Efficient and Rotation Invariant Fingerprint Matching Algorithm Using Adjustment Factor

Asif Iqbal Khan and M. ArifWani

Abstract— **This paper presents a new efficient and rotation invariant algorithm that makes use of local features forfingerprint matching. Minutiae points are first extracted from afingerprint image. Minutiae code mc, defined in this paper, is then generated for each extracted minutiae point. The proposed minutiae code is invariant to rotation of the fingerprint image.Adjustment factor (AF) is introduced to address the problem due to differences in a claimant fingerprint and a template fingerprint of the same person that may be present due to variations in inking or variations in pressure applied between a finger and the scanner. Adjustment factor is calculated from the minutiae code (mc) of the two fingerprints being matched. A two stage fingerprint matching process is proposed. During first stage only a few minutiae codes are checked to decide if the second stage of matching process is required. This makes the matching process faster. The proposed strategy is tested on a number of publicly available images (DB1 of FVC2004 database) and the results are promising.**

Keywords---- Minutiae, Bifurcation, Ridge, AFIS,Fingerprint, Minutia Code

I. INTRODUCTION

It has been more than a century now since Home Ministry Office, UK, accepted that no two individuals have the same fingerprints, and since then fingerprint recognition has become the most popular, highly reliable method for human verification and is currently being successfully used in many applications. Although fingerprint recognition is an old topic and many research has been done in *Automated Fingerprint Identification System* (*AFIS*) till date and there are many fingerprint recognition algorithms available, yet it is currently one of the most researched and matured field of biometric authentication [2]. A large number of automatic fingerprint recognition algorithms have been proposed so far and most of them have no problem in matching good quality fingerprint images, but, fingerprint matching still remains a challenge due to the difficulty in matching low-quality fingerprint images. The difficulty in matching fingerprints is mainly due to the large variability in displacement, rotation, non-linear distortion, pressure, skin condition, etc and as well as due tofeature extraction errors, due to which most of the fingerprint matching algorithms produce inaccurate results. Therefore, sometimes fingerprints from different fingers may look quite similar whereas fingerprints from the same finger may look quite different [2] and it has been reported that, 80% of the match errors in fingerprint matching is due to bad quality images in the databases [1].

Fingerprint matching is one of the essential and most important stages *in Automatic Fingerprint Identification Systems(AFIS).* Fingerprint matching algorithm should be robust enough to produce accurate results in presence of false and spurious fingerprint details (minutiae). This paper focuses on the various minutiae based fingerprint matching algorithms, their results, advantages and disadvantages.

The rest of the paper is organized as follows. Section II gives an overview of the related work done in the field of the fingerprint matching. Two selected algorithms for which results on DB1 of FVC2004 database are available are described in section III. The proposed novel fingerprint matching algorithm is discussed in Section IV. Section V gives the results of the tests conducted on DB1 ofFVC2004 database and compares the results with the two selected algorithms. Finally Section VI presents the conclusion.

II. LITERATURE REVIEW

There has been a lot of work done in the field of fingerprint recognition. Automatic fingerprint recognition has grown rapidly in the last decade and is currently being used in many applications. Although a lot of research has been done in the field of fingerprint recognition and number of approaches has been proposed over the years, still fingerprint recognition is not considered fully solved problem.

Automatic Fingerprint recognition can be divided into two main stages *Minutiae Extraction* and *Minutiae matching.* In former stage minutiae points from a fingerprint image are extracted and stored. In minutiae matching, these minutiae points are matched with other minutiae points of different fingerprint to find a match.

Typically, a minutiae point extracted from a fingerprint image is represented by a vector m = $\{x,y,\theta\}$ where $x,y$are theminutia coordinates and θ is the minutia angle, and the two minutiae points from two fingerprints are considered matching if the distance between them is smaller than a given threshold[2]. Thresholds (tolerance boxes) are used to compensate for the unavoidable errors introduced during feature extraction and to account for the small plastic distortions that cause the minutiae positions to change. This is a general approach for fingerprint matching.This type of approach is not rotation

978-1-5090-0287-0/15 $31.00 © 2015 IEEE
DOI 10.1109/ICMLA.2015.226

invariant because adjustment of any variation in rotation is done up to a limited tolerance.

Another approach was proposed by [3] which performs an inexact partial point pattern matching with $O(m2\times n2 \times log\ m)$ time complexity.The problem with this approach is that it assumes minutiae points in a template image are not close to each other and another assumption is that all minutiae points in a claimant fingerprint have a close mate in the template fingerprint, and these assumptions are not always fulfilled.

Technique proposed by *Udupa, Garg, and Sharma [4]* uses point-point approach in which transformation is applied to point p to map it with point q, if a certain condition is met and an alignment parameter is derived which is then superimposed and pairing between the remaining minutiae is determined.

Xudong Jiang and Wei-Yun Yau also proposed an algorithm based on the distance and angle between the neighbouring minutiae points [5].Another similar approach proposed in [17] uses two neighbours to form minutiae triangle structure and then obtains the similarity between triangles.The problem with this method is that it does not make relationship between local and global steps explicit.

Fingerprint matching based on Frobenius norm proposed in [6] uses Frobenius norm as a feature to describe the fingerprint.

Weiwei Zhang, Yangsheng Wang proposed a Core-Based Structure Matching Algorithm which uses a fast multi resolution based core point detection algorithm for the detection. [9] Another similar approach was proposed by SharatChikkerur and NaliniRatha in which core and delta points of a fingerprint are extracted for matching and thus improves the computational time but the problem with these core based approaches is that the extraction of singular points is usually error proneand core based algorithms fail when a fingerprint without core point or singular point is encountered [10].

Many researchers have used minutia triangles for local structure matching for fingerprint recognition [15]. Germain, Califano, and Colville [16] proposed a technique based on minutia triangles. They used the length of the sides, angles with respect to x-axis and ridge count to represent a fingerprint. George Bebis et al [18] adopted Delaunay Triangulation to the minutiae set and constructed fingerprint features. Huimin Deng [19] adopted Delaunay Triangulation to construct steady triangles to build the local structure for high reliability. However, the number of possible minutiae triangles equals the combination of three minutiae from a set of n minutiae that is potentially very large.

Technique proposed by [11] obtains global orientation field to represent the structure of a minutia. During matching two minutia lists and two orientation fieldscaptured from two fingerprint impressions are used to generate a matching score that expresses the degreeof similarity.

The author in [21] has proposed an approach for feature extraction that is based on brightness control of the pixels of a fingerprint image. A fuzzy extraction method pre-processes the fingerprint image and adjusts the brightness. The resulting image is then used to extract features for matching. The problem with this approach is that it fails when the partial print does not include structures such as core and delta.

Feng et al [22] proposed a method which is inspired from spelling correction techniques. It uses prior knowledge of fingerprints and stores dictionary of reference orientation patches. A good quality fingerprint is manually selected and its orientation fields are used to construct the dictionary of patches. This dictionary of orientation patches is used to predict the orientation of a latent fingerprint image. The main disadvantage of this method is that it requires prior knowledge of orientations and requires more storage to store the patches

.

Most of the algorithms mentioned above perform well on good quality fingerprint images but may not produce desirable results on poor quality fingerprint images. Minutiae based Fingerprint matching algorithms either work on global features in which minutiae matching is treated as point pattern matching problem determining the global alignment leading to an optimal spatial (and directional) minutiae pairing, or local features which are characterized by attributes that are invariant with respect to global transformations (rotation, translation etc.) and are suitable for matching without any global alignment. There is a need of a fast and accurate method which can withstand the challenges like rotation, partial fingerprints, missing minutia points etc. This work proposes a new novel rotation invariant fingerprint matching approach which addresses the limitations of the existing algorithms.

III. DESCRIPTION OF SELECTED FINGERPRINT MATCHING ALGORITHM

In this section, two fingerprint matching algorithms "*Local And Global Structures Based Algorithm"* and "Minutia *and Local Compatibility Based Algorithm"* are discussed in detail. The reason for selecting these two algorithms is the availability of results on fvc2004 database for comparison with our proposed method. We summarize the general matching algorithm below and then describe the two selected algorithms.

A. General Fingerprint Matching Algorithm

Typically, a minutiae point extracted from a fingerprint image is represented by a vector m = $\{x,y,\theta\}$ where x,$y$are theminutia coordinates and θ is the minutia angle (i.e. angle between ridge and x-axis). Two minutiae points from two fingerprints are considered to be matching if the difference in representation of minutiae points is smaller than a given threshold [2]. Thresholds (tolerance boxes) are used to compensate for the unavoidable errors introduced during feature extraction and to account for the small plastic distortions that cause the minutiae positions to change. A simple and common approach for minutiae matching algorithms is based on spatial distance in which each minutia is represented as a triplet m = $\{x,y,\theta\}$. After extracting the minutia points from both template and claimant fingerprints, matching is carried out as explained below.

Let *m*and *n*be the number of minutiae points extracted from the template and claimant fingerprint images T and Irespectively and let $m_1, m_2 \ldots\ldots m_m$ and $m`_1$, $m`_2 \ldots m`_n$ be the minutia points extracted from the template T and claimant fingerprints I respectively, given as

$$T = \{m_1, m_2, \ldots\ldots\ldots\ldots.m_m\},\ \ m_i = \{x_i, y_i, \theta_i\}, i = 1, \ldots m$$

$$I = \{m'_{1,}m'_2, \ldots\ldots\ldots\ldots.m'_n\},\ \ m'_j = \{x'_j, y'_j, \theta'_j\}, j = 1, n,$$

Any two minutiae points mi in T andm'$j$in I match only if the *distance* (*sd*) between them is smaller than a given threshold r_0 and the *direction difference* (*dd*) between them is smaller than some angular threshold θ_0: and is given as:

$$sd(m'_j, m_i) = \sqrt{(x'_j - x_i)^2 + (y'_j - y_i)^2} \leq r_{0,}\ and$$

$$dd\left(m'_{j,}, m_i\right) = \min\left(\left|\theta'_j - \theta_i\right|, 360^0 - \left|\theta'_j - \theta_i\right|\right) \leq \theta_0$$

After the comparison of minutiae points the result is converted into a similarity score and is calculated as

$$score = \frac{\mathrm{k}}{(\mathrm{m} + \mathrm{n})/2}$$

where$k$ represent the number of matching minutiae.

This approach described above is the general approach for fingerprint matching. However, the approach is not rotation invariant because adjustment of any variation in rotation is done up to a limited tolerance.

B. Local And Global Structures Based Algorithm

The algorithm proposed by byXudong Jiang and Wei-Yun Yau [5] is based on the distance and angle between the neighbouring minutiae points. Generally in most of the fingerprint recognition algorithms a minutiae is represented by four parameters

$Fk = (x_k, y_k, \phi_k, t_k)$

where (x_k, y_k) are the X and Y coordinates, $\phi_k$is the local ridge direction and $t_k$the minutia type (ridge-ending or bifurcation). All these parameters represent global characteristics and thus are rotation and translation dependent and as such cannot be used directly for match.

In Xudong Jiang and Wei-Yun Yau's method, for each minutia *Mk*, the relative distance *dki*, radial angle θ*ki*and minutia direction ϕ*ki*between minutia *Mk* and its *l*-nearest neighbourhood minutia *Mi*is calculated.

Thus the minutia *Mk* that describes its local structure characteristic with its *l*-nearest neighbourhood (here *l*=2) is given by:

$Flk = (d_{ki}, d_{kj}, \Theta_{ki}, \Theta_{kj}, \phi_{ki}, \phi_{kj}, n_{ki}, n_{kj}, t_k, t_i, t_j)$

Where *i* is the nearest minutiae and *j is* the 2nd nearest minutia of the minutia *k* and n_{ki}, $n_{kj}$are the number of ridges in between two minutiae.

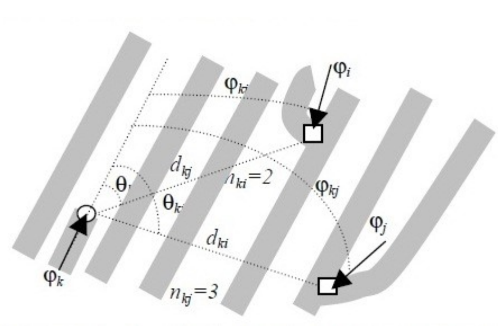

Fig 1 A minutia local structure with two nearest neighborhood.

C. Minutia and Local Compatibility Based Algorithm

The algorithm proposed [17] is based on global compatibility of local structures. Local structure comprises of a minutia and its two nearest neighbours to form a minutia triangle. A minutia is represented as

$M = (x_i, y_i, \Theta_i, t_i)$

where xi and yi are the minutia coordinates and *Θi* denotes the minutia position and ti denotes the minutia type.

A local structure of minutia p with its two nearest neighbours m and n is given by;

$V_p = (d_{pm}, d_{pn}, \Theta_{pm}, \Theta_{pn} \phi_{pm}, p_p, p_n, t_p, t_m, t_n)$

where dpm denotes the distance between minutiae p and minutiae m (see Fig 1.2), *Θ*pm denotes the relative radial angle between the directions of minutiae p and m, ϕ_{pm} denotes the relative radial angle between the directions of minutiae p and vector pm , ρ_p is ∠mpn , ρ_n is ∠mnp and t_m represents the type of minutiae *m*.

Thesimilarity between two triangles is obtained by calculating the distance between the two triangles given as

Dis (M, N) = W | V_M - V_N| , W is a weight vector

Similarity between two triangles M and N is calculated as

$$Sim(M,N) = \begin{cases} \dfrac{bl - Dis(M,N)}{bl} & \left|V_M^{(i)} - V_N^{(i)}\right| < TH_i, Dis(M,N) < bl \\ 0 & otherwise \end{cases}$$

Where THi and bl are the thresholds

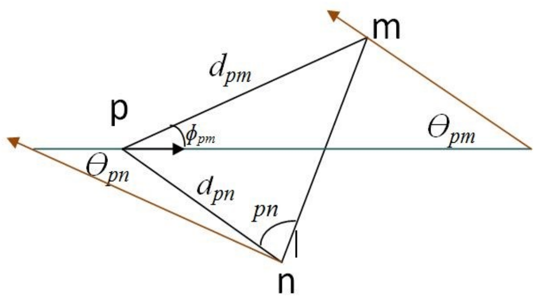

Fig 1.2 Showing minutia triangle formed by *p* with *n*and *m* as its two neighbours.

IV. PROPOSED ALGORITHM

In this paper a new fingerprint minutia matching technique is proposed, which matches the fingerprint minutiae by using the local structures of minutiae.The local structure of a minutia describes a rotation and translation invariant feature of the minutia in its neighbourhood. The local structures include all those features of a fingerprint which are independent of the position and angle of the fingerprint which include *distance between minutiae, angle between different minutiae, type of minutiaetc*. The approach uses two nearest neighbouring minutiae to define a structure for each extracted minutia and then parameter adjustment is done during adjustment phase in order to improve the accuracy. The proposed method consists of three main stages

i) Generate minutiae code: Minutiae code *mc* is a feature vector describing minutiae (explained in the subsection below). During the first stage, minutiae code *mc* is generated for all the minutiae points extracted from a fingerprint image.

ii) Calculate Adjustment Factor: In order to improve the accuracy and speed of fingerprint recognition, an adjustment factor is calculated by examining the closest mates between the two fingerprints under comparison.

iii) Calculate Matching Score: Finally matching score is calculated which shows the similarity score between the two fingerprints.

The three stages are explained in detail below.

A. Generate Minutiae Code (mc)

After minutiae extraction, which in this case has been done by using an existing method proposed in [7], minutiae code *mc* is generated for each minutia extracted from the fingerprint. The code *mc* is basically a six component feature vector describing a minutia with translation and rotation invariant features.

For each minutia *M*, the distance *dmi*,*dm*j and angle θ*m*between minutia *M* and its *two* nearest neighbourhood minutia *i, j*calculated by

$$dmi = \sqrt{(xk - xi)^2 + (yk - yi)^2}$$

$$\Theta_m = \cos^{-1}[(dmi^2 + dmj^2 - dij^2)/(2dmi * dmj)]$$

The minutia code *mc* for minutiae *m* with *i* and *j* its two closest neighbours can be given by

$$mc = (d_{mi}, d_{mj}, \Theta_m , t_m , t_i, t_j)$$

where$tm, t_i, t_j$are the types of minutiae (ridge ending or bifurcation) as shown in Fig 2.

It is easy to see that the minutiae code mc is based on local structure and is independent of the translation and rotation of the fingerprint and thus can directly be used for fingerprint matching.

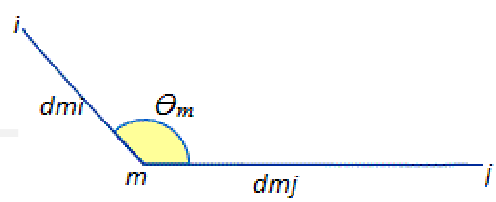

Fig 2 Example of minutia point with its two nearest neighbours

B. Calculate Adjustment Factor (AF)

Different images of the same finger may show variations in minutiae code due to differences in image orientation and resolution, thus affecting the accuracy of the matching algorithm. We introduce Adjustment Factor (AF) to handle such situations. This Adjustment Factor (AF) is used at the time of matching and it adjusts the values of minutiae code.

For a minutiae point *m* in claimant fingerprint *P*, a close mate *n* is found in template fingerprint *F*. Here *m* and *n* are close mates if they have similar minutiae codes. The similarity measure between two minutia codes is obtained by adding the differences between the respective components of the two minutia codes. Only first three components, being numerical values, are considered for obtaining the similarity measure, and the last three components, being type values, are not used for computing the similarity measure.

Let minutia codes of two minutia points *m* and *n* be represented by mc_m and mc_n, then

$$mc_m = (d_{mi}, d_{mj}, \Theta_m , t_m , t_{mi}, t_{mj}) \qquad (1)$$

$mc_n = (dn_i, dn_j, \Theta n, tn, t_{ni}, t_{nj})$

The similarity measure *sm* is calculated as

$sm=(\Delta i + \Delta j+ \Delta m)$

Where

$\Delta i = dmi - dni$
$\Delta j = dmj - dnj$
$\Delta m = \Theta_m - \Theta_n$

Two minutia points *m* and *n* are close mates if their similarity measure *sm* is less than a specified threshold *th*,

i.e. $sm<th$. (1)

th is set experimentally.

If a close mate between the claimant fingerprint and the template fingerprint exists then there is a possibility that the two fingerprints may match, and the two fingerprints need to go through a matching process that confirms that the two fingerprints match. However, the use of just one close mate to initiate the matching process between two fingerprints may not be adequate to proceed to the matching stage. This is because the chances of one close mate getting affected by various errors that may have come during extraction phase are high. Therefore, we use three close mates to decide if the two fingerprints should be sent to the matching process or not. To take care of outliers that may be present in the claimant fingerprint, a sufficient number of minutiae points N (N determined experimentally) from the claimant fingerprint are tested to obtain the three close mates. Note that three close mates may be obtained well before processing the given number of the minutiae points in which case the procedure is terminated immediately without processing the rest of the minutiae points. If three close mates are not present then the claimant fingerprint is rejected and declared not matching and no further processing is done. This makes the proposed matching method faster. If three close mates are present then Adjustment Factor is calculated before further matching process is executed.
Adjustment Factor is a three component vector that gives difference of various components of minutiae codes of two close mates and is calculated by subtracting first three corresponding components of minutiae codes of two close mates (the last three components of the minutia codes being the minutiae type are not included in the AF), i.e.

$AF = (\Delta i, \Delta j, \Delta m)$

For each of the three close mates obtained, an Adjustment Factor is calculated and the average of all the calculated AFs gives us the final Adjustment Factor (AF)

$AF = (AF1 + AF2+AF3)/3$

The final Adjustment Factor (*AF)* is then applied to all the minutiae codes (*mcs*) of the claimant fingerprint *P* before matching process is started. This is done by subtracting each component of the Adjustment Factor from the corresponding components (first three) of a minutia code. An example of an adjusted minutiae code mc_m' of minutiae code mc_m is given below.

$mc_m' = (d_{mi}', d_{mj}', \Theta_m', t_m , t_i, t_j)$

where $d_{mi}' = d_{mi} - \Delta i$
$d_{mj}' = d_{mj} - \Delta j$
$\Theta_m' = \Theta_m - \Delta m$

The adjusted minutiae codes are then used in the matching process as explained in the next section.

C. Matching and Matching Score

Matching process is a two stage process. During first stage if three minutiae codes of a claimant and a template fingerprints match then only second stage of matching process is initiated. This helps in early elimination of non-matching template fingerprints which makes the matching process faster.
The first stage of the match process is explained in the previous section which results in defining the Adjustment Factor (AF). After applying the Adjustment Factor (AF), the second stage of matching is done by comparing each minutiae point from the claimant fingerprint with all the minutiae points of the template fingerprint. Two minutiae points *m* and *n* with minutia codes mc_m' and mc_n match if the following four conditions are satisfied:

1. The last three components of minutia code mc_m' are equal to the last three components of minutiae code mc_n i.e.

   $t_m= t_n$
   $t_{mi} = t_{ni}$
   $t_{mj} = t_{nj}$

2. The absolute difference between the first nearest distance of mc_m' (i.e. d_{mi}') and first nearest distance of mc_n (i.e. d_{ni}) is less than a specified threshold *d1 (chosen experimentally) i.e*

   $abs (d_{mi}' - d_{ni}) < d1$ (2)

3. The absolute difference between the second nearest distance of mc_m' (i.e. d_{mj}') and first nearest distance of mc_n (i.e. d_{mj})is less than a specified threshold *d1 (chosen experimentally) i.e*

   $d_{mj}' - d_{nj}< d1$ (3)

4. The absolute difference between Θ_m' of mc_m' and Θ_n of $mc_n$is less than a specified threshold *d2 (chosen experimentally) i.e*

$$\Theta_m' - \Theta_n < d2 \qquad (4)$$

If any one of the above conditions is not satisfied then match fails. This process is used to obtain all the matching points. The final matching score is calculated as below:
*score=(2*k/(n1+n2)/2)*100*

where$k$ is the number of matching minutiae points, *n1, n2* are the number of minutiae points in the template and claimant fingerprints respectively.
Two fingerprints match if their matching score is greater than some given threshold T (chosen experimentally).

V. RESULTS AND DISCUSSIONS

We have tested our proposed method on DB1 of FVC 2004 dataset and BioSecure Fingerprint dataset. These datasets are considered to be the most difficult for most methods because no efforts were made to control image quality and the sensor platens were not systematically cleaned [8]. Fig 4 shows examples of images, which are of poor quality, contained in these datasets.

Justifications of using specific values for the control parameters of the proposed method are discussed below.

a) The optimal value for N (Number of minutiae points to be processed in first stage of matching to obtain the three close mates) was obtained experimentally. It was observed that processing of 50% or more number of minutiae points produce almost same result and any value below 50% degrades the final result. Thus the value of N was set as 50% of the claimant minutia points obtained and this reduces the processing time without affecting the matching results.

b) Three close mates are necessary to take the matching to the second stage. The close mates should have no or very little variation. Thus the value of similarity measure *sm* between the two minutiae points should be as small as possible to define these as close mates. Experimental results show that if *sm* is less than 6 (i.e. *th*=6 in Eq(1)), it is safe to label the two minutiae points as close mates. Hence the value of *th* was chosen as 6.

c) The values for *d1* and *d2* should be small and chosen appropriately to compensate for the local elastic deformation. The fingerprint images of FVC 2004 database have a resolution of around 500dpi. Thus a value of 10 pixels for *d1* will allow deformation of up to 0.02 of an inch which is small enough and shall not degrade the final result. Also it was observed that the inter ridge distance for the fingerprints is greater than 10 pixels; hence it is safe to choose a value of 10 for *d1*. This was also verified experimentally with FVC 2004 and BioSecure fingerprint datasets.

   The threshold value for d2 (for equation 4) can be set as follows: Let *dmi* be the first nearest distance between minutiae points i and m and *dmj* be the second nearest distance between minutiae points j and *m*, and Θ be the angle between lines mi and mj (see Fig 2). Any chord displacement at point i *say* Δd (allowed chord displacement is less than *d1* which is set to 10 as discussed earlier) will result in an angular displacement of Θ_d and is given by
   $\Theta_d = \Delta d/dmi \qquad (5)$.
   The maximum allowed angular displacement is obtained by setting maximum value for Δd. Thus the value of the threshold *d2 is* $\Delta d/dmi = 10/dmi$ (since maximum allowed value of Δd is 10 as discussed above). This was also verified experimentally with FVC 2004 and BioSecure fingerprint datasets.

The results obtained by our proposed approach are presented in *Tables I and Table II.*
Table I shows the result of the proposed algorithm onFVC 2004 database DB1 before applying the adjustment factor and *Table II* shows results after applying adjustment factor. From the results, it can be observed that by applying the adjustment factor, the False Rejection Rate (FRR) has reduced. *Table I* below shows, that before applying adjustment factor (AF) the FRR varied from 8.6% to 14.6% for different thresholds T, and *Table II* shows that after applying the adjustment factor (AF) the FRR varied from 7.3% to 14.2% and the equal error rate (EER) is around 9%.
Adjustment factor (AF) has shown no or negligible effect on the False Acceptance Rate FAR as is shown in the Table II. The reason is that Adjustment Factor is applied to improve the acceptance rate and not to improve the rejection rate of the fingerprint matching algorithm. Fig 5 shows the final graph of FAR and FRR.The results are compared with the algorithms of Feng et al [17] and Jiang [5] as shown in Table III. The EER of our proposed method is 9% which is better than both of the algorithms.
The proposed method is rotation invariant. Incorrect matches and high rejection rates are mainly due to poor quality of fingerprint images. The result of a fingerprint matching algorithm highly depends on the accuracy and efficiency of minutiae extraction and image pre-processing algorithms. Based on the experimental results, matching errors in our proposed system mainly result from poor quality images and incorrect minutiae extraction and thus add to the low verification rate. The quality of the fingerprint images is the most serious issue in fingerprint verification system and in future our efforts will focus on image enhancement schemes. Future work may also make use of an incremental system [24] that makes use of sub-space grids [25,26] for fast matching. A rule extraction system [23] may be incorporated for routing through most appropriate sub-space grids.

Fig 4 Example of poor quality fingerprint Images

TABLE I
FAR & FRR BEFORE APPLYING ADJUSTMENT FACTOR.

Threshold T	FAR	FRR
40	14.5 %	8.6 %
45	13.1%	8.9%
50	11.2 %	9.3 %
55	9.1%	9.6%
60	8.6 %	10.8 %
65	7.9 %	12.8 %
70	6.8 %	14.6 %

TABLE II
FAR &FRR AFTER ADJUSTMENT FACTOR.

Threshold T	FAR	FRR
40	14.5 %	7.3 %
45	13.1%	7.6%
50	11.2 %	8.1 %
55	9.1%	8.7%
60	8.6 %	10.1 %
65	7.9%	12.3 %
70	6.8 %	14.2 %

TABLE III
ERR COMPARISON

EER of Jiang's Algorithm[17]	EER of Yansong Feng's Algorithm	EER of Proposed Algorithm
17.5%	9.4%	9.1%

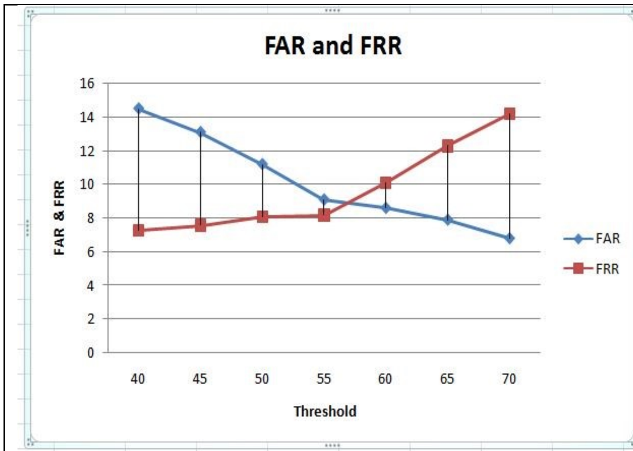

Fig 5 FAR & FMR OF THE PROPOSED METHOD

VI. CONCLUSION

In this paper we presented a new algorithm that is rotation invariant and makes use local features for fingerprint matching. Minutiae code mc was introduced to describe each minutia in a fingerprint. Minutiae code mc is based on the local features of the fingerprint and is independent of the rotation or direction of the fingerprint image. Adjustment factor (AF) was proposed for improving the performance of the proposed algorithm. Adjustment factor was calculated from the minutiae code (mc) of the two fingerprints being matched. The proposed strategy was tested on a number of images of publicly available FVC2004 database. The first version of the proposed algorithm (i.e. without adjustment factor) produced FRR between 6.6% and 16.3% for different thresholds and the second version of the proposed algorithm with adjustment factor produced the FRR between 4.6% and 15.1%.After applying Adjustment Factor (AF) the False Rejection Rate has decreased by more than one percent.

REFERENCES

[1] Maio et al. (2002a). Maio D., Maltoni D., Cappelli R., Wayman J.L. and Jain A.K., "FVC2000: Fingerprint verification competition," IEEE Transactions on Pattern Analysis Machine Intelligence, vol. 24, no. 3, pp. 402–412, 2002a.

[2] Handbook of Fingerprint Recognition, DavideMaltoni, Dario Maio, Anil K. Jain, SalilPrabhakar

[3] Bishnu A., Das S., Nandy S.C. and Bhattacharya B.B., "Simple algorithms for partial point set pattern matching under rigid motion," Pattern Recognition, vol. 39, no. 9, pp. 1662–1671, 2006.

[4] Udupa, Garg and Sharma (2001). Udupa R., Garg G. and Sharma P., "Fast and Accurate Fingerprint Verification," in Proc. Int. Conf. on

Audio- and Video-Based Biometric Person Authentication (3rd), pp. 192–197, 2001.
[5] Jiang and Yau (2000). Jiang X. and Yau W.Y., “Fingerprint Minutiae Matching Based on the Local and Global Structures,” in Proc. Int. Conf. on Pattern Recognition (15th), vol. 2, pp. 1042–1045, 2000.
[6] International Conference on Computer Science and Electronics Engineering 2012.
[7] Khan, Asif Iqbal; Wani, MohdArif, "Strategy to extract reliable minutia points for fingerprint recognition," Advance Computing Conference (IACC), 2014 IEEE International , vol., no., pp.1071,1075, 21-22 Feb. 2014
[8] Fingerprint Verification Competition (FVC2004), http://bias.csr.unibo.it/fvc2004, 2004
[9] Zhang, Weiwei, and Yangsheng Wang. "Core-based structure matching algorithm of fingerprint verification." Pattern Recognition, 2002. Proceedings. 16th International Conference on. Vol. 1. IEEE, 2002.
[10] Chikkerur, Sharat, and NaliniRatha. "Impact of singular point detection on fingerprint matching performance." Automatic Identification Advanced Technologies, 2005. Fourth IEEE Workshop on. IEEE, 2005.
[11] Chen, Z., Kuo, C.H., 1991. A topology-based matching algorithm for fingerprint authentication. In: Proc. 125th Annual IEEE Int. Carnahan Conference on Security Technology, pp. 84–87.
[12] Hrechak, A.K., McHugh, J.A., 1990. Automatic fingerprint recognition using structural matching. Pattern Recognit. 23 (8), 893–904.
[13] Jiang, X., Yau, W.Y., 2000. Fingerprint minutiae matching based on the local and global structures. In: Proc. 15th Int. Conf. Pattern Recognit. 2, pp. 1038–1041.
[14] Wahab, A., Chin, S.H., Tan, E.C., 1998. Novel approach to automated fingerprint recognition. IEE Proc. Visual Image Signal Process. 145 (3), 160–166.
[15] Yuliang He; Tian, Jie; Liang Li; Hong Chen; Xin Yang, "Fingerprint matching based on global comprehensive similarity," in *Pattern Analysis and Machine Intelligence, IEEE Transactions on* , vol.28, no.6,pp.850-862,2006.
[16] Germain, Califano and Colville (1997). Germain R., Califano A. and Colville S., “Fingerprint matching using transformation parameters,” IEEE Computational Science and Engineering, vol. 4, no. 4, pp. 42_49, 1997.
[17] Yansong Feng; Jufu Feng; Xiaoguang Chen; Zhen Song, "A NovelFingerprint Matching Scheme Based on Local Structure Compatibility," *Pattern Recognition, 2006. ICPR 2006. 18th International Conference on* , vol.4, no., pp.374,377.
[18] Bebis. G, Deaconu. T, and Georgiopoulos. M, “Fingerprint Identification using Delaunay Triangulation,” Intl Conf on Information Intelligence and Systems, 452-459, 1999.
[19] Huimin Deng, QiangHuo, “Minutiae Matching Based Fingerprint Verification Using Delaunay Triangulation and Aligned-Edge-Guided Triangle Matching,” AVBPA2005, vol 3546, pp270, 2005
[20] Jin Qi, Yangsheng Wang, “A Robust Fingerprint Matching Method”, Pattern Recognition, vol 38, pp1665-1671, 200.
[21] Lopez, Miguel, Patricia Melin, and Oscar Castillo. "A new approach for fuzzy feature extraction based on pixel's brightness." *Neural Networks (IJCNN), The 2010 International Joint Conference on*. IEEE, 2010.
[22] Feng, Jianjiang, Jie Zhou, and Anubhav K. Jain. "Orientation field estimation for latent fingerprint enhancement." *Pattern Analysis and Machine Intelligence, IEEE Transactions on* 35.4 (2013): 925-940.
[23] M. Arif Wani, “SAFARI : A Structured approach for automatic rule induction” *IEEE Transactions on Systems Man and Cybernetics journal*. Vol 31 (4): pp 650-657 Aug. 2001.
[24] M. Arif Wani, “Incremental Hybrid Approach for Microarray Classification”, *Seventh International Conference on Machine Learning and Applications*, San Diego, USA, pp. 514-520 , December, 2008.
[25] M. Arif Wani, "Microarray Classification using Sub-Space Grids", *Tenth International Conference on Machine Learning and Applications*, Hawaii, USA, pp. 389-394 , December, 2011.
[26] M. Arif Wani, “Introducing Subspace Grids to Recognise Patterns in Multidimensional Data”, *International Conference on Machine Learning and Applications,* Boca Raton, USA, pp. 33-39, 2012.