Minutiae Based Geometric Hashing for Fingerprint Database

Umarani J. [1], Viswanathan J. [2], Aman K. Gupta [1], and Phalguni Gupta [1]

[1] Department of Computer Science and Engineering, Indian Institute of Technology Kanpur, Kanpur-208 016, India.
[2] ARICENT Technologies, Gurgaon, India

E-mail: {umarani,pg}@cse.iitk.ac.in, informviswa@gmail.com, amankg@iitk.ac.in

Abstract. This paper proposes an efficient indexing technique for fingerprint database using minutiae based geometric hashing. A fixed length feature vector built from each minutia, known as Minutia Binary Pattern, has been suggested for the accurate match at the time of searching. Unlike existing geometric based indexing techniques, the proposed technique inserts each minutia along with the feature vector exactly once into a hash table. As a result, it reduces both computational and memory costs. Since minutiae of all fingerprint images in the database are found to be well distributed into the hash table, no rehashing is required. Experiments over FVC 2004 datasets prove the superiority of the proposed indexing technique against well known geometric based indexing techniques using fingerprints.

Keywords: Fingerprints, Indexing, Identification, Geometric Based Hashing, Minutia Binary Pattern.

1 Introduction

Fingerprint recognition system is used to identify a subject (human) from a large biometric database. One can do this task by searching a query image (henceforth termed as query fingerprint) against all images in the database (henceforth termed as model fingerprints) of the subjects. The process to retrieve each model fingerprint from the database and to compare it against the query fingerprint for a match is computationally inefficient. To make the process computationally efficient there is a need of an efficient indexing technique. There exist few techniques to index the fingerprint database. These techniques can be classified on the basis of the features such as singular points, directional field, local ridge-line orientations, orientation image, minutiae [1], minutiae descriptor [2], multiple features [3] and SIFT features [4]. But most matching algorithms are based on minutiae, so use of minutiae to index the fingerprint database is beneficial in many respects. In [5], geometric features from triplets are used with the help of FLASH (Fast Look up Algorithm for String Homology) hashing technique. Triplets are formed by 3 minutiae and triangles are formed using all possible combinations of minutiae. In this technique, angles formed by each triangle are used as index. The technique proposed in [6] also 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, there are large number of possible triangles which increase both memory and computational cost. A fast and robust projective matching for fingerprints has been proposed in [7]. It performs a fast match using a Geometric Hashing. However, geometric hashing may not be suitable because of its computational time and memory requirement which uses $n \times^{n} C_2$ bases pairs. In [8], geometric features from Delaunay triangles formed on the minutiae are used for indexing the fingerprints, instead of all possible combination of triangles. 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 seriously affected. Hence, this technique requires high quality of fingerprint images. This paper presents an efficient indexing technique which uses invariant spatial (distance) and directional (angle) information to index each minutia into a hash table. The technique is invariant to translation and rotation. It can be conveniently represented by a hash table containing spatial and directional information. Each minutia can be uniquely identified by its distance and angle information from its core point C and is inserted exactly once into a hash table. As a result, it eliminates multiple entries of same minutia into a hash table. So the proposed technique effectively removes the use of all possible triangles used in [6] and bases pairs in [7]; thus it reduces memory and computational complexity.

The rest of the paper is organized as follows. Feature extraction from a fingerprint image is discussed in the next section. Section 3 is discusses the proposed indexing technique while searching strategy is discussed in next section. Experimental results are analyzed in Section 5. Conclusions are given in the last section.

2 Feature Extraction

Features are extracted with the help of a series of steps such as core point detection, minutiae detection and minutia binary pattern construction. Singular point detection is defined as the point with maximum curvature of convex ridges. The singular points (SPs) are the discontinuities in the directional field [9] of a fingerprint image. There exist two types of SPs, core and delta. A core is the uppermost point of the innermost curving ridge and a delta is a point where three ridge flows meet. Many images can have more than one core point. In that case, the top most core point is used. If a core point does not exist in an image, we use high curvature point as reference point. In our experiment, core point is marked manually to minimize the error rate. Let $C = (x_c, y_c, \alpha_c)$ be the core point, detected manually where (x_c, y_c) denote the coordinates and α_c is the direction of the core point. Next, minutiae points are detected by tracing the ridges. Ridges are represented by a list of points. Two types of ridge points which are considered in our study are bifurcation and termination. Bifurcation is associated with three ridges while termination is the ridge end point. Let $M = \{m^1, m^2, \ldots, m^\circ\}$ be the detected minutiae from each model fingerprint image. Each minutia m^i is a *4*-tuple

$(x_m^i, y_m^i, \alpha_m^i, T_m^i)$ denoting their coordinates, direction and type (bifurcation or ridge end point), respectively. A representation of feature vector built from each minutia, called Minutia Binary Pattern (MBP), is constructed for the accurate match at the time of searching. Representation of MBP associates a local structure to each minutia (m^i) and is represented by *3 X 3* neighborhood of square window *W* whose base is centered at minutiae location (x_m^i, y_m^i) and it is aligned according to the minutiae direction α_m^i. Each cell is a small square window and can be uniquely identified by two indices *(a, b)*, both *a* and *b* lying between 1 and 3, denote its position in the square window. For each minutia, m^i, binary pattern is constructed by accumulating the binary values in the cells *(a,b)* of *3 X 3* square window *W* associated with m^i starting from its direction α_m^i Thus,

$$MBP = \{ m^i \in M; W(a,b); 1 \leq (a, b) \leq 3 \}$$

where *W(a,b)* is a binary value associated with cell *(a,b)* of *3 X 3* square window *W* and *M* is model fingerprint. An advantage of the MBP representation is that the value of each cell can be stored as a bit with a negligible loss of accuracy. Such a bit-based representation is particularly well suited for accurate match at the time of searching. Also, the computation of fixed length MBP value is very simple, fast and reduces storage cost.

3 Indexing

A query fingerprint may be translated or rotated relative to its respective model fingerprint in the database. Invariant spatial (distance) and directional (angle) information can be used to handle translation and rotation that are present in a fingerprint image. The proposed indexing technique is built using invariant distance and angle from core point *C* of each minutia of a model fingerprint. The technique encodes spatial (distance) and directional (angle) relationship between core point and each minutia which can be conveniently represented as a *2-D* hash table whose rows and columns are related to the spatial (distance) and directional (angle) information respectively. In order to get the two indexing elements (spatial and directional) associated with each minutia, a convex hull $Conv_{Hull}(M, \Omega)$ is obtained by considering all minutiae in *M* by adding Ω offset pixels from the original convex hull. Then a circle with center at (x_c,y_c) and radius *R* is drawn where *R* is the farthest point lying on the convex hull and (x_c,y_c) are the coordinates of the core point. This circle is divided into several sectors, each having 5° of difference. Let N_M and N_S be the number of minutiae and number of sectors of a modal fingerprint image. Note that in a fingerprint image, number of sectors is fixed while number of minutiae is varied. For each minutia mi lying in a sector S^j, the Euclidean distance, $D^i(m^i,C)$, between m^i and the core point C, is calculated as

$$INT(D^i(m^i,C)) = \sqrt{(x_m^i - x_c)^2 + (y_m^i - y_c)^2}$$

Each minutia m^i can be uniquely identified by its distance $D^i(m^i,C)$ and sector number

$S^j(m^i,C)$ from the core point C. $D^i(m^i,C)$ is the spatial contribution of minutia m^i from the core point C while $S^j(m^i,C)$ is the directional contribution of minutia m^i that lies in sector S^j from C. These two invariant information are used as indexing element to insert each minutiae of a model fingerprint along with $(\boldsymbol{M}_{id}^{i}, \boldsymbol{MBP}^{i}, \boldsymbol{\alpha}^{i})$ in hash table H where $\boldsymbol{M}_{id}^{i}$, $\boldsymbol{MBP}^{i}$ and α^i are the model fingerprint identity, minutiae binary pattern and direction of a minutia m^i respectively.

4 Searching`

During *searching*, index generated by each query minutia is used to map the same index in the hash table and vote for all (M_{id},MBP) pairs under this index I. It can be noted that due to noise present in the fingerprint images, the minutiae of the different images of the same model may be shifted or missed. In such cases, to improve the recognition performance, it is considered the minutiae not only from its mapped bin but also from its nearest bins of size k X k. For each query fingerprint minutia vote is casted and model fingerprints are sorted based on their votes and the top t models are considered as best matches.

5 Experimental Results

Most of the well known techniques for fingerprint indexing have been evaluated on FVC 2004 dataset [11]. In our experiment, DB2 and DB3 are used to evaluate the performance of the proposed indexing technique.

- FVC 2004 DB2: Second FVC 2004 dataset contains 800 fingerprints from 100 fingers (8 impressions per finger) captured using the optical sensor ”U.are.U 4000” by Digital Persona and size of image is 328 x 364 at 500 dpi.
- FVC 2004 DB3: Third FVC 2004 dataset contains 800 fingerprints from 100 fingers (8 impressions per finger) captured using thermal sweeping sensor ”FingerChip FCD4B14CB” by Atmel and size of image is 300 × 480 at 512 dpi.

To determine the performance of the proposed indexing technique, two measures, namely, *Hit Rate* and *Penetration Rate* are used. A query is regarded as *hit* if the true fingerprint is contained in the candidate list. The *Hit Rate* is the ratio of the correct queries to all queries. The *Penetration Rate* is the ratio of the size of the candidate list to the size of the whole database. Minutiae obtained from all model fingerprints are indexed using a single hash table. At the time of *searching*, it uses the indexed hash table to retrieve the set of model fingerprint identities M_{id} that are similar to the query fingerprint image. Each minutia m^j of the query fingerprint image is mapped to the hash table. Let it be mapped to the bin b of the hash table. Instead of considering only the b^{th} bin, it considers k X k neighboring bins to determine the *Hit Rate*. Fig. 1 shows *Hit Rate* against *Penetration Rate* for the proposed technique.

In order to evaluate accuracy and efficiency, the proposed indexing technique has been compared with techniques presented in [6],[8]. All these techniques are evaluated on a Quad-Core (2×2.83 GHz) workstation with 3.23 GB RAM. In all these cases, comparisons are done on FVC 2004 datasets. Memory and computational cost of all these techniques are shown in Table 1. One can observe that the proposed technique is efficient with respect to the memory and computational cost.

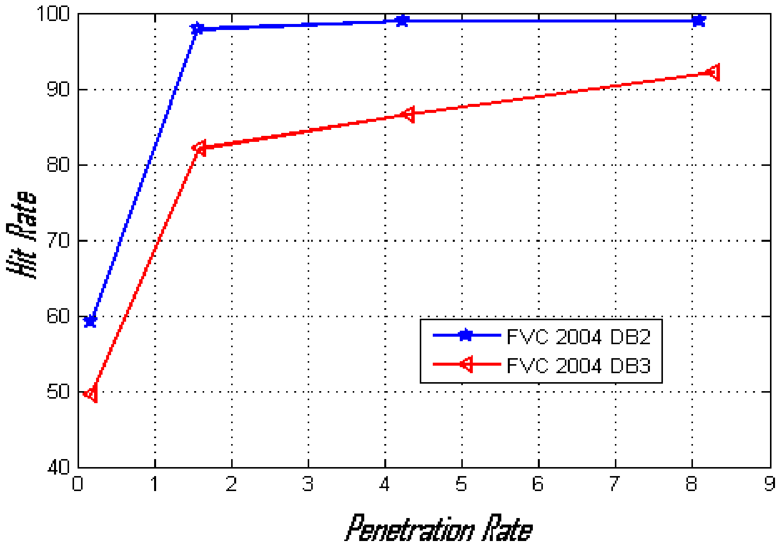

Fig. 1. Hit Rate against Penetration Rate for DB2 and DB3

Table 1. Comparison Table

Data Set	Memory Cost in MB			Computational Cost					
				Hash Table Entries			Query Time in Sec [1]		
	[6]	[8]	Proposed	[6]	[8]	Proposed	[6]	[8]	Proposed
DB2	1095	18.5	5.2	76764	447	2623	234.5	6.0	7.5
DB3	680	22.1	6.4	47465	5471	3124	265.8	6.7	8.0

6 Conclusion

This paper has proposed minutiae based geometric hashing technique to index the fingerprint database. Also, we have proposed minutia binary pattern (MBP) representation of feature vector of fixed length which is bit oriented coding, simple but very effective for the accurate search. The proposed technique performs indexing and searching in one pass with linear complexity. Unlike other geometric based indexing techniques, it inserts each minutia exactly once into a hash table. It effectively removes the use of all possible triangles proposed in [6] and bases pairs proposed in [7]; thus it reduces memory and computational complexity. Extensive experiments over FVC 2004 databases show the effectiveness of the proposed indexing technique with respect to well known geometric based indexing techniques [6][8].

---

[1] Query time includes feature extraction, searching and matching.

Acknowledgement: Authors acknowledge the support provided by the Department of Information Technology, Government of India, to carry out this work.

References

1. Raffaele, C., Matteo, F., Davide, M.: Fingerprint Indexing Based on Minutia Cylinder-Code. IEEE Transactions on Pattern Analysis and Machine Intelligence 33(5) (2011) 1051–1057
2. Feng, J.: Combining Minutiae Descriptors for Fingerprint Matching. Pattern Recognition. 41(1) (2008) 342–352
3. Boer, J.D., Bazen, A.M., Gerez, S.H.: Indexing Fingerprint Databases Based on Multiple Features. In: Proc of the 12th Annual Workshop on Circuits, Systems and Signal Processing. (2001) 300–306
4. X.Shuai, Zhang, C., Hao, P.: Fingerprint Indexing Based on Composite Set of Reduced SIFT Features. In: Proc of International Conference on Pattern Recognition. (2008) 1–4
5. Germain, R., Califano, A., Colville, S.: Fingerprint Matching Using Transformation Parameter Clustering. IEEE Computational Science and Engineering 4(4) (1997) 42 –49
6. Bhanu, B., Tan, X.: Fingerprint Indexing Based on Novel Features of Minutiae Triplets. IEEE Transactions on Pattern Analysis and Machine Intelligence 25(5) (2003) 616 – 622
7. Boro, R., Roy, S.D.: Fast and Robust Projective Matching for Fingerprints Using Geometric Hashing. In: Proc of the 4th Indian Conference on Computer Vision, Graphics and Image Processing. (2004) 681–686
8. Bebis, G., Deaconu, T., Georgiopoulos, M.: Fingerprint Identification using Delaunay Triangulation. In: Proc of IEEE International Conference on Intelligence, Information, and Systems. (1999) 452 –459
9. Bazen, A., Gerez, S.: Extraction of Singular Points From Directional Fields of Fingerprints. In: Proc of The annual CTIT Workshop on Mobile Communications. (2001) 41–44
10. Cappelli, M., Maio, D., Maltoni, D., Wayman, J.L., Jain, A.K.: FVC2004: Third Fingerprint Verification Competition. In: Proc of the First International Conference on Biometric Authentication. (2004) 1–7