An Improved Fingerprint Recognition System using the Concept of Distance Vector

Deepak Vishwakarma
Asst. Prof., IT department
IMSEC, Ghaziabad

Deepak Gupta
Asst. Prof., IT department
IMSEC, Ghaziabad

ABSTRACT
A new feature for fingerprint images is introduced. This new feature is named as Distance Vector. A Distance Vector counts the minutiae points in each row of a particular fingerprint image. A Distance Vector is associated with every fingerprint in the database. At the time of enrollment this feature is stored with the concerned fingerprint and at the time of matching this feature is matched with the Distance Vector of each fingerprint in the template database. This process increases the reliability of the fingerprint recognition task. In the first stages, image normalization and orientation of the ridges are estimated.

Keywords
Fingerprint recognition system, Distance Vector, Feature Extraction

1. INTRODUCTION
Different features of a fingerprint provide the uniqueness to the fingerprint. Before performing matching one has to identify and extract all features from the image. Feature extraction is one of the most important tasks considered in an automatic fingerprint recognition system [1]. We considered four already existing minutiae features (Ridge Dot, Ridge Ending, Ridge Bifurcation and Ridge Trifurcation) along with a new one (Distance Vector) proposed in this paper. In some cases fingerprint image comes from offline source (inked image) or directly from online source (scanned through electronic fingerprint reader). There may be imperfections in the fingerprint image. In some cases certain minutiae can be missed by the system or in some cases some spurious minutiae may take place in the feature extraction process. This problem can generate some more serious problems in determining the coordinates of each true minutiae and its relative orientation in the image. These factors decrease the reliability of the fingerprint recognition system as this system solely depends upon some comparisons and with some tolerance limits. The algorithm developed in this paper strengthens the reliability factor of already existing algorithms by introducing the concept of Distance Vector. This algorithm is tested for both offline and online fingerprint images.

2. FINGERPRINT IMAGE ENHANCEMENT
For our experiment, we used the FVC2000 DB1_B database. Each fingerprint image is of size 300x300 pixels, in 8-bit gray scale. Captured fingerprint images can have poor ridge structures. The imperfection of these fingerprints makes imperfection arises: non-uniformity of the ridge density, appearance of non printed areas and also the existence of stains. These problems can be solved by enhancement of the image. Following steps come under the process of image enhancement.

Fig 1: Original Fingerprint Image

2.1 Image normalization
This process normalizes the global statistics of the image, by reducing each image to a fixed mean and variance, although this pixel-wise operation does not change the ridge structure. The normalization factor is calculated according to the mean and variance of the image [2]. The normalized image can be estimated as

$$G(i,j) = \begin{cases} M_0 + \sqrt{\dfrac{VAR_0((I-M)^2)}{VAR}}, if\ I(i,j) > M \\ M_0 - \sqrt{\dfrac{VAR_0((I-M)^2)}{VAR}}, otherwise \end{cases}$$

Where I(i,j) denotes the gray level value at pixel (i,j). M and VAR are the estimated mean and variance of the fingerprint image I respectively whereas M0 and VAR0 are the required mean and variance respectively. G(i,j) denotes the normalized gray level intensity at pixel (i,j).

Fig 2: Normalized Fingerprint Image

As a result of normalization the contrast and brightness of the image get improved.

2.2 Orientation Field Estimation
This step determines the dominant direction of the ridges in different parts of the fingerprint image. The orientation field

represents the local orientation of the ridges in the fingerprint [8,9]. Actually, orientation tells about the direction of the ridges and this direction pays a vital role at the time of matching process. The smoothed orientation can be calculated on the basis of Least Mean Square Algorithm [3]. Following are the steps required in the process of orientation estimation.

1. Divide the input image I(i,j) into non-overlapping w x w blocks.
2. Calculate the Gradient ∂x(i,j) and ∂y(i,j) of each block. Gradient calculation is done for each pixel in the block and in X-Y coordinates.

Estimate the local orientation according to the given formulae [3]


(a) (b)


(c) (d)

Fig 3: (a) Gradient in x direction (b) Gradient in y direction (c) Double Gradient in x direction (d) Double Gradient in y direction

$$v_x(i,j) = \sum_{u=i-\frac{w}{2}}^{i+\frac{w}{2}} \sum_{v=j-\frac{w}{2}}^{j+\frac{w}{2}} 2\partial_x(u,v)\partial_y(u,v)\,,$$

$$v_y(i,j) = \sum_{u=i-\frac{w}{2}}^{i+\frac{w}{2}} \sum_{v=j-\frac{w}{2}}^{j+\frac{w}{2}} 2(\partial_x^2(u,v) - \partial_y^2(u,v))\,,$$

$$\theta(i,j) = \frac{1}{2}\tan^{-1}\left(\frac{v_y(i,j)}{v_x(i,j)}\right),$$

Here θ is the Least Square Estimate of the local ridge orientation at the block centered at pixel (i,j).

2.3 Image Enhancement using Gabor Filter

This step is required to improve the quality of the image by improving the clarity of the furrow and ridge structures in the fingerprint image. Image filtered by Gabor Filter will have less noise and it will be easier to perform recognition at the time of Feature Extraction. Gabor filter is well discussed in [5, 6, 7, 8, 10]. We have taken Even Symmetric Gabor Filter for this purpose. Ridge features of the blocks are enhanced by taking 8 different orientations. General form of a Gabor filter is as follows where f is the frequency of inter-ridge distance along an angle θ from the x-axis, σx and σy are the standard deviations of the Gaussian Envelop along

$$G(x,y;\theta,f) = exp\left\{-\frac{1}{2}\left[\frac{x_\theta^2}{\sigma_x^2} + \frac{y_\theta^2}{\sigma_y^2}\right]\right\}\cos(2\pi f x_\theta),$$

$$x_\theta = x cos\theta + y sin\theta,$$

$$y_\theta = -x sin\theta + y cos\theta,$$

x are the and y axes. We have decided the values of different parameters as follows; the average inter ridge distance is about 10 pixels so the average filter frequency is taken as f = 0.1. After that the values of δx and δy have to be decided. These values must be neither very small nor very large because if we decide small values for δx and δy, it will be ineffective to remove the noise from the image and on the other hand, if we take large values, the filter will not be able to collect ridge and valley information. So we have decided that δx = δy = 4.0 [2]. The 8 different orientations corresponding to θ are (0o, 22.5o, 45o, 67.5o, 90o, 112.5o, 135o, 157.5o) with respect to x-axis.

2.4 Binarization

After the enhancement of the fingerprint image by Gabor Filter, we have to change the image pixels to binary form. Original fingerprint image has 256 gray scale levels for its each pixel. In the Binarization step a pixel will have 2 gray level i.e. pure black and pure white. The pixel values which are larger than the decided threshold are mapped to 1 (white pixel) and others are mapped to 0 (black pixels).

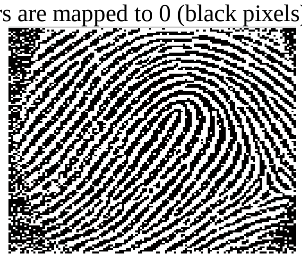


Fig 4: Binary Image

2.5 Thinning

To perform the minutiae extraction phase the thickness of each ridge must be one pixel thick. Among the existing thinning algorithms, the method proposed by Zhou, Quek and Ng [4] is adapted in our fingerprint identification system.

3. FEATURE EXTRACTION

3.1 Minutiae Extraction

We have considered four minutiae points in this step. The four minutiae features are Ridge Dot, Ridge Ending, Ridge

Bifurcation and Ridge Trifurcation. All the above minutiae points can be identified with the help of following formula [8]

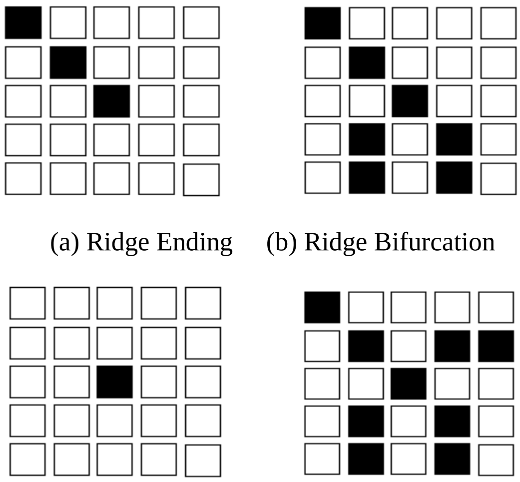
$$CN = 0.5 \sum_{i=1}^{8} |P_i - P_{i+1}|, \quad P_9 = P_1$$

Where, Pi is the pixel in the neighborhood of the currently considered pixel P. in the given method all the neighborhood pixels are scanned in the counter-clockwise direction.

P4	P3	P2
P5	P	P1
P6	P7	P8

Fig 5: Operating a Pixel

The nature of a pixel can be identified on the basis of the value of CN. Following are the CN values and corresponding ridge type.



(a) Ridge Ending (b) Ridge Bifurcation

(c) Ridge Dot (d) Ridge Trifurcation

Fig 6: Different Minutiae Points Depending upon the Value of CN

After applying the above formula we got the following information about a minutiae point:

- X-Y Coordinates of the minutiae point.
- Ridge Orientation.
- Type of the minutiae.

TABLE 1. Ridge type according to the values of CN

CN	Ridge Type
0	Ridge Dot
1	Ridge Ending
3	Ridge Bifurcation
4	Ridge Trifurcation

3.2 Distance Vector: A New Feature

This is the heart of our proposed methodology. In the minutiae extraction phase, we have checked about four different types of minutiae points i.e. Ridge Dot, Ridge Ending, Ridge Bifurcation and Ridge Trifurcation. Many times due to the poor quality of the image or the imperfections in the fingerprint image certain minutiae can be missed or in some cases some spurious minutiae may be generated that makes the verification algorithm less reliable. We define a new feature as Distance vector that verifies whether the employed algorithm performed well or not.

We defined a Distance Vector as follows; A Distance vector is a vector (a 1 x m metrics) that counts the number of minutiae points in each row of the fingerprint image. Here m is the number of rows in fingerprint image.

For an instance, suppose we have a fingerprint of size 32x32 pixels. After applying the Minutiae Extraction phase, we got the following minutiae points in the fingerprint.

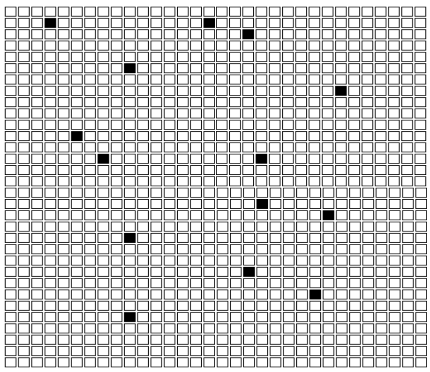


Fig 7: A 32x32 example fingerprint image having minutiae points in its different pixel positions

Black squares are used to denote the position of the minutiae points. Then the following is the Distance Vector for the shown fingerprint image;

(0,2,1,0,0,1,0,1,0,0,0,1,0,2,0,0,0,1,1,0,1,0,0,1,0,1,0,1,0,0,0,0)

In case of an image of size 128x128 the size of the Distance Vector will be 1x128. At the time of enrollment, Distance Vector of each fingerprint image will be calculated. The Distance Vector of each image will be stored with the template fingerprint image. At the time of fingerprint matching the Distance Vector of the taken fingerprint image will be compared with the stored template fingerprint image. Answer of matching depends upon the threshold value we decide for the Distance Vector. For example if we assume that there can be discrepancy of the two Distance Vectors at most at 5 places (or 5 rows) then this is our threshold value. If we want more secure system we can decrease the value of threshold to 3or 2.

4. MATCHING

Matching is done based on the minutiae based method. First of all, we identified the coordinated of the query fingerprint image along with the orientation field estimation of each minutiae point [8]. After that the position of the minutiae points are mapped to the corresponding sector [8, 9]. Matching of the Distance Vector will be performed after matching the orientation and position of the minutiae point. Depending upon the threshold values there will be a match or a mismatch. This step verifies the reliability of the performed method on the fingerprint image.

5. EXPERIMENTAL RESULTS

Using the FVC2000 DB1_B database for our experiment, the performance of the fingerprint image identification system can be evaluated by measuring its False Reject Rate (FRR) and False Accept Rate (FAR). The threshold of the matching score decides whether to reject or accept a match. The threshold is set for optimizing the performance. The simulation results are shown in table.

TABLE 2: System performance with different threshold values

	Th=5	Th=3	Th=2
FRR	24.4%	19.2%	16.7%
FAR	0.9%	1.54%	2.3%

6. CONCLUSIONS

We have developed a novel scheme for fingerprint image matching. The proposed method is based on minutiae based matching approach. Our method verifies the participation in the matching phase along with the reliability of the minutiae extraction phase itself. For improving the algorithm response, researchers are still busy in the improvement of the taken fingerprint image quality.

7. REFERENCES

[1] A. Jain, L. Hong and R. Bolle, "on-line fingerprint verification", IEEE Trans. Pattern Analysis and Matching Intelligence, Vol. 19, No. 4, PP. 302-314, April 1997

[2] L. Hong. Y. Wan and A Jain, "Fingerprint Image Enhancement Algorithm and Performance Evaluation", IEEE Trans. Pattern Analysis and Matching Intelligence, Vol. 20, No. 8, PP. 777-789, August 1998

[3] Anil K. Jain, Salil Prabhakar, L. Hong and Sarath Pankanti, "Filterbank Based Fingerprint Matching", IEEE Transactions on Image Processing, Vol. 19, No. 5, PP. 346-359, May 2000.

[4] R. W. Zhou, C. Quek, G. S. Ng, "Novel Single Pass Algorithm", Pattern Recognition Lett., IEEE Transactions on Image Processing, Vol. 16, No. 12, PP. 1267-1275, 1995

[5] Z. Wang, S. Chen, C. Bushe and X. Niu, "Performance Evaluation of Fingerprint Enhancement Algorithms", IEEE Transactions on Image Processing, PP. 389-393, 2008.

[6] Xudong Jiang, "A Study on Fingerprint Image Filtering", IEEE Transactions on Image Processing, Vol. 2, PP. 21-24, May 2009.

[7] Shunshan li , Min Wei , Haiying Tang , Tiange Zhuang and Michael H. Buonocore, "Image Enhancement Method for Fingerprint Recognition System" , IEEE Transactions on Image Processing, Sep 2005

[8] Tsong-Liang Huang, Che-Wei Liu, Jui-Peng Lin, Chien-Ying Li, Ting-Yi Kuo, "A Novel Scheme for Fingerprint Identification" , IEEE Transactions on Image Processing, 2005

[9] D.Simon-Zorita, J. Ortega-Garcia, S. Cruz-Llanas and J. Gonzalez-Rodriguez, "Minutiae Extraction Scheme for Fingerprint Recognition Systems", IEEE Transactions on Image Processing, 2001.

[10] A. Farina, Z. M., Kovacs-Vajna and A. Lione, "Fingerprint Minutiae Extraction form Skeletonized Binary Images", Pattern Recognition 32,, Vol. 5, PP. 877-889, 1999.