A New Approach To Fingerprint Recognition

Ipsha Panda
IIIT Bhubaneswar, India
ipsha23@gmail.com

Saumya Ranjan Giri
IL&FS Technologies Ltd.
Bhubaneswar, India
saumya.giri07@gmail.com

Prakash Kumar
IL&FS Technologies Ltd.
Bhubaneswar, India
raiprakas@gmail.com

Anjali Mohaptra
IIIT Bhubaneswar, India
anjali@iiit-bh.ac.in

***Abstract*—Fingerprint recognition continues to be the most widely used biometric system for security purposes. Still it is a complex pattern recognition problem. In this paper, a new approach for fingerprint recognition is developed. There are three steps: image preprocessing, minutiae extraction and minutiae matching. In the first stage the fingerprint image is preprocessed to produce a thinned image with noisy data deleted. Then for minutiae extraction, some heuristic rules have been developed based on the concept of neighboring pixels. This stage results in removal of spurious minutiae points and true minutiae points are detected. Finally an algorithm for minutiae matching has been designed to improve the performance and have accurate results.**

Keywords- ***Otsu thresholding; Central line thinning; radial distance; Minutiae***

I. INTRODUCTION

Fingerprint recognition is one of the first applications of machine pattern recognition. It is still a challenging problem due to the following reasons: smaller overlap between template and the input image due to rotation, sweat or injury on finger cause incorrect minutiae detection, non linear distortion caused by finger plasticity. A fingerprint consists of ridges and valleys. In general a ridge can either come to an end which is termed as ridge end or it can split into two ridges which is termed as bifurcation point [4]. There are three stages in fingerprint recognition. The first step i.e. image preprocessing is done to produce a noise free. second step involves application of the heuristic rules developed to extract true minutiae points and finally the proposed algorithm is applied for minutiae matching The paper is organized as follows: section II presents preprocessing stage, section III presents proposed heuristics for minutiae extraction, section IV presents matching algorithm and section V concludes and marks the future aspects.

II. IMAGE PREPROCESSI NG STAGE

The fingerprint images do need some preprocessing in order to accurately detect the minutiae points for recognition. This step is done to reduce the noise and enhance the definition of ridges against valleys. The main steps involved are shown in the figure below:

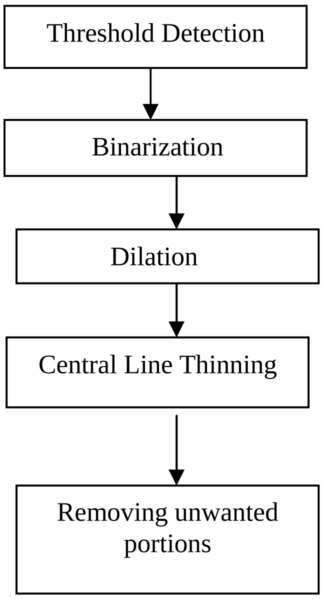

Figure 1. Image Preprocessing Step

A. Threshold detection using OTSU method [1]

The key idea of recursive Otsu thresholding is to take the set of background pixels as determined by standard Otsu thresholding and re-threshold these pixels. If let Otsu (a, b) denote the Otsu-selected threshold for all pixels in the image between values a and b, the recursive Otsu thresholding can be written as

T_1= Otsu (0, 255)

T_k = Otsu (Tk-1, 255)

The Recursive Otsu thresholding algorithm where T_1 is the threshold determined from recursion k. The following three stopping criteria for the algorithm:

1. Stop if the number of pixels added by the current recursive step exceeds the number of pixels selected by the first threshold T_1, which indicates that large portions of the true background are now being included.

2. Stop if the change in threshold between successive recursive steps is smaller than some small value d_1, which indicates that the algorithm is straining at very subtle detail (T_k-$T_{k-1} < d_1$),

3. Stop if the change in threshold between successive recursive steps is larger than some larger value d_2, which indicates that the algorithm is simply subdividing the actual background pixels. (T_k-$T_{k-1}> d_2$).

Once any one of the stopping criteria is met, the recursion is stopped and uses the result from the preceding step.

B. Binarization of an image [5]

It is a process of converting grayscale image to black and white image i.e. binary image. The threshold value that results from above step is used for this conversion. All the pixel values in the image is compared with the threshold, if the value is above threshold then the pixel value is changed to one else the value is changed to zero.

C. Dilation of an image [5]

It is a morphological process which is used to reduce the width of the ridges. It thickens object in a binary image. In this step, dilation is used to thicken the valleys in a binary fingerprint image. This results in erosion of the ridges. A conservative structuring element consisting of four ones arranged in a two-by-two square is used for the valley dilation to achieve some ridge width reduction.

D. Central line thinning of binary image [2]

This iterative step reduces the width of the ridges to one pixel at their central line. This method deletes pixels at the outer boundaries of the ridges, so long as the width is greater than one pixel. First the image is scanned for black pixels. For each pixel, a check is done to determine out of the twenty one rules which scenario is matched and following changes are done. If none of the rules matched then the middle pixel remains unchanged. This process is done continuously till no difference results in two consecutive iterations. Then the diagonal rules are implemented to produce the final thinned image.

E. Removal of unwanted portions from thinned image(using SAGAP) [3]

Short islands at the outer boundaries of the fingerprint image do not represent true ridge structure so these unwanted portions need to be removed. This step begins with detecting terminations in the thinned image. At each termination the ridge is traced one pixel at a time. The ridge is considered as island if it terminates before a maximum trace of continuous twenty pixels and it is deleted otherwise no changes are done.

Now the original image is converted to a negative image by converting the black pixels to white pixels and vice versa. These steps are repeated for the dual image constructed to get the thinned dual image After these steps the resulting image shown in Fig. 2 and Fig. 3 is given as input to the next stage i.e. minutiae extraction.

Figure 2. Original Image thinned

Figure 3. Dual Image Thinned

III. PROPOSED HEURISTICS FOR MINUTIAE EXTRACTION

After the preprocessing of fingerprint image, minutiae points are extracted from the both the thinned image resulted. Some heuristic rules have been developed which are explained in details below:

A. Detection of Ridge end and Bifurcations on the basis of neighborhood pixels:

Each of the two dual images is scanned and for each pixel count the number of neighbors the pixel has. If the count value of a pixel is one, then mark this pixel as ridge end (point ‘A’) [8], as shown in figure below. If the count value corresponds to three, then the pixel is marked as bifurcation (point ‘B’) [8] provided its neighboring pixel does not have count equal to three.

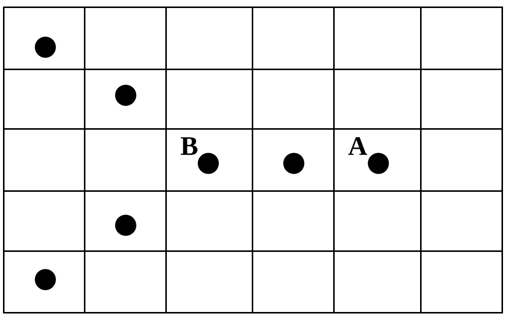

Figure 4. Detection of ridge end and bifurcation point

If its neighboring pixel have count equal to three, as shown in Fig. 5, both point ‘A’ and point ‘B’ have count equals three then point ‘A’ is considered as the bifurcation point. Point ‘A’ has no two neighboring pixels adjacent with each other (there is at least one position gap among the neighboring pixels). So point ‘B’ is not considered. The value of the pixel is changed from zero to 100 if it is a ridge end else it is changed to 200 if it is a bifurcation. After this, the ridge ends at the outer boundaries are discarded for better results.

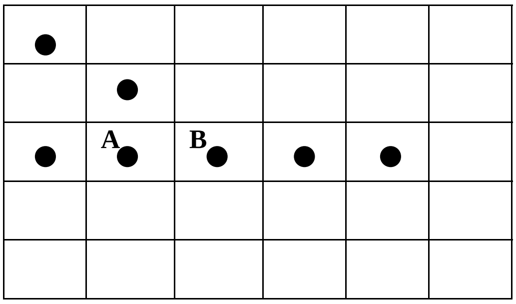

Figure 5. Bifurcation identification

B. Filtering of Detected Ridge ends

As a ridge end is detected, trace in backward direction for at least 20-25 pixels [8]. Suppose point 'A' is a ridge end as shown in figure below. If another ridge end let point 'B', is detected before the trace ends then both the points A and B are not considered as ridge end and the pixel value is changed from 100 to zero again.

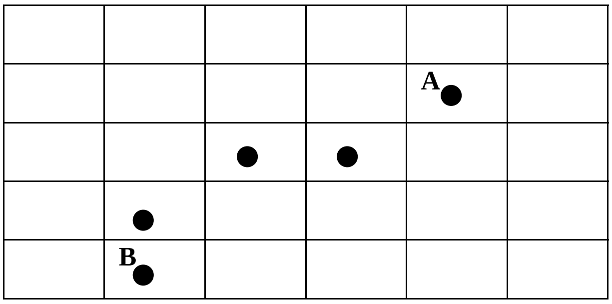

Figure 6. Removal of false ridge end

If another bifurcation point let point 'C', is detected before the trace ends then the ridge end 'A' is discarded but not the bifurcation point 'C' as it may be a spur.

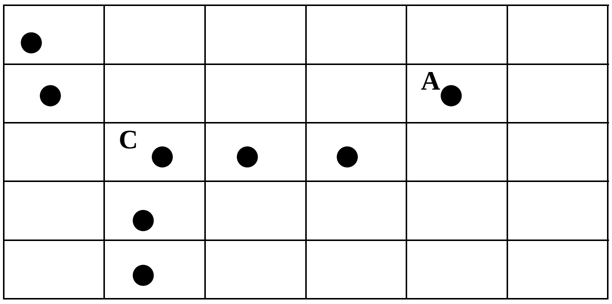

Figure 7. Removal of false ridge end when it is connected to bifurcation point

C. Filtering of the Detected Bifurcations

As a bifurcation point is detected suppose point 'A', trace in all the three directions i.e. its neighboring pixels are visited. While visiting in these directions if the branch ends before the trace is over or the trace length is less than 20-25 pixels then it is a spur and point 'A' is not a bifurcation point. As shown in figure below while tracing along the neighboring pixels point 'B' is the end point of the branch and whose trace length is less than 20-25 pixels. This shows that this branch is a spur. It is a false bifurcation point and the pixel value is changed to zero from 200.

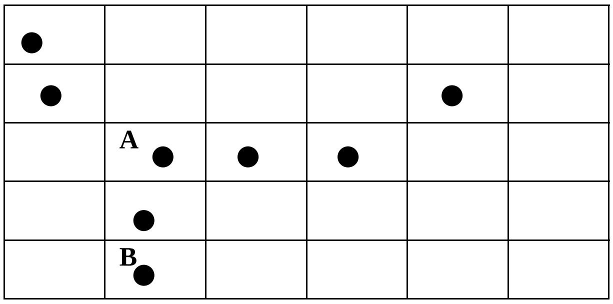

Figure 8. Removal of spurious bifurcation points

D. Removing Bridges

When two parallel ridges are joined bridges are formed as shown in figure.

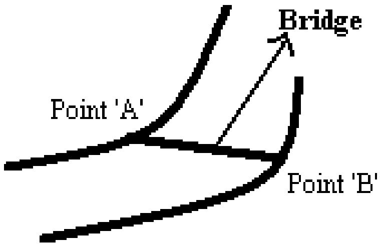

Figure 9. Bridges in a fingerprint image

The Fig. 9 shows that both point ‘A’ and point ‘B’ may be considered as bifurcation. Following the filtering process explained in above step it may so happen that when we trace in all the three directions, each branch trace may be less than 20-25 pixels then these points are not true bifurcation points. But if the trace length is greater than 20-25 pixels then these points can be considered as bifurcation points. But in actual these form the bridge, so in order to eliminate these points, need to differentiate between a true bifurcation and a false bifurcation. A true bifurcation point has one acute angle and two obtuse angles while a bridge has two acute angles and one obtuse angle. This heuristic is used to detect a bridge and then the point is discarded if it forms two acute angles and one obtuse angle.

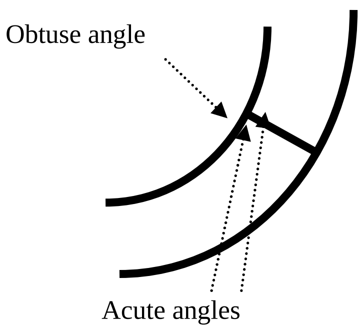

Figure 10. Angles in Bridge

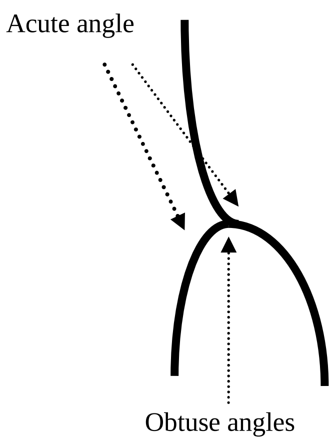

Figure 11. Angles in true bifurcation

E. Comparing with the dual image

The above steps are applied to both the images i.e. the original thinned image and the thinned dual image and find the minutiae points in each image. Then scan the original image sequentially. When a bifurcation point is detected in the original image, the particular coordinate points are searched in the dual image and a 10 X 10 window is constructed around the coordinates. If a ridge end is detected within the window then the bifurcation point in the original image is a true minutiae, else discard the bifurcation point. The same process is repeated for ridge ends in the original image i.e. bifurcation points is searched within the 10 X 10 window in dual image. Finally all the true minutiae points are extracted and stored as a template.

IV. ALGORITHM FOR MINUTIAE MATCHING

A. Method

After minutiae extraction suppose the total number of true minutiae points in the template is 5 and the total number of minutiae in the input image is 8 as shown in tables below. Then we calculate the radial distance between two minutiae points taking one of them as reference point.

Suppose point ‘A’ is the reference point in template and point ‘P’ is the reference point for input. Then the radial distance between point ‘A’ and point ‘B’ is 112 using the formula [8]:

$$\sqrt{(row_m - row_{ref})^2 + (col_m - col_{ref})^2} \quad (1)$$

TABLE I. FOR TEMPLATE

Name	Co ordinate	Radial Distance	Minutiae Type
A	79,102	0	Bifurcation
B	75,214	112	Ridge end
C	64,110	17	Bifurcation
D	157,182	111	Ridge end
E	50255	156	Bifurcation

TABLE II. FOR INPUT IMAGE

Name	Co coordinate	Radial Distance	Minutiae Type
P	75,103	0	Bifurcation
Q	87,90	18	Bifurcation
R	172,146	106	Ridge end
S	108,206	108	Ridge end
T	101,254	153	Bifurcation
U	224,228	194	Ridge end
V	187,52	123	Ridge end
W	231,133	159	Ridge end

Next we pair the minutiae points based on the type of minutiae i.e. bifurcations should be paired with bifurcations of other image and the same is for ridge ends. Suppose the reference point ‘A’ of template image is paired with the reference point ‘P’ of input image.

Then we select two minutiae points and determine angles made between these two points with the reference point as the center. Suppose we choose point ‘B’ and point ‘C’ and determine the angle between these two points with reference point ‘A’ at the center. Similarly this is repeated for all combinations and is stored.

TABLE III. ANGLES MADE BETWEEN TWO MINUTIAE POINTS FOR TEMPLATE

A	B	C	D	E
B	-	60º	46º	9º
C	60º	-	106º	51º
D	46º	106º	-	55º
E	9º	51º	-	-

TABLE IV. ANGLES MADE BETWEEN TWO MINUTIAE POINTS FOR INPUT IMAGE

P	Q	R	S	T	U	V	W
Q	-						
R	71º	-					
S	120º	48º	-				
T	128º	51º	8º	-			
U	87º	16º	32º	40º	-		
V	23º	48º	97º	105º	64º	-	
W	58º	13º	61º	69º	29º	35º	-

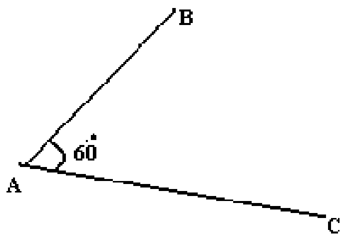

Figure 12. Arrangement of minutiae points on the image

Next we calculate the difference between the radial distances between the paired minutiae points.

TABLE V. RADIAL DISTANCE COMPARISON

Minutiae Points of Template	Radial Distance from Reference Point	Minutiae Points of Input image	Radial Distance from Reference point	Difference in radial distance
A	0	P	0	0
B	112	S	108	4
C	17	Q	18	1
D	111	R	106	5
E	156	T	153	3

Suppose the threshold value for maximum difference between radial distances allowed is 5, then all the above points are matched. So the resulted paired minutiae points based on radial distance comparison are:
A – P, B – S, C – Q, D – R , E – T

Now let us compare the angles made by each point with other points.

TABLE VI. ANGLE COMPARISON

A/P	B/S	C/Q	D/R	E/T
B/S	-			
C/Q	60º/120 º	-		
D/R	46º/48 º	106º/70º	-	
E/T	9º/8 º	51º/128º	55º/56º	-

Suppose the threshold value for angle difference is 5º. Consider the pairs A-P, B-S and C-Q. The angle made between point 'A', 'B' and 'C' is 60º and the angle between point 'P', 'Q' and 'R' is 120º. As these are the paired minutia so the angle difference must not be greater than threshold as shown in figure-11. So this results in false pairing of s B-S and C-Q.
Similarly the angle difference for each pair is calculated.
Angle DAB and angle RPS, difference is 2º.
Angle EAB and angle TPS, difference is 1º.
Angle EAD and angle TPR, difference is 1º.
Finally the correct pairs are: A – P, D – R, B – S, and E – T.

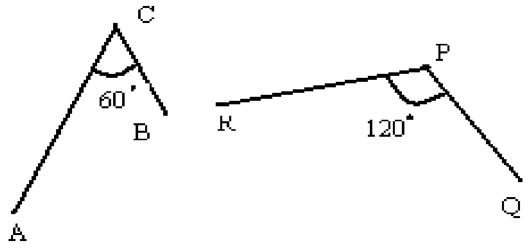

Figure 13. Arrangement of minutiae points on the template image and the input image respectively

The above matched minutiae points can be diagrammatically viewed as follows.

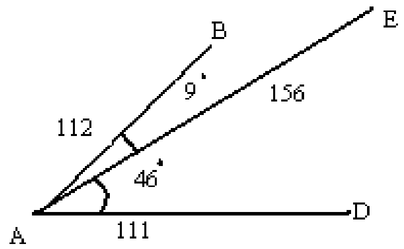

Figure 14. Arrangement of all minutiae points on the template image

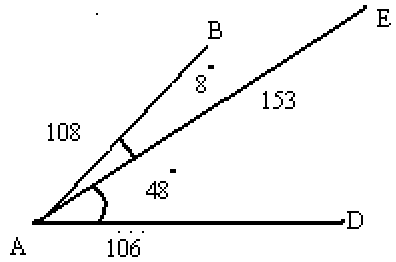

Figure 15. Arrangement of all minutiae points on the input image

These four minutiae points form similar structure in both input and template images. These steps are repeated for each pair of reference minutiae points. The highest number of matching point is the result.

B. Algorithm

1. Repeat step-2 to step-5 until all minutiae points are consumed.
2. Take one minutiae point from template and one from input image as reference and repeat step-3 to step-4 until all minutiae points are consumed.
3. Calculate radial distances for all points from the reference point for both the template and the input image.
4. Pair one minutiae point from template with one from input image if the difference in radial distance if the points ≤ certain threshold value and their type are same.
    a. If the above condition is true, then find the angle between two minutiae of an image made with the reference point. Find the same angle in the other image.

b. If difference between the angles ≤ certain threshold value, then pair minutiae points from template and input image.

5. Count the number of pairs (no repeating pairs) and this represents total number of matching minutiae points for a particular pair of reference points. Store the count value.
6. Finally select the maximum count value. This is the final matching minutiae points.

V. EXPERIMENTAL RESULTS

We have developed a java program according to the explained algorithms and seen the output is exactly what we explained in the theory. Two fingerprint images were taken one as a template image and the other as input image. Then minutiae points were detected and finally two samples were compared with the detected minutiae points.

We found all the matched points form similar structure in both samples. Lines were drawn by joining all matched points among themselves in the java program.

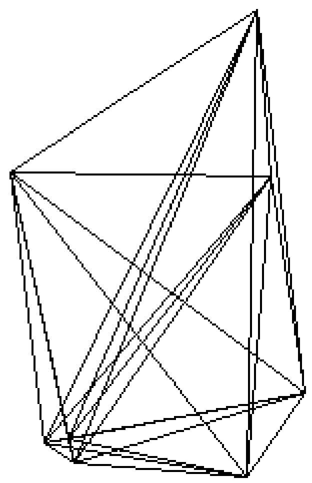

Figure 16. Arrangement of all matched minutiae points on the input image. Total 8 points are matched out of 15.

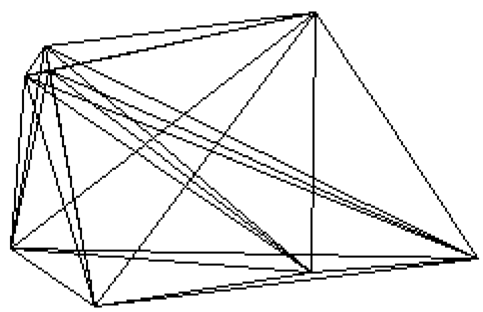

Figure 17. Arrangement of all matched minutiae points on the template image. Total 8 points are matched out of 12.

It can be observed from the above two figures that two samples are matched only when all the points form similar structure. The image in Fig. 17 has different orientation than the image in Fig. 16. This is because of the punching angle of the finger on the biometric device. The vertices in the above figures represent minutia points.

VI. CONCLUSION

We have summarized all the steps involved in a fingerprint matching system. Some of the algorithms are referred from some research papers and some are newly proposed by us. We have tested the algorithm and found it efficient to a great extent. The algorithm is tad slower than some other algorithm but gives better result. In future the time complexity of the algorithm can be reduced. In future more techniques may be identified to detect correct minutiae.

REFERENCES

[1] Oliver Nina, Bryan Morse, and William Barrett "A Recursive Otsu Thresholding Method for Scanned Document Binarization" IEEE, 2010

[2] M. Ahmed,R. Ward, "A rotation invariant rule based thinning algorithm for character recognition" IEEE Transactions on Pattern Analysis and Machine Intelligence, 24, 1672-1678.

[3] P. Kumar, S. R. Giri, G. R. Hegde and K. Verma, "A Novel Algorithm to Extract Connected Components in a Binary Image of Vehicle License Plates", IJECCT 2012, vol. 2 (2), pp. 27-32.

[4] D. Maltoni, D. Maio, A.K.. Jain, S. Prabhakar, Handbook of fingerprint recognition. New York: Springer, 2003.

[5] R.C. Gonzalez and R.E. Woods, Digital Image Processing, 3rd Edition, Prentice Hall, 2002.

[6] Xuefeng Liang and Tetsuo Asano, "Fingerprint Matching Using Minutia Polygons", 18th International Conference on Pattern Recognition ,vol. 01.

[7] F.A. Afsar, M. Arif and M. Hussain, "Fingerprint Identification and Verification System using Minutiae matching", National Conference on Emerging Technologies, 2004.

[8] T. Graig, Diefenderfer, Thesis on "Fingerprint Recognition" at Naval Postgraduate School, Monterey, California, june 2006.

[9] Ravi J, K. B. Raja, Venugopal K. R., "Fingerprint Recognition using Minutiae Score Matching", International Journal of Engineering Science and Technology vol.1,pp. 35-42,2009.

[10] Yuliang He, Jie Tian, Xiping Luo, Tanghui Zhang. "Image enhancement and minutiae matching in fingerprint verification." Elsevier, Pattern Recognition Letters ,vol. 24, pp. 1349–1360, 2003.