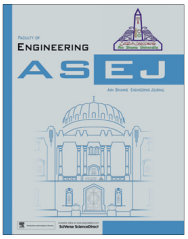
Ain Shams Engineering Journal (2013) 4, 393–408

Ain Shams University

Ain Shams Engineering Journal

www.elsevier.com/locate/asej
www.sciencedirect.com



ELECTRICAL ENGINEERING

A medium resolution fingerprint matching system

Ayman Mohammad Bahaa-Eldin *

Computer and Systems Engineering Department, Ain Shams University, Cairo, Egypt

Received 13 March 2012; revised 10 October 2012; accepted 10 October 2012
Available online 4 January 2013

KEYWORDS

Fingerprint matching;
Minutiae matching;
Feature vector distance

Abstract In this paper, a novel minutiae based fingerprint matching system is proposed. The system is suitable for medium resolution fingerprint images obtained by low cost commercial sensors. The paper presents a new thinning algorithm, a new features extraction and representation, and a novel feature distance matching algorithm. The proposed system is rotation and translation invariant and is suitable for complete or partial fingerprint matching. The proposed algorithms are optimized to be executed on low resource environments both in CPU power and memory space. The system was evaluated using a standard fingerprint dataset and good performance and accuracy were achieved under certain image quality requirements. In addition, the proposed system was compared favorably to that of the state of the art systems.

1. Introduction

Personal identification by means of biometric characteristics is now an integral part in many information systems. It has received more attention during the last decade due to the need for security in a wide range of applications. Among the biometric features, fingerprint is considered one of the most practical ones. Fingerprint recognition requires a minimal effort from the user, does not capture other information than strictly necessary for the recognition process, and provides relatively good performance. Another reason for the popularity of fingerprints is the relatively low price of fingerprint sensors, which enables easy integration into PC keyboards, smart cards and wireless hardware [1].

The general framework of a fingerprint identification system (FIS) is depicted in Fig. 1 [2]. As shown, fingerprint matching is the last step in Automatic Fingerprint Identification System (AFIS). Fingerprint matching techniques can be classified into three types:

- Correlation-based matching,
- Minutiae-based matching,
- Feature-based matching like sweat pores and 3D matching [3].

The minutiae-based matching is the most popular and widely used technique, being the basis of the human based fingerprint comparison.

Many fingerprint identification systems had been proposed, but in most of them, pre-processing, alignment and orientation are required. Also, many of these systems require high resolution fingerprint images, large memory to store templates, large memory to match, and complex processing.

* Tel.: +20 (11) 11 555750.
E-mail address: ayman.bahaa@eng.asu.edu.eg.
Peer review under responsibility of Ain Shams University.

Production and hosting by Elsevier

2090-4479 © 2012 Ain Shams University. Production and hosting by Elsevier B.V. All rights reserved.
http://dx.doi.org/10.1016/j.asej.2012.10.001

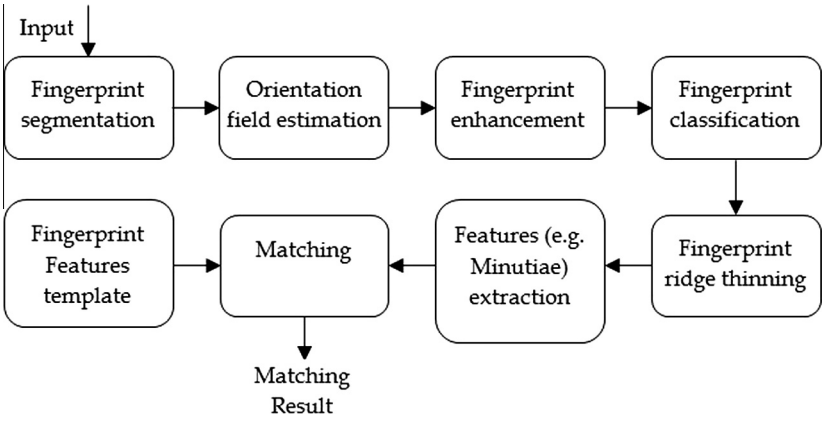


Figure 1 General block diagram for a fingerprint identification system.

This situation is not suitable for small devices with cheap fingerprint scanners, especially when massive identification processes are expected.

This work focuses on a novel feature representation scheme of fingerprints, which can be easily obtained from medium resolution (500 DPI) fingerprint scanners and that does not require any pre-orientation and alignment and in the same time requires very small memory size and simple matching processing.

This paper is organized as follows, the next section depicts related work, then the proposed fingerprint matching system is presented in Section 3. Section 4 presents the experimental results of evaluations followed by a conclusion and future work section.

2. Related work

This section presents some basic and the state of the art work in the field of fingerprint matching during the last 20 years. The focus was on the feature based matching. There is another approach by using filter banks, but it is an old one and does not give accurate results.

In [3], a hierarchical fingerprints matching method, namely Tangent Distance Sparce Weighted Random sample (TDSWR) method, using sweat pores as fingerprint features, by introducing the TD-Sparse-based method for coarse pore correspondence establishment and weighted RANdom SAmple Consensus (WRANSAC) for refinement. The proposed method measures the differences between pores based on the residuals obtained by the tangent distance and sparse representation technique, which makes the method more robust to noise and local distortions in fingerprints when compared with the existing Direct Pore matching (DP) [33] and Sparse Representation Direct Pore matching (SRDP) [34] methods. It then establishes one-to-many coarse pore correspondences, and assigns to each correspondence a weight-based on the difference between the pores in the correspondence. The final pore correspondences are obtained by adopting WRANSAC to refine the coarse pore correspondences. The experimental results demonstrated that the proposed method can more effectively establish pore correspondences and finally reduce the equal error rate (EER) by one order of magnitude in both of the two fingerprint databases used in the experiments (the best improvement on the recognition accuracy is up to 92%). However, the high computational complexity is one of the limitations of this method.

In [4], a minutia matching method was presented, describing elastic distortions in fingerprints by means of a thin-plate spline model, which is estimated using a local and a global matching stages. After registration of the fingerprints, according to the estimated model, the number of matching minutiae can be counted using very tight matching thresholds. For deformed fingerprints, the algorithm gives considerably higher matching scores compared to the rigid matching algorithms, while only taking 100 ms on a 1 GHz P-III machine. Furthermore, it was shown that the observed deformations are different from those described by theoretical models proposed in the literature.

The filter-based algorithm, in [5], uses a bank of Gabor filters [35], to capture both local and global details in a fingerprint as a compact fixed length "FingerCode". The fingerprint matching is based on the Euclidean distance between the two corresponding FingerCodes and hence it is extremely fast. However, the accuracy of the matching results is not high enough for accurate identification.

In [6], a fingerprint minutia matching technique was proposed, matching the fingerprint minutiae by using both the local and global structures of minutiae. The local structure of a minutia describes a rotation and translation invariant feature of the minutia in its neighborhood. It is used to find the correspondence of two minutiae sets and increase the reliability of the global matching. The global structure of minutiae reliably determines the uniqueness of fingerprint. Therefore, the local and global structures of minutiae together provide a solid basis for reliable and robust minutiae matching. This matching scheme is suitable for online processing for one to one matching but not on embedded devices and yet requires high resolution images.

In [7], three ideas are introduced along the following three aspects:

- Introduction of ridge information into the minutiae matching process in a simple and effective way, which solves the problem of reference point pair selection with low computational cost.
- Use of a variable sized bounding box to make their algorithm more robust to nonlinear deformation between fingerprint images.

- Use of a simpler alignment method in their algorithm.

Experiments using the Fingerprint Verification Competition 2000 (FVC2000) show that these ideas are effective and can produce around 90% accuracy.

In [8], minutia polygons are used to match distorted fingerprints. A minutia polygon describes not only the minutia type and orientation, but also the minutia shape. This allows the minutia polygon to be bigger than the conventional tolerance box, without losing matching accuracy. Furthermore, the proposed matching method employs an improved distortion model, using a Multi-quadric basis function with parameters. Adjustable parameters make this model more suitable for fingerprint distortion. Experimental results show that the proposed method is two times faster and more accurate (especially, on fingerprints with heavy distortion) than the method in [4].

In [9], a hybrid matching algorithm that uses both minutiae (point) information and texture (region) information is presented for matching the fingerprints. Results obtained shows that a combination of the texture-based and minutiae-based matching scores leads to a substantial improvement in the overall matching performance. This work was motivated by the small contact area sensors provided for the fingertip and, therefore, sense only a limited portion of the fingerprint. Thus, multiple impressions of the same fingerprint may have only a small region of overlap. Minutiae-based matching algorithms, which consider ridge activity only in the vicinity of minutiae points, are not likely to perform well on these images due to the insufficient number of corresponding points in the input and template images.

In [10], an efficient method for minutiae-based fingerprint matching is proposed, which is invariant to translation, rotation and distortion effects of fingerprint patterns. The algorithm is separated from a prior feature extraction and uses a compact description of minutiae features in fingerprints. The matching process consists of three major steps: Finding pairs of possibly corresponding minutiae in both fingerprint patterns, combining these pairs to valid tuples of four minutiae each, containing two minutiae from each pattern, and the third step is the matching itself. It is realized by a monotonous tree search that finds consistent combinations of tuples with a maximum number of different minutiae pairs. This approach has reasonable and scalable memory requirements and it is computationally inexpensive.

In [11], a minutiae indexing method is proposed to speed up the fingerprint matching, which narrows down the search space of minutiae to reduce the computational effort. An orderly sequence of features is extracted to describe each minutia, and the indexing score is defined to select minutiae candidates from the query fingerprint for each minutia in the input fingerprint. This method can be applied in both minutiae structure-based verification and fingerprint identification. Experiments are performed on a large-distorted fingerprint database (FVC2004 DB1) to guarantee the validity of this method.

In [12], the design and implementation of an online fingerprint verification system was described. This system operates in two stages: minutia extraction and minutia matching. In this method, the authors improved the minutia extraction algorithm proposed by Ratha et al. [13], to be faster and more reliable, and it was implemented for extracting features from an input fingerprint image, captured with an online inkless scanner. For minutia matching, an alignment-based elastic matching algorithm has been developed. This algorithm is capable of finding the correspondences between minutiae in the input image and the stored template without resorting to exhaustive search. It has the ability of adaptively compensating for the nonlinear deformations and inexact pose transformations between fingerprints.

In [14], a minutia matching algorithm, which is a modified version of the Jain et al.'s [12] algorithm is proposed. The algorithm can better distinguish two images from different fingers and is more robust to nonlinear deformation. Experiments performed using a set of fingerprint images captured with an inkless scanner shows that the algorithm is fast and has high accuracy.

In [15], a fingerprint minutiae matching algorithm was proposed, which is fast, accurate and suitable for the real time fingerprint identification system. In this algorithm, the core point is used to determine the reference point and a round bounding box is used for matching. Experiments performed using a set of fingerprint images captured with a scanner showed that the algorithm is faster and more accurate than Xiping Luo's [14] algorithm.

In [16], a minutia matching method based on a global alignment of multiple pairs of reference minutiae was proposed. These reference minutiae are commonly distributed in various fingerprint regions. When matching is performed, these pairs of reference minutiae are to be globally aligned, and those region pairs far away from the original reference minutiae will be aligned more satisfactorily. Experiment shows that this method leads to improvement in system identification performance.

In the previous paragraphs, several algorithms and matching systems were presented. The performance achieved by those algorithms regarding accuracy, speed, and storage requirements are neither suitable for trusted personal identification nor suitable for restricted computing environments such as embedded systems and smart cards. Hence the next section presents several contributions to enhance the matching accuracy and to eliminate as much as possible of computing and processing, while keeping the fingerprint image feature vector as small as possible to save storage and memory space.

3. Proposed fingerprint matching system

As depicted in Fig. 1, the first 4 steps are pre-processing steps where segmentation, orientation, enhancements and classifications are performed prior to matching. In our proposal these steps are not needed, so the first step is thinning. Feature extraction step is done on 3 steps mentioned after as steps 2, 3 and 4. Finally the matching is identified as step 5. So he proposed fingerprint matching system consists of 5 steps,

1. Thinning of the fingerprint image.
2. Core point detection.
3. Minutiae extraction.
4. Feature vector construction.
5. Distance based matching.

Each step will be described in details in the following subsections.

3.1. Thinning algorithm

Most of the current thinning algorithms used in fingerprint pre-processing operations need the step of binarization before thinning as in [2,4,7,17–21].

Binarization causes the following problems:

1. A significant amount of information may be lost during the binarization process.
2. Binarization is time consuming and may introduce a large number of spurious minutiae.
3. In the absence of an a priori enhancement step, most of the binarization techniques do not provide satisfactory results when applied to low-quality images.

The effect of these spurious minutiae is to decrease the recognition accuracy and, hence, increase the false match and the false non-match rates. Motivated by this analysis, a new thinning algorithm is proposed in this section. This new thinning algorithm works directly on the gray-scale images without the binarization step. The basic idea of this algorithm was given in a previous work of the author [36].

Generally, the gray values of pixels of ridges in the fingerprint image gradually decrease going from an edge towards the center of the ridge line, then, increase again going towards the other edge. This represents the definition of a local minimum. The idea is to capture this local minimum line to convert the ridge of (e.g. 5) pixels wide into one pixel wide.

According to the resolution of the used fingerprints (500 DPI), making a window size of 8×8 pixels for scanning to be very suitable as it contains at least one complete ridge if it exists. There is a difference in the procedure of searching for a local minimum between a horizontal (or near horizontal) ridge (referred to as type1) and a vertical (or near vertical) ridge (referred to as type2), see Fig. 2, for the fingerprint image (Fig. 2a) taken from the standard fingerprint image database FVC2000 [22]. Fig. 3 presents the steps of identifying block types where Fig. 4 shows the steps of the thinning algorithm. For type1 ridges, the block is scanned column by column, during this scan, the local minimum is searched for in each column, and in a new block initialized by zeros, all local minima in those columns will be replaced with ones. For type2 ridges, the block is scanned row by row, during this scan, the local minimum is searched for in each row, and in a new block initialized by zeros, and all local minima in those rows will be replaced with ones. The fingerprint image is divided into blocks of 8×8 pixels and each block is processed one at a time. Fig. 5 shows the gray levels of 2 blocks to illustrate how each type can be determined and thinned. Fig. 6a shows the result of thinning the fingerprint in Fig. 2.

It is interesting to point out that after implementing the thinning algorithm; most bifurcations are separated with one or two pixels, i.e. not connected. Therefore, a complete scan of the resulted thinned image must be performed. A window of 3×3 pixels is then used and separations of less than 2 pixels are connected. The results of the enhancements are shown in Fig. 6b.

Comparing the results of this thinning algorithm to those of [23], a much accurate result is obtained along with much less H breaks and spikes. The overall computational performance of the proposed thinning algorithm is double of the computational performance of [23], since no pre-processing steps (orientation field, segmentation, enhancement and binarization) are needed.

The effect of using the proposed thinning algorithm over those requiring a binarization step was found to increase the overall recognition rate by a ratio of 7% to 9%, when tested on the FVC2000 dataset. This enhancement originated from

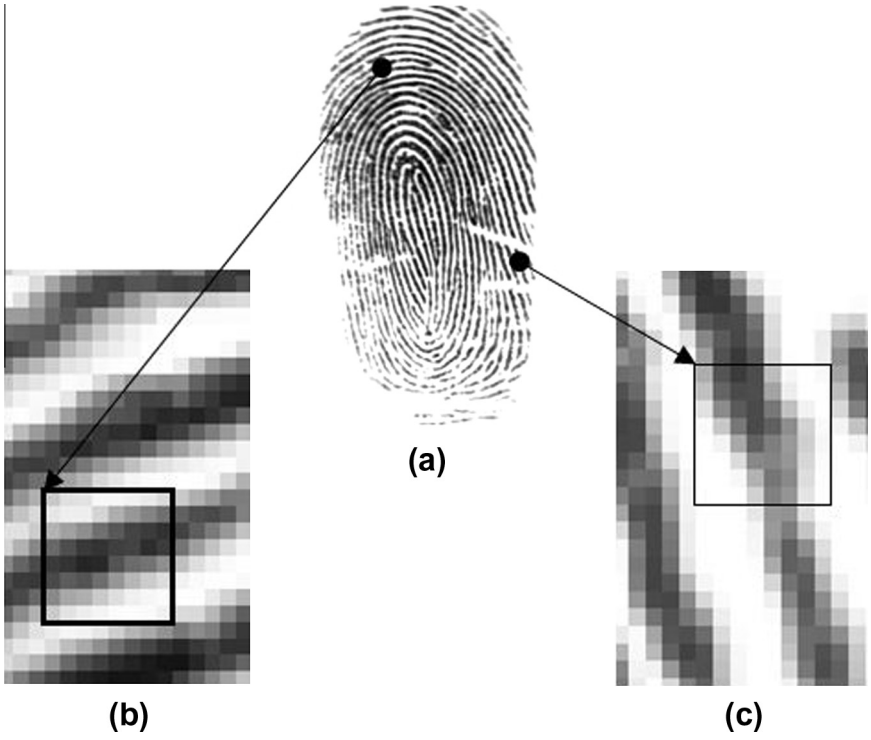


Figure 2 (a) The fingerprint image, (b) a type1 (horizontal) zoomed block of size 8×8, (c) a type2 (vertical) zoomed block of size 8×8.

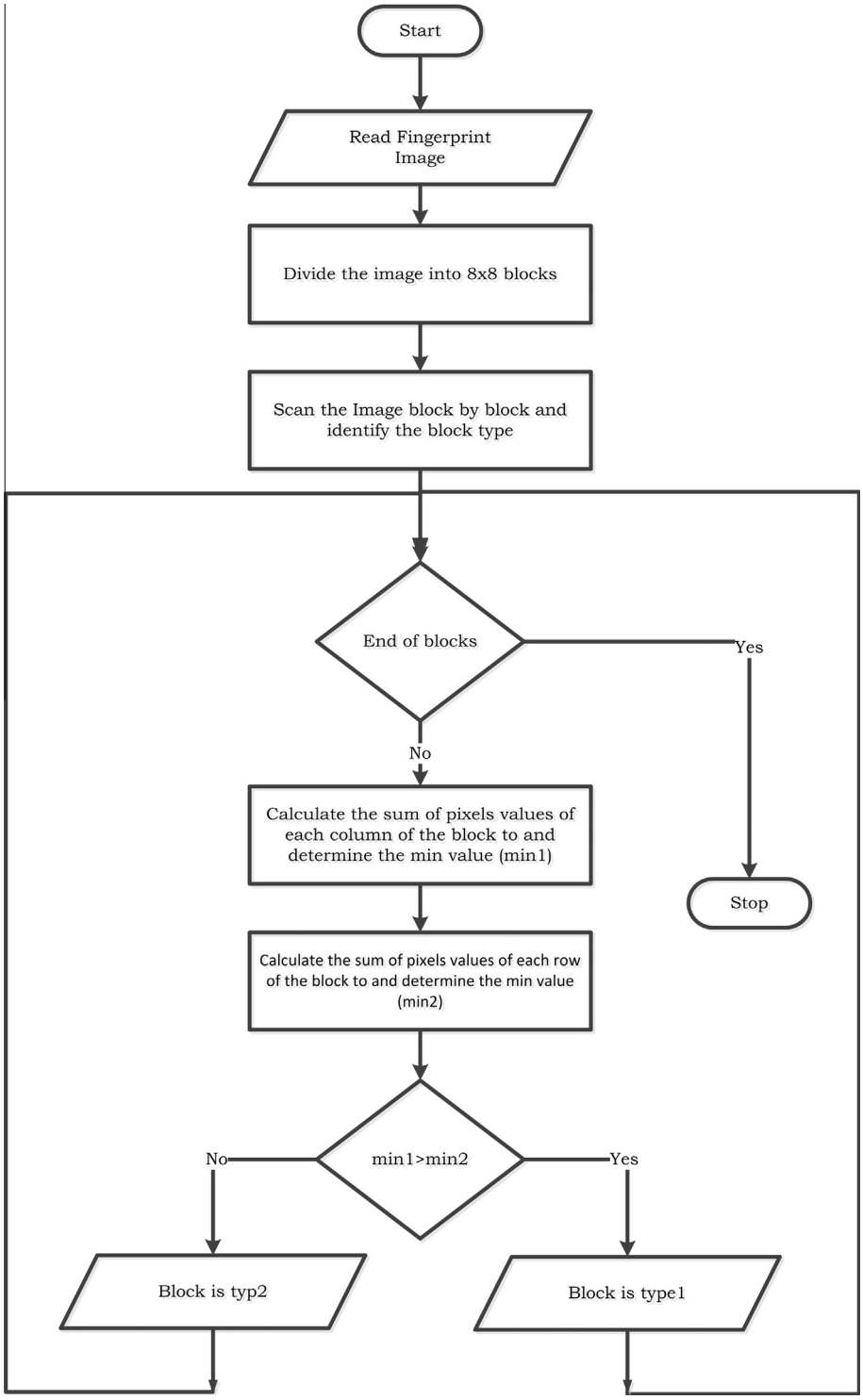


Figure 3 The steps of discrimination between type1 and type2 blocks.

the elimination of the noisy pixels and the spurious minutiae presented by the binarization step.

3.2. Fingerprint minutiae extraction and core point detection

Now, the fingerprint image is thinned into one pixel wide ridges, this image is processed to extract the feature vector for later matching. Consider the following definitions,

Inputs: X = fingerprint image, T = fingerprint template PID = person identity
Output: R = result of matching, then the 4 cases of R can be defined A

$$\forall X \in image(PID) \wedge X \equiv T \rightarrow R = truematch$$
$$\exists X \notin image(PID) \wedge X \equiv T \rightarrow R = falsematch$$
$$\exists X \notin image(PID) \wedge X \neq T \rightarrow R = truenonmatch$$
$$\forall X \in image(PID) \wedge X \neq T \rightarrow R = falsenonmatch$$

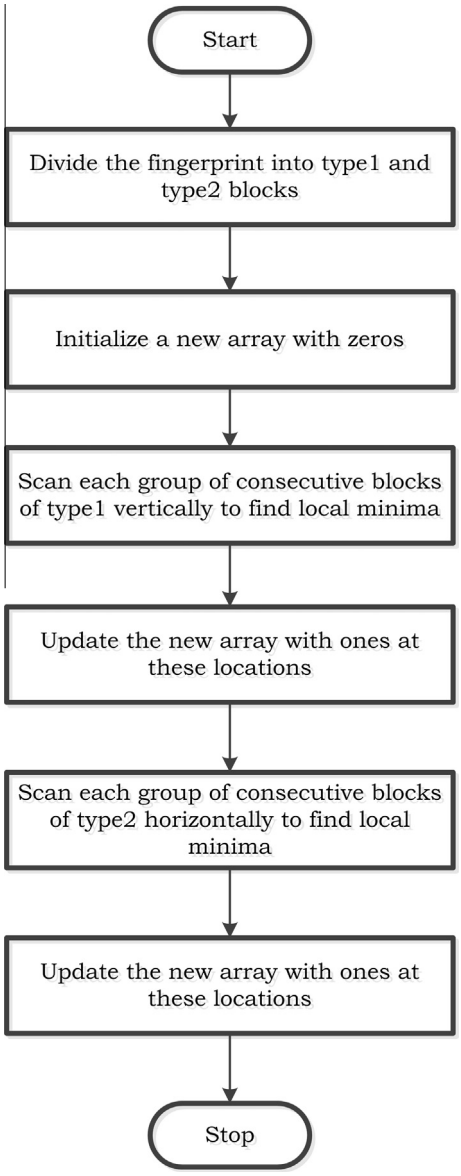


Figure 4 A flowchart of the proposed algorithm.

The goal of matching is to increase both the true match and the true non-match and to decrease the false ones. To achieve a rotation and translation independent matching algorithm, all minutiae positions will be relative to the fingerprint core point.

3.2.1. Core (singular) point detection

The core (or singular) point of a fingerprint is defined as "the point of the maximum curvature on the convex ridge" [24], which is usually located in the central area of fingerprint. The reliable detection of the position of a reference point can be accomplished by detecting the maximum curvature. While the method in [26] gives accurate results, it has a very high computational cost. The method of [25] is used to determine the core point as follows:

For each overlapping block in an image.

- For each overlapping block in an image, generate and reconstruct a ridge orientation image by computing gradients of the pixels in the block, a ridge frequency image through obtaining the FFT value of the block, and an energy image by summing the power of FFT value.
- Apply the corresponding complex filter $h = (x + iy)^m g(x, y)$ centered at the pixel orientation in the orientation image, where $g(x,y) = e^{-\frac{x^2+y^2}{2\sigma^2}}$ indicate the order of the complex filter and a Gaussian window, respectively.
- Reconstruct the filtered image by composing filtered blocks.
- The maximum response of the complex filter in the filtered image can be considered as the reference point. Since there is only one output, the unique output point is taken as the reference point.

Note that in some cases the output of the previous method ius the top left corner of the image, and this means the image does not contain a core point. Image acquisition and segmentation limits can generate fingerprint images with no core point. In addition, Arch class contains fingerprint images with no core point.

A singular point, as defined in [30–32] can be used as the reference point. In the fingerprint images with available core point, the singular point will be eventually the core point while in other images and types of fingerprints, the singular point will serve as the reference point of the image.

The singular point detection algorithm in [30] is described as follows:

- Determine the core-delta path by determining the high angular variation points in the vertical direction. This mapping makes it possible to highlight the points with an angular variation closed to $\pi/2$ in the directional map.
- For each kernel (i,j), the differences computed among each directional map element (i,j), and its 8_neighbors are used to detect the zones with the highest vertical differences.
- Finally, the point having the maximum angular difference is selected.

3.2.2. Minutiae extraction

As described in [1], minutiae are extracted from the thinned fingerprint image.

A simple image scan generates a crossing number for each pixel. 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 8-neighborhood of p:

$$cn(p) = \sum_{i=1}^{8} P_{i \bmod 8} - P_{i-1} \quad (1)$$

where P_i is a the binary value of the pixel (i) (0 for background and 1 for a ridge pixel) and P_0, P_1, …, P_7 are the pixels belonging to an ordered sequence of pixels defining the eight neighborhood of p. For the values of $cn(p)$, the pixel p would be an intermediate ridge point if $cn(p) = 2$, a ridge ending minutia if $cn(p) = 1$, a bifurcation minutia if $cn(p) = 3$, or a more complex minutia (e.g., crossover) if $cn(p) > 3$.

201	224	227	248	247	241	233	218	1839
231	239	238	213	175	152	132	112	1492
179	161	143	112	89	69	60	50	863
105	79	58	52	64	67	46	60	**531**
59	52	38	41	84	108	116	143	641
74	93	85	111	151	183	204	221	1122
138	168	175	202	223	240	247	251	1644
203	227	238	245	252	253	251	241	1910
1190	1243	1202	1224	1285	1313	1289	1296	

(a) A type 1 (horizontal block) where the minimum sum value for rows is found at row 4 (531) and the minimum sum value for columns is found at column 1 (1190)

174	84	60	77	189	247	255	249	**1335**
192	101	75	78	164	238	255	254	1357
226	139	91	75	120	193	243	255	1342
251	200	120	75	97	142	213	250	1348
255	244	169	98	127	154	208	247	1502
255	255	220	137	133	150	215	251	1616
255	255	248	179	119	138	221	254	1669
255	255	255	209	129	149	228	255	1735
1863	1533	1238	**928**	1078	1411	1838	2015	

(b) A type 2 (vertical block) where the minimum sum value for rows is found at row 1 (1335) and the minimum sum value for columns is found at column 4 (928)

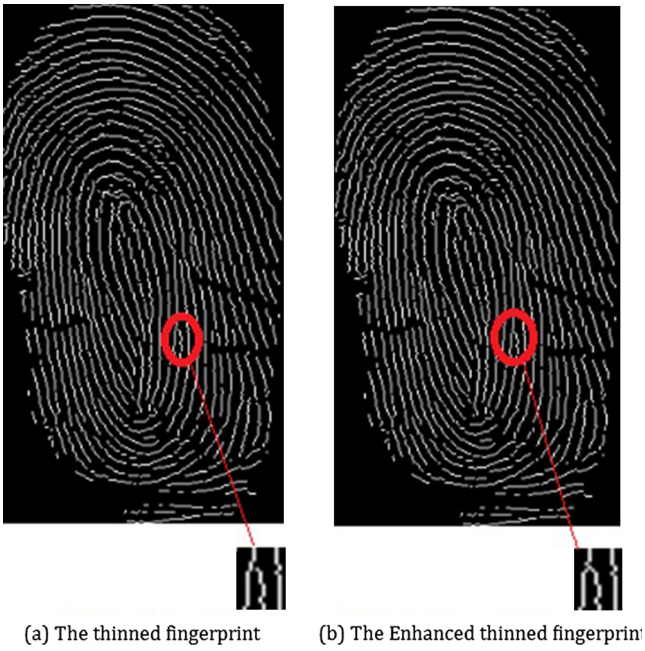
0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	1
0	0	0	0	1	1	1	0
1	1	1	1	0	0	0	0
0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0

(c) The corresponding block of figure (a) after horizontal ridge identified by ones

0	0	1	0	0	0	0	0
0	0	1	0	0	0	0	0
0	0	0	1	0	0	0	0
0	0	0	1	0	0	0	0
0	0	0	1	0	0	0	0
0	0	0	0	1	0	0	0
0	0	0	0	1	0	0	0
0	0	0	0	1	0	0	0

(d) The corresponding block of figure (b) after vertical ridge identified by ones

Figure 5 Illustration of the thinning algorithm.



(a) The thinned fingerprint (b) The Enhanced thinned fingerprint

Figure 6 Enhancement of thinned fingerprint.

Each minutia position is converted from (x, y) format to a distance (r) from the core point position (x_0, y_0). Hence, we have for each minutia: its distance from the core point (r) and the type of the minutiae (type1 if it is a termination and type2 if it is a bifurcation). Fig. 7 shows the result of this minutiae extraction stage, where the termination minutiae are represented by circles and the bifurcation minutiae by diamonds together with the core point of the fingerprint by an asterisk.

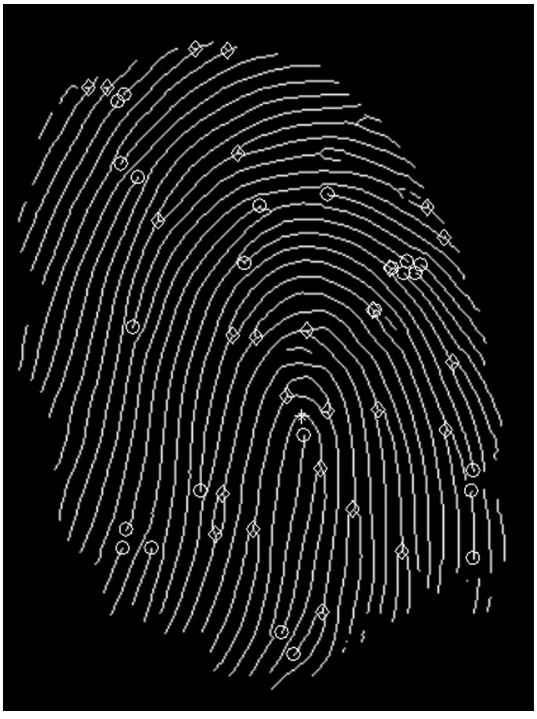


Figure 7 Core point (asterisk), terminations (circles) and bifurcations (diamonds).

3.3. Fingerprint features representation

In this proposed work, the features of the fingerprint will be represented by the number of minutiae of each type within a

specific distance from the core point. This is achieved by dividing the fingerprint image into concentric tracks around the core point. The track width is chosen to be 10 pixels as in a 500 DPI image resolution, a ridge is around 5 pixels wide and the inter-ridge distance is also around 5 pixels. A vector of $N \times 2$, where N is the maximum number of tracks, can be constructed around the core point and can be obtained as follows:

$$N = \left\lfloor \frac{\min(xc, y_c, W - x_c, H - y_c)}{10} \right\rfloor \quad (2)$$

where x_c and y_c are the coordinates of the core point from the top left corner of the image, W is the width of the fingerprint image and H is its height.

This type of features' selection has many advantages, namely:

1. They distinct fingerprints with high accuracy, as the number and position of minutiae are the main features used to identify fingerprints manually.
2. These features can be applied for full or partial fingerprints as long as the core point can be detected, in an accurate way, and enough number of tracks can be constructed around the core point.
3. The features are translation and rotation invariant and no pre-alignment or processing is needed to extract the features.
4. The core point detection and the possible number of tracks are used as judgment parameters to accept the fingerprint impression as a template or as a challenge for identification.

Fig. 8 shows the feature vector of the fingerprint extracted minutiae.

3.4. Vector distance matching algorithm

The proposed matching algorithm is based on calculating the distance between the feature vector of the fingerprint (called the challenge fingerprint) and a set of pre-stored templates in the FIS database. This can be done in 2 scenarios,

1. In a one to one matching, a fingerprint feature vector is compared to the set of pre-stored template(s) vector(s) of a specific person where the person ID is given as an input to the matching system. If the (average) distance falls below a certain threshold, the finger print is matched
2. In a one to many matching, a fingerprint feature vector is compared to the set of the pre-stored templates vectors. The minimum (average) distance between the fingerprint image and the set of templates for a specific user is identified. If this (average) distance falls below a certain threshold, the finger print is matched to the owner of the template set, otherwise no match is achieved.

For this system to work properly, an enrollment phase is required where each person is requested to scan his finger more than one time (typically 3–8 times). For each finger impression, the core point is detected, and the feature vector is constructed. As mentioned before, the successful detection of the core point and the construction of a number of tracks around it, that exceeds a certain value, are used as quality measures to accept the finger impression or reject it. The person ID is also requested (This ID can be the name, a unique number, or anything else unique) and this ID is stored in the database. Finally, each impression feature vector is stored under this ID and is called a matching template.

3.4.1. Feature vector difference calculation

To calculate the distance between 2 vectors, the absolute difference between each corresponding cells is calculated as $D = |V1 - V2|$. Remember that the value of each cell presents the number of features of either type1 or type2, in a 10 pixel wide track around the core point. As the challenge fingerprint feature vector and the pre-stored templates feature vectors dimension may vary in the number of tracks, the least number of columns of the 2 vectors will be the number of columns, in the resulting absolute distance vector. As shown later, in the experimental results, 14 tracks is the minimum number of tracks to be used. Fig. 9 shows eight different images of the same fingerprint presented in Fig. 2 (shown as (9-a)) from the FVC2000 dataset. Fig. 10 shows an example of calculating of absolute distance between one impression of the fingerprint and a pre-stored template.

3.4.2. Feature vectors mean distance calculation

The sum of the absolute differences vector (D) is calculated, resulting in a pair of numbers that represent the distance between the two feature vectors (the challenge vector and the template vector). The same process is applied for all the templates stored in the database related to the same user. Since in the used dataset, eight images are stored for each fingerprint, one of them will be used as a challenge and the other seven will be used as templates. Table 1 shows the difference pair for each template.

The geometric mean is used to calculate the average distance between the challenge fingerprint and the set of templates. The geometric mean was used to give equal weights to templates of the same finger. It is possible to find some stored templates that represent partial fingerprints resulting in smaller distance, as the number of tracks to be compared will be small, and others of full fingers giving larger distances. The geometric mean overcomes this issue by giving all templates the same weight regardless of being complete or partial. A simple mean will prefer to match with partial fingerprints since they always produce smaller distances. The geometrical mean of n numbers is calculated as the nth root of the multiplication of the n numbers. In this case,

$$\text{gm1} = \sqrt[7]{25 \times 24 \times 24 \times 26 \times 27 \times 17 \times 69} = 27.487960 \quad \text{and} \quad \text{gm2} = \sqrt[7]{9 \times 9 \times 10 \times 12 \times 11 \times 7 \times 6} = 8.919279$$

Type 1	0	1	0	0	0	0	0	2	0	1	6	3	3	4	1	0	0	1	1	0	0	1	1	0
Type 2	0	2	0	1	1	2	3	2	2	2	1	1	0	1	2	1	0	0	0	0	0	0	4	1

Figure 8 A feature vector for the sample fingerprint.

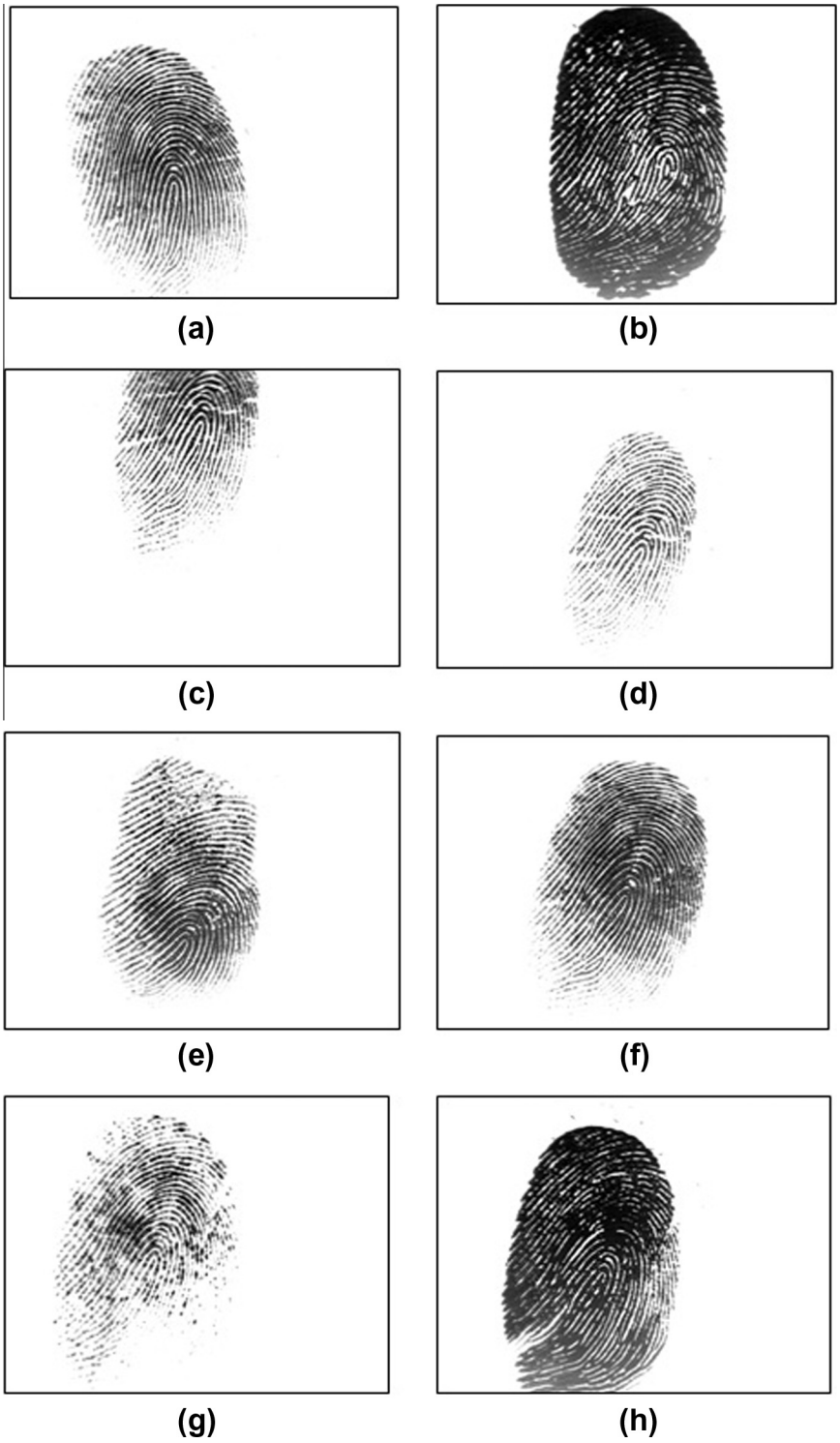


Figure 9 Different impressions of the same fingerprint.

4. Experimental results

The set of proposed algorithms (thinning, core point detection, minutiae extraction, and feature vector matching) are subjected to a set of experimental results using the FVC2000 data set [22].

The main objectives of these experiments are to identify three main threshold values; the minimum number of tracks used for successful matching (MinTRACKS) and the threshold values of the 2 means calculated (Threshold1 and Threshold2). Also, several performance evaluation criteria including error rates, time, and memory requirements will be calculated and presented.

Type 1	1	1	1	0	1	1	6	5	2	4	3	2	5	1	3	2	0	2	4	4	3	1	1
Type 2	0	2	0	0	2	1	1	1	1	0	1	0	1	1	1	1	2	0	1	1	2	0	1

(a)

Type 1	1	1	1	0	3	3	4	0	0	0	1	1	1	2	1	2
Type 2	0	0	0	1	1	0	0	1	1	0	0	1	0	1	0	0

(b)

ΔT1	0	0	0	0	2	2	2	5	2	4	2	1	4	1	2	0
ΔT2	0	2	0	1	1	1	1	0	0	0	1	1	1	0	1	1

(c)

Figure 10 (a) Feature vector of fingerprint impression in Fig. 9c, (b) Feature vector of pre-stored template of fingerprint in Fig. 9a (c) Absolute difference vector of Fig. 9a and b where only 16 columns are used for calculations.

Table 1 Distance pair for a challenge impression and a set of 7 templates.

Template	1	2	3	4	5	6	7
Type 1 sum	25	24	24	26	27	17	69
Type 2 sum	9	9	10	12	11	7	6

4.1. Evaluation dataset

The database (Fingerprint Verification Competition) FVC2000 [22] is used during experiments. FVC2000 consists of four different databases (DB1, DB2, DB3 and DB4) which were collected by using the following sensors/technologies:

- DB1: low-cost optical sensor "Secure Desktop Scanner" by KeyTronic.
- DB2: low-cost capacitive sensor "TouchChip" by ST Microelectronics.
- DB3: optical sensor "DF-90" by Identicator Technology.
- DB4: synthetic fingerprint generation.

Each database is 110 fingers wide (w) and 8 impressions per finger deep (d) (see Table 2) (880 fingerprints in all); fingers from 101 to 110 have been made available to the participants to allow parameter tuning before the submission of the algorithms; the benchmark is then constituted by fingers numbered from 1 to 100. Since not all the fingerprint images of the FVC2000 matches the required quality of having a core point and a minimum number of tracks, Special quality requirements were applied. All the selected images must adhere to the following conditions:

- Orientation, all the selected images were correctly oriented with minimum rotation.
- Full fingerprint, all the selected images were of full fingerprints.
- Core point, all images had a core point near the middle of the image.

Applying the mentioned conditions, 20 fingerprints with 3 impressions each are used for the evaluation of the proposed system, however, results for the entire dataset are then presented for comparison issues.

4.2. Performance criteria

The ultimate measure of utility of a fingerprint system for a particular application is the recognition rate. This can be described by two values [27]:

- The false acceptance rate (FAR), also known as False Match Rate (FMR), which is the ratio of the number of instances of pairs of different fingerprints, found to (erroneously) match to the total number of match attempts.
- The false rejection rate (FRR), AKA False Non-Match Rate (FNMR), is the ratio of the number of instances of pairs of the same fingerprint found not to match to the total number of match attempts.

FAR and FRR trades off against one another. That is, a system can usually be adjusted to vary these two results for the particular application, however, decreasing one increase the other and vice versa. FAR is also called false match rate or Type II error and FRR is also called false non-match rate or Type I error. These are expressed as values in $[0, 1]$ interval or as percentage values.

Some other "compact" indices are also used to summarize the accuracy of a verification system [1]:

Table 2 Description of the FVC2000 data set.

	Sensor type	Image size	Set A ($w \times d$)	Set B ($w \times d$)	Resolution (DPI)
DB1	Low-cost optical sensor	300×300	100×8	10×8	500
DB2	Low-cost capacitive sensor	256×364	100×8	10×8	500
DB3	Optical sensor	448×478	100×8	10×8	500
DB4	Synthetic generator	240×320	100×8	10×8	About 500

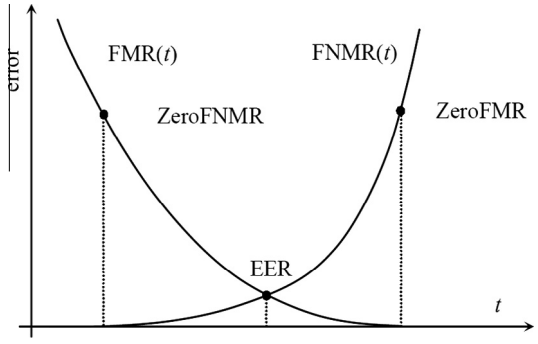


Figure 11 An example of FMR(t) and FNMR(t) curves, where the points corresponding to EER, ZeroFNMR, and ZeroFMR are highlighted.

- Equal-Error Rate (EER) denotes the error rate at the threshold t for which false acceptance rate and false rejection rate are identical: FAR (t) = FRR (t) (see Fig. 11). In practice, because the matching scores distributions are not continuous (due to the finite number of matched pairs and the quantization of the output scores), an exact EER point might not exist: In this case, instead of a single value, an interval should be reported [22].
- ZeroFNMR is the lowest FMR at which no false non-matches occur (see Fig. 11).
- Failure To Capture (FTC) rate is associated with the automatic capture function of a biometric device and denotes the percentage of times the device fails to automatically capture the biometric when it is presented to a sensor.
- Failure To Enroll (FTE) rate denotes the percentage of times users are not able to enroll in the recognition system. There is a tradeoff between the FTE rate and the accuracy (FAR and FRR) of a system. FTE errors typically occur when the recognition system performs a quality check to ensure that only good quality templates are stored in the database (Core point and minimum number of tracks in this case) and rejects poor quality templates. As a result, the database contains only good quality templates and the system accuracy (FMR and FNMR) improves.

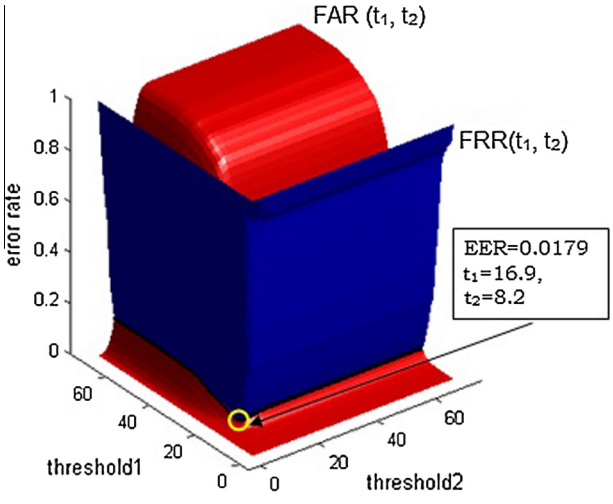


Figure 12 FRR and FAR surfaces where the intersection between the two surfaces is drawn with a solid line.

- Failure To Match (FTM) rate is the percentage of times the input cannot be processed or matched against a valid template because of insufficient quality.

Not all these measures are used to evaluate the proposed matching system as they are related more to the sensor than user behavior (alignment of the finger, pressure on the sensor surface, senor type and technology of manufacturing), namely FTC and FTE are not evaluated for this work.

4.3. Calculation of the threshold values

The 3 values to calculate for a suitable matching are MinTRACKS, Threshold1 and Threshold2.

Using the selected 60 images, it was found that the minimum number of tracks that can be constructed around the core point is 14. Other images had more tracks, so MinTRACKS is taken to be 14. Of course experiments can be done by reducing the number of tracks and calculation of the performance measure to find the minimum number of tracks, however it had been shown that reducing only one track (from 14 to 13) the performance is highly affected. Each fingerprint template (minutiae table) T_{ij}, $i = 1,\ldots,20$, $j = 1,\ldots,3$, is matched against the fingerprint images (minutiae tables) of F_i, and the corresponding Genuine Matching Scores (GMSs) are stored. The number of matches (denoted as NGRA – Number of Genuine Recognition Attempts [22]) is $20 \times 3 = 60$.

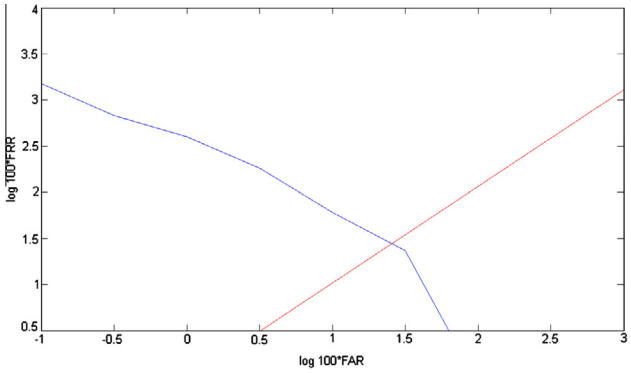


Figure 13 ROC curve.

Table 3 Comparison with Liu et al. method.

	Proposed method	Liu et al. method [2]
Method	Minutiae based	Sweat pores based
Accuracy (%)	98	92
Image resolution (DPI)	500	1200+
Complexity	Low	High

Table 4 Results for FVC2000 dataset.

Dataset	FAR	FRR	t_1	t_2
DB1	0.2113	0.2105	18	7
DB2	0.1835	0.15	18	9
DB3	0.1325	0.1525	29	12
DB4	0.1844	0.1788	10	5

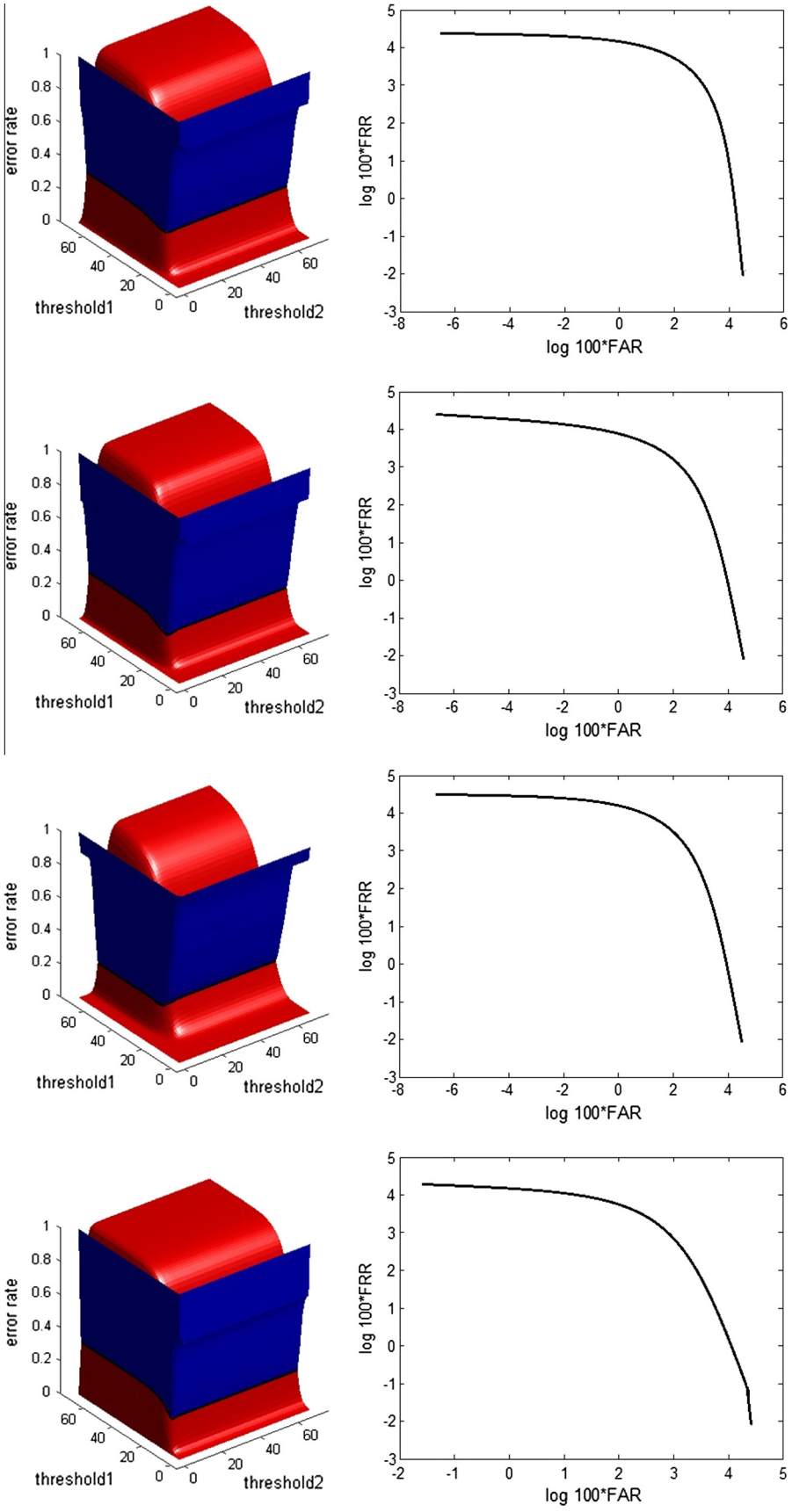


Figure 14 FAR and FRR against threshold and ROC curves for the FVC2000 datasets.

Now, FAR and FRR will be evaluated for different values of the Threshold1 and the Threshold2 where a match occurs if gm1 $< t_1$ and gm2 $< t_2$

$$\mathrm{FRR}(t_1, t_2) = \frac{\mathrm{NGMS1s} > t_1 \vee \mathrm{NGMS2s} > t_2}{\mathrm{NGRA}} \tag{3}$$

where NGMS1s is the number of genuine matching scores computed from gm1 values, NGMS2s is the number of genuine matching scores computed from gm2 values, and NGRA is the number of genuine recognition attempts which is 60.

$$\mathrm{FAR}(t_1, t_2) = \frac{\mathrm{NIMS1s} \leqslant t_1 \wedge \mathrm{NIMS2s} \leqslant t_2}{\mathrm{NIRA}} \tag{4}$$

where NIMS1s is the number of imposter matching scores computed from gm1 values, NIMS2s is the number of imposter matching scores computed from gm2 values, and NIRA is the number of imposter recognition attempts which is 1140 (19×60). The result will be 2D surfaces, FAR (t_1, t_2) and FRR (t_1, t_2) as shown in Fig. 12.

It is shown that EER = 0.0179, where it corresponds to the threshold values of t_1 = 16.92 and t_2 = 8.21. The values of thresholds are not always integers because it is not necessary for the two surfaces to intersect at integer values of thresholds. To determine the integer values of thresholds that corresponds to error rates FRR and FAR, the four possible combinations of thresholds around the two given before are tested and the two values combination that gives the minimum difference between FRR and FAR (because EER is defined as the point where FRR and FAR are equal) are considered as the thresholds t_1 and t_2 that will be used for that database for any later fingerprint recognition operation. The four possible combinations that threshold values t_1 and t_2 can take are: (16, 8), (16, 9), (17, 8), and (17, 9). The combination (17, 8) gives the minimum difference between FAR and FRR. So, when these thresholds are used in the proposed matching algorithm, the result is that FRR = 0.0167 and FAR = 0.0184.

4.4. Performance evaluation

4.4.1. Evaluation using the reduced dataset

From the previous experiments, the proposed algorithms achieve an EER of approximately 0.02 equivalent to an accuracy of 98% when applied to the test dataset for the threshold values described (14, 17, 8).

This dataset contains 20 fingerprints with 3 impressions each chosen from the FVC200 DB1_A. Those impressions are chosen for oriented complete fingerprint images with core points detected at the middle (ideal case). The True Acceptance Rate is: TAR = 1 − FAR = 1 − 0.0184 = 0.9816, and the True Rejection Rate is: TRR = 1 − FRR = 1 − 0.0167 = 0.9833. Because EER is the error where false match rate equal false non-match rate a chi-square test is: $X^2 = \mathrm{FMR}^2 + \mathrm{FNMR}^2 = 2(\mathrm{EER})^2 = 2 * (0.0179)^2 = 0.00064$.

ZeroFMR is defined as the lowest FNMR at which no False Matches occur and ZeroFNMR as the lowest FMR at which no False Non-Matches occur [22]. Because now the FRR (FNMR) and FAR (FMR) are drawn as 2D surfaces, all locations of FAR points having zero values are determined and the minimum value of the corresponding FRR values at these locations is the ZeroFAR. Also, to calculate the ZeroFAR value, all locations of FRR points having zero values are determined and the minimum value of the corresponding FAR values at these locations is the ZeroFRR.

$$\mathrm{ZeroFMR} = 0.3167 \quad \text{at } t_1 = 14 \quad \text{and} \quad \mathrm{t2} = 5,$$

$$\mathrm{ZeroFRR} = 0.0316 \quad \text{at } \mathrm{t_1} = 16 \quad \text{and} \quad \mathrm{t_2} = 10.$$

A ROC (Receiving Operating Curve) is given where FNMR is plotted as a function of FMR; the curve is drawn in log–log scales for better comprehension [22].

To draw the curve in the positive portions of x- and y-axis, FMR and FNMR values are multiplied by 100 before applying the logarithm on them. Fig. 13 shows the ROC curve of the proposed matching algorithm. To get one curve, only one column of the FAR matrix is drawn against one column of the FRR matrix, after multiplying with 100 and applying the logarithmic on both. The red line corresponds to the EER.

The achieved accuracy is very high (98%) compared to that of [3] which is 92% with much less required resolution and lower complexity as shown in Table 3.

4.4.2. Evaluation using FVC2000 dataset

In the previous section, selected images of fingerprints with core points found near the center of the image and 14 tracks constructed around this core points as a minimum number of tracks resulted in 98% EER.

Using the entire dataset of FVC2000 without the core point and minimum tracks restriction to evaluate the performance resulted in lower accuracy. The lower performance measures values are expected since all fingerprint impressions including those which didn't pass the quality measures are used. The overall accuracy of the proposed algorithm ranges from 79% to 86% as shown in Table 4. Fig. 14 shows both the error surfaces and ROC curves for the different datasets.

It is important to mention that some images did not contain a core point and in these experiments, we did not use the singular point as a reference point.

Table 6 Results for FVC2004 dataset.

Dataset	FAR	FRR	t_1	t_2
DB1	0.1571	0.136	21	7
DB2	0.091	0.1	17	9
DB3	0.1219	0.122	20	10
DB4	0.0899	0.1	15	7

Table 5 Description of the FVC2004 data set.

	Sensor type	Image size	Set A ($w \times d$)	Set B ($w \times d$)	Resolution (DPI)
DB1	Optical sensor	640×480	100×8	10×8	500
DB2	Optical sensor	328×364	100×8	10×8	500
DB3	Thermal sweeping sensor	300×480	100×8	10×8	512
DB4	Synthetic generator	288×384	100×8	10×8	About 500

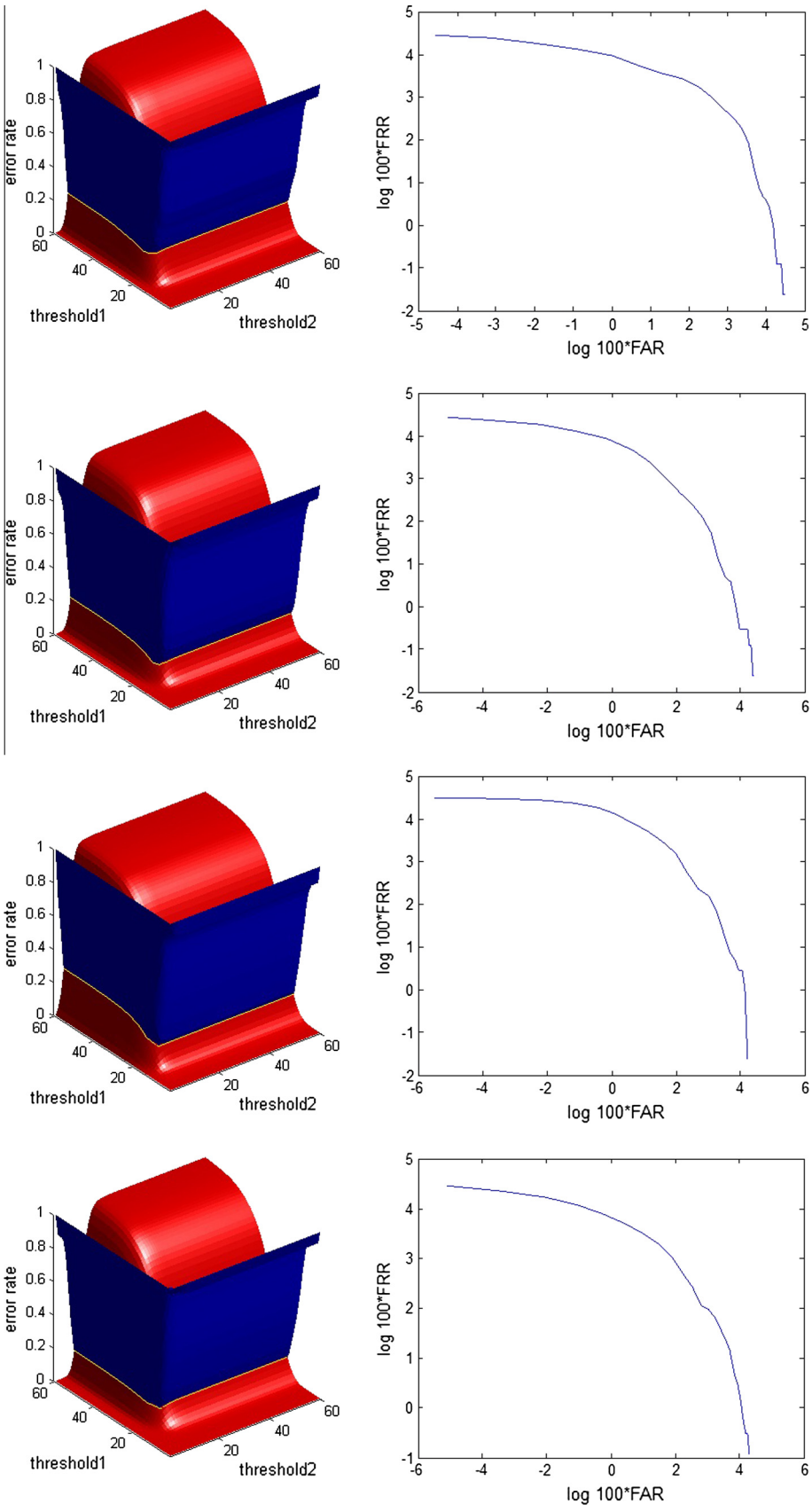


Figure 15 Results of FVC2004.

4.4.3. Evaluation using FVC2004 dataset

After the FVC2000 dataset was introduced, 2 other versions, FVC2002 and FVC2004 [1], were also available for the sake of evaluation. The main advancement of FVC2004 is the usage of new sensors, especially DB3 where a thermal sweep sensor was used to produce a 512 DPI fingerprint images.

4.4.3.1. Evaluation using core point. Table 5 shows the description of the dataset while Table 6 shows the results of the evaluation of FVC2004 datasets.

The overall accuracy of the proposed algorithm over this dataset is highly improved than of FVC2000. This is expected since better fingerprint impressions were obtained using better sensors and in most of the cases, the core point was correctly detected. Fig. 15 shows both the error surfaces and ROC curves for the different datasets.

The comparison of the proposed algorithm and that of [28,29] which is also minutiae based algorithm, is shown in Table 7.

The results of the algorithm can be even improved if the threshold values for the minimum number of tracks is increased, however in this case some of the fingerprint images will be rejected and a Failure to Enroll (FTE) value should be calculated.

4.4.3.2. Evaluation using singular point. There are situations in which the core is not present. Arch type fingerprint have no core and delta points. In this case, the system fails to enroll the image and this is counted as a false match or a false non-match. However the method presented in [30,31] can improve this situation by introducing another reference point for the mentioned situations where a core point is not detected.

The singularity point is used as a reference point and the algorithm is applied. Table 8 shows the results after applying the singularity point detection. It is also noticed that there is an improvement in the accuracy for all datasets except for DB4, where the results did not change. This is because DB4 is a synthetic one with all images having a core point.

4.4.4. Time and storage performance

It can be shown that the proposed system also performs efficiently from the point of view of consumed time and required space to store a feature vector.

Table 7 Comparison with Li et al. method.

	Proposed method	Li et. al. method [28,29]
Method	Minutiae based	Minutia based
Minimum EER	0.0899 (DB4)	0.1357 (DB3)
Maximum EER	0.1571 (DB1)	0.18903 (DB1)

Table 8 Results for FVC2004 dataset with Conti et al. singularity detection.

Dataset	FAR	FRR	t_1	t_2
DB1	0.1362	0.1152	20	7
DB2	0.0682	0.092	16	8
DB3	0.112	0.098	21	10
DB4	0.0899	0.1	15	7

The total size required to store a feature vector is N bytes where N is the MinTRACKS value used. Only 1 byte is required to store the value of the count of the 2 types of minutiae per track as 4 bits are enough to store each value. In our case only 14 bytes are used which is very compact compared with other minutiae based matching methods where at leases three values for each minutia are needed $\langle x, y, \theta \rangle$.

There are several optimizations done in the algorithms, namely:

- The complexity of binarization is totally removed as direct gray level images are processed for thinning.
- The matching is done on small feature vectors with simple math operations.

Also the time required to scan a finger print and match it to a template is very low. When tested on a 2 GHz PC, a time of 1.3 ms is required to match the challenge fingerprint feature vector against a template set of a specific user. MATLAB R2009b was used for implementation and evaluation.

5. Conclusion and future work

A novel proposed system for fingerprint matching is introduced. Several contributions in the thinning process, feature extraction and matching are given. The system is evaluated against a standard dataset and the matching, storage, and time performances are found to outperform other methods.

The proposed system is very suitable for medium resolution type of finger prints produced by commercial sensors and is expected to perform better when higher resolutions are used.

In the future work, the system is to be evaluated against higher resolution datasets and to be compared with the modern methods like sweat pores based matching and 3D matching for evaluation.

Acknowledgement

The author would like to thank Prof. Ashraf Saad Hussein, Ain Shams University, for his valuable comments and suggestions.

References

[1] Maltoni D, Maio D, Jain AK, Prabhakar S. Handbook of fingerprint recognition. second ed. London: Springer-Verlag; 2009.
[2] Ji L, Yi Z. Fingerprint orientation field estimation using ridge projection. Pattern Recogn 2008;41:1491–503.
[3] Liu F, Zhao Q, Zhang D. A novel hierarchical fingerprint matching approach. Pattern Recogn 2011;44:1604–13.
[4] Bazen AM, Gerez SH. Fingerprint matching by thin-plate spline modeling of elastic deformations. Pattern Recogn 2003;36:1859–67.
[5] Jain AK, Prabhakar S, Hong L, Pankanti S. Filterbank-based fingerprint matching. IEEE Trans Image Process 2000;9(5):846–59.
[6] Jiang X, Yau WY. Fingerprint minutiae matching based on the local and global structures. In: Proc int conf on, pattern recognition (15th), vol. 2; 2000. p. 1042–5.

[7] Yuliang H, Tian J, Luo X, Zhang T. Image enhancement and minutiae matching in fingerprint verification. Pattern Recogn 2003;24:1349–60.
[8] Liang X, Asano T. Fingerprint matching using minutia polygons. In: Proc int conf on, pattern recognition (18th), vol. 1; 2006. p. 1046–9.
[9] Jain A, Ross A, Prabhakar S. Fingerprint matching using minutiae and texture features. In: Proc int conf on image processing (ICIP) Thessaloniki, Greece; 2001. p. 282–5.
[10] Eckert G, Müller S, Wiebesiek T. Efficient minutiae-based fingerprint matching. In: IAPR conference on machine vision applications, Tsukuba Science City, Japan; 2005. p. 554–7.
[11] Zhang Y, Tian J, Cao K, Li P, Yang X. Improving efficiency of fingerprint matching by minutiae indexing. In: 19th International conference on pattern recognition, Tampa, FL; 2008.
[12] Jain A, Hong L, Bolle R. On-Line Fingerprint Verification. IEEE Trans Pattem Anal Mach Intell 1997;19(4):302–13.
[13] Ratha NK, Chen SY, Jain AK. Adaptive flow orientation-based feature extraction in fingerprint images. Pattern Recogn 1995;28(11):1657–72.
[14] Luo X, Tian J, Wu Y. A minutia matching algorithm in fingerprint verification. In: 15th ICPR int conf on pattern recognition, Barcelona, Spain, vol. 4; 2000. p. 833–6.
[15] Jie Y, fang Y, Renjie Z, Qifa S. Fingerprint minutiae matching algorithm for real time system. Pattern Recogn 2006;39:143–6.
[16] Zhu E, Yin J, Zhang G. Fingerprint matching based on global alignment of multiple reference minutiae. Pattern Recogn 2005;38:1685–94.
[17] Luping J, Zhang Y, Lifeng S, Xiaorong P. Binary fingerprint image thinning using template-based PCNNs. IEEE Trans Syst, Man, Cybernet – Part B: Cybernet 2007;37(5):1407–13.
[18] Khazaei H, Mohades A. Fingerprint matching and classification using an onion layer algorithm of computational geometry. Int J Math Comput Simul 2007;1(1):26–32.
[19] Farina A, Kovacs-Vajna ZM, Leone A. Fingerprint minutiae extraction from skeletonized binary images. Pattern Recogn 1999;32:877–89.
[20] Ravi J, Raja KB, Venugopal KR. Fingerprint recognition using minutia score matching. Int J Eng Sci Technol 2009;1(2):35–42.
[21] Humbe V, Gornale SS, Manza R, Kale KV. Mathematical morphology approach for genuine fingerprint feature extraction. Int J Comput Sci Security 2007;1(2):45–51.
[22] Maio D, Maltoni D, Cappelli R, Wayman JL, Jain AK. FVC2000: fingerprint verification competition. IEEE Trans Pattern Anal Mach Intell 2002;24(3):402–12.
[23] Lam L, Lee SW, Suen CY. Thinning methodologies: a comprehensive survey. IEEE Trans Pattern Anal Mach Intell 1992;14(9):869–85.
[24] Liu M, Jiang XD, Kot A. Fingerprint reference-point detection. EURASIP J Appl Signal Process 2005;4:498–509.
[25] Yang JC, Park DS. Fingerprint verification based on invariant moment features and nonlinear BPNN. Int J Control, Autom, Syst Dec. 2008;6(6):800–8.
[26] Nilsson K, Bigun J. Localization of corresponding points in fingerprints by complex filtering. Pattern Recogn Lett 2003;24:2135–44.
[27] O'Gorman L. An overview of fingerprint verification technologies. Elsevier Inform Security Tech Rep 1998;3(1):21–32.
[28] Liu Lingli, Li L. Preprocessing and minutiae extraction of fingerprint image. J Comput Eng Applic 2006;32(16):190–2.
[29] Lv Yu-hua, Li L. Extracting fingerprint minutiae feature combined with structure information. J Comput Eng Applic 2009;45(7):184–6.
[30] Conti V, Militello C, Sorbello F, Vitabile S. A frequency-based approach for features fusion in fingerprint and iris multimodal biometric identification systems. IEEE Trans Syst, Man, Cybernet 2010;40(4):384–95.
[31] Conti V, Militello C, Sorbello F, Vitabile S. A multimodal technique for an embedded fingerprint recognizer in mobile payment systems. Int J Mobile Inform Syst 2009;5(2):105–24.
[32] Awady A, Baba K. Singular point detection for efficient fingerprint classification. Int J New Comput Arch Applic (IJNCAA) 2012;2(1):1–7.
[33] Zhao Q, Zhang L, Zhang D, Luo N. Direct pore matching for fingerprint recognition. In: Proceedings of ICB'09; 2009. p. 97–606.
[34] Liu F, Zhao Q, Zhang L, Zhang D. Fingerprint pore matching based on sparse representation. In: Proceedings of the 20th international conference on, pattern recognition; 2010.
[35] Feichtinger Hans G, Strohmer Thomas. Gabor analysis and algorithms. Birkhäuser; 1998. ISBN 0-8176-3959-4.
[36] Saleh Amira M, Bahaa-Eldin Ayman M, Wahdan AA. A modified thinning algorithm for fingerprint identification systems. In: International conference on computer engineering and systems, 2009. ICCES 2009. p. 371–6.

Ayman Mohammad Bahaa-Eldin is an associate professor, Computer and Systems Eng. Dept. Ain Shams University. Dr. Bahaa-Eldin received his B.Sc., M.Sc. and Ph.D. in Computer Engineering from Ain Shams University in 1995, 1999, and 2004 respectively. He was visiting professor in several local and international universities. His fields of research are cryptography, computer networks, and computer and network security. He published more than 30 papers in refereed international journals and conferences, and participated in the reviewing process of many international journal and conferences.