Pattern Recognition 67 (2017) 110–126

Contents lists available at ScienceDirect

Pattern Recognition

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

Fingerprint indexing based on minutiae pairs and convex core point

Javad Khodadoust[a,*], Ali Mohammad Khodadoust[b]

[a] Department of Computer Engineering and Information Technology, Payame Noor University (PNU), Qeshm, Iran
[b] Department of Electrical Engineering, Sadjad University of Technology, Mashhad, Iran

ARTICLE INFO

Article history:
Received 19 September 2016
Revised 17 December 2016
Accepted 13 January 2017
Available online 17 January 2017

Keywords:
Fingerprint indexing
Minutiae
Clustering
Singular points
Candidate list reduction
Ellipse

ABSTRACT

Fingerprint identification is an important issue for identifying fingerprints and plays a key role in the fingerprint recognition systems. However, performing a fingerprint identification over a large database can be an inefficient task due to the lack of scalability and high computing times of fingerprint matching algorithms. Fingerprint indexing is a key strategy in automatic fingerprint identification systems (AFISs) which allows us to reduce the number of candidates, the search space, and the occurrences of false acceptance in large databases. In this paper, an efficient indexing algorithm is proposed using minutiae pairs and convex core point which employs k-means clustering and candidate list reduction criteria to improve the identification performance. Our proposal can effectively reduce the search space and number of candidates for fingerprint matching, and thus achieves higher efficiency and significantly improves the system retrieval performance. Experimental results over some of the fingerprint verification competition (FVC) and the national institute of standards and technology (NIST) databases prove the superiority of the proposed approach against some of the well known indexing algorithms.

1. Introduction

Fingerprints are physiological characteristics of biometric recognition and have been extensively used in both forensic and non-forensic applications [49]. A fingerprint is the pattern of ridges (single curved segments) and valleys (regions that lies in between ridges) on the surface of a fingertip [41,74]. Fig. 1 shows an example of gray-scale image of fingerprint and the details extracted from it.

A fingerprint recognition system may be called either a verification system or an identification system. The aim of fingerprint verification systems is to focus on matching the fingerprint of a person with his/her stored fingerprint templates, i.e., matching module works in one-to-one comparison mode. In these systems, the speed of a comparison mainly depends on how the similarity measure is evaluated between a query print (probe) and the fingerprint stored in the database (gallery), whereas the selection of the similarity measure usually depends on representation of the features. On the other hand the aim of the fingerprint identification systems is to establish the identity of a person, given a query print and a database of enrolled fingerprints of different individuals to seek a match. In these systems, matching module works in one-to-many comparison mode [22,39,49,52,70]. If the number of templates in a database are N, thus identification systems can be viewed as a series of N one-to-one comparisons [49,58].

Most of the well known fingerprint matching algorithms are fast and quite accurate to deal with small databases [59], but nowadays the size of most of modern fingerprint databases is very large and need a search on all gallery fingerprints which affect both accuracy and efficiency of fingerprint matching [7,22,52]. If there is a need for finding a person among these databases, it must be done in reasonable time, often shorter than a few seconds. Within this context, the bottleneck step in the identification process is the matching algorithm, because it must be performed once per each gallery to determine which one has the most similary with the probe.

There are various methods to reduce the search time and computational complexity, and to speed up the matching process in the literature [52,58]. Generally, reducing the number of comparisons and increasing the speed of a comparison can efficiently speed up the matching process [70].

Fingerprint pre-selection techniques can significantly reduce the search space [49]. These techniques can be classified into two main groups: exclusive classification, a fixed number of classes, and continuous classification, fingerprint indexing [7,10,46]. In exclusive classification, a fingerprint database is divided to a fixed number of classes and the query print is compared with those belonging

* Corresponding author.
E-mail addresses: khodadoust.j@gmail.com (J. Khodadoust), khodadoust.a.m@gmail.com (A.M. Khodadoust).

http://dx.doi.org/10.1016/j.patcog.2017.01.022
0031-3203/

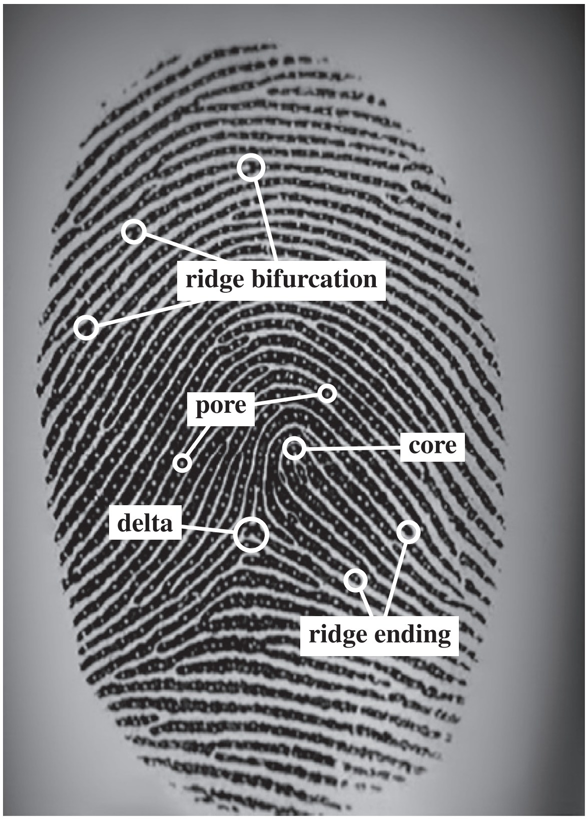


Fig. 1. Gray-scale image of fingerprint (FVC2006 DB2_A, 135_11) [21] and the details extracted from it.

to the same class [46]. However, exclusive classification has two drawbacks:

- Even with fixed number of classes involved exclusive classification is still a challenging problem due to small inter-class variability (much similarity between two images from different fingers), large intra-class variability (large variability in different impressions of the same finger), poor quality fingerprints, and the presence of noise [49].
- The number of classes in which the search space is divided are small and fingerprints are unevenly distributed among them. For instance in Galton-Henry classification scheme, that consists of five major classes: whorl, left loop, right loop, arch, and tended arch, about 93% of the fingerprints belong to three classes right loop, left loop, and whorl [7,10,22,23,49,52].

Therefore, using exclusive classification will not effectively reduce the number of potential candidates to a minimum.

Fingerprint indexing works by associating a query print with index codes or feature vectors that best describe their features [32]. Each extracted feature vector is used to conform an index which is stored in an index table. When a query is performed in the retrieving stage, the number of correspondences between the indices of the query and gallery fingerprints are computed [52].

Extraction of discriminative and reliable features is crucial for fingerprint indexing. Fingerprint indexing algorithm is usually at first is used to quickly choose a subset of candidates and then accurate but slower matching algorithm is employed to determine the final result [63]. Fingerprint indexing causes reduction of the search space of AFIS without missing accuracy [46,52] and can be classified into three approaches [63]:

- *Level-1 indexing*: These approaches use level-1 features, such as the ridge flow structures, pattern types, orientation field, ridge frequency field and singular points (core and delta), and are commonly faster than level-2 indexing approaches.
- *Level-2 indexing*: These approaches are based on level-2 features (minutiae).
- *Combined indexing*: These approaches combine both level-1 and level-2 features to arrive at a better performance.

Other solutions to deal with large databases are related to hardware and parallel computing. Jiang and Crookes [37] presented an efficient implementation of fingerprint matching on field-programmable gate arrays (FPGAs), which takes advantage of an early jump-out (EJO) mechanism to speed up the large database retrieval. Peralta et al. [59] proposed a two-level distributed and parallel framework that allows a parallel search through the fingerprint database, providing an increased system performance. Gutiérrez et al. [28] presented a graphics processing unit (GPU) fingerprint matching system based on minutia cylinder-code (MCC). GPUs provide large-scale parallelism on computing platforms.

In this paper we introduce a new fingerprint indexing algorithm, a combined indexing approach, based on the minutiae pairs and convex core point which employs k-means clustering and candidate list reduction criteria to speed up the search and to improve the accuracy in large databases. k-means clustering and candidate list reduction criteria can effectively reduce the search space and number of candidates in fingerprint matching, and thus achieve higher efficiency and significantly improve the system retrieval performance. Experiments over six data sets demonstrate that the proposed algorithm outperforms some of the best state-of-the-art indexing algorithms.

The remainder of the paper is organized as follows. Section 2 presents a review of some of the related work in fingerprint indexing. Section 3 deals with image enhancement and feature extraction. Section 4 discusses the proposed indexing approach. Section 5 presents experimental results and analysis. Finally, Section 6 draws some conclusions.

2. Related work

Many fingerprint indexing approaches have been reported in the literature. We classify these approaches according to the level-2 features used for indexing into the following categories: non-minutia based approaches, minutia single based approaches, minutiae double based approaches, minutiae triplet based approaches, minutiae quadruplet based approaches, minutia K-Plet based approaches, and minutia cylinder-code (MCC) based approaches. In general, the difference between indexing methods is mainly in the selection of the features and in the retrieving stage. Table 1 compares the proposed algorithm with representative minutiae based fingerprint indexing algorithms.

2.1. Non-minutia based approaches

These indexing approaches are independent of the minutiae. Liu and Yap [46] proposed a representation of orientation fields (OFs) based on a set of polar complex moments (PCMs) for fingerprint indexing. PCMs can describe fingerprint ridge flow structures and are tolerant to spurious orientations in noisy fingerprints. Cappelli [7] described a fingerprint indexing algorithm based on a set of features obtained from ridge orientation and frequency. The main advantage of this algorithm is that the searching speed is high.

Since minutiae contain more discriminative information of fingerprints compared to ridge orientations and frequencies, level-2 indexing approaches should be more accurate [63].

Table 1
Representative minutiae-based fingerprint indexing algorithms.

Algorithm	Local structure topology	Topological features	Indexing scheme	Distortion
Proposed approach	Doubles	All minutiae+convex core point	Clustering	Insensitive
Jayaraman et al. [36]	Singles	All minutiae	Hashing	Insensitive
Tiwari and Gupta [66]	Singles	All minutiae	Hashing	Insensitive
Wang and Hu [69]	Doubles	All minutiae	Quantization	Insensitive
Germain et al. [25]	Triplets	All triangles	FLASH	Insensitive
Bhanu and Tan [4]	Triplets	All triangles	Hashing	Insensitive
Bebis et al. [2]	Triplets	Delaunay triangles	Hashing	Sensitive
Liang et al. [44]	Triplets	Low-order Delaunay triangles	Hashing	Insensitive
Muñoz-Briseño et al. [52]	Triplets	High-order Delaunay triangles	FLASH	Insensitive
Gago-Alonso et al. [22]	Triplets	Delaunay triangles+redundant triangles	Hashing	Insensitive
Uz et al. [68]	Triplets	Delaunay triangles	Hashing	Sensitive
Iloanusi et al. [33]	Quadruplets	All convex quadrilaterals	Clustering	Sensitive
Cappelli et al. [9]	Cylinders	Minutiae cylinders	LSH	Insensitive
Wang et al. [70]	Cylinders	Minutiae cylinders	LSH+Hashing	Insensitive
Su et al. [63]	Cylinders	Minutiae cylinders	LSH+Inverted index table	Insensitive
Mansukhani et al. [50]	K-Plets	All stellar structures with K spokes of variable length	Tree	Insensitive
Bai et al. [1]	K-Plets	All stellar structures with K spokes of variable length	Tree	Insensitive

2.2. Minutia single based approaches

These approaches are based on minutiae singles. Jayaraman et al. [36] proposed an indexing algorithm using features extracted from minutiae singles. They proposed an indexing technique which uses spatial and directional information to index each minutia into a single hash table. Each minutia identified by its spatial (distance) and directional (angle) information from its core point and is inserted exactly once into a hash table. They also proposed a minutiae binary pattern (MBP) representation of the fixed length feature vectors built from each minutia for an accurate match at the time of searching. Tiwari and Gupta [66] proposed an indexing algorithm similar to Jayaraman et al. [36] algorithm. They use a coaxial Gaussian track code (CGTC), a fixed length feature vector built from each minutia, instead of the MBP in [36].

Let $M = \{m_i\}_{i=1}^{n}$ be a set of the selected minutiae, where m_i is ith minutia and n is the number of minutiae. These indexing approaches allow to consider only n minutiae during indexing, thus whose complexities are $O(n)$. However the accuracy of this indexing approach is generally lower than the indexing approaches based on two or more minutiae due to the fact that the number of distinctive information in one minutia is less than that in the relation between two or more minutiae.

Jayaraman et al. [36] and Tiwari and Gupta [66] indexing strategies are invariant to rotation and translation and whose complexity is $O(n)$. However, these indexing strategies have several drawbacks. The performance of these strategies relies heavily on the core point. In some of the fingerprints, i.e. arch and twin loop, there are no core point or more than one core point. Furthermore, the core point in noisy images could not be present or false core point can be detected. Jayaraman et al. [36] and Tiwari and Gupta [66] also used several sectors that each minutia lying in a sector. When a minutia placed in boundary between two sectors, some problems arise in this case and their algorithms do not have robust strategies to deal with this problem. Belhadj et al. [3], Gupta and Gupta [27], and Zhu et al. [76] proposed a core point detection algorithm that can solve problems of core point detection in [36,66] and can improve their algorithms.

2.3. Minutiae double based approaches

One of the indexing approaches is based on Minutiae doubles. Wang and Hu [69] proposed an indexing algorithm using features extracted from minutiae pairs. Their feature vectors are invariant to any geometric transformations of the fingerprints. They used a binary bit-string for indexing. Their scheme has the properties of non-invertibility, revocability, and multiple template independence. Because the number of minutiae pairs are $n(n-1)/2$, thus they should consider $O(n^2)$ minutiae pairs during matching.

2.4. Minutiae triplet based approaches

These approaches are based on Minutiae triplets and triangles formed by them. The use of triangles was first introduced by Germain et al. [25] and used as descriptor in their indexing approach. They considered all possible minutiae triplets that can be formed with the minutiae. To increase true correspondences, pairwise registration is carried out between each probe and template fingerprint using transformation parameter clustering. Bhanu and Tan [4], Choi et al. [13], and Biswas et al. [5] improved the Germain et al. [25] algorithm by using new feature of minutiae triplets. Because the number of minutiae triplets are $n!/(n-3)!3!$, thus they should consider $O(n^3)$ minutiae triplets during indexing, which would be very time-consuming.

Bebis et al. [2] used the triangles resulting from the Delaunay triangulation of the minutiae set M. Delaunay triangulation is a special type of triangulation and minimizes the maximum angle of each triangle, i.e., tries to avoid long skinny triangles when triangulating points [6]. Delaunay triangulation allows to consider only $O(n)$ minutiae triplets during indexing.

An important issue when employing Delaunay triangulation for indexing is that the distortions such as Minutiae shifts (displacements), spurious minutiae, and missing minutiae could change the triangulation locally by eliminating important triangles or introducing spurious triangles. To address these issues, Liang et al. [44], Muñoz-Briseño et al. [52], and Gago-Alonso et al. [22] proposed an extension of Delaunay triangulation. The main problem of Liang et al. [44] algorithm is that the number of generated triangles are low, thus the identification accuracy will be reduced. Also, some geometric features are lost, affecting correctness of the fingerprint indexing. In general, this algorithm only reduces the negative effect caused by Minutiae displacements. The main drawback of Muñoz-Briseño et al. [52] and Gago-Alonso et al. [22] approaches is the selected feature vectors and effect of distortion (Minutiae shifts) on these feature vectors. They used ridge count to generate feature vectors. Problems arise when the ridge-counting line (triangle side) is parallel to the ridge structures, the line may meet the same ridge at no point, at one point, at two points, or at more than two points, due to skin deformation. Also, if there are large ridge-gaps (scars) in the intersections of ridges and ridge-counting line, the ridge count feature will be wrong.

Uz et al. [68] proposed a matching hierarchically based on minutiae quality. In this algorithm matching is performed hierarchically, starting at the top level of the Delaunay hierarchy which contains high quality minutiae only and moving down to lower levels and ending at the bottom level of the Delaunay hierarchy which contains all possible minutiae. The main drawback of Uz et al. [68] approach is the use of Delaunay triangulation that was proposed by Bebis et al. [2]. This triangulation is more sensitive to distortion and noise, and does not have robust strategy to deal with missing or spurious minutiae.

2.5. Minutiae quadruplet based approaches

These approaches are based on Minutiae quadruplets. There are three types of irregular quadruplets [33]: concave, convex, and reflex (crossed). In these approaches, the convex quadruplets are used to index fingerprint images. Iloanusi [32] and Iloanusi et al. [33] used minutiae quadruplets and clustering technique to increase the indexing performance. Lovasz et al. [47] proved that the minimum number of convex quadrilaterals determined by n points in general position in the plane are $(0.37501)\binom{n}{4}+O(n^3)$, thus they should consider $O(n^4)$ minutiae quadruplets during indexing, which would be very time-consuming.

2.6. Minutia K-Plet based approaches

Minutia K-Plet based approaches are based on K-nearest neighbors of each minutia and are an extension of the nearest neighbor (NN) local based structure. Local structures in these approaches are formed by a central minutia and K-nearest neighbors of each minutia. Minutia K-Plet was first proposed by Chikkerur et al. [11] to encode the local neighborhood of each minutia. Since each minutia chooses K other minutiae from its local neighborhood, the number of K-Plets are $O(n)$. Minutia K-Plet based technique is invariant to rotation and translation.

Mansukhani et al. [50], instead of using the graph in [11], employed a tree based matching algorithm. The search time of their matching algorithm only depends on the index tree configuration and is independent of the number of enrolled templates. Bai et al. [1] proposed a multipath indexing strategy based on minutia K-Plet and 3-level trees. The main drawback of K-Plet is that it does not ensure a connection between all parts of the fingerprint. Ham et al. [29] have shown superiority of the Delaunay triangulation in comparison with K-Plet. K-Plet needs at least 4 edges by point without warranting a connected graph, whilst Delaunay triangulation ensures connectivity using approximately 2.6 edges by point [29].

2.7. Minutia cylinder-code (MCC) based approaches

These approaches are based on minutia cylinder-code (MCC) proposed by Cappelli et al. [8]. MCC combines the advantages of both NN and fixed radius local structures, and encodes spatial and directional relationships between the central minutia and its neighborhood by projecting its neighbor minutia to a three dimensional (3D) space in the form of a cylinder. The cylinder is set up by using the direction difference between minutiae as the height and the radius as the base. Cappelli et al. [9] used the MCC to generate fixed length binary vectors. In the retrieving stage, a local sensitive hashing (LSH) algorithm is used for finding similarities between the binary vectors. LSH is a generic probabilistic framework and is effective in finding approximate nearest neighbor of high dimensional vector. In general, data objects or feature vectors that have 10 or more dimensions are considered high dimensional vectors [30]. The basic constituent of an LSH scheme is a family of hash functions $H = H_1, H_2, \ldots, H_l$ which are likely to produce hash collisions for similar items [63,72]. In their algorithm, hash functions are randomly generated. However, their algorithm do not have robust strategies to deal with missing or spurious minutiae especially in low quality or small fingerprints.

Wang et al. [70] proposed a theoretical framework to learn compact binary hash codes from high-dimensional binary representations of fingerprints. Compact binary codes usually are exploited to speed up the search in large-scale applications. They also developed a hierarchical indexing scheme that combines geometric hashing and the LSH (Geo-LSH). Su et al. [63] proposed a learning-based fingerprint pose estimation algorithm which can register fingerprints into a unified finger coordinate system. They combined pose estimation algorithm with an improved LSH algorithm for MCC descriptor.

If the number of minutiae templates of a gallery are t and each template contains m minutiae, LSH and geometric hashing are used to index $O(tm)$ basis points, defined by minutiae, and dictionaries, respectively. However, MCC has very high-dimensional feature vectors. Furthermore, LSH is an approximate searching method which is not quite applicable for the cases requiring high accuracy rate.

3. Image enhancement and features extraction

3.1. Preprocessing and fingerprint image enhancement

The first processing step after fingerprint image acquisition is segmentation, that is, dividing a fingerprint image into foreground and background regions. Several fingerprint image segmentation techniques have been proposed in the literature. In this paper, we employ fingerprint image segmentation proposed by Fahmy and Thabet [15]. After carrying out the segmentation, fingerprint image enhancement has been employed to reduce noise and to increase the contrast between ridges and valleys in the gray-scale fingerprint images before further processing. The quality of the fingerprint images strongly affects the performance of AFISs [64]. Enhance process is performed on both the gallery and probe images. A fingerprint enhancement algorithm should not result in any spurious ridge structures because spurious ridge structure affects the accurate extraction of minutiae that cause degradation of the fingerprint matching accuracy.

Various fingerprint enhancement techniques have been proposed in the literature. In this paper, we use the Gabor filtering proposed by Hong et al. [31]. Gabor filtering is one of the most extensive techniques to fingerprint enhancement. Fig. 2 shows results of the fingerprint image enhancement and quality estimation on a gray-scale fingerprint image. Segmentation and quality estimation are addition in our case, not included in [31].

Various fingerprint image quality estimation methods, dependent on how we use the quality value, have been proposed in the literature. Fingerprint quality estimation is one of the most important processes for AFISs. Fingerprint features such as minutiae in high-quality regions must be treated with higher priority and scored differently, compared to features in low-quality regions; the fingerprint features, which are detected in low-quality region, are unreliable and prone to errors. Without quality estimation, the AFISs accuracy cannot be further improved [60]. Fingerprint quality relies on various factors such as sensor types, finger conditions, and capture types. In this paper, we employ the NIST fingerprint image quality (NFIQ) [53] to estimate the quality of fingerprint image. The NFIQ is the most widely used method to estimate the quality of fingerprint image. The method computes four maps in each block: direction map, low contrast, low flow, and high curve. These maps are summarized into one quality map, containing five levels of quality (i.e., an integer in the range of 0 to 4 where 0 and 4 represents the lowest and highest quality, respectively [71].

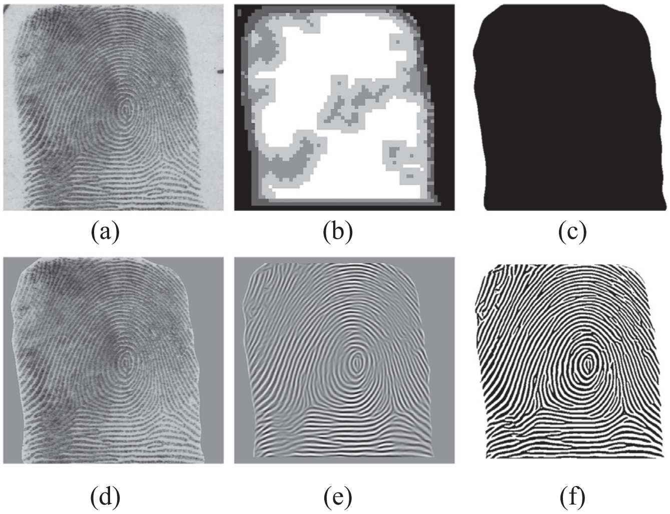


Fig. 2. Fingerprint image enhancement and quality estimation. (a) original image (NIST SD4, f0258_04) [54], (b) quality map of (a), (c) the region of interest (ROI) of (a), (d) segmented image, (e) filtered image by the Gabor filtering , and (f) binarized image.

3.2. *Minutiae extraction*

Fingerprint minutiae extraction is a critical issue in fingerprint recognition. Several minutiae extraction techniques have been proposed in the literature. Generally, minutiae are extracted from the skeletonized fingerprint image. Thinning is a morphological operation that reduces the line thickness to 1 pixel, resulting in a skeleton image. Thinning can be applied on both gray-scale and binary images. Ideally, the width of the skeleton should be strictly 1 pixel but this is not always true. Furthermore, some methods use advanced techniques for minutiae extraction which does not require thinning operations. Fronthaler et al. [16] employed parabolic symmetry to detect the position and direction of a minutia. Liu and Cao [45] proposed an approach to extract minutiae from orientation and frequency fields. In this paper, we use the enhanced thinning algorithm proposed by Zhao and Tang [75] to make sure that the ridges are 1 pixel width (see Fig. 7 (b) and (c)).

The output of a minutiae extraction stage is a set of minutiae. Each minutia is represented by its location coordinate, direction, and qualitiy, forming a 4-tuple $m_i = (x_i, y_i, \theta_i, q_i)$. In order to extract minutiae, the concept of crossing number is used [75]. The crossing number $cn(p)$ of a pixel p in a binary image is defined as half the sum of the differences between pairs of the adjacent pixels in the eight-neighborhood of p, is then computed for both the gallery and query fingerprints in the foreground region.

Other than the method used to extract the minutiae, two types of errors can be produced [57]: missing (non-detected real minutiae) and spurious minutiae (non-existing detected minutiae). Such errors might be produced by poor-quality images, but also due to scars or creases in the fingerprint pattern, or the use of thinning stage. Regarding erroneous minutiae, a missing minutia can only be recovered (extracted) by improving the minutiae extraction method; otherwise, a spurious minutia can be detected and removed from the minutiae set by postprocessing techniques.

Because the spurious minutiae are more frequent than missing ones, postprocessing techniques can be considered in order to prune them and improve the results of the matching process. We use the postprocessing techniques [75] and convex hull filter [57] to remove the false minutiae on the edge of the fingerprint foreground to extract the true minutiae.

3.3. *Singular points detection*

Singular points (SPs) play a key role in AFISs. SPs of fingerprint, defined as where the ridge curvature is the highest or the OF is discontinuous, are essential for enrollment and identification [76]. Any SP can be classified as either core or delta which is defined as a concentrate region where the ridge curvature converges to a local maximum or local minimum, respectively [27]. There are two core point types in fingerprint images, i.e., upper (convex) and lower (concave) core points [43,76]. In this paper, we use convex core point detection algorithm was proposed by Zhu et al. [76]. They proposed a walking algorithm which can walk directly to the SPs without scanning the whole fingerprint image. Because the walking algorithm is a fast method for detecting fingerprint SPs, it can be used in real time AFISs.

Note that the success of walking algorithm relies on good estimation of the OF. Thus a better estimation of the OF improves the performance of the walking algorithm. In this paper, we use the OF estimation scheme proposed by Turroni et al. [67]. This scheme modifies the existing techniques, gradient-based method used in walking algorithm, and it can lead to relevant accuracy improvements. Furthermore, Zhu et al. [76] recommended the combination of the walking algorithm and the AMF, angle matching index (AMI) combined with convergence index (COIN) filter, based method [61] for applications which care more about accuracy and the combination of the walking algorithm and the Poincaré index (PI) based method [49] for applications requiring both accuracy and efficiency. The PI method has several drawbacks. It has a problem when the SPs is near the border. Furthermore, PI method is very sensitive to noise and behaves badly when images have poor quality [3]. The AMF method is an accurate method for SPs detection. However, this method is time-consuming.

Walking algorithm, PI method, AMF method, and their combinations do not have robust strategies to deal with the arch-type fingerprints. Furthermore, in some cases, walking algorithm cannot detect all SPs (see Fig. 3(d)). Since the walking algorithm is designed to be fast, not too accurate, we use the combination of the walking algorithm and SPs detection algorithm, orientation-deviation (OD) combined with extended Poincaré index (EPI), proposed by Belhadj et al. [3] to make sure that the convex core point is stable and unique. Belhadj et al. [3] algorithm is capable of detecting the convex core point in the arch-type fingerprints. Furthermore, this algorithm is more robust to noise and less sensitive to partial fingerprints and location of singularities at borders.

4. Indexing features and technique

After performing the fingerprint features extraction, we obtain the true minutiae and convex core point from the fingerprint image. In this section, we propose an indexing algorithm that uses minutiae pairs and the convex core point u, combined with a strategy that employs a weighting method to take into account the significance and importance of feature vectors.

By pairing up any two minutiae m_i and m_j, in the set M, a minutiae pair (m_i, m_j) can be constructed. Hence, our weighting method is performed by assigning a weight to take into account the significance and importance of each feature vector in retrieving process. In our algorithm, minutiae qualities have four values q_1, q_2, q_3, or q_4 where q_1, q_2, q_3, and q_4 correspond to high-, medium-, low-, and lower-quality minutiae, respectively.

We construct minutiae pairs based on all possible combination of the minutiae, which contain high-, medium-, low-, and lower-quality minutiae. First, we only consider the high-quality minutiae

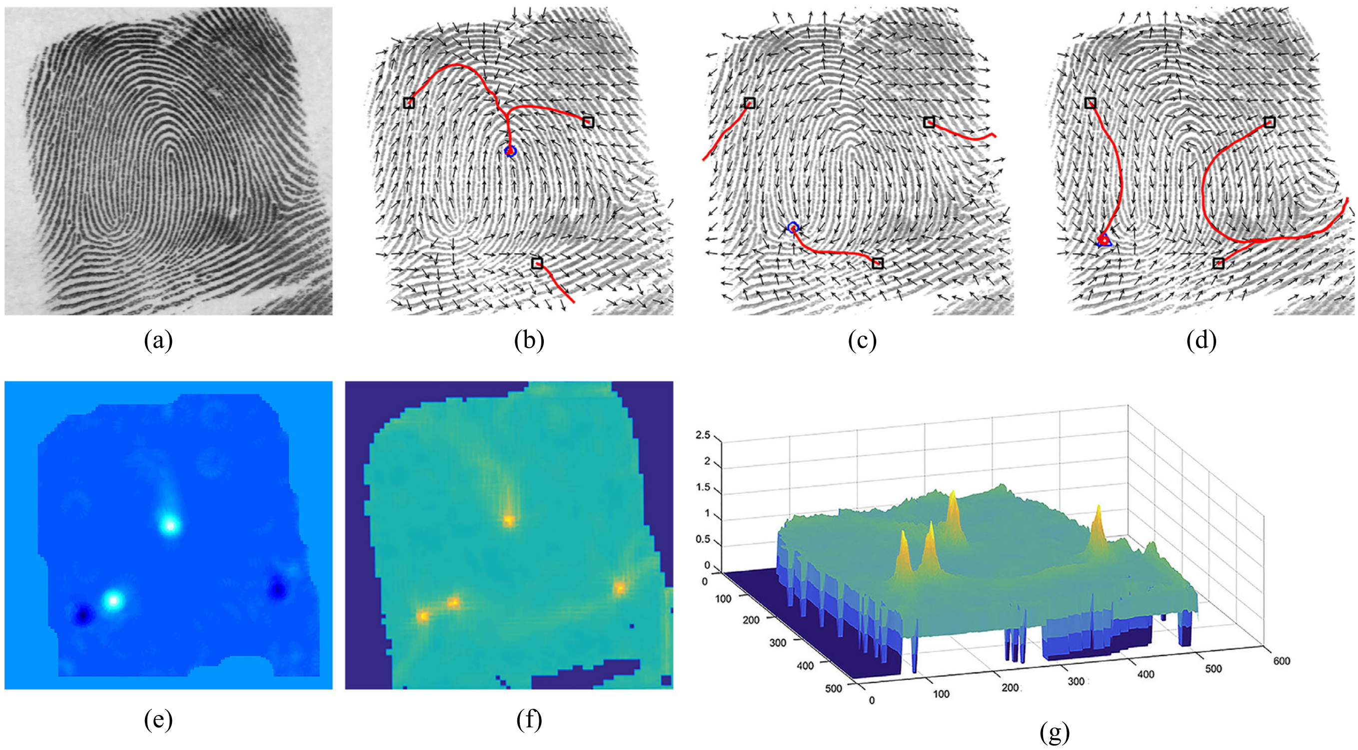


Fig. 3. Singular points detection: (a) fingerprint image (f1522_03) from NIST SD4 [54], (b) walking from the given starting points on walking directional field WDF_{uc} (convex core point detection) [76], (c) WDF_{lc} (concave core point detection), (d) WDF_d (delta point detection), (e) AMI index image [61], (f) 2D orientation field energy map (OFEM) [3], and (g) 3D OFEM. Starting points in top row are marked with black square (□).

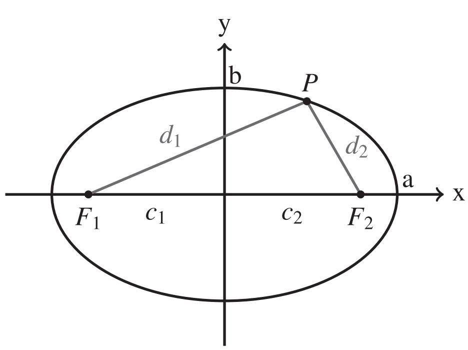


Fig. 4. The ellipse and some of its mathematical properties.

and then construct their minutiae pairs. For medium-, low-, and lower-quality minutiae and their combinations also do the same operation. Finally, we consider all minutiae pairs and convex core point and we generate triangles of related to them. In the triangles, each line segment constructed from minutiae pairs is a side of a triangle. In order to construct the triangles, we calculate the distance from any minutia belongs to a minutiae pair to convex core point by Euclidean distance in Eq. (5). For each triangle, we generate an ellipse and we adapt any vertices pair (minutiae pair) belongs to a side of triangle with focal points of ellipse. Other vertex of triangle (convex core point) has been placed on the curve of ellipse.

Many of the different definitions can be found in the literature for ellipse. An ellipse is a curve on a plane that surrounds two focal points (focus or foci) such that the sum of the distances to the two focal points is constant for every point on the curve [51]. This property is illustrated in Fig. 4. In the figure, the distances of point p on the curve to the negative (F_1) and positive (F_2) focal points are correspondingly denoted by d_1 and d_2. The property states that:

$$d_1 + d_2 = 2a \tag{1}$$

Ellipses have two perpendicular and symmetric axes. These axes intersect at the center of the ellipse due to this symmetry. The larger of axis, which corresponds to the larger distance between antipodal points on the ellipse, is called the major axis. The smaller axis, and the smaller distance between antipodal points on the ellipse, is called the minor axis. The equation of an ellipse whose major and minor axes coincide with the Cartesian axes is:

$$\left(\frac{x}{a}\right)^2 + \left(\frac{y}{b}\right)^2 = 1 \tag{2}$$

where x, y are the coordinates of any point on the ellipse, a, b are the radius along the x-axis and y-axis, respectively. For an ellipse given by Eq. (2), with $a > b > 0$, the focal points are points that lie along the x-axis in the positions $F_1 = (c_1, 0)$ and $F_2 = (c_2, 0)$ with

$$a^2 = b^2 + c^2 \tag{3}$$

The distance from the center to either focal point is the fixed value c. The length of the whole major axis is $2a$, the distance between the focal points is $2c$, and the distance across the ellipse in the shorter direction is $2b$. Fig. 8 shows the overall scheme of our indexing algorithm.

4.1. Creating the index

Indexing features and the retrieval strategy are two essential aspects of fingerprint indexing. We apply the utility of features extracted from ellipse to construct feature vector $\mathbf{V_e}$. The feature vector $\mathbf{V_e}$ has five dimensions and therefore may also be referred to as a point in five-dimensional (5D) space. The 5D point associated to minutiae pair (m_i, m_j), from true minutiae set, and convex core

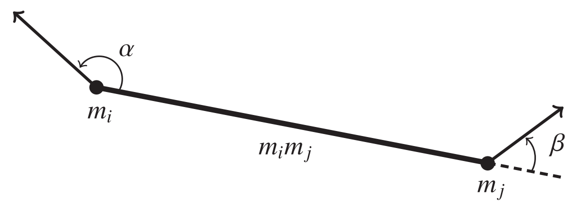


Fig. 5. The pair (α, β) formed by line segment $m_i m_j$.

point u is represented by an ellipse with major axis A=2a and minor axis B=2b whose two minutiae m_i and m_j (the vertices pair corresponding to the side $m_i m_j$ of triangle $\triangle m_i m_j u$) are adapted to the focal points of ellipse (see Figs. 4 and 6(c)). The feature vector $\mathbf{V_e}$ is constructed as follows:

$$\mathbf{V_e} = (A, B, \theta, \alpha, \beta) \tag{4}$$

where A is the sum of the sides $m_i u$ and $m_j u$ of $\triangle m_i m_j u$, B denotes minor axis of ellipse that is calculated with the Eq. (3), θ is the angle between side $m_i m_j$ and the side $m_i u$ of $\triangle m_i m_j u$, α is the angle between the reference direction of the side $m_i m_j$ and the orientation of m_i in the counter-clockwise direction, and β is defined analogously (see Fig. 5). According to Fig. 6, the feature vector $\mathbf{V_e}$ is constructed as follows:

For two minutiae m_i and m_j, from minutiae set M, and convex core point u, sides d_{iu} and d_{ju} are obtained by Euclidean distance:

$$\begin{aligned} d_{ij} &= \sqrt{(x_i - x_j)^2 + (y_i - y_j)^2} \\ d_{ju} &= \sqrt{(x_j - x_u)^2 + (y_j - y_u)^2} \\ d_{iu} &= \sqrt{(x_i - x_u)^2 + (y_i - y_u)^2} \end{aligned} \tag{5}$$

where (x_i, y_i), (x_j, y_j), and (x_u, y_u) are coordinates (x, y) of minutiae m_i, m_j, and convex core point u, respectively. According to ellipse property:

$$A = 2a = d_{iu} + d_{ju}$$

$$C = 2c = d_{ij}$$

According to Eq. (3):

$$B = 2b = \sqrt{(d_{iu} + d_{ju})^2 - (d_{ij})^2}$$

Also, according to cosine rule:

$$\theta = \arccos \frac{(d_{iu})^2 + (d_{ij})^2 - (d_{ju})^2}{2 d_{iu} d_{ij}}, \quad 0 < \theta < 180^\circ$$

According to Eq. (6) and Fig. 5, α and β are calculated as follows:

$$\alpha = arctan \frac{Y}{X}$$

$$\beta = \alpha + \theta_j - \theta_i$$

$$\begin{bmatrix} X \\ Y \end{bmatrix} = \begin{bmatrix} \cos\theta_i & \sin\theta_i \\ \sin\theta_i & -\cos\theta_i \end{bmatrix} \begin{bmatrix} x_j - x_i \\ y_j - y_i \end{bmatrix} \tag{6}$$

where θ_i and θ_j are orientations of two minutiae m_i and m_j, respectively.

Due to distortion, it is reasonable to consider the tolerance areas. A tolerance area is decided by cells. The feature vector $\mathbf{V_e}$ is a point in the continuous space; $\mathbf{V_e}$ is discretized into $N_E = N_A \times N_B \times N_\theta \times N_\alpha \times N_\beta$ cells. Each cell can be uniquely identified by five indices (i, j, k, r, s) that denote its position in the discrete space, with $i \in I_A = \{n \in \mathbb{N}, 1 \le n \le N_A\}$, $j \in I_B = \{n \in \mathbb{N}, 1 \le n \le N_B\}$, $k \in I_\theta = \{n \in \mathbb{N}, 1 \le n \le N_\theta\}$, $r \in I_\alpha = \{n \in \mathbb{N}, 1 \le n \le N_\alpha\}$, and $s \in I_\beta = \{n \in \mathbb{N}, 1 \le n \le N_\beta\}$.

Let $\delta_A = A/N_A$, $\delta_B = B/N_B$, $\delta_\theta = 180/N_\theta$, $\delta_\alpha = 360/N_\alpha$, and $\delta_\beta = 360/N_\beta$. Note that $\delta_A = 2p - 1$ and $\delta_B = 2p' - 1$, where $\{p, p'\} \in \mathbb{N}$. Also let

$$d\varphi_i = \left(i - \frac{1}{2}\right)\delta_A \tag{7}$$

$$d\varphi_j = \left(j - \frac{1}{2}\right)\delta_B \tag{8}$$

$$d\varphi_k = \left(k - \frac{1}{2}\right)\delta_\theta \tag{9}$$

$$d\varphi_r = \left(r - \frac{1}{2}\right)\delta_\alpha \tag{10}$$

$$d\varphi_s = \left(s - \frac{1}{2}\right)\delta_\beta \tag{11}$$

be the center of the cells with indices i, j, k, r, s. For each cell (i, j, k, r, s), a numerical value $E_n(i, j, k, r, s)$ or discrete feature vector $\mathbf{V_d}$ is obtained.

Because $\mathbf{V_d}$ is discrete and fixed-length vector therefore making it feasible and beneficial for fingerprint indexing based on clustering. Clustering refers to partitioning of data objects or observations into relatively homogeneous clusters. The data objects in a cluster are similar to one another, and dissimilar to data objects

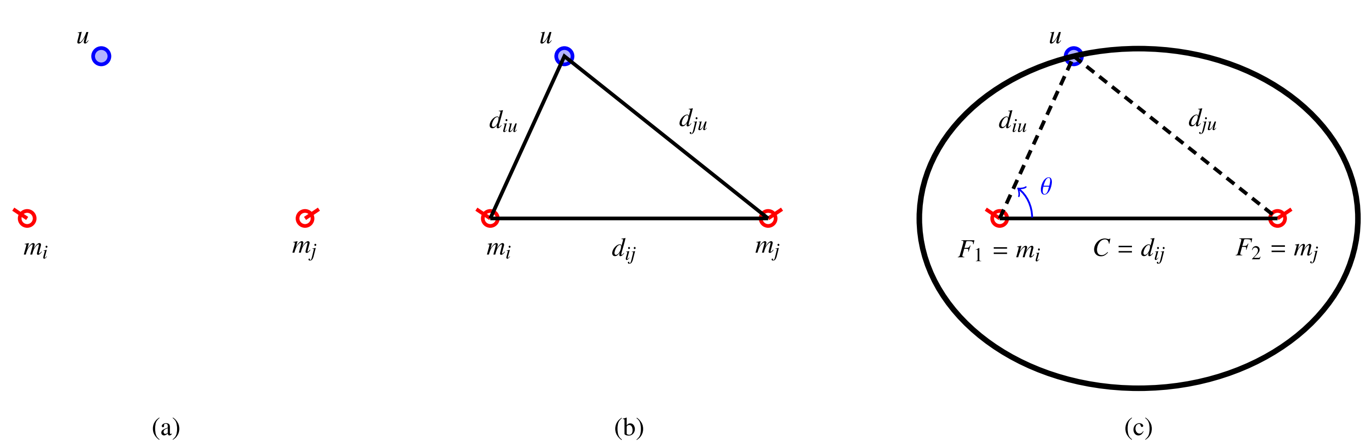


Fig. 6. Generation of the feature vector $\mathbf{V_e}$: (a) two minutiae m_i and m_j from minutiae set M and convex core point u, (b) creating triangle $\triangle m_i m_j u$, (c) creating ellipse.

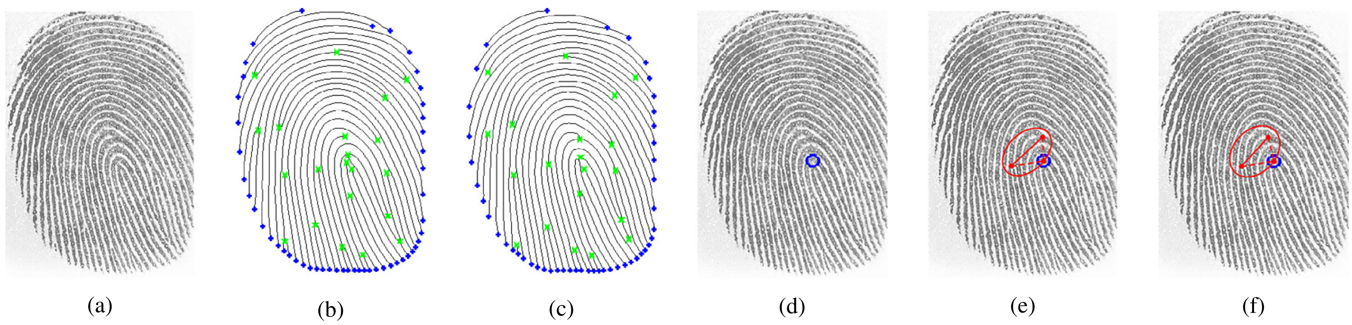


Fig. 7. Feature extraction: (a) original fingerprint image 3_3 from FVC2000 DB2_A [18], (b) extracted minutiae based on valleys (c) extracted minutiae based on ridges (d) detected convex core point, (e) extracted ellipse by (b) and (d), and (f) extracted ellipse by (c) and (d). Border minutiae, true minutiae, and convex core point are marked with blue plus (+), green cross (×), and blue circle (∘), respectively.

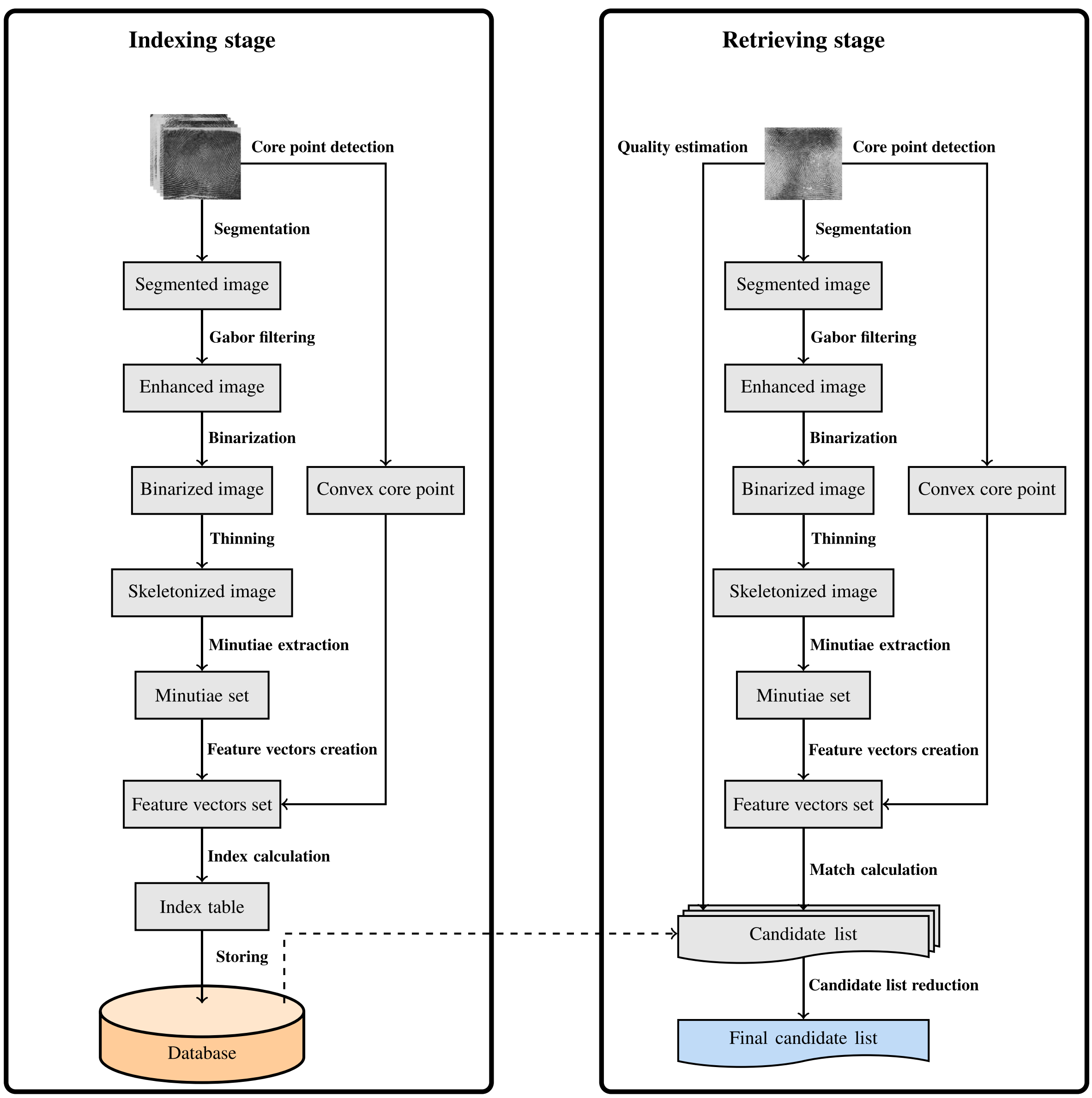


Fig. 8. Proposed fingerprint indexing and retrieving.

in other clusters [30]. Partitioning clustering methods are one of the clustering methods that organize the data objects of a data set into several joint, in overlapping approaches, or disjoint, in exclusive approaches, clusters [30,42]. The main algorithms of partitioning clustering are k-means [48], k-medoids [40] and their variants. These algorithms partition data objects into k clusters requiring the user to specify the number of clusters.

The k-means clustering algorithm is a distance based algorithm and is one of the most widely used algorithms for clustering. The main reasons for popularity of k-means clustering algorithm are efficiency, ease of implementation, simplicity, and empirical success [35]. However, k-means algorithm is sensitive to outliers and often terminates at a local optimum and is not guaranteed to converge to the global optimum [30]. Outliers are data objects that do not comply with the general model or behavior of the data. To overcome the local optimum problem, Han et al. [30] suggested running the k-means algorithm multiple times with different initial cluster centers. The complexity of the k-means algorithm is *O(nkt)*, where *n*, *k*, and *t* are the total number of data objects, clusters, and iterations, respectively ($t \ll n, k \ll n$).

The K-medoids algorithm is modification of k-means algorithm to diminish sensitivity of k-means algorithm to outliers. The main drawback of k-medoids algorithm is that its computational complexity is high. The complexity of each iteration in the k-medoids algorithm is $O(k(n-k)^2)$.

The number of clusters is an important parameter for clustering algorithms like k-means. This parameter also controls the proper granularity of cluster analysis. It can be regarded as finding a good balance between accuracy and compressibility in cluster analysis [30]. Cross-validation technique [30] and x-means [56] method are well-known techniques for computing the number of clusters in the k-means clustering algorithms.

In our proposal, we employ k-means clustering algorithm proposed by Khanmohammadi et al. [42]. They proposed a hybrid method that combines k-harmonic means (KHM) and overlapping k-means algorithms (OKM). Their method uses the output of KHM method to initialize the cluster centers of OKM method. Even though they used their clustering method in medical domain but it could be applied to any other domain. We also use cross-validation technique to determining the number of clusters. This technique divides the data set *D* into *n* parts. Next, uses *n*-1 parts to build a clustering model, and uses the remaining part to test the quality of the clustering. For any $k > 0$, we repeat this process *n* times to derive clusterings of *k* clusters by using each part in turn as the test set. The overall quality measure is obtained from the average of the quality measure. Finally, we compare the overall quality measure with respect to different values of *k*, and select the best number of clusters.

There are many methods for measuring the quality of a clustering. In this paper, we employ silhouette coefficient proposed by Rousseeuw [62]. Suppose data set *D* is partitioned into *k* clusters, $C_1, C_2, \ldots, C_k$. For each data object $o \in D$, we formally calculate *a(o)* as follows:

$$a(o) = \frac{\sum_{o' \in C_i : 1 \le i \le k, o \ne o'} dist(o, o')}{|C_i| - 1} \tag{12}$$

Similarly *b(o)* is calculated as follows:

$$b(o) = \min_{C_j : 1 \le j \le k, j \ne i} \left\{ \frac{\sum_{o' \in C_j} dist(o, o')}{|C_j|} \right\} \tag{13}$$

The silhouette coefficient of *o*, *s(o)*, is then defined as:

$$s(o) = \frac{b(o) - a(o)}{\max\{a(o), b(o)\}} \tag{14}$$

The value *s(o)* is between −1 and 1. When $s(o) = 1$, the cluster containing *o* is the preferable case. However, when $s(o) < 0$, the cluster is a bad situation and should be avoided.

4.2. Retrieving the candidate list

A retrieval strategy is a method used to select fingerprints from a gallery with similar feature vectors to a probe under consideration that makes the candidate list.

Fast look-up algorithm for structural homology (FLASH) [25], vantage point (VP) trees [72], and locality-sensitive hashing (LSH) [9,72] are suitable for high-dimensional data. FLASH is a high-dimensional indexing algorithm akin to geometric hashing that has been used for similarity searching in a number of other domains. VP trees are a promising class of tree structures for high-dimensional data and are constructed in a top-down manner [72]. LSH is a generic probabilistic framework and is based on the idea that if two vectors are similar, then after projecting into a lower-dimensional subspace, they will remain similar [9,72]. Indyk and Motwani [34] and Gionis et al. [26] introduced an LSH scheme based on bit sampling and explained how to map other vector spaces into a Hamming space to use the bit sampling scheme more generally.

Most of clustering algorithms for high-dimensional data do not work properly and traditional distance measures can be ineffective on high-dimensional data. High-dimensional data clustering methods can be classified into two groups: dimensionality reduction methods and subspace clustering methods. In our proposal, the dimensionality is not high and we can employ traditional clustering algorithms.

In cluster search, each probe is compared with the cluster mean, center of clusters, vectors to the close clusters followed by searching of the retrieved clusters. Fingerprint search is finally performed on a cluster to find the templates close to the query print.

In this paper, we employ KHM-OKM method in [42] for retrieving the candidate list. This method includes four main steps as following:

1. Find the center of clusters using the KHM method as follows:

$$\vec{z_j} = \frac{\sum_{i=1}^{n} m(\vec{z_j}|\vec{v_i}) w(\vec{v_i}) \vec{v_i}}{\sum_{i=1}^{n} m(\vec{z_j}|\vec{v_i}) w(\vec{v_i})} \tag{15}$$

   where, $\vec{v_i}$ is an n-dimensional vector, $\vec{z_j}$ is a cluster center, $m(\vec{z_j}|\vec{v_i})$ is the membership of vector $\vec{v_i}$ to centroid of the cluster *j* calculated by Eq. (16) , and $w(\vec{v_i})$ is the weight associated with vector $\vec{v_i}$ calculated by Eq. (17)

$$m(\vec{z_j}|\vec{v_i}) = \frac{||\vec{v_i} - \vec{z_j}||^{-p-2}}{\sum_{j=1}^{k} ||\vec{v_i} - \vec{z_j}||^{-p-2}} \tag{16}$$

   where *p* is a free parameter (typically $p \ge 2$)

$$w(\vec{v_i}) = \frac{\sum_{j=1}^{k} ||\vec{v_i} - \vec{z_j}||^{-p-2}}{\left(\sum_{j=1}^{k} ||\vec{v_i} - \vec{z_j}||^{-p}\right)^2} \tag{17}$$

2. Initialize the OKM method based on the results of previous step.
3. Obtain the final cluster centers using the OKM method as follows:

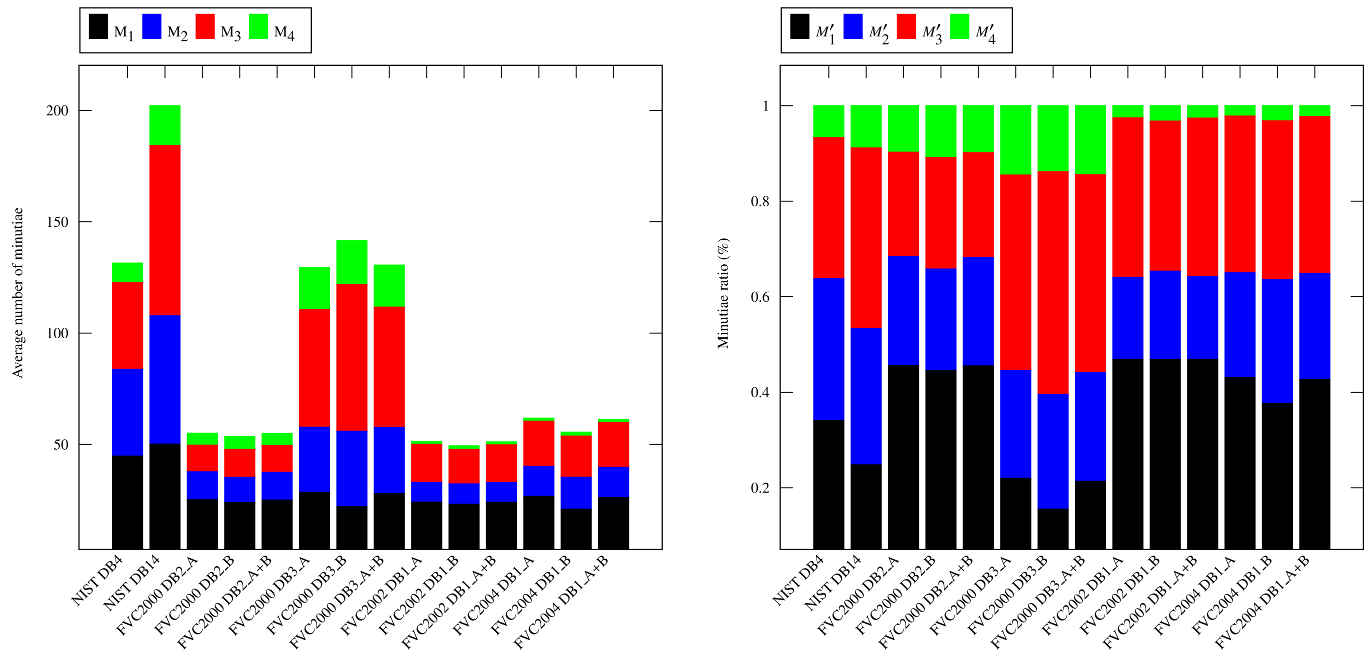
$$\vec{z_j} = \frac{1}{\sum_{v_i \in C_j} \frac{1}{\delta_i^2}} \sum_{v_i \in C_j} \frac{1}{\delta_i^2} \cdot \left(\delta_i \times \vec{v_i} - \sum_{\vec{z_j} \in C(\vec{v_i})/\vec{z_i}} \vec{z_j} \right) \tag{18}$$

   Where $C(\vec{v_i})$ is the list of all clusters that $\vec{v_i}$ belongs to and it is updated, and δ_i corresponds to the total number of clusters that $\vec{v_i}$ belongs to (in this case $\delta_i = |C(\vec{v_i})|$).
4. Identify clusters that each vector belongs to using Eq. (19)

$$m(\vec{z_j}|\vec{v_i}) = \frac{||\vec{v_i} - \vec{z_j}||^2}{\sum_{j=1}^{k} ||\vec{v_i} - \vec{z_j}||^2} \tag{19}$$

In the offline stage, KHM-OKM method is used to store all the feature vectors of each fingerprint in clusters.

In the online stage, given a prob, for each of its feature vectors corresponding clusters are found and a similarity score, or distance measure, is obtained for each fingerprint belongs to a gallery. Then, measured similarity scores are multiplied to quality weights. In our proposal, feature vectors have ten weights w_1, w_2, w_3, w_4, w_{12}, w_{13}, w_{14}, w_{23}, w_{24} or w_{34} where w_1, w_2,



(a) Average number of minutiae.

(b) Minutiae ratio (%).

Fig. 9. Comparison of the average number of minutiae obtained using different quality thresholds.

w_3, w_4, w_{12}, w_{13}, w_{14}, w_{23}, w_{24} and w_{34} correspond to quality of the minutiae pairs high-quality and high-quality, medium-quality and medium-quality, low-quality and low-quality, lower-quality and lower-quality, high-quality and medium-quality, high-quality and low-quality, high-quality and lower-quality, medium-quality and low-quality, medium-quality and lower-quality, low-quality and lower-quality, respectively ($w_1 > w_{12} > w_2 > w_{13} > w_{23} > w_3 > w_{14} > w_{24} > w_{34} > w_4$ and $w_1 + w_2 + w_3 + w_4 + w_{12} + w_{13} + w_{14} + w_{23} + w_{24} + w_{34} = 1$).

Finally, we apply the candidate list reduction criteria as proposed by Cappelli et al. [10] to reduce the candidate list.

Various approaches are proposed to reduce the candidate list in the literature. These approaches can be classified into two reduction criteria: fixed threshold and top ranking. The fixed threshold only selects fingerprints with indexing score higher than a fixed threshold. The top ranking retain a fixed number of candidates with the highest scores. Cappelli et al. [10] proposed two reduction criteria: variable threshold on score difference and variable threshold on score ratio. These two reduction criteria are similar. The only difference is that variable threshold on difference is based on score differences instead of score ratios. Their reduction criteria allow a significant reduction of the candidate list with great improvement of the indexing performance.

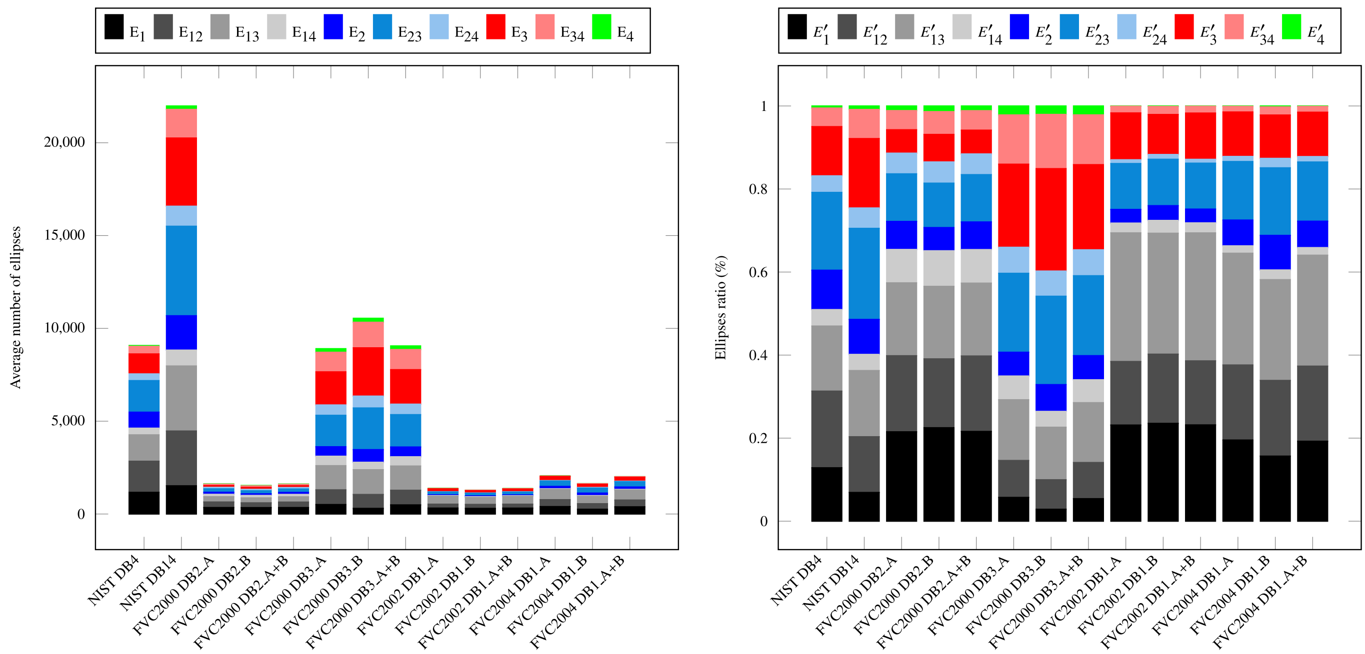
5. Experiment results and analysis

An exhaustive experimental evaluation has been carried out in this research to study the performance of the proposed algorithm. This evaluation has been performed on several data sets in order to evaluate the effectiveness and robustness of the proposed algorithm with respect to distortions and low-quality fingerprint images. Fig. 9 displays average number of minutiae obtained using different quality thresholds to filter the minutiae. In Fig. 9(a), M_1 is minutiae set belongs to high quality region of the fingerprint image, whereas M_4 is minutiae set belongs to lower quality region of the image. In Fig. 9(b), M'_1 is computed as the ratio between M_1 and M, i.e. M_1/M. M'_2, M'_3, and M'_4 are obtained as M'_1. Also, Fig. 10(a) displays average number of ellipses obtained using different quality thresholds. E_1 is ellipses set obtained using minutiae belong to M_1. E_2, E_3, and E_4 are obtained as E_1. E_{12} is ellipses set obtained using minutiae belong to M_1 and M_2. E_{13}, E_{14}, E_{23}, E_{24}, and E_{34} are obtained as E_{12}. In Fig. 10(b), E'_1 is computed as the ratio between E_1 and E, i.e. E_1/E. E'_2, E'_3, E'_4, E'_{12}, E'_{13}, E'_{14}, E'_{23}, E'_{24}, and E'_{34} are obtained as E'_1. Note that E is ellipses set obtained using all minutiae belong to M.

5.1. Databases

To compare the accuracy and performance of our algorithm with other state-of-the-art algorithms, different types of fingerprint databases applied in previous studies are used, including:

- *NIST DB4* [54]: The NIST DB4, or special database 4 (SD4), has been provided by the NIST (national institute of standards and technology) and contains 2000 fingers and 2 impressions per finger, *F* and *S*, yielding a total of 4000 fingerprints. Each fingerprint is rolled against a paper card and then scanned into digital form.
- *NIST DB14* [55]: The NIST DB14, or SD14, has been also provided by the NIST and contains 27,000 fingers and 2 impressions per finger, *F* and *S*, yielding a total of 54,000 fingerprints. Each fingerprint of this database such as NIST DB4 is captured. Also, the size of each of the fingerprint images in this database is greater than NIST DB4.
- *FVC2000 DB2* [18]: The second FVC2000 data set has two different sets *A* and *B*. Set *A* contains the fingerprint images from the first 100 fingers, while set *B* has the images from the other 10 fingers, 8 impressions per finger. Each data set is captured using **TouchChip** sensor, a low-cost capacitive sensor, by ST Microelectronics.
- *FVC2000 DB3* [18]: The third FVC2000 data set has the same type and number of fingerprint images as FVC2000 DB2 but at different size. Each data set is captured using **DF-90** sensor, an optical sensor sensor, by Identicator Technologyan.



(a) Average number of ellipses.

(b) Ellipses ratio (%).

Fig. 10. Comparison of the average number of ellipses obtained using different quality thresholds.

Table 2
Brief description of the used data sets in our experiment.

Data set	Number of fingerprints	Impressions per finger	Size of fingerprints (pixel)	Format of fingerprints	Resolution (dpi)
NIST DB4	2000	2	512 × 512	PNG	500.38
NIST DB14 (reduced)	2700	2	832 × 768	WSQ	500.38
FVC2000 DB2_A + B	110	8	256 × 364	TIF	500
FVC2000 DB3_A + B	110	8	448 × 478	TIF	500
FVC2002 DB1_A + B	110	8	388 × 374	TIF	500
FVC2004 DB1_A + B	110	8	640 × 480	TIF	500

- *FVC2002 DB1* [19]: The first FVC2002 data set has the same type and number of fingerprint images as FVC2000 DB2 but at different size. Each data set is captured using **TouchView II** sensor, an optical sensor sensor, by Identix.
- *FVC2004 DB1* [20]: The first FVC2004 data set has the same type and number of fingerprint images as FVC2000 DB2 but at different size. Each data set is captured using **V300** sensor, an optical sensor sensor, by CrossMatch.

Table 2 presents the traits of the data sets used, showing the number, size, format, and resolution of fingerprints. Note that fingerprint images of NIST DB14 is compressed with an implementation of the wavelet scalar quantization (WSQ) compression specification.

5.2. Hardware and software environment

The experiments were performed on a desktop PC with an Intel®Core^{TM}2 Quad Q9400 CPU at 2.66 GHz processor running Microsoft Windows 10 x64 having 6 GB of RAM. All versions of our algorithm were implemented with MATLAB and C++ programming languages.

5.3. Evaluation setup and parameters

Our indexing method has been evaluated on the six data sets described in the Section 5.1. On each data set, a fast and accurate segmentation algorithm [15] has been used to segment fingerprint images. Images of NIST SD4 and SD14 data sets often include lines, handwritten notes, stamps, etc. that can affect the performance of fingerprint recognition. In order to reduce the incidence of spurious minutiae, Cappelli et al. [9] removed a 50 pixel border of fingerprint images. This approach improves the segmentation performance on images of NIST SD14 data set (see Fig. 11(f–j)). Note that lines and handwritten notes in images of NIST SD4 data set are often located on images, thus Cappelli et al. [9] scheme is often ineffective in NIST SD4 data set. On the other hand, in NIST SD4 data set, the border removed by Cappelli et al. [9] scheme may contains some true minutiae and it can have a negative effect on the performance of fingerprint recognition. In our proposal, we only apply Cappelli et al. [9] scheme on NIST SD 14 data set.

In our experiments, we test three state-of-the-art segmentation algorithms. Yang et al. [73] algorithm can lead to satisfactory results on the latent fingerprint images and can be used on images of NIST SD14 data set. However, Yang et al. [73] algorithm causes the loss of some true minutiae. Thai and Gottschlich [65] algorithm obtains good results on FVC data sets but cannot lead to satisfactory results in NIST SD4 and SD14 data sets. Furthermore, this algorithm is very time-consuming and is not suitable to be used in real-time systems. The results show that the segmentation algorithm proposed in [15] outperforms the other algorithms. This algorithm can reach the desirable efficiency and accuracy on both FVC and NIST

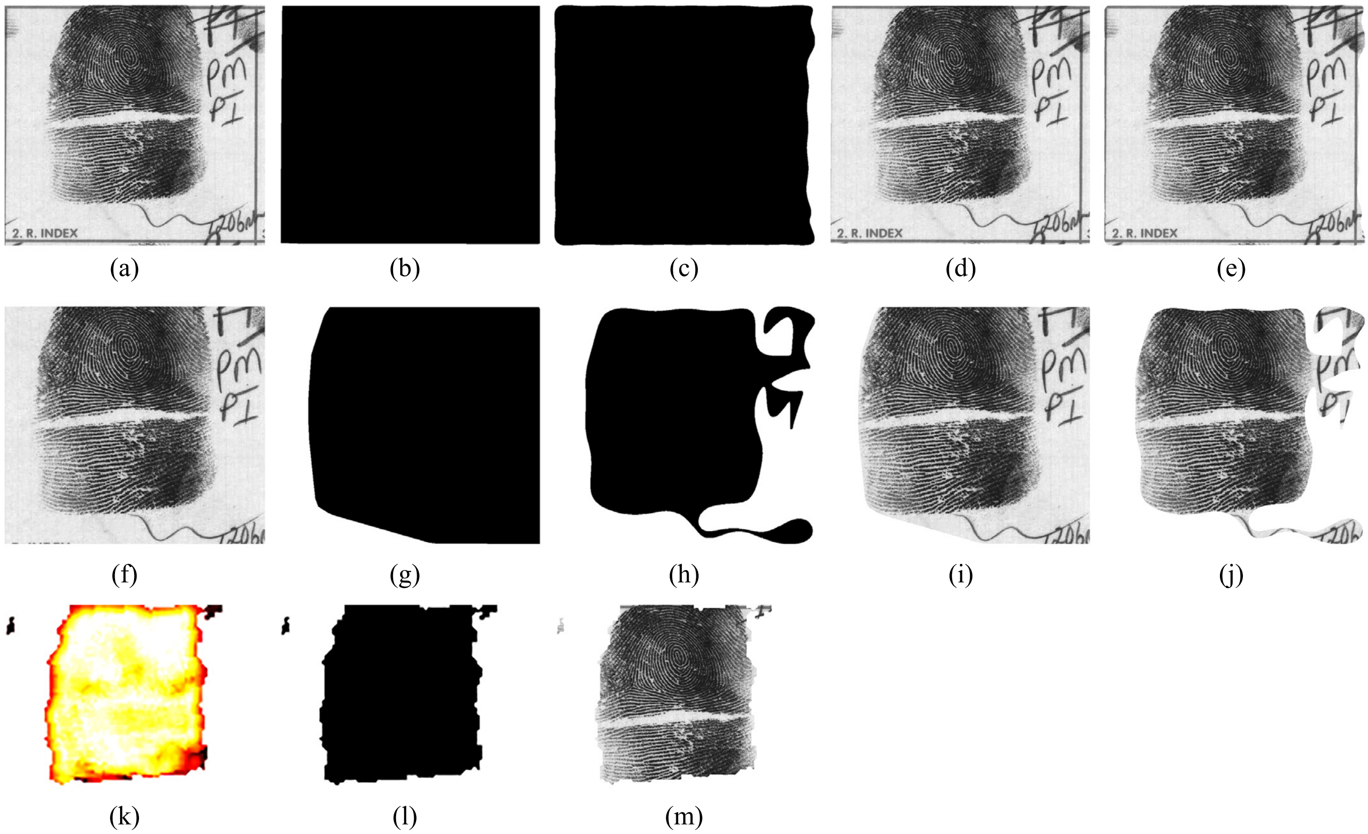


Fig. 11. Comparison of fingerprint segmentation algorithms: (a) original fingerprint image (f0026752 from NIST SD14) [55], (b) ROI of (a) by Thai and Gottschlich [65] algorithm, (c) ROI of (a) by Fahmy and Thabet [15] algorithm, (d) segmented image of (a) [65], (e) segmented image of (a) [15], (f) obtained result by removing a 50-pixel border of image (a), (g) ROI of (f) by Thai and Gottschlich [65] algorithm, (h) ROI of (f) by Fahmy and Thabet [15] algorithm, (i) segmented image of (f) [65], (j) segmented image of (f) [15], (k) confidence map of (a) by Yang et.al [73] algorithm, (l) ROI of (a) [73], and (m) segmented image of (a) [73].

databases. Fig. 11 demonstrates the superiority of Fahmy and Thabet [15] algorithm in comparison with other algorithms.

After carrying out the segmentation and image enhancement, minutiae extraction and convex core point detection algorithms have been employed to create feature vectors for all fingerprints. Although segmentation of fingerprint images can remove some spurious minutiae, very spurious minutiae may still be remained in fingerprint images, specially in the fingerprint images of NIST SD14 data set. In our proposal, Zhao and Tang [75] algorithm has been employed to remove spurious minutiae. Note that Zhao and Tang [75] algorithm is applied on valleys, but we apply this algorithm on ridges, because fingerprint image enhancement is employed before binarization process (see Fig. 7(b) and (c)). Our experimental results show, in some cases, Zhao and Tang [75] algorithm alone cannot remove all border minutiae in fingerprint boundary, specially in the fingerprint images of NIST SD14 data set. We also evaluate the effect of convex hull filter [57] on minutiae set obtained from minutiae extraction stage. Similar to Zhao and Tang [75] algorithm, convex hull filter alone cannot remove all border minutiae. In order to remove more of border minutiae, both filtering algorithms are employed. First, we apply convex hull filter on the minutiae set and then we employ Zhao and Tang [75] algorithm on the remaining minutiae. The quality of most of the border minutiae is lower than 10. Thus, we only consider convex hull filter CH-10. Furthermore, we have chosen the boundary as 10 pixels.

In order to detect of convex core point, Zhu et al. [76] and Belhadj et al. [3] algorithms are combined. Zhu et al. [76] algorithm is a fast and accurate algorithm. However, in some cases, this algorithm fails to detect singularities. In our experiments, we evaluate the effect of Zhu et al. [76] algorithm in combination with Qi and Liu [61] and Belhadj et al. [3] algorithms. The results show that Qi and Liu [61] algorithm and Belhadj et al. [3] algorithm have similar results, in accuracy, but Belhadj et al. [3] algorithm is more faster than Qi and Liu [61] algorithm. Furthermore, a problem arises with the arch-type fingerprints that do not have any SP. Among the three behind SPs detection algorithms, Belhadj et al. [3] algorithm is capable of detecting a reference point from arch-type fingerprints. In our experiment, we have chosen the convex core point tolerance as 25 pixels.

In order to estimate quality of fingerprints, we use NFIQ [53]. In our experiments, we have chosen quality thresholds t_1, t_2, and t_3 as 50, 25, 10, respectively. After carrying out the minutiae extraction and convex core point detection, the ellipse properties have been utilized to create feature vectors. The tolerance values δ_A, δ_B, δ_θ, δ_α, and δ_β are 9, 7, 5, 7, 7, respectively. Furthermore, the values of weights related to feature vectors quality w_1, w_2, w_3, w_4, w_{12}, w_{13}, w_{14}, w_{23}, w_{24} and w_{34} are 0.3, 0.2, 0.125, 0.11, 0.1, 0.075, 0.05, 0.025, and 0.015, respectively.

Before analysing the indexing, the robustness of the proposed feature vectors have been evaluated. The proposed feature vectors can be expressed as a directional graph whose nodes are minutiae and whose edges are feature vectors. Thus, the graph matching methods can be adopted to utilize the proposed feature vectors in fingerprint matching. In this experiment, we apply Chikkerur et al. [11] method, with K = 8. Table 3 compares the proposed technique, K-Plet coupled with our feature vectors, with some of the matching algorithms published in the literature. True positive rate (TPR) is percentage of test fingerprints that are correctly identified in the database, when only the best matching score is retrieved. The TPR is the error obtained when using a rank of 1. R100 is the

Table 3
Statistical tests for the identification measures. Best results in the row are highlighted with bold font.

Matcher	TPR	R100	ZeroR
Our feature vectors + Chikkerur [11]	3.73	3.65	4.71
Choi [12]	4.78	6.53	7.42
Chikkerur [11]	12.11	8.88	7.54
Bozorth3 [53]	**1.94**	8.96	7.26
Jiang [38]	3.86	3.52	**4.42**
MCC [8]	3.02	**2.11**	4.88
Friedman p-value	9.58×10^{-5}	9.57×10^{-5}	0.07

Table 4
The number of clusters and clustering quality.

Data set	Number of clusters (k)	Silhouette coefficient
NIST DB4	485	0.89
NIST DB14 (reduced)	623	0.78
FVC2000 DB2_A + B	118	0.93
FVC2000 DB3_A + B	196	0.86
FVC2002 DB1_A + B	102	0.90
FVC2004 DB1_A + B	167	0.91

lowest rank that allows an error lower than 1%. ZeroR is lowest rank that does not allow errors. The optimum value for R100 and ZeroR is 1 [58]. In order to establish a fair comparison between the techniques and to detect significant differences, statistical tests can be used [58]. Friedman and analysis of variance (ANOVA) tests [14] can be used to measure the differences between the techniques with a multiple comparison analysis. Unfortunately, ANOVA, a parametric test, has several drawbacks [14], thus we do not consider ANOVA.

In our experiments, we use the non-parametric and Friedman tests described in [14,17,24]. The Friedman test ranks the algorithms for each data set separately, the best performing algorithm getting the rank of 1 [14]. The Holm procedure, which was described in [14], is employed to find out which algorithms are distinctive. A p-value provides information about whether a statistical hypothesis test is significant or not [24]. Table 3 displays the power of our feature vectors in combination with matching algorithm proposed by Chikkerur et al. [11].

The indexing approach described in Section 4 has been applied to the resulting feature vectors. Table 4 reports the number of clusters, k, and average silhouette coefficient reached in the experiments for each data set. If average silhouette coefficient is more than 0.7, the quality of clustering considered to be excellent. Since the number of feature vectors of each data set differ from other data sets, value k is different for each data set.

Finally, candidate list reduction criteria proposed by Cappelli et al. [10] has been applied to reduce candidates. In most cases, variable threshold on score ratio criteria is more effective than the variable threshold on score difference criteria. Thus we apply variable threshold on score ratio criteria to reduce the candidate list. Note that the proposed algorithm has been designed and implemented using the parallelism in the calculation and search in order to speed up the computations and searches. Table 5 shows overall time of our indexing algorithm.

There are various methods to evaluate the performance of fingerprint indexing algorithms. For instance, trade-off between correct index power (CIP) and penetration rate (PR) [1,4,22,52], indexing accuracy (IA) and PR [46], error rate (ER) and PR [9,63], and hit rate (HR) and PR [32,33,36,70]. In our experiments, we have used the trade-off between ER and PR to illustrate the results. ER is the percentage of probes that are not found (measuring the accuracy), while PR is the proportion of database that the system has to search (measuring the efficiency). It is clear that for high performance gain, ER should be low.

Table 5
Average execution time for different stages of our system.

Stages in fingerprint indexing	Time (s)
Segmentation	0.391
Fingerprint enhancement and quality assessment	1.270
Binarization and thinning	0.331
Minutiae extraction and postprocessing	2.445
Convex core point detection	0.583
Extraction of feature vectors from an image	2.936 (offline)
Index space creation	172.104 (offline)
Gallery feature vectors' assignments to clusters	0.027 (offline)
Mapping of a probe's feature vectors to clusters	0.027 (online)
Candidate list retrieval for a probe	0.045×10^{-3} (online)
Candidate list reduction	0.032 (online)

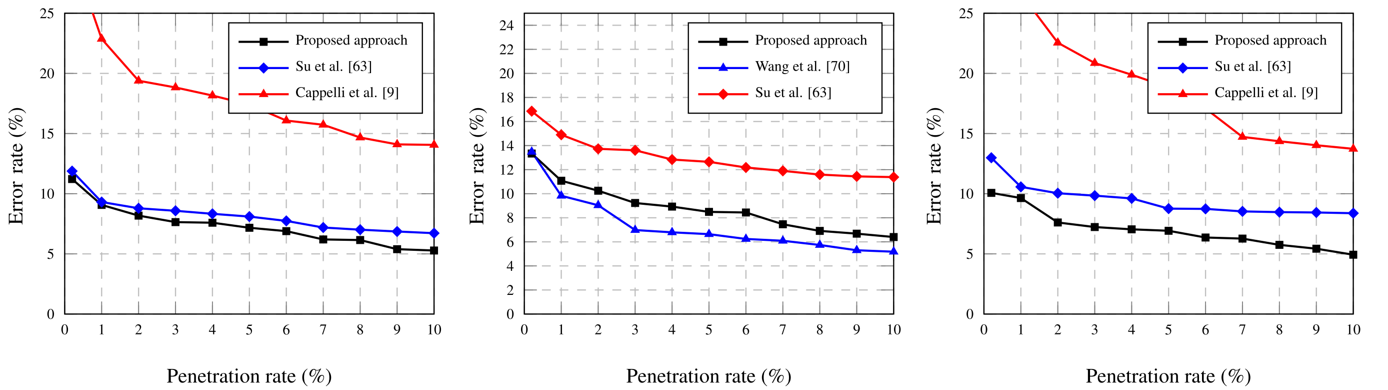
Table 6
The data sets and indexing methods for which published results are available.

Data set	Methods with published results
NIST DB4	Germain et al. [25] Bhanu and Tan [4] Cappelli et al. [9] Muñoz-Briseño et al. [52] Gago-Alonso et al. [22] Bai et al. [1] Su et al. [63]
NIST DB14	Cappelli et al. [9] Muñoz-Briseño et al. [52] Gago-Alonso et al. [22] Bai et al. [1] Wang et al. [70] Su et al. [63]
FVC2000 DB2	Iloanusi et al. [33] Cappelli et al. [9] Gago-Alonso et al. [22] Su et al. [63]
FVC2000 DB3	Iloanusi et al. [33] Cappelli et al. [9] Muñoz-Briseño et al. [52] Gago-Alonso et al. [22] Su et al. [63]
FVC2002 DB1	Germain et al. [25] Liang et al. [44] Mansukhani et al. [50] Iloanusi et al. [33] Wang and Hu [69] Cappelli et al. [9] Muñoz-Briseño et al. [52] Gago-Alonso et al. [22] Wang et al. [70] Su et al. [63]
FVC2004 DB1	Liang et al. [44] Mansukhani et al. [50] Iloanusi et al. [33] Gago-Alonso et al. [22] Tiwari and Gupta [66] Jayaraman et al. [36]

Table 7 displays the statistical tests results of indexing methods over all the data sets described in the Section 5.1. In this experiment, we only have focused on the TPR measures.

5.4. Results

Experiments on the six data sets described in the Section 5.1 demonstrate that our representation of the fingerprints provides more robust and significant information than other solutions and clearly outperforms most of other methods. According to Table 7, the proposed algorithm, Wang et al. [70] algorithm, and Su et al. [63] algorithm are the most accurate algorithms, with statistically significant differences with respect to the other algorithms. Also, Fig. 12 shows the superior accuracy of our pro-



(a) Indexing performance on NIST DB4. **(b)** Indexing performance on NIST DB14 (reduced). **(c)** Indexing performance on FVC2000 DB2.

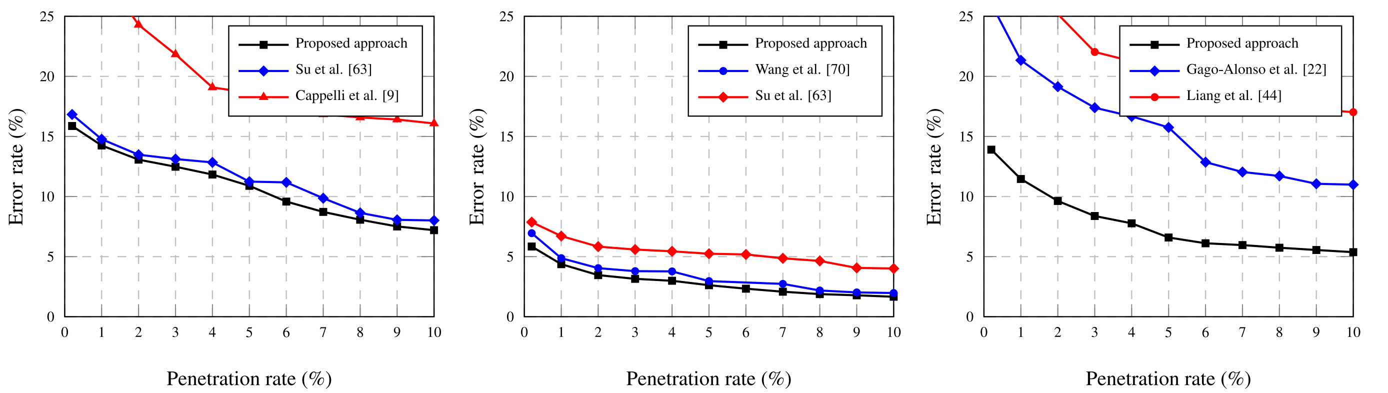


Fig. 12. Comparison with some of state-of-the-art algorithms in NIST and FVC databases.

Table 7
Statistical tests for the indexing methods. Best result is highlighted with bold font.

Indexing algorithm	TPR
Proposed	2.18
Jayaraman et al. [36]	4.21
Tiwari and Gupta [66]	4.18
Germain et al. [25]	7.84
Bhanu and Tan [4]	5.13
Bebis et al. [2]	5.99
Liang et al. [44]	3.96
Muñoz-Briseño et al. [52]	3.18
Gago-Alonso et al. [22]	3.23
Uz et al. [68]	4.92
Wang and Hu [69]	4.54
Iloanusi et al. [33]	5.01
Cappelli et al. [9]	3.28
Wang et al. [70]	**2.15**
Su et al. [63]	2.67
Mansukhani et al. [50]	4.27
Bai et al. [1]	4.11
Friedman p-value	9.12×10^{-5}

posal over some of the best state-of-the-art methods, except for Wang et al. [70] algorithm in NIST SD14 data set. However, Wang et al. [70] algorithm is more accurate only for very small values of penetration. Note that since the error rates of some indexing algorithms are larger than Liang et al. [44], we do not include them in the Fig. 12. We only consider three curves that correspond to the top three performers, according to Table 6.

All the implementations of the identification and indexing algorithms, excepting that proposed in [8,9,11,38,53], were developed by us and they are only based on the specifications and descriptions given by the respective authors. This step of our work was very tedious due to the fact that the papers do not provide all the details to achieve a perfect implementation of the idea presented. Thus, the results of the evaluated matching and indexing algorithms may differ from the results in the original papers. We have tried that the implementations of existing methods be exactly similar to their implementations but the results of the experiments display that there is a little difference between our results and the results of the original papers. The main reasons for this difference are the used platforms, i.e., using both data sets, type A and type B, standard database FVC instead of using only type A, the difference between the hardwares used, difference between softwares used, lack of information related to the values of the parameters employed in papers, etc.

Finally, the main reasons for higher performance are due to the robust features extracted from minutiae and convex core point, the robust k-means clustering, and the robust candidate list reduction criteria.

6. Conclusions

In this paper, we have introduced a novel fingerprint indexing algorithm based on minutiae details and convex core point. This algorithm uses a new representation of the fingerprints based on the triangles and their deep relations to ellipses, which is invariant to rotation and translation and is robust to distortions. Also, we have employed a k-means clustering algorithm and candidate list reduction criteria to increase performance of our indexing algorithm.

Experiments on some of the data sets of the FVC and NIST databases show that our approach outperforms some of the best state-of-the-art indexing algorithms reported in the literature. It has been observed that for different data sets, different indexing algorithms may be the most accurate. This paper can help researchers in the field to develop new indexing algorithms.

Our future work includes k-means clustering algorithms that are independent of determination of value k by user. We also plan to combine the proposed feature vectors and MCC descriptor for indexing of latent palmprints.

Acknowledgments

The authors would like to thank the anonymous reviewers for their valuable comments and suggestions, which have contributed a lot towards improving the content and presentation of this paper.

References

[1] C. Bai, T. Zhao, W. Wang, M. Wu, An efficient indexing scheme based on k-plet representation for fingerprint database, in: Proceedings International Conference on Intelligent Computing, 2015, pp. 247–257.
[2] G. Bebis, T. Deaconu, M. Georgiopoulos, Fingerprint identification using delaunay triangulation, in: Proceedings International Conference Information Intelligence and Systems, 1999, pp. 452–459.
[3] F. Belhadj, S. Akrouf, S. Harous, S.A. Aoudia, Efficient fingerprint singular points detection algorithm using orientation-deviation features, J. Electron Imag. 24 (3) (2015) 033016–1–033016–13.
[4] B. Bhanu, X. Tan, Fingerprint indexing based on novel features of minutiae triplets, IEEE Trans. Pattern Anal. Mach. Intell. 25 (5) (2003) 616–622.
[5] S. Biswas, N. Ratha, G. Aggarwal, J. Connell, Exploring ridge curvature for fingerprint indexing, in: Proceedings of the 2nd IEEE International Conference on Biometrics: Theory, Applications and Systems (BTAS), 2008, pp. 1–6.
[6] G.R. Bradski, A. Kaehler, Learning OpenCV: Computer Vision with the OpenCV Library, 1st, O'Reilly Media, 2009.
[7] R. Cappelli, Fast and accurate fingerprint indexing based on ridge orientation and frequency, IEEE Trans. Syst. Man Cybern. 41 (6) (2011) 1511–1521.
[8] R. Cappelli, M. Ferrara, D. Maltoni, Minutia cylinder-code: a new representation and matching technique for fingerprint recognition, IEEE Trans. Pattern Anal. Mach. Intell. 32 (12) (2010) 2128–2141.
[9] R. Cappelli, M. Ferrara, D. Maltoni, Fingerprint indexing based on minutia cylinder-code, IEEE Trans. Pattern Anal. Mach. Intell. 33 (5) (2011a) 1051–1057.
[10] R. Cappelli, M. Ferrara, D. Maltoni, Candidate list reduction based on the analysis of fingerprint indexing scores, IEEE Trans. Inf. Forensics Secur. 6 (3) (2011b) 1160–1164.
[11] S. Chikkerur, A.N. Cartwright, V. Govindaraju, K-plet and coupled BFS: a graph based fingerprint representation and matching algorithm, in: Proceedings International Conference on Biometrics (ICB 2006), 2005, pp. 309–315.
[12] H. Choi, K. Choi, J. Kim, Fingerprint matching incorporating ridge features with minutiae, IEEE Trans. Inf. Forensics Secur. 6 (2) (2011) 338–345.
[13] K. Choi, D. Lee, S. Lee, J. Kim, An improved fingerprint indexing algorithm based on the triplet approach, in: Proceedings of the 4th International Conference Audio and Video based Biometric Person Authentication (AVBPA03), 2003, pp. 584–591.
[14] J. Demšar, Statistical comparisons of classifiers over multiple data sets, J. Mach. Learn. Res. 7 (2006) 1–30.
[15] M.F. Fahmy, M.A. Thabet, A fingerprint segmentation technique based on morphological processing, in: Proceedings IEEE International Symposium on Signal Processing and Information Technology (ISSPIT), 2013, pp. 000215–000220.
[16] H. Fronthaler, K. Kollreider, J. Bigun, Local features for enhancement and minutiae extraction in fingerprints, IEEE Trans. Image Process. 17 (3) (2008) 354–363.
[17] M. Friedman, The use of ranks to avoid the assumption of normality implicit in the analysis of variance, J. Am. Stat. Assoc. 32 (1937) 675–701.
[18] FVC2000: The First Fingerprint Verification Competition. http://bias.csr.unibo.it/fvc2000/. Accessed 2 December 2016.
[19] FVC2002: The Second Fingerprint Verification Competition. http://bias.csr.unibo.it/fvc2002/. Accessed 2 December 2016.
[20] FVC2004: The Third International Fingerprint Verification Competition, http://bias.csr.unibo.it/fvc2004/. Accessed 2 December 2016.
[21] FVC2006: The Fourth International Fingerprint Verification Competition, http://bias.csr.unibo.it/fvc2006/. Accessed 2 December 2016.
[22] A. Gago-Alonso, J. Hernández-Palancar, E. Rodríguez-Reina, A. Muñoz-Briseño, Indexing and retrieving in fingerprint databases under structural distortions, Expert Syst. Appl. 40 (8) (2013) 2858–2871.
[23] M. Galar, J. Derrac, D. Peralta, I. Triguero, D. Paternain, C. Lopez-Molina, S. García, J.M. Benítez, M. Pagola, E. Barrenechea, H. Bustince, F. Herrera, A survey of fingerprint classification part I: taxonomies on feature extraction methods and learning models, Knowl. Based Syst. 81 (2015) 76–97.
[24] S. García, F. Herrera, An extension on statistical comparisons of classifiers over multiple data sets for all pairwise comparisons, J. Mach. Learn. Res. 9 (2008) 2677–2694.
[25] R.S. Germain, A. Califano, S. Colville, Fingerprint matching using transformation parameter clustering, IEEE Comput. Sci. Eng. 4 (4) (1997) 42–49.
[26] A. Gionis, P. Indyk, P. Motwani, Similarity search in high dimensions via hashing, in: Proceedings 25th International Conference on Very Large Data Bases (VLDB), volume 99, 1999, pp. 518–529.
[27] P. Gupta, P. Gupta, A robust singular point detection algorithm, Appl. Soft Comput. 29 (2015) 411–423.
[28] P.D. Gutiérrez, M. Lastra, F. Herrera, J.M. Benítez, A high performance fingerprint matching system for large databases based on GPU, IEEE Trans. Inf. Forensics Secur. 9 (1) (2014) 62–71.
[29] M.I. Ham, Y.B. Pereira, E.B.G. Reyes, A multiple substructure matching algorithm for fingerprint verification, in: Proceedings 12th Iberoamericann Congress on Pattern Recognition (CIARP 2007), 2007, pp. 172–181.
[30] J. Han, M. Kamber, J. Pei, Data Mining: Concepts and Techniques, third ed., Elsevier, 2011.
[31] L. Hong, Y. Wan, A. Jain, Fingerprint image enhancement: algorithm and performance evaluation, IEEE Trans. Pattern Anal. Mach. Intell. 20 (8) (1998) 777–789.
[32] O.N. Iloanusi, Fusion of finger types for fingerprint indexing using minutiae quadruplets, Pattern Recognit. Lett. 38 (2014) 8–14.
[33] O. Iloanusi, A. Gyaourova, A. Ross, Indexing fingerprints using minutiae quadruplets, in: IEEE Computer Society Conference/Computer Vision and Pattern Recognition Workshops (CVPRW), 2011, pp. 127–133.
[34] P. Indyk, R. Motwani, Approximate nearest neighbors: towards removing the curse of dimensionality, in: 30th Annual ACM Symposium on Theory of Computing, 1998, pp. 604–613.
[35] A.K. Jain, Data clustering: 50 years beyond k-means, Pattern Recognit. Lett. 31 (8) (2010) 651–666.
[36] U. Jayaraman, A.K. Gupta, P. Gupta, An efficient minutiae based geometric hashing for fingerprint database, Neurocomputing 137 (2014) 115–126.
[37] R.M. Jiang, D. Crookes, FPGA-based minutia matching for biometric fingerprint image database retrieval, J. Real-Time Image Process. 3 (3) (2008) 177–182.
[38] X. Jiang, W.Y. Yau, Fingerprint minutiae matching based on the local and global structures, in: Proceedings 15th International Conference on Pattern Recognition, 2000, pp. 1038–1041.
[39] H. Kasban, Fingerprints verification based on their spectrum, Neurocomputing 171 (2016) 910–920.
[40] L. Kaufman, P. Rousseeuw, Finding Groups in Data: An Introduction to Cluster Analysis, Wiley, 1990.
[41] M.A.U. Khan, T.M. Khan, D.G. Bailey, Y. Kong, A spatial domain scar removal strategy for fingerprint image enhancement, Pattern Recognit. 60 (2016) 258–274.
[42] S. Khanmohammadi, N. Adibeig, S. Shanehbandy, An improved overlapping k-means clustering method for medical applications, Expert Syst. Appl. 67 (2017) 12–18.
[43] T.H. Le, H.T. Van, Fingerprint reference point detection for image retrieval based on symmetry and variation, Pattern Recognit. 45 (9) (2012) 3360–3372.
[44] X. Liang, A. Bishnu, T. Asano, A robust fingerprint indexing scheme using minutia neighborhood structure and low-order delaunay triangles, IEEE Trans. Inf. Forensics Secur. 2 (4) (2007) 721–733.
[45] E. Liu, K. Cao, Minutiae extraction from level 1 features of fingerprint, IEEE Trans. Inf. Forensics Secur. 11 (9) (2016) 1893–1902.
[46] M. Liu, P.T. Yap, Invariant representation of orientation fields for fingerprint indexing, Pattern Recognit. 45 (7) (2012) 2532–2542.
[47] L. Lovász, K. Vesztergombi, U. Wagner, E. Welzl, Convex quadrilaterals and k-sets, in: J. Pach (Ed.), Towards a theory of geometric graphs. Volume 342 of Contemporary Mathematics Series, AMS, 2004, pp. 139–148.
[48] J. MacQueen, Some methods for classification and analysis of multivariate observations, in: Proceedings of the 5th Berkeley symposium on mathematics statistics and probability California, 1967, pp. 281–297.
[49] D. Maltoni, D. Maio, A.K. Jain, S. Prabhakar, Handbook of fingerprint recognition, second ed., Springer, 2009.
[50] P. Mansukhani, S. Tulyakov, V. Govindaraju, A framework for efficient fingerprint identification using a minutiae tree, IEEE Syst. J. 4 (2) (2010) 126–137.
[51] A. Mazer, The Ellipse: A Historical and Mathematical Journey, John Wiley & Sons, 2010.
[52] A. Muñoz-Briseño, A. Gago-Alonso, J. Hernández-Palancar, Fingerprint indexing with bad quality areas, Expert Syst. Appl. 40 (5) (2013) 1839–1846.
[53] NIST Biometric Image Software (NBIS): https://www.nist.gov/services-resources/software/nist-biometric-image-software-nbis. Accessed 2 December 2016.
[54] NIST Special Database 4: NIST 8-Bit Gray Scale Images of Fingerprint Image Groups (FIGS), https://www.nist.gov/srd/nist-special-database-4. Accessed 2 December 2016.
[55] NIST Special Database 14: NIST Mated Fingerprint Card Pairs 2 (MFCP2), https://www.nist.gov/srd/nist-special-database-14. Accessed 2 December 2016.
[56] D. Pelleg, A. Moore, X-means: extending k-means with efficient estimation of the number of clusters, in: Proceedings of the 17th international conference on machine learning (ICML'00), 2000, pp. 727–734.
[57] D. Peralta, M. Galar, I. Triguero, O. Miguel-Hurtado, J.M. Benitez, F. Herrera, Minutiae filtering to improve both efficacy and efficiency of fingerprint matching algorithms, Eng. Appl. Artif. Intell. 32 (2014) 37–53.
[58] D. Peralta, M. Galar, I. Triguero, D. Paternain, S. Garca, E. Barrenechea, J.M. Benitez, H. Bustince, F. Herrera, A survey on fingerprint minutiae-based local matching for verification and identification: taxonomy and experimental evaluation, Inf. Sci. 315 (2015) 67–87.
[59] D. Peralta, I. Triguero, R. Sanchez-Reillo, F. Herrera, J.M. Benitez, Fast fingerprint identification for large databases, Pattern Recognit. 47 (2) (2014) 588–602.

[60] K. Phromsuthirak, V. Areekul, Fingerprint quality assessment using frequency and orientation subbands of block-based fourier transform, in: Proceedings International Conference on Biometrics (ICB), 2013, pp. 1–7.
[61] J. Qi, S. Liu, A robust approach for singular point extraction based on complex polynomial model, in: Proceedings of the 2014 IEEE Conference on Computer Vision and Pattern Recognition Workshops (CVPRW), 2014, pp. 78–83.
[62] P.J. Rousseeuw, Silhouettes: a graphical aid to the interpretation and validation of cluster analysis, J. Comput. Appl. Math. 20 (1987) 53–65.
[63] Y. Su, J. Feng n, J. Zhou, Fingerprint indexing with pose constraint, Pattern Recognit. 54 (2016) 1–13.
[64] P. Sutthiwichaiporn, V. Areekul, Adaptive boosted spectral filtering for progressive fingerprint enhancement, Pattern Recognit. 46 (9) (2013) 2465–2486.
[65] D.H. Thai, C. Gottschlich, Global variational method for fingerprint segmentation by three-part decomposition, IET Biom. 5 (2) (2016) 120–130.
[66] K. Tiwari, P. Gupta, Indexing fingerprint database with minutiae based coaxial gaussian track code and quantized lookup table, in: Proceedings IEEE International Conference on Image Processing (ICIP), 2015, pp. 4773–4777.
[67] F. Turroni, D. Maltoni, R. Cappelli, D. Maio, Improving fingerprint orientation extraction, IEEE Trans. Inf. Forensics Secur. 6 (3) (2011) 1002–1013.
[68] T. Uz, G. Bebis, A. Erol, S. Prabhakar, Minutiae-based template synthesis and matching for fingerprint authentication, Comput. Vision Image Underst. 113 (9) (2009) 979–992.
[69] S. Wang, J. Hu, Alignment-free cancelable fingerprint template design: a densely infinite-to-one mapping (DITOM) approach, Pattern Recognit. 45 (12) (2012) 4129–4137.
[70] Y. Wang, L. Wang, Y.M. Cheung, P.C. Yuen, Learning compact binary codes for hash based fingerprint indexing, IEEE Trans. Inf. Forensics Secur. 10 (8) (2015) 1603–1616.
[71] C.I. Watson, M.D. Garris, E. Tabassi, C.L. Wilson, R.M. McCabe, S. Janet, K. Ko, User's Guide to NIST Biometric Image Software (NBIS), National Institute of Standards and Technology, 2007.
[72] C. Winter, M. Steinebach, Y. Yannikos, Fast indexing strategies for robust image hashes, Digital Invest. 11 (2014) S27–S35.
[73] X. Yang, J. Feng, J. Zhou, S. Xia, Detection and segmentation of latent fingerprints, in: Proceedings IEEE International Workshop on Information Forensics and Security (WIFS), 2015. http://dx.doi.org/10.1109/WIFS.2015.7368604.
[74] M. Zahedi, O.R. Ghadi, Combining gabor filter and FFT for fingerprint enhancement based on a regional adaption method and automatic segmentation, Signal Image Video Process. 9 (2) (2015) 267–275.
[75] F. Zhao, X. Tang, Preprocessing and postprocessing for skeleton-based fingerprint minutiae extraction, Pattern Recognit. 40 (4) (2007) 1270–1281.
[76] E. Zhu, X. Guo, J. Yin, Walking to singular points of fingerprints, Pattern Recognit. 56 (2016) 116–128.

Javad Khodadoust received his B.Sc. and M.Sc. degrees in computer engineering from the Payame Noor University (PNU), Iran, in 2011 and 2014, respectively. His research interests include pattern recognition, image processing, computer vision, biometrics, and machine learning.

Ali Mohammad Khodadoust received his B.Sc. degree in electrical engineering from Khavaran Institute of Higher Education (KHI), Iran, in 2010 and his M.Sc. degree in electrical engineering from Sadjad University of Technology, Iran, in 2013. His research interests include image processing, signal processing, and wireless systems.