Journal of Computer Science 8 (9): 1467-1472, 2012
ISSN 1549-3636

Impact of Algorithms for the Extraction of Minutiae Points in Fingerprint Biometrics

[1]Sudha S. Ponnarasi and [2]M. Rajaram
[1]Department of EEE, Information and Communication Engineering,
Government College of Engineering, Tirunelveli, India
[2]Department of EEE, Electrical and Electronics Engineering,
Anna University of Technology, Tirunelveli, India

Abstract: Problem statement: As the importance of automatic personal identification applications increases, biometrics particularly fingerprint identification is the most consistent and greatly acknowledged technique. A very important step in automatic fingerprint recognition system is to extract the minutiae points from the input fingerprint images automatically and quickly. **Approach:** Fingerprints from the database FVC2002 (DB1-a) is used for experimental purpose. The minutiae points from 100 fingerprints were detected. It is proposed to use Minutiae Detection using Crossing Numbers (MDCN) and Minutiae Detection using Midpoint Ridge Contour Method (MDMRCM). Finally the performance of minutiae extraction algorithms using the number of minutiae detected in both the cases were analysed. **Results:** The result shows that the avearge performance of MDCN method for minutiae points detection is 88% and for MDMRCM method is 92%. **Conclusion:** The performance of MDMRCM is better than MDCN method. MDMRCM method extract more minutiae points than MDCN method. It consumes lesser time to get the output and the false minutiae points were not detected. And hence MDMRCM method is considered to be a superior than MDCN method.

Key words: Bifurcation, artificial neural network, minutiae extraction, minutiae, crossing numbers, midpoint ridge contour

INTRODUCTION

Biometrics and fingerprints: Personal identification are usually divided into three types as shown in Table 1, by what one owns (e.g., a credit card or keys), by something you know (e.g., a password or a PIN code) or by physiological or behavioural characteristics. The last method is referred to as biometrics and the six most commonly used features include face, voice, iris, signature, hand geometry and of course fingerprint identification (Maltoni *et al*., 2009).

It has been established and is commonly known, that everyone has a unique fingerprint (Yager and Amin, 2004) which do not change over time. Each person's finger has its own unique pattern, hence any finger could be used to successfully identify a person.

A fingerprint autopsy: A fingerprint's surface consists of ridges and furrows. The exact pattern of these ridges and furrows (or valleys) that makes the fingerprint unique. The features of a fingerprint can be divided into three different scales of detail, of which the coarsest is the classification of the fingerprint. The classification of fingerprints can be traced back to 1899. A british policeman, Sir Edward Richard Henry, introduced the classification system. It is called henry classification system. He introduced five different types of fingerprints; right loop, left loop, whorl, arch, tented arch. Its pictorial representation is given in Fig. 1. This classification system is still in use today but has been extended to include more types, for example double loop, central pocket loop and accidental.

The second scale of fingerprint details consists of features at ridge level. The discontinuities (endings, bifurcations,) that interrupt the otherwise smooth flow of ridges are called minutiae and analysis of them, their position and direction, is what identifies a person. Many courts of law consider a match with 12 concurring points (the 12-point rule) present in a clear fingerprint as adequate for unique positive identification.

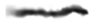
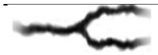
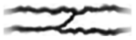
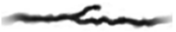
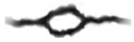
Some of the most common minutiae are presented in Table 2. The more unusual a minutiae is the more significance it has when used for identification.

Corresponding Author: Sudha, S. Ponnarasi, Department of EEE, Information and Communication Engineering, Government College of Engineering, Tirunelveli, India

Table 1: Identification methods

Method	Examples
What you know	Password, PIN code, user id
What you have	Cards, keys, badges
What you are (biometrics)	Fingerprint, face, voice, iris, signature, hand geometry

Table 2: Examples of minutiae types

	Ending (or termination)
	Bifurcation
	Independent (or short) ridge
Dot	
	Bridge
	Spur (or hook)
	Eye (or island)

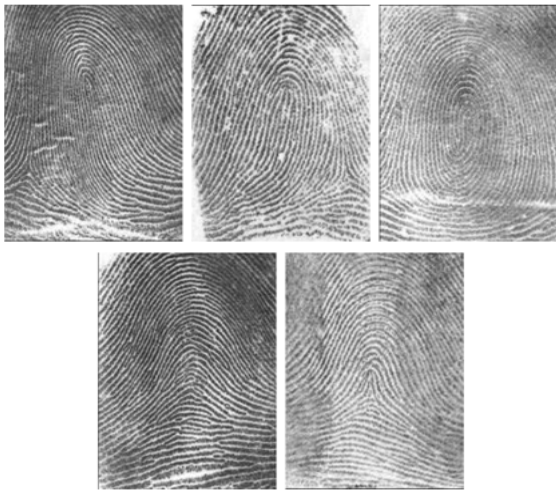


Fig. 1: Different fingerprint types of the henry classification system. (Top to bottom, left to right: right loop, left loop, whorl, arch, tented arch)

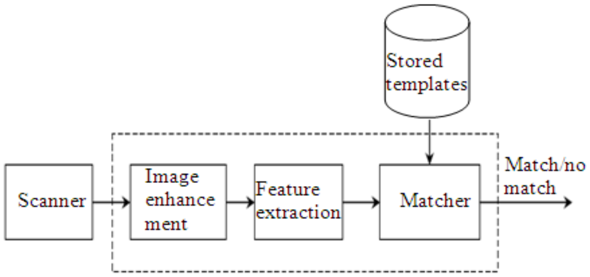


Fig. 2: A generic Automatic Fingerprint Identification System

Automatic Fingerprint Identification System (AFIS): For contemporary applications the fingerprint identification/verification process is undertaken automatically. Such a system is commonly known as Automatic Fingerprint Identification System (AFIS).

A generic AFIS consists of five different stages; fingerprint acquisition, image enhancement, feature extraction, matching and decision, which is illustrated in Fig. 2.

MATERIALS AND METHODS

Sambasiva *et al*. (2008) proposed fingerprint identification technique using a gray level watershed method to find out the ridges present on a fingerprint image by directly scanned fingerprints or inked impression. Hastings (2007) developed a ridge enhancement technique for fingerprint images. And this is implemented bt oriented diffusion method.

When the small irregular and discontinuous ridges were removed the intensity of the image changes regularly towards the ridges and valleys. But it is assured that the good and clear ridges and valleys were retained.

Jinwei *et al*. (2006) proposed a method for fingerprint verification in which both minutiae and model based orientation field is used. It gives robust discriminatory information other than minutiae points. The final conclusion of the matchers based upon the orientation field and minutiae are put together and hence done the fingerprint matching.

Kumari and Suriyanarayanan (2008) proposed a method to identify the performance measure of local operators in fingerprint.By using the various local operators namely sobel, roberts, prewitt, canny and Log, the edges of fingerprint images were detected. Then it is again segmented to obtainseparate segments. Sonavane and Sawant (2007) presented a method by introducing a special domain fingerprint enhancement technique for noisy images. This is a structure similarity measure approach. In this method the input fingerprint image is divided into a set of filtered images and then the orientation field is estimated.

Kukula *et al*. (2009) purposed a method to investigate the effect of human interaction on fingerprint matching performance, image quality and minutiae count. Three images of the right index fingers of 75 participants for both optical and capacitance fingerprint sensors were collected. To identify the variations in minutiae counts and image quality various tests like descriptive statistics, analysis of variance and Kruskal-Wallis nonparametric tests were conducted. It is proved that there is significant difference in image quality based on the force level and each sensor technology, yet there is no significant difference in minutiae count based on the force levels of the capacitance sensor. The image quality, shown to be effected by force and sensor type, is one offactors that affects the matching performance.

Girgis *et al.* (2007) proposed a method to describe a fingerprint matching based on lines extraction and graph matching principles. In this method generic algorithm was employed for accurate fingerprint matching. From the experimental results the effectiveness of the algorithm can be demonstrated. Luping and Zhang (2008) proposed a method for the estimation of fingerprint orientationfield. And this is implemented using ridge protection in four steps, first of all preprocessing of the fingerprint image was carried out, secondly, the primary ridge of fingerprint block were determined, thirdly the block direction was estimated and at last the estimated orientation field was corrected.

Kekre *et al*. (2010) method is image-based in which feature vectors of a fingerprint are extracted after sectorization of the cepstrum of a fingerprint. They are matched with those stored in the database.

The minutia points: The fingerprint signal in its raw form contains the necessary data for successful identificationhidden amongst a lot of irrelevant information. Thus image enhancing processes will remove noise and other clutter before the next step of localising and identifying distinct features, so called feature extraction. Today's AFISes commonly identify only ridge endings and bifurcations as distinct features (Maltoni *et al*., 2009). This mainly because all other type of minutiae can be expressed using only these two main types and they are by far the most common. Algorithms often return too many features, some of which are not actual minutiae; hence some kind of post-processing is necessary to remove these spurious minutiae.

A typical feature extraction algorithm is shown in Fig. 3. It involves five operations; (I) orientation estimation, with the purpose to estimate local ridge directions (II) ridge detection, which separate ridges from the valleys by using the orientation estimation resulting in a binary image (III) thinning algorithm/skeletonization, giving the ridges a width of 1 pixel, (IV) minutiae detection, identifying ridge pixels with three ridge pixel neighbours as ridge bifurcations and those with one ridge pixel neighbour as ridge endings and (V) post processing, which removes spurious minutiae.

The minutia detection methods under evaluation:

Minutiae Detection using Crossing Numbers (MDCN): Most of the fingerprint minutia extraction methods are thinning-based method. In which each ridge is converted to one pixel wide. That is known as skeletonization. By locating the ridge end points and bifurcation points minutia points are detected (Shi and Govindaraju, 2006).

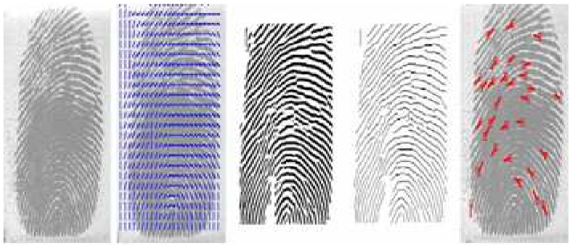


Fig. 3: Example of minutiae extraction algorithm; (I) input fingerprint, (II) orientation field, (III) extracted ridges, (IV) skeletonized image and (V) extracted minutiae

Table 3: Crossing number's property

CNProperty	
0	Isolated point
1	Ending point
2	Connective point
3	Bifurcation point
4	Crossing point

The concept of Crossing Number (CN) is widely used for extracting the minutiae (Jain *et al*., 1997). The crossing number for a pixel P is:

P4	P3	P2
P5	P	P1
P6	P7	P8

$$CN = 1/2\sum_{i=1}^{8}\left|P_i - P_{i+1}\right.$$

where, Pi is the binary pixel value in the neighborhood of P with Pi = (0 or1) and P9 = P1.

The skeleton image of fingerprint is scanned and all the minutiae are detected using the following properties of CN as shown in Table 3.

Then depending on the value of CN the minutiae point which have ending or bifurcation property are taken into consideration as shown in Fig. 4.

In the proposed method, the minutiae points locations and their considered direction from the 8 directions (N, S, W, E, NE, NW, SE, SW) are recorded then they used to construct the database depending of the number of recorded minutiae point and their direction.

The next step, the points with the code of it is direction will be stored in database which representing of fingerprint image, this will be as the first layer of the network which associated with the components of the input vector and fed forward back propagation neural network.

In this study, the skeleton image of the fingerprint has been scanned and all types of points in the image depending on the CN values from the Table 3 have been found. The location (x, y) and the direction of each point with CN value equal to (1and 3) were recorded in the matrix.

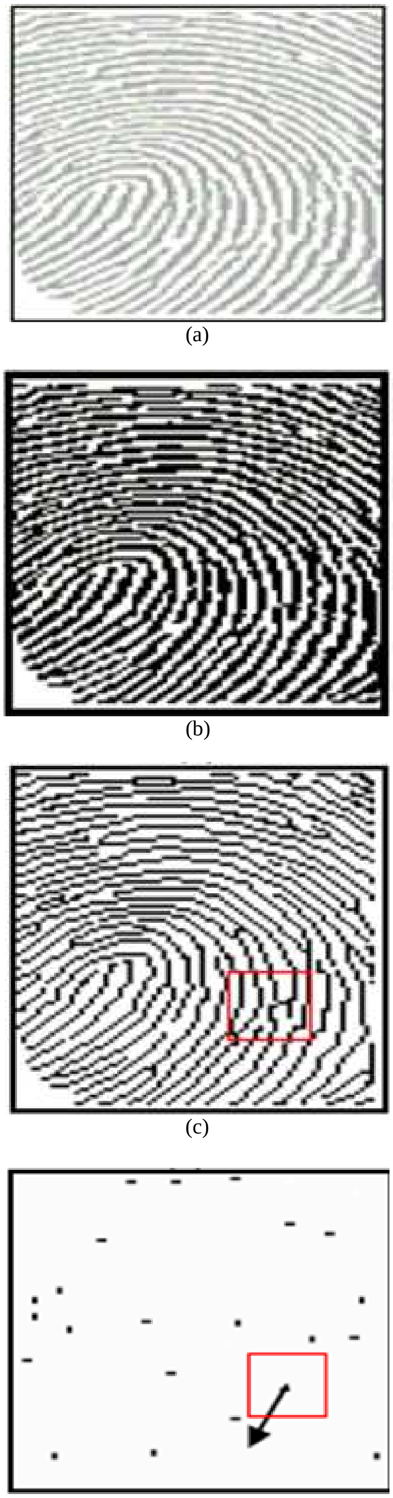


Fig. 4: (a) original image in gray, (b) binarization image to black-white(c) thinning black-white image and (d) determine the minutia point with direction SW

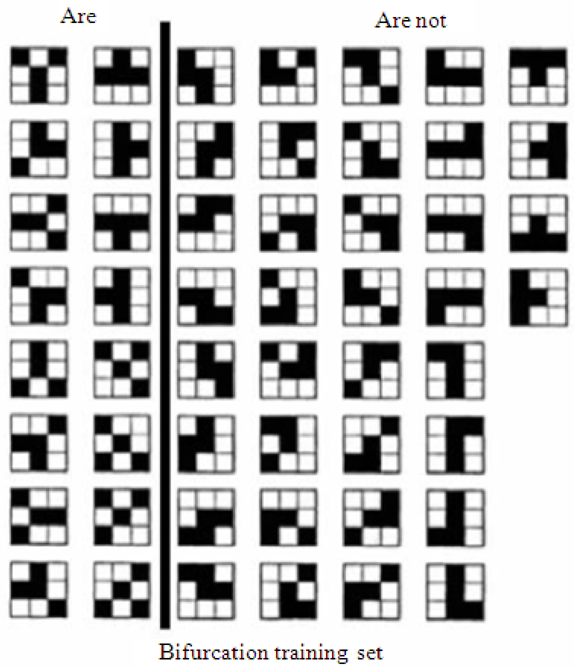


Fig. 5: A bifurcation training set grouped by desired output

First the processing of selecting points is start randomly, for each random point P(i, j), the proposed algorithm will search for the points from the recorded points in the matrix depending on the direction of the point P(i, j), for example if considered direction is N for point P so the searching process is done for points located at (x, y) where x< i also for NE and NW taken the same points, while the considered direction is S, SE and SW the searching process is done for point located where x>i as shown in the Fig. 3, this process will be repeated for all the points have the above condition.

where, for direction E the condition y <j is taken and for direction W the condition y > j is taken.

The function used to determine the value for each node of the input layer in the neural network is:

$$G(P(i,j)) = abs((j-i) * D$$

where, D is value for direction code. The bifurcation training set is illustrated in Fig. 5. The pseudo code for determine the direction of each ending and bifurcation points is as following:

```
Suppose P is the checked point and P1-P8 are
neighborhood pixels
If CN = 3 then
    If P1 and P3 and P7 = 1 then direction = W
    Else if P1 and P3 and P5 = 1 then direction = S
    Else if P1 and P7 and P5 = then direction = N
    Else if P3 and P5 and P7 = 1 then direction = E
    Else if P4 and P3 and P5 = 1 then direction = SE
    Else if P3 and P2 and P1 = 1 then direction = SW
    Else if P3 and P5 and P6 = 1 then direction = NE
    Else if P4 and P8 and P5 = 1 then direction = NW
End if
If CN = 1 then
    If P1 = 1 then direction = W
    If P1 = 1 then direction = W
    If P3 = 1 then direction = S
    If P7 = 1 then direction = N
    If P5 = 1 then direction = E
    If P4 = 1 then direction = SE
    If P2 = 1 then direction = SW
    If P6 = 1 then direction = NE
    If P8 = 1 then direction = NW
```

Minutiae Detection usingMidpoint Ridge Contour Method (MDMRCM):

Midpoint Ridge Contour Algorithm:

- Initialize current scanning position (x, y)-(0, 0)
- While x-end, y-end not comes; scan the image from current position left to right and top to bottom
- Place contour on current pixel (black) found in the image, store that point as start ridge minutiae position
- Calculate the length vector in each contour direction
- Find the maximum (Max-in) and minimum (Min-in) length vector in each direction
- Direction of minimum length vector will be width of the ridge line, count the pixels along the width
- Store all pixel positions along the width of the ridge in to an array Wd (1-n)
- Check contour positions for all pixels in an array from Wd (2) to Wd (n-1)
- If any of these pixels are white in color then store that pixel as minutiae position
- If minutiae shows ending of ridge line then store that point as end-ridge position; change the color of current ridge (between start-ridge to end-ridge) to red color go to step 2
- Else move to the pixel at the center in the array and move to the next pixel in to direction Max-in go to Step 4

Back propagation training algorithm is as follows:

Step 1: initialize the weights.

Step 2: while stopping condition is false, execute steps 3-10.

Step 3: For each training pair X:$_t$, do step 4-9

Step 4: Each input unit X_i, I = 1, 2, 3,.nreceives the input signal, x and broadcast it to the next layer.

Step 5: for each hidden layer neuron denoted as Z_j j = 1, 2, 3, ..,p, $Z_{mj} = V_{oj} + \sum X_I V_{ij}$ $Z_j = f(Z_{mj})$

Step 6: For each output neuron $Y_{k,}$ K = 1, 2..,m $y_{ink} = W_{ok} + \sum Z_j W_{jk}$ $y_k = f(y_{ink})$

Step 7: Compute δ_k for each output neuron, $Y_{k,}$ $\delta_k = (t_k - y_k)\, f(y_{ink})$ $\Delta W_{jk} = \alpha\delta_k Z_j \Delta W_{ok} \alpha\delta_k$ since $Z_0 = 1$

Step 8: for each hidden neuron, $\delta_{inj} = \sum\delta_k W_{jk}$ j-11, 2, 3…,P $\delta_j = \delta_{inj} f(Z_{inj})$ $\Delta V_{ij} \alpha\delta_j X_i$ $\Delta V_{oj} = \alpha\delta_j$

Step 9: W_{jk} (new) = W_{jk} (old) + ΔW_{jk} V_{ij} (new) = V_{ij} (old) + ΔV_{ij}

Step 10: Test for stopping condition.

Once the recognition program has been fully trained, each 3×3 pixel scan of the fingerprint image of the locations identified by “Midpoint ridge Contour” algorithm is passed to the program to identify as either an accepted minutia or not. If the 3×3 scan has centered onto an accepted minutia, an output of “1” results, otherwise an output of “0”results. For each input scan that resembles minutiae, its position in relation to the core point of the fingerprint is recorded. This information is useful in the fingerprint matching process.

RESULTS AND DISCUSSION

Minutiae points from 100 fingerprints in database FVC2002 (DB1-a) has been detected using MDCN and MDMRCM techniques and compared their performance. The comparison on 12 fingerprints out of 100 is given in the Table 4. Table values shows that MDMRCM is much efficient and can extract minutiae points in much better way and in greater number than MDCN based method. The MDMRCM method takes less time and detects no false minutiae. And hence MDMRCM method is considered to be a superior than MDCN method.

Advantages of MDMRCM method:

- Much efficient
- Less time required

No false minutiae

Table 4: Comparison between MDCN and MDMRCM method

Fingerprint imaqge	Actual no. of minutiae	MDCN	MDMRCM
1-1.tif	42	36	36
2-1.tif	35	32	34
3-1.tif	38	34	36
4-1.tif	29	27	28
5-1.tif	45	40	43
6-1.tif	35	30	31
7-1.tif	44	39	41
8-1.tif	46	42	44
9-1.tif	37	34	36
10-1.tif	39	32	34
11-1.tif	32	28	30
12-1.tif	41	37	39

Limitations and future work: Extracting minutiae from more than one finger is time consuming. For improvisation neuro fuzzy technique can be implemented in future. And more over the impact of age is to be calculated from finger prints.

CONCLUSION

The MDMRCM algorithm determine solutions for the major issues like efficiency, speed and accuracy in the detection of minutiae points in finger print biometrics. In this study, two minutiae detection algorithms such as MDCN and MDMRCM have been compared. Thus it concludes that MDMRCM algorithm is much efficient, faster and accurate than MDCN algorithm.

REFERENCES

Girgis, M.R., A.A. Sewisy and R.F. Mansourc, 2007. Employing generic algorithms for precise fingerprint matching based on line extraction. I. J. Graphics, Vision Image Proces., 7: 51-59.

Hastings, R., 2007. Ridge enhancement in fingerprint images using oriented diffusion. Proceedings of the 9th Biennial Conference of the Australian Pattern Recognition Society on Digital Image Computing Techniques and Applications, Dec. 3-5, IEEE Xplore Press, Glenelg, Australia, 245-252. DOI: 10.1109/DICTA.2007.4426803

Jain, A.K., L. Hong and R. Bolle, 1997. On-line fingerprint verification. IEEE trans. Patt. Anal. Mach. Intel., 19: 302-314. DOI: 10.1109/34.587996

Jinwei, Gu., J. Zhou and C. Yang, 2006. Fingerprint recognition by combining global structure and local cues, IEEE Trans. Image Proces., 15: 1952-1964. DOI: 10.1109/TIP.2006.873443

Kekre, H.B., T. Sarode and R. Vig, 2010. Fingerprint identification using sectorized cepstrum complex plane. Int. J. Comput. Appli., 8: 12-15.

Kukula, E.P., C.R. Blomeke, S.K. Modi and S.J. Elliott, 2009. Effect of human-biometric sensor interaction on fingerprint matching performance, image quality and minutiae count. Int. J. Comput. Appli. Technol. DOI: 10.1504/IJCAT.2009.024079

Kumari,V. V. and N. Suriyanarayanan, 2008. Performance measure of local operators in fingerprint detection. Acad. Open Inter. J., 23: 1-7.

Luping, J. and Y. Zhang, 2008. Fingerprint orientation field estimation using ridge projection. J. Patt. Recog., 41: 1491-1503. DOI: 10.1016/j.patcog.2007.09.003

Maltoni, D., D. Maio, A.K. Jain and S. Prabhakar, 2009. Handbook of Fingerprint Recognition. 2nd Edn., Springer, London, ISBN-10: 9781848822535, pp: 496.

Sambasiva Rao. G., C. NagaRaju, L.S.S. Reddy, E.V. Prasad, 2008. A novel fingerprints identification system based on the edge detection. Inter. J. Comput. Sci. Network Security, 8: 394-397.

Shi, Z. and V. Govindaraju, 2006. A chaincode based scheme for fingerprint feature extraction. Patt. Recog. Lett., 27: 462-468. DOI: 10.1016/j.patrec.2005.09.003

Sonavane, R. and B.S. Sawant, 2007. Noisy fingerprint image enhancement technique for image analysis: A structure similarity measure approach. J. Comput. Sci. Netw. Secur., 7: 225-230.

Yager, N and A. Amin, 2004. Fingerprint classification: A review. Patt. Anal. Appl., 7: 77-93. DOI: 10.1007/s10044-004-0204-7