Minutiae Point Extraction and Data Creation Methodology

Ishdeep Singla[1] Yadav Gupta[2]

[1]P. G. Student [2]Assistant Professor

[1]Department of Information Technology [2]Department of Computer Science And Engineering

[1,2]Chandigarh University, Mohali, Punjab, 140413

Abstract— In biometrics fingerprint is the widely and oldest technology used as identification. Points are extracted from the fingerprints. It is known that everyone has unique fingerprint, but no mathematical proof has been presented which can present as. It's hard to match some fingerprint patterns with whole world's patterns. In this paper a mathematical proof on the basis of adjacent minutiae points distance calculation has been presented. This calculation shows that every fingerprint is unique. Further by using extracted minutiae a database has been designed.

Keywords -- Minutiae points, Fingerprint database, Biometrics.

I. INTRODUCTION

This paper is representing a method that how database of fingerprint is created. This paper covers Techniques using that techniques a small size database has been created. Moreover searching speed to verify and identify a person has been enhanced by using B-Tree.

II. FINGERPRINT FEATURES

A. Local Features

The fingerprint local features are those attributes that give the minutiae details about the fingerprint pattern [1]. Minutiae further provide various ways that the ridges can be discontinuous. A ridge can suddenly end (termination), or can divide into two ridges (bifurcation) as shown in figure 1. There are 40-100 minutiae point in a good quality image [8]. And in a fingerprint image of 300x300 pixels the distance between two fingerprints vary between 1-113 pixels. With these features and numerical figures local features has become more suitable to compare fingerprints [4]. There are many methods like cross number are available to extract the minutiae points.

B. Fingerprint Classification

It's obvious that with the increase database size complexity and automatic comparison time will also increase. So to reduce the search time and computational complexity, there is a need to classify fingerprint in a precise and consistent manner which will help to reduce search time with less number of comparisons. According to Galton-Henry classification (Galton, 1892 and Henry, 1900) classification, we classify fingerprint images into 5 major classes: plain arch, tented arch, left-loop, right-loop and whorl (a plain and twin loop, respectively).

Arch In whole fingerprint arch covers only 5 % of the portion. These consist of ridges that run majorly in horizontal manner can say from left to right as shown in figure 4. There are two types of arches – Plain arches and Tented arches. Generally, plain arch has no singular points. While tented arch have one core and one delta [2].

Loop Loops cover 60-70 % of whole fingerprint pattern. As the name suggests set of the ridges enters on either side of the fingerprint, bends, touches or crosses the line running from the delta to the core and run back in same direction of the side where the ridge or ridges entered as shown in figure 4.0. Each loop pattern has is one delta and one core. There can be left loop or right loop.

Whorl 25-35 % of fingerprint pattern is covered by whorl. In a whorl, more than one ridges moves through at least one circuit [3]. A whorl pattern always consists of two or more deltas. There are two types of whorl plain whorl and double whorl [5]. A plain whorl is the pattern which consists of some ridges which make or partially make a complete circuit with two deltas. Double loop whorl consists of two separate and distinct loops.

III. FINGERPRINT MATCHING TECHNIQUES

There a lot of techniques for matching a fingerprint. There are three most popular methods for matching fingerprints [1] are described below.

A. Correlation-based matching

In this method one fingerprint image is superimposed on other. The correlation between corresponding pixels is computed for different alignments and on the basis of these correlations and computations decision is made.

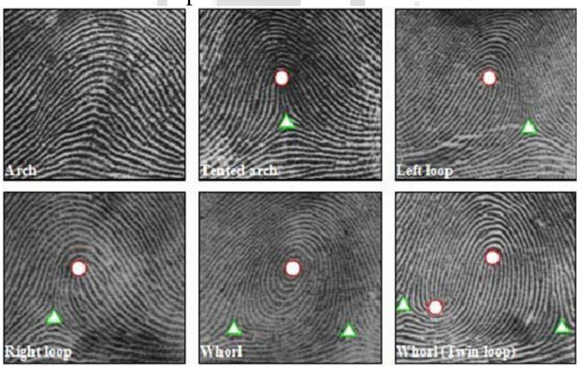

Fig. 1: Fingerprint Classes [5]

B. Pattern-based (or image-based) matching

In pattern based algorithms the basic fingerprint patterns (arch, whorl, and loop) are used to compare fingerprints, between a previously stored template and a candidate fingerprint. This requires that the images be aligned in the same orientation [9]. To do this, the algorithm finds a central point in the fingerprint image and centres on that. In a pattern-based algorithm, the template contains the type, size, and orientation of patterns within the aligned fingerprint image. The candidate fingerprint image is graphically compared with the template to determine the degree to which they match.

C. Minutiae-Based Matching

This is the most popular and widely used technique, being the basis of the fingerprint comparison made by fingerprint examiners. Minutiae are extracted from the two fingerprints and stored as sets of points in the two-dimensional plane [11]. Minutiae-based matching essentially consists of finding the alignment between the template and the input minutiae sets that result in the maximum number of minutiae pairings. In this thesis we have implemented a minutiae based matching technique. This approach has been intensively studied, also is the backbone of the current available fingerprint identification products [10].

IV. MINUTIAE EXTRACTION

A minutiae point matching is the best approach for the matching of fingerprints. The work of minutiae extraction includes some important steps that are Ridge Thinning [13], Minutiae Marking, False Minutiae Removal and Minutiae Representation.

A. Ridge Thinning

The main aim of this step is to convert the redundant pixels of ridge into one pixel wide. This will be very helpful in finding minutiae points and to implement minutiae point algorithm [8]. In Matlab there has one very popular morphological thinning function to perform this task [13].
bwmorph (binaryImage,’thin’,Inf)
The thinned image is then filtered, again using MATLAB’s three morphological functions to remove some H breaks, isolated points and spikes (See Figure 2)
bwmorph(binaryImage,’hbreak’,k) → For H breaks
bwmorph(binaryImage,’clean’,k) → For isolated points
bwmorph(binaryImage,’spur’,k) → Spikes

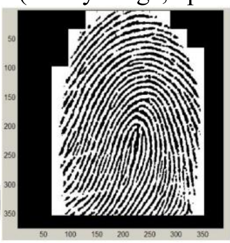
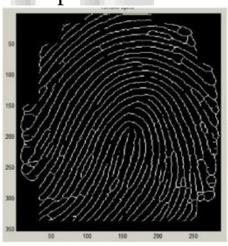

Fig. 2(a) Binarized image **fig. 2(b): Thinning image**

V. MINUTIAE MARKING

The name of this algorithm is Crossing Number (CN). It is implemented thinned image. Iteratively a 3x3 pixels wide picture is selected from thinned image then check that if the central pixel is a ridge branch and the central pixel is 1 and has exactly three neighbours of 1’s, then its **bifurcation** (See Figure 3).

0	1	0
0	1	0
1	0	1

Fig. 3: Ridge Bifurcation

If there one central 1 with exactly one 1in its neighbourhood, then it’s a **ridge ending**.(See Figure 4).

0	0	0
0	1	0
0	0	1

Fig. 4: Ridge Termination

VI. CALCULATIONS FOR UNIFICATION OF THE FINGERPRINT

This mathematical calculation is based upon minutiae points. In a good quality image there are 40-100 minutiae points [6]. Now the whole methodology is described in following steps.

– Firstly apply the process like cross-number [9] method on good quality fingerprint[12] image (fig. 5) and find all minutiae points.

Fig. 5: Fingerprint

– Process the minutiae point image and resize the image to 300x300 pixels (fig. 6)

Fig. 6: Minutiae Pattern (300x300 Pixels)

– From the output image (fig. 6) compute the gap (distance) between two adjacent minutiae points and start computing from top to bottom and from left to right (fig 7). Then it can be noticed that the distance between two adjacent minutiae points vary from 1 Pixel to 113 Pixels (1mm to 10mm).

Fig. 7: Calculating Distance of Two Adjacent Minutiae Points

– As it has been already discussed that In a good quality fingerprint image there can be 40-100 minutiae points [7] and distance vary between 1-113 Pixels. But to make calculations easy the median of both quantities have been taken. That is 70 minutiae points and 57 pixels distance.
– By applying simple mathematics the total number of patters can be calculated as $57^{\wedge 70}$ and it can be noticed very clearly that this count is much more than the whole universe’s people’s count. In simple words it can be said that the fingerprint can have $57^{\wedge 70}$ different patterns. Example: As the whole

world's population 7.5 Billion approximately. Which is extremely less than 57^{70}.

VII. Creation of Small Size Database

The Suggested method varies according to size of company that is number of people in a company. In this thesis three different categorizations have been presented.

A. For Small Size Company having less number of personal (1 to 1000)

Method for the creation of the database

- Take input minutiae to the system and track it's all the minutiae points and then start drawing lines top to down and left to right starting from first minutiae point to second so on till end. (Figure 8 showing then minutiae pattern and figure 9 showing minutiae track pattern)
- Now measure the distances between two adjacent minutiae in millimeters. Measure lengths from above pattern in mm are 0.5, 0.6, 0.7, 0.3, 0.8,1.1,0.9,0.5, 1.0, 0.5,0.6
- Now write measure length in 05 06 07 03 08 11 09 05 10 05 06 called "trace" (say).

1). Sum

Fig. 8: Trace Packet

- Apply B tree algorithm on Sum.

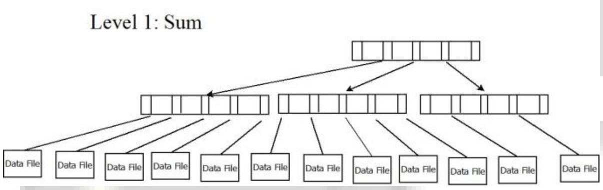

Fig. 9: Single Stage B Tree

Single Stage B Tree:

The maximum distance between two minutiae points can lie between 1mm to 10mm. As already discussed number of minutiae points lies between 40 - 70. To make calculation easy the averages can be taken that is average distance between minutiae points is 5mm and the average number of minutiae points is 55. Hence it is clear that the sum the 55 minutiae points will lie between 45x1 = 45mm to 45x 5=225mm. Means now to create B- Tree keys are 45, 46, 47, 48, 49, 50,. . . . . , 225.

B. For Medium Size company have personal size lies between Thousand – million

Method for the creation of the database:

Take input minutiae to the system and track it's all the minutiae points and then start drawing lines top to down and left to right starting from first minutiae point to second so on till end. (Figure 8 showing then minutiae pattern and figure 9 showing minutiae track pattern)

Now measure the distances between two adjacent minutiae in millimeters. Measure lengths from above pattern in mm are 0.5, 0.6, 0.7, 0.3, 0.8,1.1, 0.9,0.5, 1.0, 0.5,0.6 Now write

1.)Even 's Sum	2.)Odd's Sum	3.)Multiple of 3's Sum

Fig. 10: Trace Packet

- Apply B+ tree algorithm on Evens' Sum by considering as first level, Odd's Sum by considering as second level and Multiple of 3's sum by considering as third level B+ Tree as shown in Figure 11.
- *First Level B+ Tree*
- As the average minimum distance between two adjacent minutiae points can be 5mm and maximum average distance is 10mm. Hence it is clear that the sum the 70 minutiae points will lie between 45x1 =45mm to 45x 5=225mm.
- But for the construction of B+ Tree half minutiae numbers are considered so range will reduce to 45mm to 113mm.
- *Second Level B+ Tree*
- Now as shown in figure, the average minimum distance between two adjacent minutiae points can be 5mm and maximum average distance is 10mm. Hence it is clear that the sum the 70 minutiae points will lie between 45x1 =45mm to 45x 5=225mm.
- But for the construction of B+ Tree half minutiae numbers are considered so range will reduce to 45mm to 113mm.
- *Third Level B+ Tree*
- As the average minimum distance between two adjacent minutiae points can be 5mm and maximum average distance is 10mm. Hence it is clear that the sum the 45 minutiae points will lie between 45x1 =45mm to 45x 5=225mm.
- But for the construction of B+ Tree will include 1/3 of total minutiae numbers and reduced rage will be 15mm to 75mm.

C. For Large Size company have personal size between millions- billions

The size of the trace can be further increase and the level of the search tree can also be increase according to population.

VIII. Implement and Testing

After finding the Minutiae points and the distance of the each adjacent Minutiae points' pair now this code is used to implement B-Tree which is designed in C Language

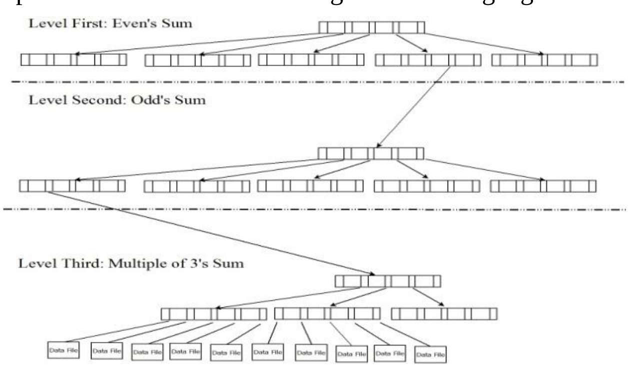

Fig. 11: Three Stage Leveling of B+

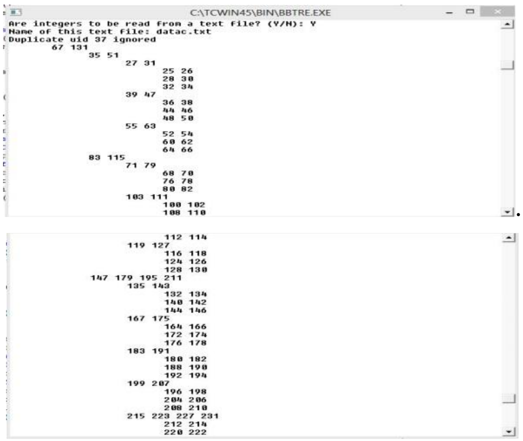

Fig. 12: Snapshot of Input Window

Testing With Different Test is used to implement B-Tree which is designed in C Language.

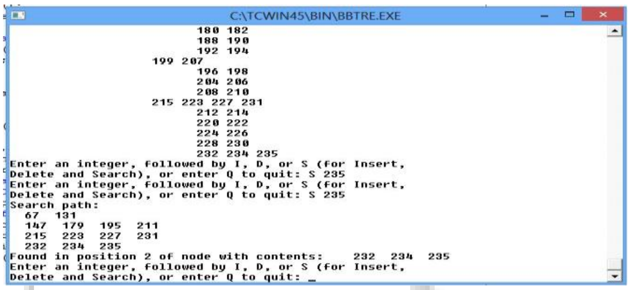

Fig. 13: Snapshot of Testing & Result

A. Testing and Implementations of B-Tree

Testing test is present for First Level Searching technique only because this same algorithm has to be repeated according to the problem. The input is taken as text file contain the all possible combination of sum of distances between adjacent minutiae points as shown in fig. 12.

This testing and Output belong to First Level Technique, But this same algorithm can be repeated for Second and Third level Searching Techniques. After applying this technique just there is requirement of applying *Linear or Binary* searching algorithm on very small size data. Searching speed of this algorithm is in nanoseconds.

IX. Conclusions

The database creation from a fingerprint, solve the problem of identification and verification of a person. Moreover searching speed is also increased due to the use of B Tree algorithms. By using these technique there is no need of carrying identity and swap cards.

References

[1] Ardovini, L. Cinque, F. Della Rocca, and E. Sangineto. A semi-automatic approach to photo identification of wild elephants. Pattern Recognition and Image Analysis, pages 225–232, 2007.

[2] B.M. Mehtre and B. Chatterjee,“Segmentation of fingerprint images–a composite method”, Pattern Recognition, 22(4):381–385, 1989.

[3] Cappelli, R.; Ferrara, M.; Maltoni, D. Minutia cylinder-code: A new representation and matching technique for fingerprint recognition. IEEE Trans. Pattern. Anal. Mach. Intell. 2010, 32, 2128–2141.

[4] Cappelli, R.; Maio, D.; Maltoni, D.; Wayman, J.L.; Jain, A.K. Performance evaluation of fingerprint verification systems. IEEE Trans. Pattern. Anal. Mach. Intell. 2006, 28, 3–18.

[5] Jain, A.K.; Feng, J. Latent fingerprint matching. IEEE Trans. Pattern. Anal. Mach. Intell. 2011,33, 88–100.

[6] Jain, A.K.; Feng, J.; Nandakumar, K. Fingerprint matching. Computer 2010, 43, 36–44.

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

[8] Prabhakar, S.; Ivanisov, A.; Jain, A.K. Biometric recognition: Sensor characteristics and imagequality. IEEE Instrum. Meas. Mag. 2011, 14, 10–16.

[9] S. Belongie, J. Malik, and J. Puzicha. Shape matching and object recognition using shape contexts. IEEE Trans. on Pattern Analysis and Machine Intelligence, pages 509–522, 2002.

[10] T. Burghardt and N. Campbell. Individual animal identification using visual biometrics on deformable coat patterns. In Proceedings of the 5th International Conference on Computer Vision Systems, Berlin,Germany.. Accessed, volume 9. Citeseer, 2007.

[11] Tan, X.; Bhanu, B. Fingerprint matching by genetic algorithms. Pattern Recogn. 2006, 39, 465–477.

[12] Y. Bulatov, S. Jambawalikar, P. Kumar, and S. Sethia. “Hand recognition using geometric classifiers”, ICBA’04, Hong Kong, China, pages 753–759, July 2004. Z. Sun, T. Tan, Y. Wang, and S.Z. Li, "Ordinal palmprint representation for personal identification", Proc. EEE Computer Vision and Pattern Recognition (CVPR), vol. 1, pp. 279-284, 2005.

[13] Zhao, Q.; Zhang, D.; Zhanga, L.; Luoa, N. High resolution partial fingerprint alignment using pore-valley descriptors. Pattern Recogn. 2010, 43, 1050–1061.