Neurocomputing 137 (2014) 115–126

Contents lists available at ScienceDirect

Neurocomputing

journal homepage: www.elsevier.com/locate/neucom

An efficient minutiae based geometric hashing for fingerprint database ☆

Umarani Jayaraman *, Aman Kishore Gupta, Phalguni Gupta

Department of Computer Science and Engineering, Indian Institute of Technology Kanpur, Kanpur-208 016, India

ARTICLE INFO

Article history:
Received 20 September 2012
Received in revised form
14 February 2013
Accepted 28 February 2013
Available online 13 March 2014

Keywords:
Fingerprints
Indexing
Identification
Geometric based hashing
Minutia binary pattern

ABSTRACT

This paper proposes an efficient indexing technique for fingerprint database using minutiae based geometric hashing. This technique consists of two stages, known as *indexing* and *searching*. For an accurate match at the time of searching, it has proposed a fixed length feature vector built from each minutia, known as Minutia Binary Pattern. Unlike any existing geometric based indexing technique, 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.

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 subjects. Generally, the process to retrieve each model fingerprint from the database and to compare it against the query fingerprint for a match is computationally inefficient. A fingerprint image has the following characteristics:

- Number of minutiae extracted from a fingerprint of any subject at any two time instants may not be same.
- There may be too many minutiae in a fingerprint; some of them may be false. Also, there is a possibility of missing some true minutiae.
- 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.

Existing fingerprint indexing techniques can be classified on the basis of the methods of extracting features such as singular points [1], directional field [2], local ridge-line orientations [3–7], orientation image [8,9], minutiae [10–12], minutiae descriptor [13], multiple features [14], matching scores [15,16], SIFT features [17] and line features [18]. But most of the matching algorithms are based on minutiae; so use of minutiae to index the fingerprint database is beneficial in many respects.

Minutiae based techniques derive robust geometric features from triangles formed by triplets of minutiae and use hashing techniques to perform the search. An indexing technique for fingerprints has been proposed in [19] where geometric features obtained from triangles formed by triplets of minutiae are used for indexing. Triangles are formed between all possible combinations of minutiae triplets. During *indexing* stage, nine indexing keys *viz* the length of three sides (L_i), the ridge count between each pair of vertices (R_i) and minutiae angles encoded in a transformation-invariant fashion (θ_i) where $i=1,2,3$ have been considered. The index generated by the FLASH (Fast Lookup Algorithm for String Homology) framework serves as the key to identify triangles that resemble one another. At *searching* stage, index $I=((L_i, R_i, \theta_i)$: $i=1,2,3)$ generated from all possible triangles of a query fingerprint is used to retrieve all triangles stored in the table that are labeled with the same index. Matching is performed between triangles based on the transformation parameter clustering. Parameters such as translation in X,Y direction, rotation θ, along with the fingerprint ID are used to perform the matching. To reduce the false correspondences, geometrical constraints from minutiae triangles *viz* maximum length of three sides, median and

☆Preliminary version of this manuscript [Paper Id. 1265] has been accepted in International Conference on Intelligent Computing (ICIC'12), 2012 and sub-selected for Neurocomputing.

* Corresponding author. Tel.: +91 512 2597579; fax: +91 512 2597647.
E-mail address: umarani@cse.iitk.ac.in (U. Jayaraman).

http://dx.doi.org/10.1016/j.neucom.2013.02.059
0925-2312/

minimum angles, triangle handedness, type, direction and ridge count minutiae density are considered in [20]. Since triangles are formed using all possible minutiae, it increases both memory and computational cost.

In [21], a fast and robust projective matching technique has been proposed for fingerprints. The non-linear distortion and noise added in the fingerprint images at the time of image capturing are modeled as transformation in the projective domain. Matching of the fingerprints involves computation of the homography matrix for the projective transformation, mapping of the minutia points by this homography and finally computation of the points that match each other within a pre-determined threshold. It makes use of Geometric Hashing [22] for fast match. This technique consists of two phases viz. *preprocessing* and *recognition* phases. Assume, fingerprint of a subject contains a maximum of n minutiae and there are N fingerprints in the database. Let b be the bases *i.e.* the number of feature points used to obtain the hash table through geometric hashing. The value of b is determined by the class of invariants. For example, $b=2,3,4$ are, for a similarity transformation, an affine transformation and a perspective transformation respectively [23]. In *preprocessing* phase, all model fingerprints are represented in similarity invariant way. For each ordered pair ($b=2$) of minutiae of a model fingerprint, a coordinate system is defined and the coordinates of remaining $n-2$ minutiae of the model fingerprint are recomputed based on this coordinate system. The recomputed coordinates are used as the index of a hash table where entry of the form (bases, model) is recorded. Hence, there are totally $n \times {}^{n}C_2$ possible entries of bases pairs into a hash table. At the time of *recognition* phase, an arbitrary ordered bases (minutiae) is chosen from the query fingerprint. The coordinates of the remaining minutiae of the query fingerprint are recomputed in terms of the coordinate system defined by this bases. The recomputed coordinates of each minutia are used as the index into hash table and for each entry (bases, model) in the hash table, a vote is casted. If an entry (bases, model) scores large number of votes, then this bases corresponding to the one is selected from the query fingerprint. However, geometric hashing may not be suitable because it uses $n \times {}^{n}C_2$ bases pair where n is the maximum number of minutiae of a fingerprint.

In [24] geometric features obtained from Delaunay triangles of minutiae have been used for indexing the fingerprints. It can be shown that if n is the number of minutiae, Delaunay triangulation produces $O(n)$ 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. Fig. 1 shows an example of both fingerprints which are taken at two different time instants from the same person. Note that there is significant change in their Delaunay triangles. Some of the well known geometric based indexing techniques for fingerprint have been summarized in Table 1.

This paper presents an efficient indexing technique which uses invariant spatial (distance) and directional (angle) information to index each minutia into a hash table. It has proposed fixed length feature vector built from each minutia, known as Minutia Binary Pattern (MBP), for the accurate match at the time of searching. The technique is robust to translation and rotation. It can be conveniently represented by a hash table which contains spatial and directional information. Each minutia can be uniquely identified by its distance and angle information from its core point and is

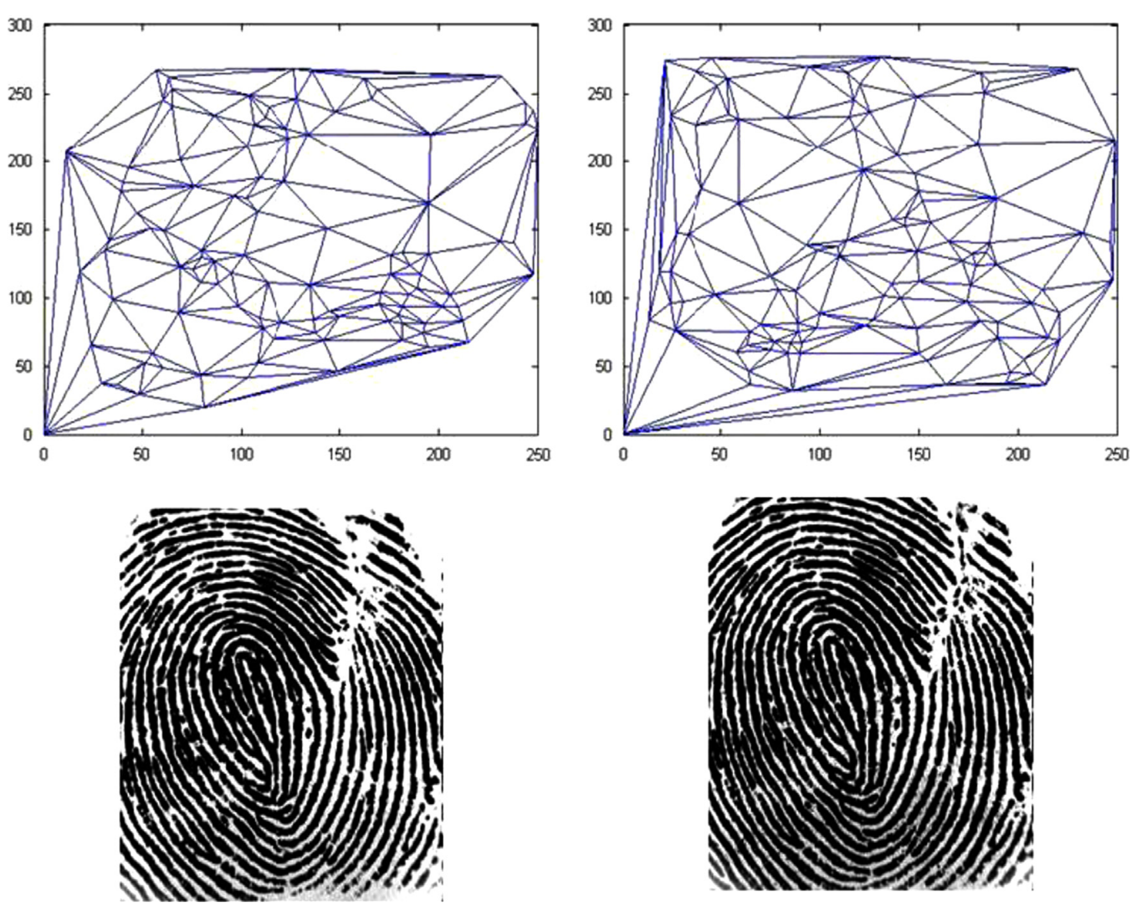

Fig. 1. Delaunay triangulation: Both fingerprints are taken at different time from same person.

Table 1
Existing geometric based indexing techniques used in fingerprint.

Author	Index elements	Comments
All triplets		
Germain et al. [19]	Length of three sides Local orientation Ridge count between 2 vertices	Triangle formed for all possible minutiae and there are ${}^{n}C_3$ triangles which increases memory and computation cost
Bhanu et al. [20]	Max length of 3 sides Median and Minimum angles Triangle handedness, type, direction ridge count, minutiae density	
Geometric hashing		
Rintu et al. [21]	Minutiae co-ordinate position	There are $n \times {}^{n}C_2$ possible bases pair entries which increases computation cost
Delaunay triangulation		
Bebis et al. [24]	Ratio of min to to max length Ratio of median side to max side Cosine of the angles between 2 sides	Delaunay triangulation is more sensitive to noise and distortion

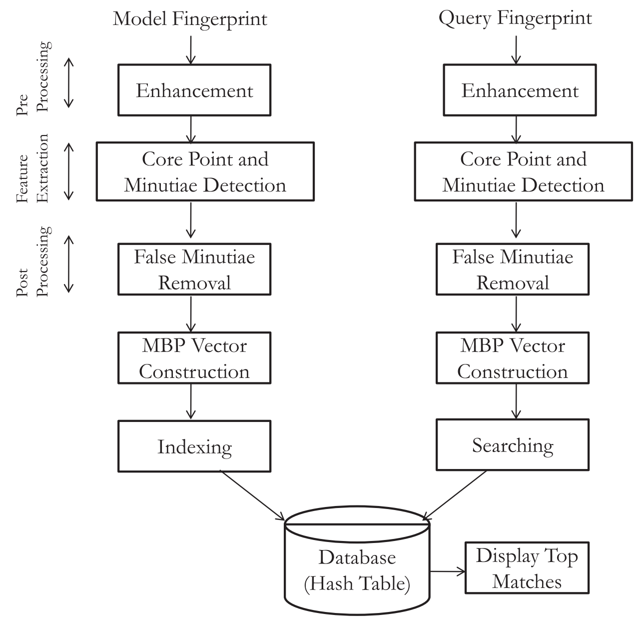

Fig. 2. Block diagram of proposed system.

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 [20] and bases pairs in [25]; as a result, it reduces memory and computational complexity.

The paper is organized as follows. Next section discusses the feature construction technique to obtain features from minutiae of a fingerprint image. Section 3 discusses the proposed indexing technique. Experimental results are analyzed in the next section. Conclusions are given in the last section.

2. Pre-processing and feature extraction

This section discusses the method of extracting features from the fingerprint. Minutiae points have been used to construct binary patterns around each minutiae, named as Minutiae Binary Pattern (MBP) which are used for matching. Fig. 2 shows the block diagram of the proposed system.

2.1. Fingerprint image enhancement

For a gray scale fingerprint image, different techniques to reduce the noise and to increase the contrast between ridges and valleys have been implemented in spatial domain or frequency domain. In this paper, Gabor filter based enhancement technique has been used to enhance the fingerprint images. It consists of several major steps like normalization, segmentation, orientation image estimation, frequency image estimation, Gabor filtering, binarization, morphological closing operation and thinning. Fingerprint image is normalized to a pre-specified mean and variance. The main purpose of normalization is to reduce the variation in gray-level values along ridges and valleys. Segmentation is the process of separating valid ridges and valleys from the background regions containing no valid information. The technique based on variance thresholding has been used to perform the segmentation.

The orientation estimation of a fingerprint image defines the local orientation of the ridges around each local neighborhood.

Table 2
Fingerprint Image Enhancement by Gabor Filter.

S.No	Poor fingerprints	Enhanced fingerprints
1		
2		
3		
4		

We have used least mean square based orientation estimation [26]. In a local neighborhood where there does not exist minutiae and singular point, the gray levels along ridges and valleys can be modeled as a sinusoidal shaped wave along a direction normal to the local ridge direction. Orientation and frequency estimation have been used to construct the even-symmetric Gabor filter. Since Gabor filters satisfy frequency-selective and orientation-selective properties, we have used these filters for orientation and frequency estimation. These properties allow the filters to be tuned to give the maximal response to ridges at a specific orientation and frequency in the fingerprint image. The gray scale image is binarized to improve the contrast of ridges and valleys in a fingerprint image. Morphological closing operation on the binary image helps to close the holes which are present within valid ridges. Thinning on the binarized image is used to erode the foreground pixels so that it becomes width of single pixel. Table 2 shows enhanced images of the poor quality images. One can observe that the cut mark in fingerprint image is connected in the enhanced fingerprint image.

2.2. Core point detection

Core point detection is the most challenging task. The core point or reference point is defined as the point with maximum curvature of convex ridges. Two different filters are used: one for the core type and other one for the delta type. Filtering is applied to complex images, *i.e.* the orientation tensor field in different scales [27]. The orientation tensor field image $z(x,y)=(f_x+if_y)^2$ is often used to represent the global structure in a fingerprint where f_x and f_y are the derivative of the original image in the x-direction and in the y-direction respectively. The complex filter response is $c=\mu\ \exp\{i\alpha\}$ where μ is a certainty measure of symmetry and α is the geometric orientation of the symmetric pattern. By using certainty measures, μ_1 and μ_2, for core and delta point symmetry respectively, we can identify core and its position (x_j, y_j) if $|\mu_1|>T_1$ and delta if $|\mu_2|>T_2$ where T_1 and T_2 are thresholds. Fig. 3 shows the steps involved in detecting the core point in a fingerprint while Fig. 4 shows the detected core points marked with dot for various types of fingerprints.

2.3. Minutiae extraction

In order to detect minutiae from a fingerprint, the crossing number (CN) concept [28] is used. We filter the thinned fingerprint image and compute the number of one-value of each 3×3 window. If the central is 1 and has only 1 one-value neighbor, then the central pixel is a ridge ending or termination. If the central is 1 and has 3 one-value neighbors, then the central pixel is a bifurcation. If the central is 1 and has 2 one-value neighbors, then the central pixel is a usual pixel. For each extracted minutia point, the following information is stored: (i) x and y coordinates, (ii) minutia orientation, θ and (iii) type of minutia (ridge ending or bifurcation).

2.4. False minutiae removal

Due to the presence of noise, the thinned binary image contains a large number of spurious minutiae having no valid information which affect the matching performance of the system. These spurious minutiae can be eliminated by (i) suppressing the internal spurious minutiae and (ii) suppressing the extrema minutiae. The internal spurious minutia can be suppressed if the distance between (i) a termination and a bifurcation are smaller than a predefined threshold T, (ii) two bifurcations are smaller than T, (iii) two terminations are smaller than T. In our experiment we have chosen T as 8. To suppress the extrema minutiae, we have determined a region of interest (ROI) which can be extracted by applying the closing operation followed by an erosion on the binary image. We can suppress minutiae external to this ROI. Fig. 5 shows the result of minutiae extraction before and after post-processing. Minutiae locations are marked by red circle for ridge ending and green circle for ridge bifurcation. All angles are measured w.r.t positive x-axis.

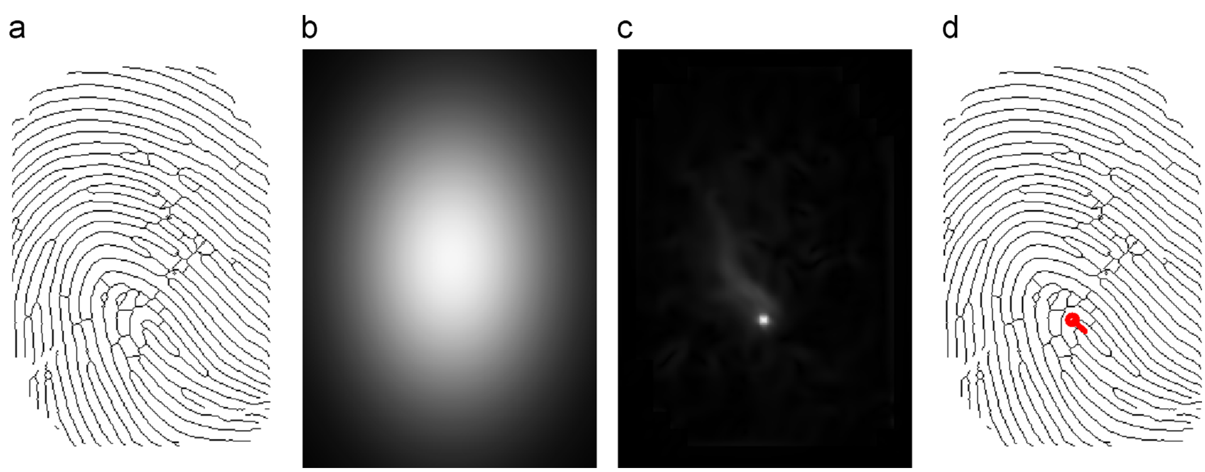

Fig. 3. Core point detection in a fingerprint. (a) Enhanced fingerprint, (b) resized maps, (c) modified core map and (d) marked core point.

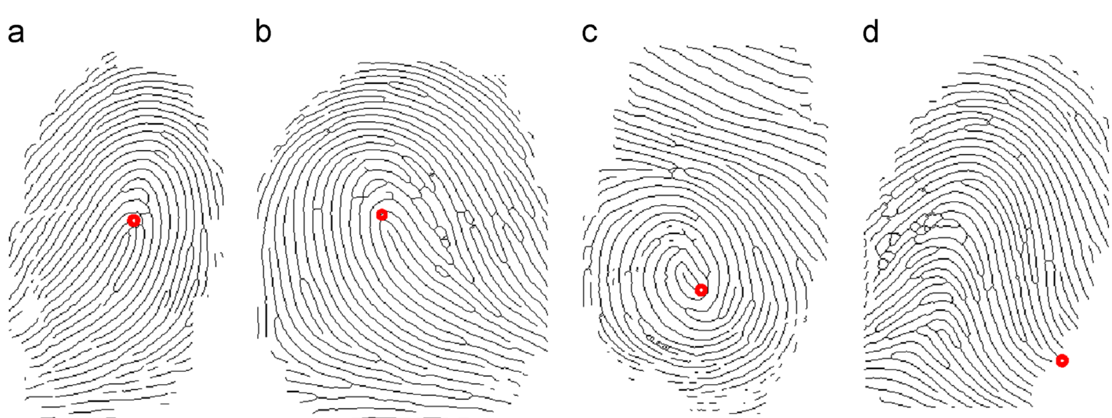

Fig. 4. Detected core points on different types of fingerprints. (a) Left loop, (b) right loop, (c) whorl and (d) arch.

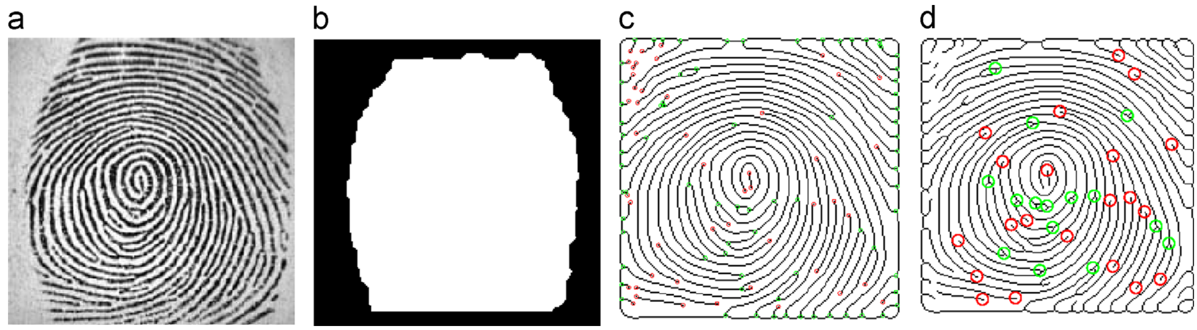

Fig. 5. Result of minutiae extraction. (a) Original fingerprint, (b) ROI, (c) minutiae before post processing and (d) minutiae after post processing.

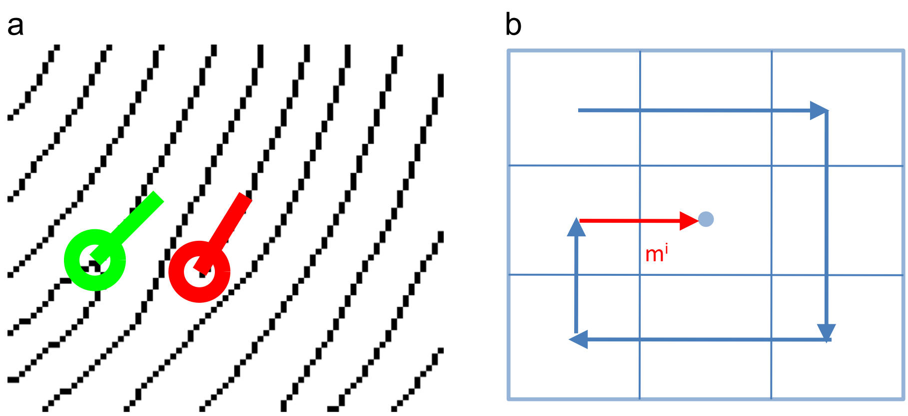

Fig. 6. MBP vector construction. (a) Termination and bifurcation and (b) direction of MBP vector.

2.5. Minutia binary pattern construction

Minutia Binary Pattern (MBP) is a vector derived from each minutia (m^i) by considering 3×3 window W as shown in Fig. 6. Each cell of W is a small square window and can be uniquely identified by two indices, (a,b) denoting its position in the square window, $1 \le (a,b) \le 3$. For each minutia, m^i, binary pattern is constructed by accumulating the binary values from all cells (a,b) in 3×3 square window W associated with m^i. Thus,

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

where $W(a,b)$ is a binary value associated with cell (a,b) 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 well suited for accurate match at the time of searching. Also, computation of fixed length MBP vector is very simple, fast and requires less space to store.

3. Proposed indexing technique

The model fingerprints in the database are indexed into a single 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. It recognizes the query fingerprint by searching only indexed and its neighboring hash bins.

3.1. Indexing

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^i_m, y^i_m, θ^i_m, T^i_m) denoting their x and y coordinates, direction and type (ridge ending or bifurcation) which is shown in Fig. 6(a). It can be noted that a query fingerprint may be translated or rotated relative to their respective model fingerprint in the database. Invariant spatial (distance) and directional (angle) information can be used to handle translation and rotation present in a fingerprint. The proposed technique is built using invariant distance and angle from the 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 by 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 is drawn centered at (x_c, y_c) whose radius R is the farthest point lying on the convex hull where (x_c, y_c) are the coordinates of the core point. This circle is divided into 72 distinct sectors, each at 5° of distance. For each minutia m^i 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

$$INT(D^i(m^i, C)) = \sqrt{(x^i_m - x_c)^2 + (y^i_m - y_c)^2} \tag{1}$$

Given the sector number S^j in which the minutia m^i is lying and the core point C, the minutia can be uniquely identified by its distance $D^i(m^i, C)$ and sector number $S^j(m^i, C)$. $D^i(m^i, C)$ is the spatial contribution of minutia m^i from the core point C while S^j of m^i with respect to C is the directional contribution of minutia m^i. These two invariant information are used as indexing element to insert each minutia of a model fingerprint along with M_{id} and MBP vector in the hash table H where M_{id} and MBP are the model fingerprint identity and minutiae binary pattern of a minutia m^i respectively.

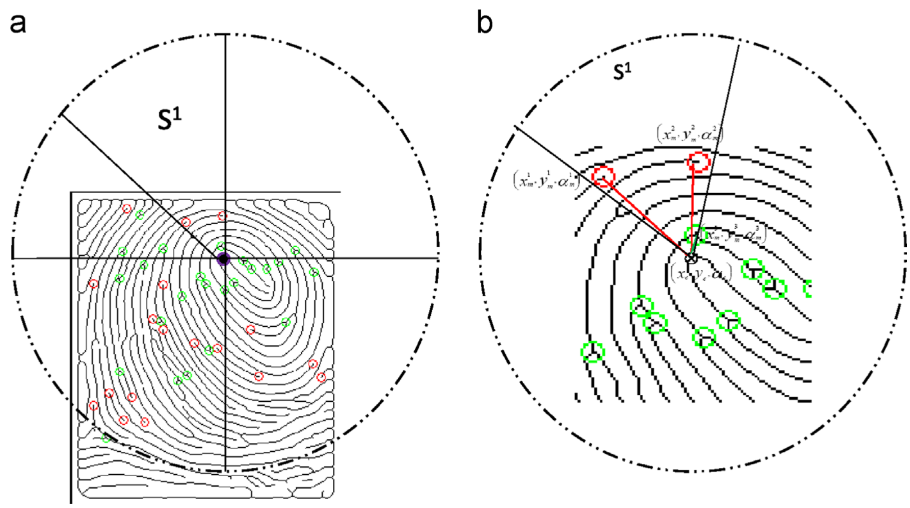

Fig. 7. Indexing steps. (a) Fingerprint is divided into sectors and (b) Indexing elements in a sector.

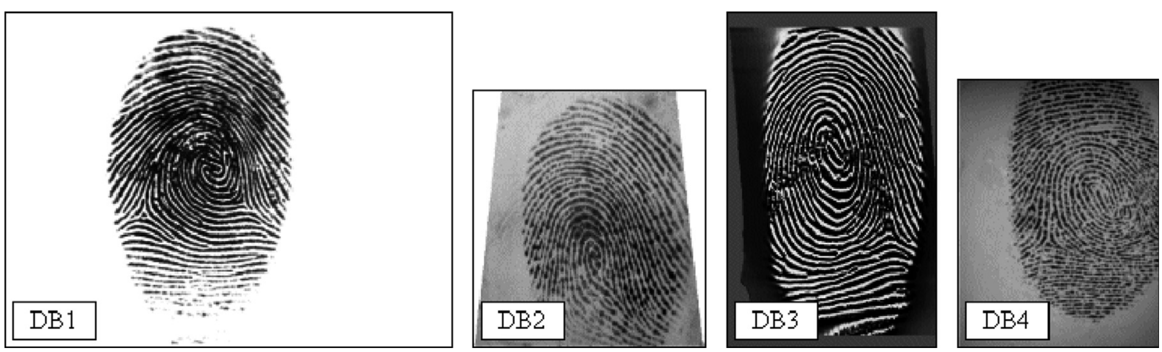

Fig. 8. FVC 2004 sample dataset.

Table 3
Core point accuracy for 25 pixels difference.

FVC-2004 Database 800 Fingerprints	Missed core point No	False core point No	Accuracy (%)
DB1	0	148	81.50
DB2	18	141	80.12
DB3	30	191	72.37

The computation of two indexing elements is given in Fig. 7. The algorithm for indexing the model fingerprints in the database is shown in Algorithm 1.

Algorithm 1. Indexing.

For each model M in the database do the following

1: Extract the minutiae from a model fingerprint image. Assume, there are o such minutiae.
2: Detect the core point $C=(x_c, y_c)$ of the model fingerprint image.
3: Construct the minutia binary pattern (MBP) around each minutiae $m^i; 1 \le i \le o$.
4: Obtain a convex hull $Conv_{Hull}(M, \Omega)$ by considering all minutiae in M by adding Ω offset pixels.
5: Draw a circle centered at C with radius R which is the farthest point lying on the convex hull.
6: Divide this circle into 72 sectors, each having 5° of difference. Let $S^j(m^i, C)$ be the sector S^j in which m^i is lying and C is the core point.
7: Compute the Euclidean distance, $D^i(m^i, C)$, between each minutia m^i and core point C lying in sector S^j.
8: Use $[D^i(m^i, C), S^j(m^i, C)]$ as an index to insert each minutia m^i along with model fingerprint identity, M_{id} and MBP vector into a 2-D hash table H.

3.2. Searching

During searching, it searches the generated hash table against minutiae of a query. Let $Q=\{m^1, m^2, \ldots, m^n\}$ be the n minutiae extracted from a query fingerprint. It can be noted that due to noise present in the fingerprints, the minutiae extracted from different fingerprints of the same subject may be shifted or missed. In such cases, to improve the recognition performance, it considers the minutiae not only from its mapped bin but also from its $(x \times y)$ neighboring bins. Let q be the mapped index in the hash table H for a minutia, m^k where $1 \le k \le n$. It has considered $(x \times y)$ neighbors of q in H. Let z be such a neighboring bin of q. There may be some minutiae of different model fingerprints from the database lying in the bin z of hash table H. Let c be a minutia of a model fingerprint lying in z. Then the Hamming distance between q and c, $\forall c \in H(z)$ can be obtained by

$$d(c) = d_H(MBP^{(q)}, MBP^{(c)}) \tag{2}$$

where d_H denotes the *Hamming distance* and $MBP^{(q)}, MBP^{(c)}$ are the *Hamming metric* space of dimension d (*i.e.* the space of binary vectors of length d under the standard Hamming metric). In this paper, d is 9.

For each minutia m^k of the query fingerprint Q, its corresponding candidate set S_k contains the model fingerprint identities $M_{id}(c)$ such that $d(c)$ satisfies (2) for all in $H(z)$. Thus, there are n such candidate sets $S_1, S_2, \ldots, S_n$ for given n minutiae in a query fingerprint Q. These candidate sets $\{S_1, S_2, \ldots, S_n\}$ are concatenated and the number of occurrences of each model fingerprint identity M_{id} is determined. Let S be the set of the form as $S = \{M_{id}, l\}$ where l is the number of occurrence of each model fingerprint identity M_{id}. First t model fingerprint identities having the largest number of occurrences from the set S are considered as the top t matches against the query fingerprint Q. The algorithm for recognition is given in Algorithm 2.

Algorithm 2. Searching.

For a query Q do the following
1: Extract the minutiae from a query fingerprint. Assume, there are n such minutiae.
2: Detect the core point $C = (x_c, y_c)$ of the query fingerprint image.
3: Construct the minutia binary pattern (MBP) around each minutiae $m^k; 1 \le k \le n$.
4: For each minutia, m^k, find the indices $D^k(m^k, C), S^j(m^k, C)$.
5: Map the indices $D^k(m^k, C), S^j(m^k, C)$ of a minutia m^k into a hash table to access the appropriate bin.
6: Compute the difference of minutiae minutia binary pattern (MBP) vector between query's minutia, m^k, and all minutiae that found in the bin.
7: Retain the model fingerprint identities M_{id}'s in its corresponding candidate set S_k if it exactly match.
8: Accumulate the results into the candidate set $S_1, S_2, \ldots, S_n$.
9: Cast a vote based on their occurrence of their model identities (M_{id})'s by concatenating the candidate set $S_1, S_2, \ldots, S_n$ to declare the top t best matches.

4. Experimental results

4.1. Dataset

We have evaluated our technique on FVC 2004 dataset [29]. Some of the sample fingerprint images are shown in Fig. 8.

- FVC 2004 DB1: This dataset contains 800 fingerprints from 100 fingers (8 impressions per finger) captured using the optical sensor "V300" by CrossMatch and image is of 640 × 480 at 500 dpi.
- FVC 2004 DB2: This 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 each image is 328 × 364 at 500 dpi.
- FVC 2004 DB3: This one consists of 800 fingerprints from 100 fingers (8 impressions per finger) captured using thermal sweeping sensor "FingerChip FCD4B14CB" by Atmel and size of each image is 300 × 480 at 512 dpi.

For each subject in the dataset, first image is considered for searching and the remaining (7) images are used for indexing.

4.2. Core point detection accuracy

We have manually marked the position of the core point and have obtained the same using the core point detection algorithm as discussed in Section 2.2. The detected core point is considered to be false if its position is away by more than 25 pixels from its true location which has been marked manually. Table 3 shows the number of missed and falsely detected core points by the algorithm. In our experiment, missed core point fingerprints are also considered as falsely detected core point and we replace these fingerprints by the correct one which have been marked manually known as *semi-automatic dataset*. Actual dataset which contains both missed and falsely detected fingerprints is known as *automatic dataset*. Results of the proposed technique have been analyzed on these two different datasets. The core point detection algorithm fails to detect the core point due to the following reasons.

- Fingerprint does not contain core at all (missed case)
- Fingerprint images are of poor quality (false case)
- Fingerprint of Arch type (false case)

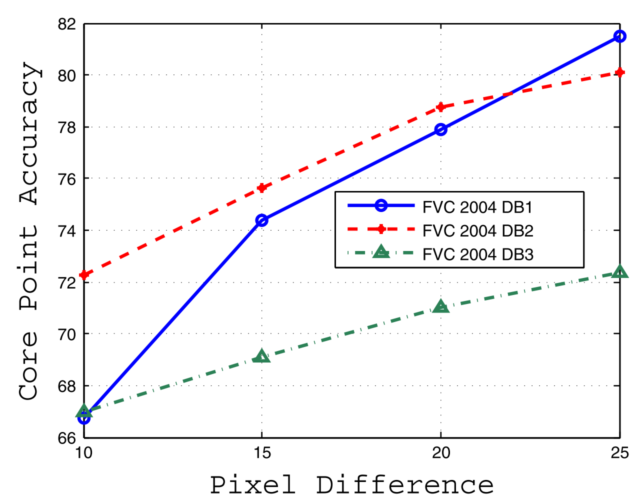

Fig. 10. Core point accuracy against distance.

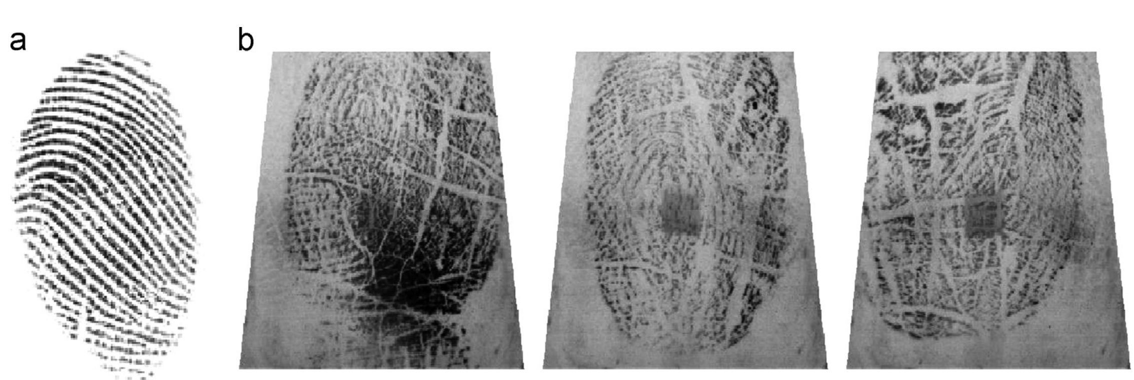

Fig. 9. FVC 2004 database. (a) No core and (b) Some poor quality images.

The algorithm fails to detect the core point on arch type of fingerprints to some extent. The results of the fingerprint classification in [30] have shown that there are about 65% of the total fingerprints which contain left or right loops, about 30% of the fingerprints which contains whorl. Remaining fingerprints have arch and tented arch. So number of failures in detecting the core point on arch type fingerprint images may be very few. Fig. 9 shows some of the fingerprint images having no core and poor quality images of FVC-2004 database. Accuracy of the core point detection for various pixel differences from the true location is shown in Fig. 10.

4.3. Performance measurements

To determine the performance of the proposed indexing technique, three measures, namely, *Hit Rate* and *Penetration Rate* have been used. These measures are defined as follows.

- *Hit rate* (H_r) is the rate that the number of query images is correctly identified in the top t best matches out of total number of query images. More clearly, let X be the number of queries correctly identified and L be the total number of queries made, then *Hit Rate* is given by $H_r = (X/L) \times 100\%$.
- *Penetration rate* (P_r) is defined by $P_r = (1/X\sum_{i=1}^{X} d_i/N) \times 100\%$ where X is the number of query images correctly identified, d_i is the number of images retrieved for ith query image and N is the database size.
- *Cumulative match characteristic* (CMC) curve is the relationship between the probability of identification against various ranks.

4.4. Results

The proposed indexing technique has been evaluated on the three datasets of FVC 2004. On each dataset, a state-of-art minutiae extraction algorithm has been used to create ISO/IEC 19794-2 templates [31] for all fingerprints. Minutiae obtained from all model fingerprints are indexed using a single hash table. At the time of searching, it uses the hash table to retrieve the set of model fingerprint identities M_{id} that are similar to the query fingerprint image. Each minutia m^k of the query fingerprint image is mapped to

Table 4
Optimum sector number chosen from Fig. 11.

Dataset	Rank-1 recognition rates		Optimum sector no (x)	
	Automatic (%)	Semi-automatic (%)	Automatic	Semi-automatic
DB1	63	75	3	5
DB2	64	68	3	3
DB3	60	80	5	3

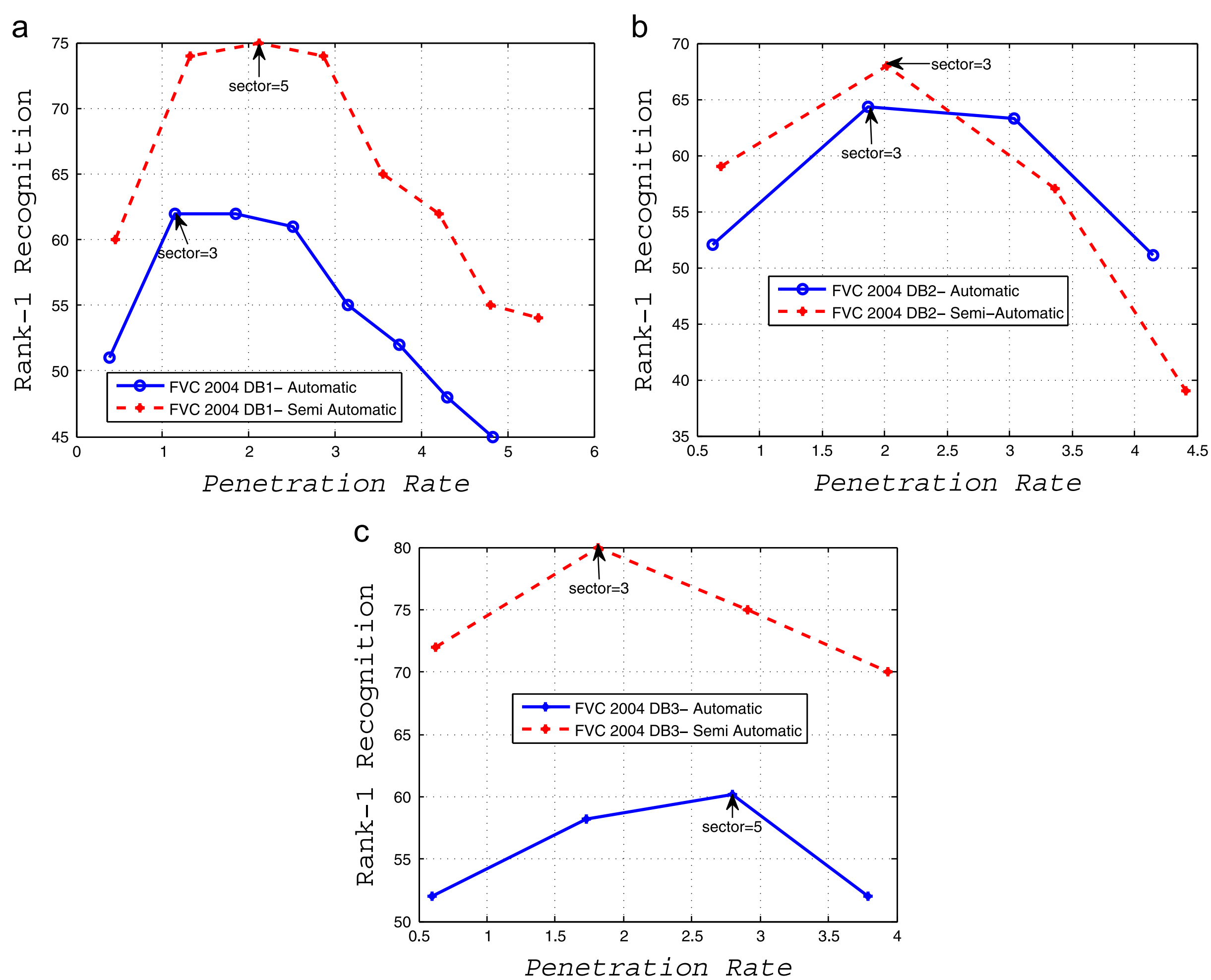

Fig. 11. Rank-1 recognition against *Penetration Rate*. (a) DB1, (b) DB2 and (c) DB3.

the hash table. Difference between the MBP vectors of the query minutia and all minutiae in the mapped bin is computed. If MBP vectors exactly match then model identities corresponding to these minutiae are considered for the candidate set S. A vote is cast for the model identities to declare the top t matches. Let it be mapped to the bin b of the hash table. Instead of considering only the bth bin, it considers its $x \times y$ neighboring bins to improve the performance of the recognition results where x refers to the number of sectors (angle in degree) while y refers to the distance (in pixels) respectively. Both x and y refer to the row and column of the 2-D hash table.

4.5. Parameter selection (x and y)

The objective is to determine the optimum number of neighboring bins ($x \times y$) to be considered such that we get the high *Hit Rate* with low *Penetration Rate*. Hence these two parameters are varied one by one, (*i.e.*) x is varied followed by y. Since we have considered distance of 25 pixels as tolerance value to detect the core point accuracy, we have first varied the x (sector number) by keeping the y (distance) as constant (*i.e.*25) to obtain the rank-1 recognition. Fig. 11(a) shows rank-1 recognition against *Penetration Rate* obtained for various x values for automatic and semi-automatic DB1 datasets. One can observe that as x increases, the *Penetration Rate* also increases. The value x is considered as an optimum for which high *Hit Rate* (rank-1 recognition) is achieved with low *Penetration Rate*. Optimum x value to achieve maximum rank-1 recognition rates (63% and 75%) are 3 and 5 for DB1 automatic and semi-automatic datasets respectively. Similarly, optimum x value to achieve maximum rank-1 recognition rates have been obtained for DB2 and DB3. Results are shown in Fig. 11 and Table 4.

These optimum x values have been used in each dataset to obtain the suitable value of y for the best *Hit Rate* and *Penetration Rate*. Fig. 12 shows *Hit Rate* and *Penetration Rate* for various datasets. For all cases, *Hit Rates* are obtained for the top 10 best matches. It has been observed that *Hit Rate* and *Penetration Rate* are increased with the increase of y value. Thus, there is a trade-off between *Hit Rate* and *Penetration Rate*. The y value is considered as an optimum when these rates are stable. The performance difference between automatically and semi-automatically detected core points

Table 5
Hit Rate and *Penetration Rate* for top 10 best matches From Fig. 12.

Dataset	Optimum x		Optimum y		*Hit Rate*		*Penetration Rate*	
	Auto	Semi	Auto	Semi	Auto (%)	Semi (%)	Auto (%)	Semi (%)
DB1	3	5	25	19	83.0	96	1.40	2.61
DB2	3	3	25	21	86.7	95	2.16	2.33
DB3	5	3	25	21	84.6	94	3.01	1.95

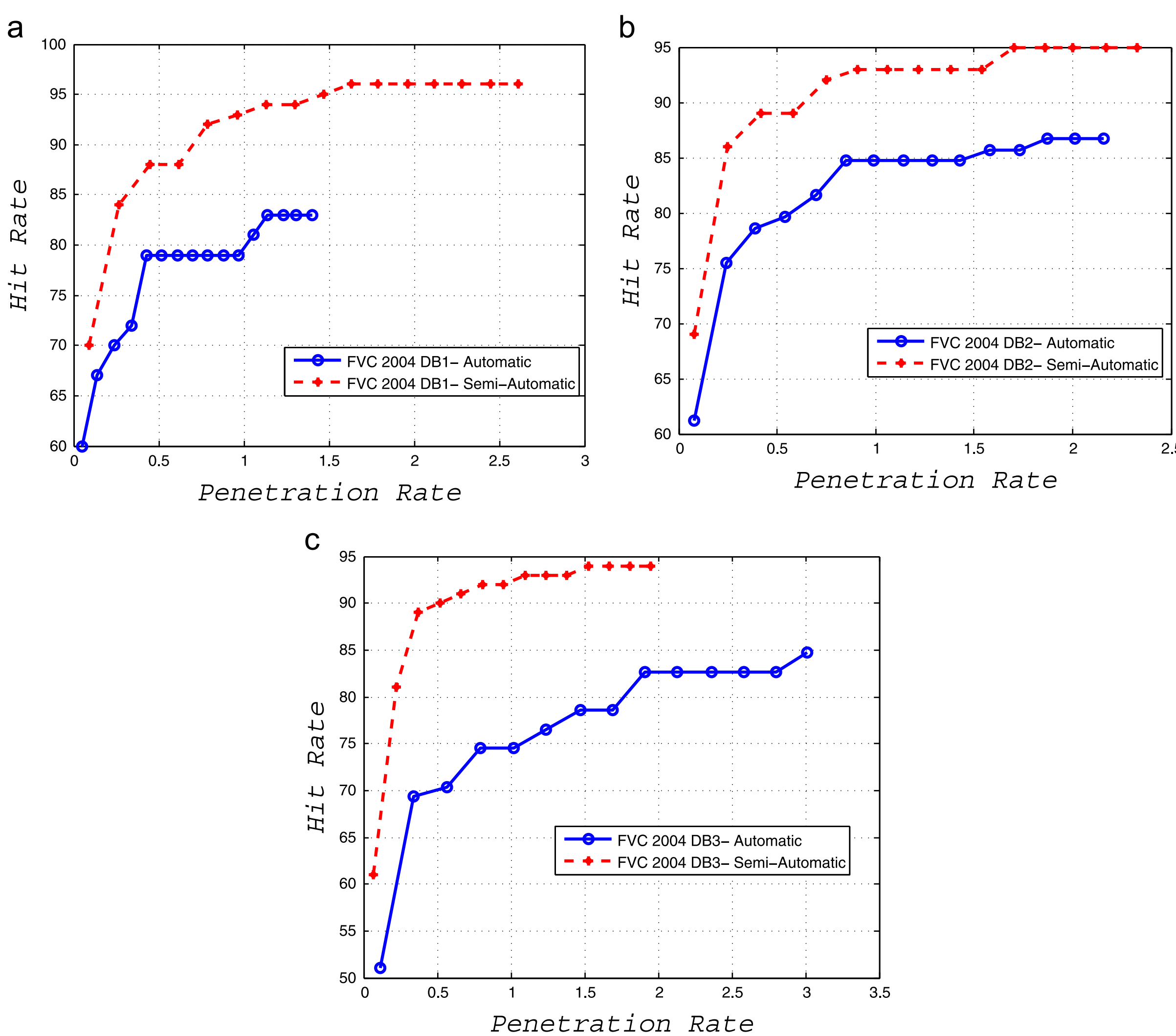

Fig. 12. *Hit Rate* against *Penetration Rate*. (a) DB1, (b) DB2 and (c) DB3.

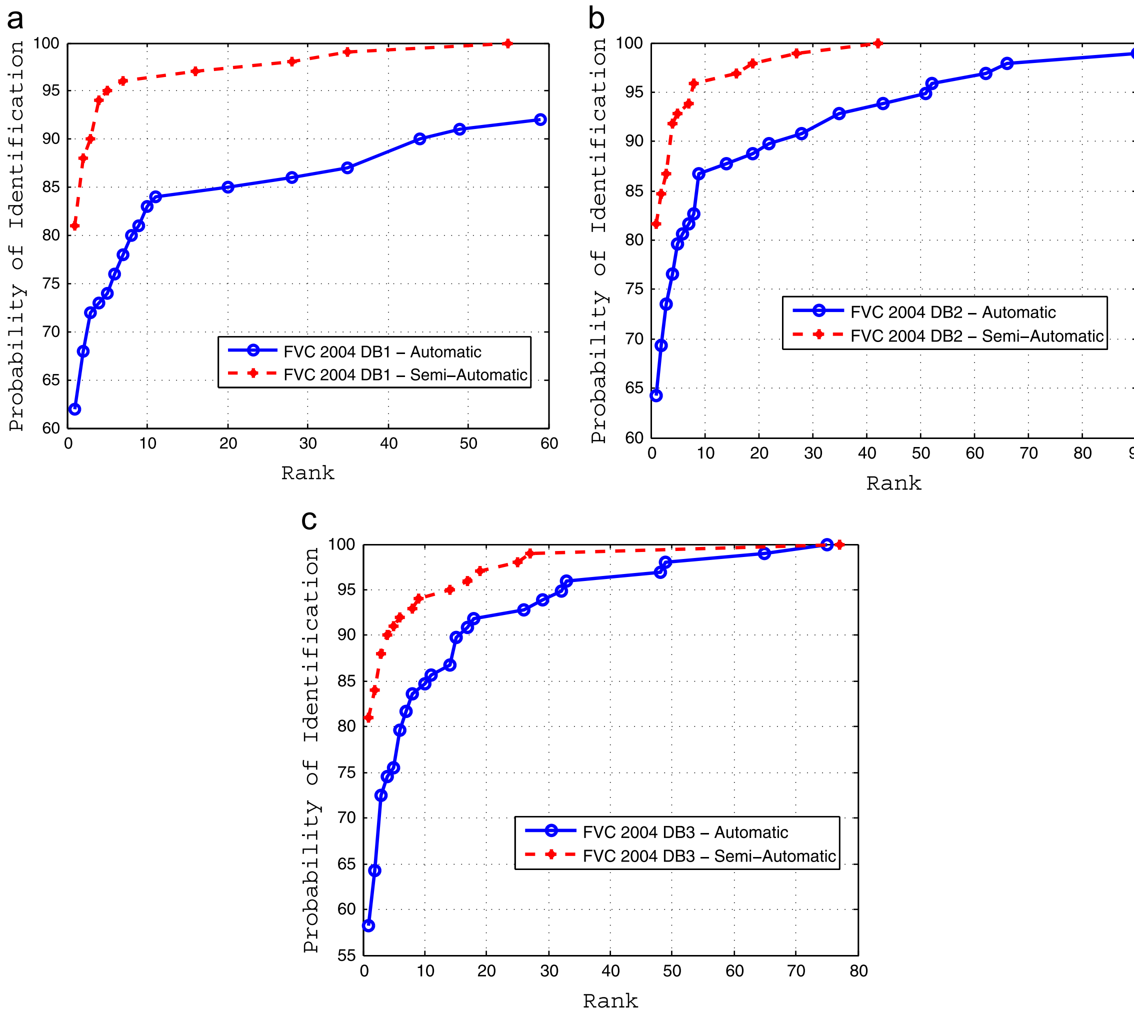

Fig. 13. CMC Curve. (a) DB1, (b) DB2 and (c) DB3.

Table 6
Indexing performance: *Penetration Rates* at various *Hit Rates*.

FVC 2004 DB1	95%	99%	100%
All Triplets	8.1%	27.2%	40.9%
Delaunay Triangles	7.2%	18.1%	32.7%
LoD Triangles	3.6%	10.0%	20.9%
Quadruplets	–	11.8%	12.0%
Proposed Automatic	2.26%	–	–
Proposed Semi-Automatic	1.55% (top 5)	2.61% (top 35)	2.61% (top 55)

for different datasets can be observed in Fig. 12. The best *Hit Rate* and *Penetration Rate* where core points are detected automatically and semi-automatically for different datasets are shown in Table 5.

These optimum x and y values have been used to obtain the probability of identification accuracy against various ranks. Fig. 13 shows the CMC curve for different datasets. For automatic dataset, maximum accuracies which are obtained at rank 59 for DB1, 90 for DB2 and 75 for DB3 are 92.0%, 98.9% and 100% respectively. One can observe that 100% accuracy is achieved in semi-automatic dataset at rank 55,42 and 77 for DB1, DB2 and DB3 respectively.

4.6. *Performance comparison*

The proposed technique has been compared with four other fingerprint indexing techniques; *viz.* All Triplets [20], Delaunay [24], Low-order Delaunay [32], Quadruplets [33]. For this comparison, we follow the testing platform of Liang et al. [32] and use the three images indexed as 2,3 and 4 out of eight images of each subject for indexing while the remaining five images for searching. Since we have considered five images for searching, the average results are reported. These techniques are evaluated on a Quad-Core (2 $\times$ 2.83 GHz) workstation with 3.23 GB RAM. In all these cases, comparisons are done on FVC2004 DB1 dataset. Table 6 shows the average *Penetration Rates* for various *Hit Rates*. The results reported in Table 6 of the proposed technique are obtained for top t matches. It shows that the technique requires less *Penetration Rate* to achieve the same *Hit Rate*. This might be explained by the fact that the number of triangles produced in all triplet based techniques is $O(n^3)$. Therefore, its search space in the database is larger than those of other techniques [32] which increases the *Penetration Rate*. Although the Delaunay and the LOD Delauany based techniques produce $O(n)$ triangles, but it affects the topology of Delaunay triangulation seriously due to noise and distortion which makes the technique unstable. However, distance and angle used in the proposed technique are not affected by any distortion and noise. Hence, the proposed technique is more stable.

5. Conclusions

This paper has proposed a 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. It is a bit oriented coding, simple but very effective for the accurate match at the time of searching. At the time of indexing, minutiae of all model fingerprints are extracted and inserted into a single hash table H along with their respective model identity (M_{id}) and MBP vector. During searching, minutiae of query Q are mapped on hash table to access the stored minutiae in the bin and its neighboring bins. The model identities M_{id}'s whose MBP vectors are exactly matched, are retained in the candidate set. These M_{id}'s of all mapped query minutiae are concatenated and occurrence of each M_{id} is found out. M_{id}'s having the t largest number of occurrences are selected as the top t best matches.

The proposed indexing technique performs *indexing* and *searching* in one pass with linear complexity. Unlike other techniques, this technique inserts each minutia exactly once into a hash table. It effectively removes the use of all possible triangles proposed in [20] and bases pairs proposed in [25]; thus it reduces memory and computational complexity. For the semi-automatic dataset, the proposed technique has achieved 96% *Hit Rate* with 2.61% *Penetration Rate* for DB1 dataset. Similarly, *Hit Rate* and *Penetration Rate* achieved for DB2 are 95% and 2.33% respectively while these for DB3 are found to be 94% and 1.95% respectively.

Acknowledgments

This is to acknowledge that this work has been supported by the Ministry of Communication and Information Technology (MCIT), Government of India, Delhi, India.

References

[1] T. Liu, G. Zhu, C. Zhang, P. Hao, Fingerprint indexing based on singular points, in: Proceedings of International Conference on Image Processing, 2005, pp. 293–296.
[2] R. Cappelli, D. Maio, D. Maltoni, Indexing fingerprint databases for efficient 1:N matching, in: Proceedings of International Conference on Control, Automation, Robotics and Vision, Invited Paper, 2000.
[3] L. Alessandra, M. Dario, M. Davide, Continuous versus exclusive classification for fingerprint retrieval, Pattern Recognit. Lett. 18 (10) (1997) 1027–1034.
[4] R. Cappelli, D. Maio, D. Maltoni, A multi-classifier approach to fingerprint classification, Pattern Anal. Appl. 5 (2) (2002) 136–144.
[5] X. Jiang, M. Liu, A.C. Kot, Fingerprint retrieval for identification, IEEE Trans. Inf. Forensics Security 1 (4) (2006) 532–542.
[6] S.O. Lee, Y.G. Kim, G.T. Park, A feature map consisting of orientation and interridge spacing for fingerprint retrieval, in: Proceedings of International Conference on Audio and Video Based Biometric Person Authentication, 2005, pp. 184–190.
[7] C. Raffaele, L. Alessandra, M. Dario, M. Davide, Fingerprint classification by directional image partitioning, IEEE Trans. Pattern Anal. Mach. Intell. 21 (5) (1999) 402–421.
[8] J. Li, W.Y. Yau, H. Wang, Fingerprint indexing based on symmetrical measurement, in: Proceedings of International Conference on Pattern Recognition, 2006, pp. 1038–1041.
[9] L. Manhua, J. Xudong, K.A. Chichung, Fingerprint retrieval by complex filter responses, in: Proceedings of International Conference on Pattern Recognition, 2006, pp. 1042–1048.
[10] X. Liang, T. Asano, A. Bishnu, Distorted fingerprint indexing using minutiae detail and delaunay triangle, in: Proceedings of the 3rd International Symposium on Voronoi Diagrams in Science and Engineering, 2006, pp. 217–223.
[11] C. Raffaele, F. Matteo, M. Davide, Fingerprint indexing based on minutia cylinder-code, IEEE Trans. Pattern Anal. Mach. Intell. 33 (5) (2011) 1051–1057.
[12] N.K. Ratha, K. Karu, S. Chen, A.K. Jain, A real-time matching system for large fingerprint databases, IEEE Trans. Pattern Anal. Mach. Intell. 18 (8) (1996) 799–813.
[13] J. Feng, Combining minutiae descriptors for fingerprint matching, Pattern Recognit. 41 (1) (2008) 342–352.
[14] J. deBoer, A.M. Bazen, S.H. Gerez, Indexing fingerprint databases based on multiple features, in: Proceedings of the 12th Annual Workshop on Circuits, Systems and Signal Processing, 2001, pp. 300–306.
[15] G. Aglika, R. Arun, A novel coding scheme for indexing fingerprint patterns, in: Proceedings of the Joint IAPR International Workshop on Structural, Syntactic, and Statistical Pattern Recognition, 2008, pp. 755–764.
[16] T. Maeda, M. Matsushita, K. Sasakawa, Characteristics of the identification algorithm using a matching score matrix, in: Proceedings of First International Conference on Biometric Authentication, 2004, pp. 330–336.
[17] X. Shuai, C. Zhang, P. Hao, Fingerprint indexing based on composite set of reduced SIFT features, in: Proceedings of International Conference on Pattern Recognition, 2008, pp. 1–4.
[18] W. Jia, Y.-H. Zhu, L.-F. Liu, D.-S. Huang, Fast palmprint retrieval using principal lines, in: International Conference on Systems, Man and Cybernetics (SMC '09), 2009, pp. 4118–4123.
[19] R. Germain, A. Califano, S. Colville, Fingerprint matching using transformation parameter clustering, IEEE Comput. Sci. Eng. 4 (4) (1997) 42–49.
[20] B. Bhanu, X. Tan, Fingerprint indexing based on novel features of minutiae triplets, IEEE Trans. Pattern Anal. Mach. Intell. 25 (5) (2003) 616–622.
[21] R. Boro, S.D. Roy, Fast and robust projective matching for fingerprints using geometric hashing, in: Proceedings of the 4th Indian Conference on Computer Vision, Graphics and Image Processing, 2004, pp. 681–686.
[22] H.J. Wolfson, Model based object recognition by geometric Hashing, in: Proceedings of European Conference on Computer Vision, 1990, pp. 526–536.
[23] M. Costa, R. Haralick, L. Shaprio, Optimal affine invariant point matching, in: Proceedings of International Conference on Pattern Recognition, 1990, pp. 233– 236.
[24] G. Bebis, T. Deaconu, M. Georgiopoulos, Fingerprint identification using delaunay triangulation, in: Proceedings of IEEE International Conference on Intelligence, Information, and Systems, 1999, pp. 452–459.
[25] Y. Lamdan, J. Schwartz, H. Wolfson, Affine invariant model based object recognition, IEEE Trans. Robot. Autom. 6 (5) (1990) 578–589.
[26] L. Hong, Y. Wan, A. Jain, Fingerprint image enhancement: algorithm and performance evaluation, IEEE Trans. Pattern Anal. Mach. Intell. 20 (8) (1998) 777–789.
[27] K. Nilsson, J. Bigun, Localization of corresponding points in fingerprints by complex filtering, Pattern Recognit. Lett. 24 (13) (2003) 2135–2144.
[28] B.M. Mehtre, Fingerprint image analysis for automatic identification, Mach. Vis. Appl. 6 (2–3) (1993) 124–139.
[29] M. Cappelli, D. Maio, D. Maltoni, J.L. Wayman, A.K. Jain, FVC2004: Third fingerprint verification competition, in: Proceedings of the First International Conference on Biometric Authentication, 2004, pp. 1–7.
[30] R. Cappelli, D. Maio, D. Maltoni, L. Nanni, A two-stage fingerprint classification system, in: Proceedings of the 2003 ACM SIGMM Workshop on Biometrics Methods and Applications, 2003, pp. 95–99.
[31] ISO/IEC 19794-2:2005, in: Information Technology Biometric Data Interchange Formats Part 2: Finger Minutiae Data, 2005.
[32] X. Liang, A. Bishnu, T. Asano, A robust fingerprint indexing scheme using minutia neighborhood structure and low-order delaunay triangles, IEEE Trans. Inf. Forensics Security 2 (4) (2007) 721–733.
[33] O. Iloanusi, A. Gyaourova, A. Ross, Indexing fingerprints using minutiae quadruplets, in: 2011 IEEE Computer Society Conference on Computer Vision and Pattern Recognition Workshops (CVPRW), 2011, pp. 127–133.

Umarani Jayaraman is currently a research scholar in the Department of Computer Science and Engineering, Indian Institute of Technology Kanpur. She has received her M.E in Computer Science and Engineering from Anna University. She published two journal papers and six conference papers in the field of Large Scale Biometric Identification. She also been associated with the project sponsored by the Department of Information Technology, Government of India, Delhi, India. Her research interest includes Digital Image Processing, Biometrics and Pattern Recognition.

Aman Kishore Gupta has received B.Tech Degree in Computer Science and Engineering from Maharana Pratap Engineering College Kanpur, India in 2010. He is currently working as a "Research Associate" in the project on the Development of Biometrics System at Indian Institute of Technology Kanpur, India, since July 2010. The project has been sponsored by the Department of Information Technology, Government of India, Delhi, India. His research interests include Image Processing, Biometrics and Pattern Recognition.

Phalguni Gupta is associated with Department of Computer Science and Engineering, Indian Institute of Technology Kanpur, India, where currently he is a professor. He received his Ph.D. degree in computer science and engineering from Indian Institute of Technology Kharagpur, India, in 1986. Prior to joining IIT Kanpur, he was with the Image Processing and Data Product Group of the Space Applications Centre (ISRO), Ahmedabad, India (1983–1987) and was responsible for correcting image data received from Indian Remote Sensing Satellite. His research interest includes sequential algorithms, parallel algorithms, online algorithms, image processing, biometrics, identity and infrastructure management. Currently, he is responsible for several research projects in the area of image processing, biometrics and parallel processing. He has published over 150 peer-reviewed journals and conference papers. He is also the co-author of six books. He is a member of the Association Computing Machinery (ACM) and recipient of 2007 IBM Faculty Award.