Optimized Minutiae–Based Fingerprint Matching

Neeta Nain *, Deepak B M, Dinesh Kumar, Manisha Baswal, and Biju Gautham

Abstract—**We propose a new minutiae-based approach to match fingerprint images using similar structures. Distortion poses serious threats through altered geometry, increases false minutiae, and hence makes it very difficult to find a perfect match. This algorithm divides fingerprint images into two concentric circular regions – inner and outer – based on the degree of distortion. The algorithm assigns weightages for a minutiae–pair match based on the region in which the pair exists. The implementation of the algorithm on the standard FVC DB shows robust performance.**

Keywords:* *Distortion, minutiae–based structures, matching, mass–centroid concept

1 Introduction

Biometric recognition refers to the use of distinctive physiological (e.g., fingerprint, face, retina, and iris) and behavioral (e.g., gait, signature) characteristics, called biometric identifiers or automatically recognizing individuals. Biometrics offers reliable means of authentication, and greater security and convenience [1] than traditional methods of personal recognition; these attributes cannot be easily shared or stolen.
The existing approaches for fingerprint matching are: minutiae–based, and correlation-based. The former has several advantages over the latter such as lower time complexity, better space complexity, less requirement of hardware etc. The uniqueness of a fingerprint is due to unique pattern shown by the locations of the minutiae points – irregularities of a fingerprint – ridge endings, and bifurcations.
The non–linear distortion in the fingerprint images makes it very difficult to handle matching as it changes the geometrical position of the minutiae points. The regions, that are affected, shift the geometry of the minutiae and hence pose a potential threat to acceptance of a genuine match. The distortion is due to the pressure applied on the scanner, the static friction, the skin moisture, elasticity, and rotational effects etc. [2] that occur during the acquisition. On the sensor, if the force is not applied orthogonally to the surface, elastic deformations are formed [3]. The level of distortion increases from the center towards the outer regions. As we move from the center, towards the boundary of a fingerprint image, the distortion due to rotation, static friction etc. increases. In other words, the distortion is more towards the boundaries than at the center.
Different algorithms exist that handle distortion in fingerprint images. [4]–[6] are based on the distortion detection, [7]–[9] are bounding–box–based, [10]–[11] local similarity, [3] deformation model, and others [12]–[13].
[14] uses both minutiae and texture features; aligning is based on the maximum number of corresponding pairs, and matching is based on the sum of the squared differences between the feature vector. [14] method does not take into account the images with large contact area. The algorithm [3] employs a thin–plate spline model to describe non–linear distortion between the corresponding minutiae pairs. It tries to find a match, which sometimes, may be forced even if they belong to different fingerprint images hence increasing the false acceptance rate.
In this paper, the algorithm employs a quick aligning stage after the extraction of the binary image. This method considers aligning the binary image by taking the mass–centroid[1] of the binary image, which is divided into two regions – upper and lower halves. This method is used to achieve a fast alignment and to set a reference, which gives an excellent approximation of the alignment that has to be done at a later stage using the minutiae points. When the angle of rotation to be achieved is large, this method is extremely efficient and saves a lot of time, which otherwise is spent in finding the correspondences to find an alignment. Calculating the alignment and correspondences between the minutiae points proves costly and time consuming especially when the translational and rotational parameters are large.

2 Algorithm

The algorithm has two stages. In the first stage, the minutiae points are extracted, and in the second stage, the aligning and the matching of the fingerprint images are done. The algorithm is designed to reduce time taken in aligning, immediately after the calculation of the binary image.

2.1 Minutiae Extraction

The method proposed by Hong et al. [15] is used for efficient minutiae extraction. The so obtained thinned

*Date of Submission: 9^{th} of March, 2008. Address of the corresponding author: Department of Computer Engineering, Malaviya National Institute of Technology, Jaipur–302017, India. Telephone/Fax No.: +91 141 2529140/+91 141 2713285

[1] Centroid is calculated for the black pixels.

ISBN:978-988-98671-9-5

binary image is used to extract the minutiae points using the crossing number method [17]–[19]. P is the set of minutiae points so obtained.
Even after the efficient extraction of the thinned binary image, there may be a few false minutiae. This is due to the amount of noise, moisture on the finger, dust on the scanner etc. About 10% of the images that are acquired are of poor quality [15]. The next stage is to eliminate the false minutiae from the set P. If not eliminated, the false minutiae aid the increase in false acceptance rate. The algorithm employs the method described in [16] to eliminate the false minutiae. It is observed that the genuine minutiae points are quite far away from one another. False minutiae points seem to accumulate very closely.

2.2 Region separation and alignment

The region separation and the optimized alignment stage make this algorithm novel. The fast alignment is sought as an approximate alignment stage, and the region separation is devised in order to handle the non–uniform distortion.

2.2.1 Fast alignment—an optimization

The devised algorithm has a fast alignment stage after the extraction of the binary image. It is based on the mass–centroid concept which uses the non–minutiae features. The algorithm divides the binary image horizontally into two halves – upper and lower. The centroid for the black pixels is calculated for each such region. Then, the centroids are joined to obtain a line. The whole image is now rotated so as to make this line parallel to the Y–axis. This method is applied on all the images – template and test images. This is shown in Fig. 1. This stage proves to be very effective especially when the alignment angle is very large, and sets a reference parameter for all the images. Although this method is an approximate alignment, it saves time in further alignment processes. Hence, the alignment stage based on the minutiae points becomes very fast and computationally much easier at a later stage; the differences in the corresponding pairs come out to be much smaller at this stage than what they would be without this stage. The algorithm [20] uses minutiae–centered circular regions to find correspondences in order to align. But, when the angle of rotation is large, this comes out to be computationally expensive.

Figure 1. The left binary image is the one without the mass–centroid–based alignment. The right binary image is rotated based on the mass–centroid concept.

2.2.2 Region separation—sector formation to handle distortion

The degree of distortion at the regions close to the center is lesser as compared to the regions that are away from the center. In other words, the distortion is not uniform throughout the fingerprint image. For example, rotational effects are lesser at the central regions as compared to the regions away from the center owing to the elasticity. When the force applied on the scanner is not orthogonal, the central regions show lesser amount of distortion than any other. When the degree of distortion is high, the aligning is time consuming. The algorithm [21] uses a triangular–based matching by joining the neighborhood minutiae points, it does not consider the non–uniform distortion that is present throughout the image. Hence, finding the alignment becomes very costly when all the minutiae points are considered.
The algorithm separates the fingerprint image into two concentric circular regions – inner (R_i) and outer (R_o) – as shown in the Figure 2. For aligning the image post region separation, the algorithm considers only the minutiae in the inner regions, as distortion is comparatively lesser in R_i. The diameter D of the inner circle is taken as half of the length of the minor axis μ of the elliptical fingerprint image. For the values 0.4μ and 0.3μ, the number of minutiae is insufficient to find an accurate alignment, hence, we chose $D = 0.5\mu$ as justified in Table 1.

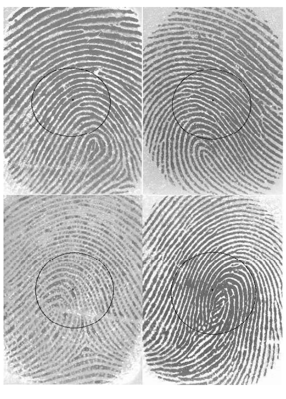

Figure 2. The inner circular regions are shown in the figure[2].

Value of D	Average number of minutiae
0.3μ	5.8
0.4μ	7.66
0.5μ	10.55

Table 1: Selection of the parameter D. The average number of minutiae is calculated on 20 random images.

The minutiae points in R_i are P_i, and in R_o are P_o. The set P_{pi} stores every possible pair of the set P_i; similarly, P_{po} is calculated.

2.3 Alignment

Since the distortion, which is the cause for false minutiae, is more in R_o than in R_i, the algorithm considers the minutiae points in the inner region to find an alignment with the template image. Alignment based on the singular point detection is more prone to errors [22]. Since the algorithm has already aligned the images using the mass-centroid concept, the angle of aligning in this stage is very small. The parameters that we used to find an alignment are:

- Ridge count r between any two minutiae points. This is calculated by tracing the line and calculating the number of 1–to–0[3] transactions in the binary image. The ridge count is a rotational–and–scaling independent parameter.
- The orientation o of the minutiae points at the ends of a line. This is independent of scaling.
- The type of minutiae t at the end points of the line formed using the minutiae pair. For example: if both the end points have a bifurcation or a ridge ending, or the pair has a bifurcation and a ridge. This is also independent of rotation and scaling.
- The angle a of the line joining the minutiae points with respect to a reference line, say the X–axis. This is also a scaling-independent factor.

The method that we have adopted to align the image is outlined below:

1. Minutiae pairs are formed by using all the minutiae P_i in R_i. We call this set as P_{pi} for an input image, and similarly minutiae pairs P_{pi}^{t}[4] for the template images is calculated.

2. Let there be M and N minutiae pairs in the sets P_{pi} and P_{pi}^t respectively. Every pair in P_{pi} is compared with every pair in P_{pi}^t using the parameters mentioned under Section 2.3. The detailed proposed algorithm for aligning is summarized as Algorithm 1.

Algorithm 1 Alignment

Require: P_{pi}: Minutiae pairs in R_i, P_{pi}^t: Minutiae pairs in R_i^t
M {Number of minutiae pairs in R_i}, N {Number of minutiae pairs in R_i^t}
Initialization
$Angle = 0$; {an array of unknown size}

```
procedure Alignment
  for (i = 1 : M) do
    for (j = 1 : N) do
      if (r_i == r_j^t) then
        if (o_i == o_j^t) then
          if (t_i == t_j^t) then
            DiffAngle = (a_i − a_j^t)
          end if
        end if
      end if
      if (DiffAngle ≤ ThreshAngle) then
        Add DiffAngle to Angle
      end if
    end for
  end for
end procedure
```

The value of the $ThreshAngle$ is taken as 2 degrees. This is very small because, an approximate alignment is already carried out on the binary fingerprint image. The mass–centroid alignment sets a reference and hence saves time in finding the correspondence among the minutiae pairs. The comparison of orientation, type of minutiae, and ridge count values in the respective arrays for the minutiae pairs is based on the pattern: lowest value of the X–coordinate to highest, and lowest value of the Y–coordinate to the highest, in the Cartesian co–ordinate system. For every image, the same order is followed. Hence, the order of finding correspondences is maintained.

3. The angle by which the image is to be rotated is taken as the average of the array $Angle$. Since most of the values in this array are small in magnitude, and close to one another, the mean does not affect the overall results. Since the outer minutiae P_o are not considered, the computational complexity is reduced by a great factor as the minutiae–pair comparison is of $O(n^2)$.

[2] The circle is drawn from the center of the image in the algorithm. The circular regions in this figure are not scaled.

[3] The value 1 stands for the white pixel and black is represented by the value zero.

[4] The superscript t is used to show that the parameter is for the template image. The parameters without this subscript are for the test images.

2.4 Minutiae–pair Matching

After the alignment stage, the test image is aligned in accordance with the template image. In order to account for the distortion in the fingerprint images, the algorithm divides the region R_o into several sectors as shown in Figure 3. In each such sector, independent of other sectors, all possible minutiae pairs are calculated and their parameters are stored. If the number of sectors is small, say four or less, the non–uniform distortion affects the results since the structures constructed using the minutiae become large; smaller structures show greater tolerance when distortion is involved. In our algorithm, the number of sectors was chosen to be eight. Since the minutiae pairs are used in matching, and the structures formed by these pairs are small enough, the effect of distortion in the global structure is eliminated as far as possible. [23] uses the set of all possible triangles as input to Fuzzy Feature Match based algorithm. If the triangles constructed by joining far away points such that the level of distortion in the global perspective is high, the algorithm proves to be costly in finding feature matches. Also, since similar structures are used for matching, scaling does not play a threat in finding a match. While the algorithm also uses the minutiae pairs in R_i, the matches in R_i are given more weight–age than the matches in R_o. This is again due to the amount of distortion in R_o.
The matching is done sector-wise. The minutiae pairs are confined to a particular sector in R_o and R_i. For every sector, the number of pairs that match is calculated.

Figure 3. The images were divided into several sectors[5] in order to handle distortion.

Let there be U_k and V_k minutiae pairs in k^{th} sector i.e., S_k and S_k^t[6] respectively. Sector–wise matching includes both sectors (inner and outer) from a single image.

The detailed proposed algorithm for matching is summarized as Algorithm 2. The values C_i and C_o store the counts of the number of minutiae pairs that are matched in R_i and R_o respectively. A match is possible when e% of the pairs in R_i and f% of the pairs in R_o match. Here,

[5]Sectors are not measured to be equal in the rough figure.
[6]This denotes the inner and outer sector of the template image.

Algorithm 2 Sector–wise Matching

Require: P_{pi}: Minutiae pairs in R_i, P_{po}: Minutiae pairs in R_o for bot the images
U_k {The number of minutiae pairs in k^{th} sector of R_i} V_k {The number of minutiae pairs in k^{th} sector of R_i^t}
Initialization
$C_i = C_o = 0$; {an array of unknown size}
procedure Sector-wise Matching
  for $(i = 1 : U_k)$ **do**
    for $(j = 1 : V_k)$ **do**
      if $(r_i == r_j^t)$ **then**
        if $(o_i == o_j^t)$ **then**
          if $(t_i == t_j^t)$ **then**
            $C_i = C_i + 1$ if the pair $\in R_{ki}$
            $C_o = C_o + 1$ if the pair $\in R_{ko}$
          end if
        end if
      end if
    end for
  end for
end procedure

$e > f$. In our case, the value e was set to around 74% – found experimentally, and the value f was set to around 30%.

3 Results

The fingerprint database that we have used is the publicly available standard FVC-2004 fingerprint data – http://bias.csr.unibo.it/fvc2004/download.asp. The level of distortion in FVC–2004 DB is relatively higher than that in FVC–2000 and FVC–2002. We have tried the algorithm on a total of 320 fingerprint images that belong to 40 distinct fingers – eight copies of each finger. The algorithm was implemented on Matlab R20072. The algorithm was tested on a Pentium–4 processor (3 GHz), 1GB RAM system. It was observed that the average time taken for the extraction of the binary image was close to 8 seconds.
The average time taken for the matching is very low. The matching was done without the mass–centroid concept and with the mass–centroid concept, and the results were compared with each other. It is seen that the mass-centroid concept increases the speedness by several hundreds. This is shown in Table 2.

Mass-centroid	Time taken In Seconds
Without	0.2
With	around 0.00005681

Table 2: Classification results

The time taken without the mass–centroid concept is large in comparison with the other one due to the aligning stage which takes time. The fast aligning method approximates the alignment angle and also increases the speedness of the matching stage. The ROC curve hence obtained is shown in Figure 4.

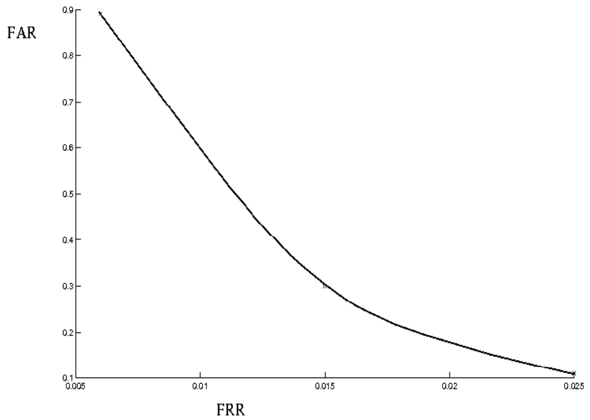

Figure 4. The Receiver–operating Curve at various thresholds.

4 Conclusions and future work

The mass–centroid concept is extremely fast and hence saves valuable time in finding the alignment and match. The time factor by which it improved the algorithm is several hundreds. Further work would be directed towards increasing the accuracy of heavily distorted images.

References

[1] A. Jain, L. Hong, and R. Bolle, On-line Fingerprint Verification, *IEEE Trans. Pattern Analysis and Machine Intelligence*, vol. 19, no. 4, Apr. 1997.

[2] A. Ross, S. Dass, and A K. Jain, A deformable model for fingerprint matching, *Patter Recogn.*, vol. 38, no. 1, pp. 95-103, Jan. 2003.

[3] A. M. Bazen and S. H. Gerez, Fingerprint Matching by thin-plate spline modeling of elastic deformations, *Pattern Recogn.*, vol. 36, no. 8, pp. 1859-1867, 2003.

[4] N. K. Ratha, R. M. Bolle, Effect of controlled acquisition on fingerprint matching, *in Proc. 14th ICPR, 1998*, vol. 2, pp. 1659-1661.

[5] C. Dorai, N. Ratha, and R. Bolle, Detecting dynamic behavior in compressed fingerprint videos: distortion, *in Proc. CVPR, Hilton Head, SC.*, Jun. 2000, pp. 2320-2326.

[6] H. Chen, J. Tian, and X. Yan, Fingerprint matching with registration pattern inspection, *Proc. AVBPA, Lect. Notes Comput. Science. 2688. New York, Springer*, 2003, pp. 327-334.

[7] A. K. Jain, L. Hong, and R. Bolle, On-line fingerprint verification, *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 19, no. 4, pp. 302313, Apr. 1997.

[8] X. P. Luo, J. Tian, and Y. Wu, A minutia matching algorithm in fingerprint verification, *in Proc. 15th ICPR*, Sep. 2000, vol. 4, pp. 833836.

[9] D. Lee, K. Choi, and J. Kim, A robust fingerprint matching algorithm using local alignment, *in Proc. 16th ICPR, Aug. 2002*, vol. 3, pp. 803806.

[10] N. Ratha, R. Bolle, V. Pandit, and V. Vaish, Robust fingerprint authentication using local structural similarity, *in Proc. 5th IEEE Workshop on Applied Computer Vision, Dec. 2000*, vol. 1, pp. 2934.

[11] X. Jiang and W. Y. Yau, Fingerprint minutiae matching based on the local and global structures, in Proc. 15th ICPR, Sep. 2000, vol. 2, pp. 10381041.

[12] A. Senior and R. Bolle, Improved fingerprint matching by distortion removal, *IEICE Trans. Inf. Syst., Special issue on Biometrics*, vol. E84-D, no. 7, pp. 825831, Jul. 2001.

[13] X. J. Chen, J. Tian, and X. Yang, A new algorithm for distorted fingerprints matching based on normalized fuzzy similarity measure, *IEEE Trans. Image Process.*, vol. 15, no. 3, pp. 767776, Mar. 2006.

[14] Anil Jain, Arun Ross, and Salil Prabhakar, Fingerprint matching using minutiae and texture features, *in Proc. of Intnl. Conf. on Image Processing ICIP*, pp. 282-285, Thessaloniki, Greece, Oct. 7-10, 2001.

[15] Hong, L., Wan, Y., and Jain, A. K. Fingerprint image enhancement: Algorithm and performance evaluation. *IEEE Transactions on Pattern Analysis and Machine Intelligence* 20, 8 (1998), pp 777-789.

[16] Tico, M., and Kuosmanen, P. An algorithm for ngerprint image post-processing. *In Proc. of the Thirty-Fourth Asilomar Conference on Signals, Systems and Computers (November 2000)*, vol. 2, pp. 17351739.

[17] Amengual, J. C., Juan, A., Prez, J. C., Prat, F., Sez, S., and Vilar, J. M. Real-time minutiae extraction in ngerprint images. *In Proc. of the 6th Int. Conf. on Image Processing and its Applications (July 1997)*, pp. 871875.

[18] Mehtre, B. M. Fingerprint image analysis for automatic identication. *Machine Vision and Applications* 6, 2 (1993), 124139.

[19] S. Kasaei, M. D., and Boashash, B. Fingerprint feature extraction using block direction on reconstructed images. *In IEEE region TEN Conf., digital signal Processing applications, TENCON (December 1997)*, pp. 303306.

[20] Haiyong Chen, Hongwei Sun, Kwok-Yan Lam, A fast and elastic fingerprint matching algorithm using minutiae-centered circular regions, *Intl. Conf. on Emerging Security Information, Systems and Technologies, IEEE*, Oct. 14-20, 2007.

[21] Z. M. Kovacs-Vajna, A fingerprint verification system based on triangular matching and dynamic time warping, *IEEE Trans. on Pattern Analysis and Machine Intelligence*, vol. 22, no. 11, pp. 1266-1276, Nov. 2000.

[22] Sharat Chikkerur and Nalini Ratha, Impact of Singular Point Detection on Fingerprint Matching Performance, *AUTOID '05: Proceedings of the Fourth IEEE Workshop on Automatic Identification Advanced Technologies, IEEE Computer Society*, ISBN 0-7695-2475-3, pp. 207-212, 2005.

[23] Xinjian Chen, Jie Tian, Xin Yang, and Yangyang Zhang, An Algorithm for Distorted Fingerprint Matching Based on Local Triangle Feature Set, *IEEE Trans. on Information Forensics and Security*, vol. 1, no. 2, June 2006.