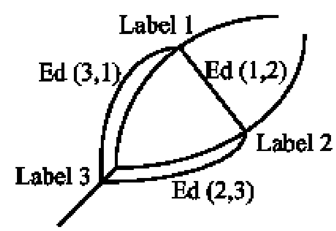


Figure 5. Selecting appropriate location for bifurcation's direction

The distance between locations associated with label 1 and label 2 is smallest so the location associated with label 3 will be used to calculate the 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 these coordinates and they tell 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. Ridges are counted by counting the pixels that have value '0' along a straight line because 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' type vector. We assign a degree of similarity to central minutia on the basis of the number of neighbors that are matched between input and template images. Table I explains the concept effectively.

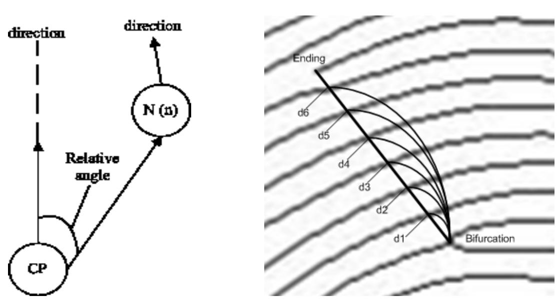
TABLE I. ASSIGNMENT OF DEGREE TO THE MINUTIA ON THE BASIS OF NUMBER OF MATCHED FEATURE VECTORS THAT ARE BELONGED TO DIFFERENT NEIGHBORS

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) for 1,2,3	3
4	FVM2(input)=FVM2(temp) for 1,2,3,4	4
5	FVM2(input)=FVM2(temp) for 1,2,3,4,5	5

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. If the degree of similarity is greater than or equal to '1' then the minutia will be selected for second stage matching. 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.

B. *Forming Final (confirmed) List Using Second Stage Matching*

This stage forms a final list of minutiae that is 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 presented in Fig. 6(b) effectively.



(a) (b)

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 the 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.

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

Figure 7. Elements of FVM3

First two elements in this vector type are calculated in the same way as in first stage. Finding orientation is already discussed in (3). Difference between orientations of central point and the concerned neighbor will be compared between input and template images [8]. We can understand the building of last element in feature vector (FVM3) from Fig. 6(b). We may find multiple distances when we go from center point to corresponding neighbor. The number of distances that we find would depend upon the number of ridges in the straight path from center point to concerned neighbor. Careful selection of threshold is important to make this feature scale invariant. All of these distances from input image are stored and compared with the distances from template image.

Because of the stares effect in the ridges, detection of some ridges may miss. To eliminate this problem, the 8-connected neighbors of every pixel on the straight line are checked. 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. The minutia which fulfills the criteria mentioned in Table II will make its contribution in final score calculation. All these features of a minutia from two images must match for its consideration in final score. The matching score is calculated by using the formula given in (4).

$$score = \frac{MP * MP}{NI * NT} \tag{4}$$

Where 'MP' is the number of points matched between two images. 'NI' and 'NT' are the total number of points (minutiae) detected in input and template fingerprints. The formula given in (4) is similar to one used in [9,10].

Fuzzy logic is incorporated in this algorithm and a confidence level is attached with each of the elements in 'FVM3'. Assignment of a confidence level to each element is done empirically as it is attached after executing the algorithm multiple times. Table II describes that how our algorithm decides about acceptance or rejection of each minutia.

TABLE II. DECISION ABOUT ACCEPTANCE OR REJECTION OF A MINUTIA BASED ON FUZZY LOGIC

Distance to central point	Relative angle	Difference 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 sum of confidence levels is greater than '70' then decision will be in the favor of minutia and it will be accepted. 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.

V. EXPERIMENTAL RESULTS

There are multiple ways of representing the performance of an algorithm but we have used equal error rate and ROC curve to depict it. Database (DB1_A) of FVC 2002 [11] 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. FAR (false accept rate) and FRR (false reject rate) were plotted at different thresholds to calculate EER (equal error rate); it is a point where FAR and FRR are equal.

In [12], the method for calculating these two error rates is available. Fig. 8 shows ROC (receiver operating characteristic) curves of [1] and proposed algorithm in which FRR is plotted as a function of FAR. The graph is plotted in log-log scale for better comprehension. 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% which was 11.42 % in case of unimproved (previous) approach.

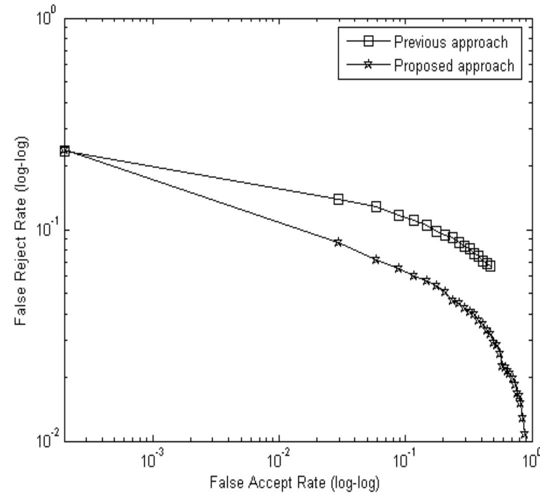


Figure 8. ROC curves of the previous and proposed algorithm

The comparison of proposed approach with other existing approaches is described in tabular form in Table III.

TABLE III. COMPARISON BETWEEN DIFFERENT FINGERPRINT MATCHING ALGORITHMS

Algorithm	Equal Error Rate (%)	Accuracy (%)
Proposed	6.851	93.15
Wahab et. al [1]	11.42	88.58
BioGina	7.95	92.05
Digital Finger pass Co.	8.27	91.73
PA-16 [11]	16.28	83.72
PA-22 [11]	17.34	82.66
PA-25 [11]	35	65
PA-03 [11]	50	50

VI. CONCLUSION

Minutiae based algorithm has been proposed in this approach 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 (FVM3) in the second stage is assigned a confidence level empirically. 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

[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, 1998, pp. 160–166.

[2] D. Maltoni, D. Maio, A.K. Jain, and S. Prabhakar, Handbook of Fingerprint Recognition, Springer, New York, 2003.

[3] Weiping Chen and Yongsheng Gao, "A Minutiae-Based Fingerprint Matching Algorithm Using Phase Correlation," 9th Biennial Conference of the Australian Pattern Recognition Society on Digital Image Computing Techniques and Applications (DICTA), 2007, pp. 233–238.

[4] Hrechak A. and McHugh J., "Automated Fingerprint Recognition Using Structural Matching," Pattern Recognition, vol. 23, no. 8, 1990, pp. 893–904.

[5] Chen Z. and Kuo C.H.,"A Topology-Based Matching Algorithm for Fingerprint Authentication," 25th Int. Carnahan Conf. on Security Technology, 1991, pp. 84–87.

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

[7] 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, 1998, pp. 777–789.

[8] Shenglin Yang and Ingrid M. Verbauwhede, "A Secure Fingerprint Matching Technique," 2003 ACM SIGMM workshop on Biometrics methods and applications (WBMA '03), Berkeley, California, USA, November 8, 2003, pp. 89–94.

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

[10] JuCheng Yang et al., "Fingerprint Matching using Global Minutiae and Invariant Moments," 2008 Congress on Image and Signal Processing, vol. 4, 2008, pp. 599–602.

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

[12] http://Bias.Csr.Unibo.It/Fvc2000/Downloads/fvc2000_report.pdf, the Fingerprint Verification Competition Organization.