An Efficient Fingerprint Indexing Scheme

Arjun Reddy, Umarani Jayaraman, Vandana Dixit Kaushik and P. Gupta

Abstract This paper proposes an efficient geometric-based indexing scheme for fingerprints. Unlike other geometric-based indexing schemes, the proposed indexing scheme reduces both memory and computational costs. It has been tested on IITK database containing 2,120 fingerprints of 530 subjects. Correct Recognition Rate is found to be 86.79 % at top 10 best matches. Experiments prove its superiority against well-known geometric-based indexing schemes.

Keywords Biometrics · Fingerprint indexing · Core point · Minutiae

1 Introduction

Fingerprint recognition system is used to recognize the identity of a subject. Identification can be done by searching all images in the database (henceforth termed as *models*) against a image (henceforth termed as *query*). To make an efficient process, there is a need of an efficient indexing technique. A fingerprint has the following characteristics:

A. Reddy (✉) · U. Jayaraman · P. Gupta
Indian Institute of Technology, Kanpur, India
e-mail: areddy@cse.iitk.ac.in

U. Jayaraman
e-mail: umarani@cse.iitk.ac.in

P. Gupta
e-mail: pg@cse.iitk.ac.in

V. D. Kaushik
Harcourt Butler Technological Institute, Kanpur, India
e-mail: vandanadixitk@yahoo.com

B. V. Babu et al. (eds.), *Proceedings of the Second International Conference on Soft Computing for Problem Solving (SocProS 2012), December 28–30, 2012*, Advances in Intelligent Systems and Computing 236, DOI: 10.1007/978-81-322-1602-5_77, © Springer India 2014

- Number of minutiae extracted from a fingerprint of a subject at any two time instants may not be same.
- There are too many minutiae in a fingerprint; some may be false.
- Number of minutiae of any two fingerprints may not be same.
- There may be partial occlusion in a fingerprint of a subject and it may overlap with some other subjects that are not present in the database.
- A query fingerprint may be rotated and translated with respect to the corresponding model fingerprints in the database.

Most of the available fingerprint indexing schemes can be classified on the basis of the features such as singular points [1], directional field [2], local ridge-line orientations [3], orientation image [4], minutiae [5], minutiae descriptor, multiple features, and SIFT features. Since most matching algorithms use minutiae, the use of minutiae is especially beneficial. These schemes derive robust geometric features from triplets of minutiae and use hashing techniques to perform the search.

A prominent geometric-based indexing technique for fingerprints is proposed by Germain et al. [6] in which geometric features from triplets are used with the help of Fast Look up Algorithm for String Homology (FLASH) hashing technique. It does clustering using transformation parameter where all the fingerprints in that bin represent a hypothesis match between the three points in the query fingerprint and those in the fingerprints of database. The best coordinate transformation that matches query triplet and model triplet is calculated with the information. The transformation should be such that squared distance between the points of query triplet and model triplet is minimum. This transformation parameter reduces false matches.

The scheme in [7] uses geometric features from minutiae triplets where the triplet features are maximum length of three sides, median and minimum angles, triangle handedness, type, direction and ridge count minutiae density. Since triangles are formed using all possible minutiae, this increases both memory and computational cost. A fast and robust projective matching for fingerprints has been proposed in [8]. It performs a fast match using a Geometric Hashing [9] which needs large computational time and memory.

Bebis et al. [10] have used the geometric features from Delaunay triangles formed on the minutiae for indexing the fingerprints, instead of all possible combination of triangles. It can be shown that for a given set of minutiae, the Delaunay triangulation produces linear number of triangles. This compares favorably to the number of all possible combinations of triangles/ bases pairs considered in approaches [6–8]. However, the major issue with Delaunay triangulation is that it is more sensitive to noise and distortion. For example, if some minutiae are missed or added (spurious minutiae), the structure of Delaunay triangulation gets affected.

This paper presents an efficient indexing scheme which uses geometric information from triangles formed on minutiae to index model fingerprints. It assumes that the uncertainty of feature locations associated with minutiae feature and shear does not affect the angles of a triangle arbitrarily. Triangles are invariant to translation and rotation. It reduces the number of possible triangles by taking minutiae to form triangles within the specified region R from its core point C and is inserted exactly once into a hash table. So it effectively removes the use of all triangles/bases pairs used in [6–8] reducing memory and computational complexity.

The paper is organized as follows. Next section discusses feature extraction technique from a fingerprint image. Section 3 discusses the proposed indexing scheme. Experimental results are analyzed in the next section. Conclusions are given in the last section.

2 Feature Extraction

Feature extraction is a series of steps such as minutiae detection and core point detection [11]. Let $M = \{m_1, m_2, \ldots, m_o\}$ be the detected minutiae from each model fingerprint image. Each minutia m_i is a 4-tuple $(x_{mi}, y_{mi}, \alpha_{mi}, T_{mi})$ which denotes their coordinates, direction, and type. Let $C = (x_c, y_c, \alpha_c)$ be the core point, detected through [11] where (x_c, y_c) denote the coordinates and α_c is the direction of the core point. The proposed indexing scheme overcomes some of the issues and constraints in [6, 7] by considering small number of possible triangles instead of all possible triangles. In a model fingerprint, it considers the minutiae to form triangles within the specific range R from its core point C reducing computational and memory cost. Further, it introduces some additional features to reduce false correspondences. The triplet features used in the proposed indexing scheme are:

1. Sides of the triangles s_1, s_2, s_3: Sides of the triangle is considered in certain order as the longest side, the medium side and the smaller side. If any two or three sides are similar, this system does not consider them because this type of triplets is negligible in number and their exclusion does not affect the results.
2. Regions of vertices r_1, r_2, r_3: Considering core point C as the center, concentric circles are drawn with radius $r, 2r, 3r, 4r$... till the circle's outer line is completely outside fingerprint boundary. The optimal value of r is found out with the help of experiments. Let p_1, p_2, p_3 be three vertices of triplet and r_1, r_2, r_3 be their respective regions of vertices and d_1, d_2, d_3 be their respective distances from core point C. Then the value of r_i is given by $r_i = \frac{di}{r} + 1, i = 1, 2, 3$.
3. Triangle type λ: Each minutia is either termination or bifurcation. If $\lambda_1, \lambda_2, \lambda_3$ are three vertices to indicate whether they are termination or bifurcation point, then λ for the triplet can be used as one more attribute for indexing component. Since λ_1, λ_2 and λ_3 can have values either 0 or 1, based on termination or bifurcation, λ can be any value between 0 and 7 and is calculated by $\lambda = 4\lambda_1 + 2\lambda_2 + \lambda_3$.
4. Orientation φ : Sometime all the above features may be same for triplets of two triangles. They can be differentiated by their orientation which can be calculated using cross-product between the longest side and the medium side. Orientation has two values, $+1$ or -1.

3 Proposed Indexing Scheme

The proposed indexing technique consists of two stages known as indexing and searching. During indexing, the model fingerprints in the database are indexed into a hash table. For any new fingerprint image, it can be added into a hash table without affecting the performance of the searching algorithm and without modifying the existing hash table. During searching, it recognizes the query fingerprint by searching only indexed hash bins.

3.1 Indexing

For each model fingerprint consisting of core point C and minutiae, concentric circles are drawn with radius r, $2r$, $3r$, $4r$... till the circle's outer line is completely outside fingerprint boundary. The value of r is found experimentally. All possible triangles within various circles are determined. The index $I = (s_1, s_2, s_3, r_1, r_2, r_3\lambda, \varphi)$ is generated from each triangle to select an appropriate bin in hash table. At this bin, model fingerprint ID is added. At the time of indexing, it keeps track of number of triangles generated from each model fingerprint. Let T_i be the total number of triangles generated from a model M_i used for indexing. Steps for indexing are given in Algorithm 1.

```
Algorithm 1: Indexing
For each model fingerprint $M_i$ do the following
1. Generate all possible triangles for minutiae of fingerprint.
2. Consider only those triangles whose all sides are in the specifies range $R$.
3. Store numbers of triangles considered in indexing and store it in $T_i$ where $T_i$ indicates total number of triangles of fingerprint $M_i$ considered for indexing.
4. For each triplet do the following
   a) Generate index $I = (s_1, s_2, s_3, r_1, r_2, r_3, \lambda, \varphi)$.
   b) Access appropriate bin of the hash table.
   c) Add model fingerprint ID of the image to this bin.
```

```
Algorithm 2: Searching
For the given query $Q$ do the following
1. Generate all possible triangles from the minutiae of the query image.
2. Consider only those triangles whose all sides are in the specified range $R$. Let there be $T$ such triangles.
3. For each triangle, do the following
   a) Generate index $I = (s_1, s_2, s_3, r_1, r_2, r_3, \lambda, \varphi)$.
   b) Access appropriate bin of the hash table.
   c) Increment count $C_i$ for all model fingerprints whose ID is in that bin.
   d) Calculate $S_i$, score of image $M_i$, by formula $S_i = \frac{Ci}{Ti}$.
   e) Obtain largest $t$ $S_i$'s that are to be considered for critical search.
```

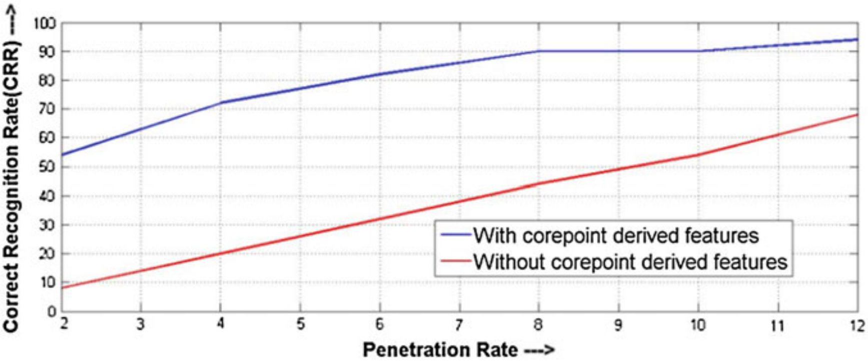

Fig. 1 Performance graph: effect of core point generated features

3.2 *Searching*

For a given query fingerprint, all possible triangles are generated using minutiae similar to indexing stage. Then only those triangles are considered whose all sides of triangle fall within the range R. Let there be T such triangles. From these triangles, an index $I = (s_1, s_2, s_3, r_1, r_2, r_3, \lambda, \varphi)$ is generated to find an appropriate bin of hash table and to count each model fingerprint ID in that bin C_i. Same process is repeated for remaining query's triangles. Finally, score S_i is calculated by $S_i = \frac{C_i}{T_i}, i = 1, 2, \ldots, N$where N denotes the number of fingerprints in the database. Largest t S_i's which are considered for top t best matches are used for critical search to find out the exact match. Steps for searching are given in Algorithm 2.

4 Experimental Results

The proposed indexing scheme has been tested on IITK fingerprint database of 2,120 images acquired from 530 subjects. Every person has given four fingerprints of the same finger, at different instant of times. Four datasets have been created to carry out various experiments. DB1 contains 2120 fingerprints, DB2 has 1,336 fingerprints while DB3 and DB4 contains 668 fingerprints and 200 fingerprints respectively. In all these datasets, three impressions of each finger are used for indexing and remaining one impression is used for searching.

Figure 1 shows the performance of the proposed indexing scheme with respect to core point and database sizes. DB4 dataset is used for performing this experiment. The *CRR* is very high when core point is considered to obtain additional features from triangles. This *CRR*, when core point is used, is close to the *CRR* of the existing schemes like [6] and [7] . But the proposed scheme has achieved this performance without using transformation parameter cluster and imposing any geometrical constraints. Experiments have also been conducted on all datasets DB1, DB2, DB3, DB4. It is found that there is no drastic difference in *CRR* for different datasets and as database size is increasing, *CRR* is coming down for a given *Penetration Rate*, slightly.

5 Conclusion

This paper has proposed a fingerprint-based indexing scheme which uses core point. This has helped to overcome various issues and constraints of well-known schemes. It has been tested on IITK database containing 2,120 fingerprints of 530 subjects. Accuracy is found to be 86.79 % at top 10 best matches. This accuracy has been achieved without using clustering based on transformation parameter and without imposing geometrical constraints as in existing schemes. However, it may not perform well when a fingerprint does not contain any core point.

References

1. Liu, T., Zhu, G., Zhang, C., Hao, P.: Fingerprint indexing based on singular points. In: Proceedings of International Conference on Image Processing, pp. 293–296 (2005)
2. Cappelli, R., Maio, D., Maltoni, D.: Indexing fingerprint databases for efficient 1:N matching. In: Proceedings of International Conference on Control, Automation, Robotics and Vision (2000)
3. Alessandra, L., Dario, M., Davide, M.: Continuous versus exclusive classification for fingerprint retrieval. Pattern Recogn. Lett. **18**(10), 1027–1034 (1997)
4. Li, J., Yau, W.Y., Wang, H.: Fingerprint indexing based on symmetrical measurement In: Proceedings of International Conference on Pattern Recognition, pp. 1038–1041 (2006)
5. Liang, X., Asano, T., Bishnu, A.: Distorted fingerprint indexing using minutiae detail and delaunay triangle. In: Proceedings of the 3rd International Symposium on Voronoi Diagrams in Science and, Engineering, pp. 217–223 (2006)
6. Germain, R.S., Califano, A., Colville, S.: Fingerprint matching using transformation parameter clustering. IEEE Comput. Sci. Eng. **4**(4), 42–49 (1997)
7. Bhanu, Bir, Tan, Xuejun: Fingerprint indexing based on novel features of minutiae triplets. IEEE Trans. Pattern Anal. Mach. Intell. **25**(5), 616–622 (2003)
8. Boro, R., Roy, S.D.: Fast and robust projective matching for fingerprints using geometric hashing. In: Proceedings of the 4th Indian Conference on Computer Vision, Graphics and Image Processing, pp. 681–686 (2004)
9. Haim, J.: Wolfson and Isidore Rigoutsos. Geometric Hashing: An overview, IEEE Computational Science and Engineering **4**(4), 10–21 (1997)
10. Bebis, G., Deaconu, T., Georgiopoulos, M.: Fingerprint identification using delaunay triangulation. In: Proceeding of IEEE International Conference on Intelligence, Information, and Systems, pp. 452–459 (1999)
11. Jain, A., Prabhakar, S., Hong, L.: A multichannel approach to fingerprint classification. IEEE Trans. Pattern Anal. Mach. Intell. **21**(4), 348–359 (1999)