



A novel ant colony optimization algorithm for large-distorted fingerprint matching

Kai Cao^{a,1}, Xin Yang^{b,1}, Xinjian Chen^{c,1}, Yali Zang^b, Jimin Liang^a, Jie Tian^{a,b,*}

^a Life Sciences Research Center, School of Life Sciences and Technology, Xidian University, Xi'an 710071, China

^b Institute of Automation, Chinese Academy of Sciences, Beijing 100190, China

^c Radiology and Imaging Sciences Department, Clinical Center, National Institute of Health, MD 20892, USA

ARTICLE INFO

Article history:

Received 15 November 2010

Received in revised form

20 April 2011

Accepted 24 April 2011

Available online 14 May 2011

Keywords:

Distortion

Fingerprint matching

Minutiae pairing

Minutia similarity

Ant colony optimization

ABSTRACT

Large distortion may be introduced by non-orthogonal finger pressure and 3D–2D mapping during the process of fingerprint capturing. Furthermore, large variations in resolution and geometric distortion may exist among the fingerprint images acquired from different types of sensors. This distortion greatly challenges the traditional minutiae-based fingerprint matching algorithms. In this paper, we propose a novel ant colony optimization algorithm to establish minutiae correspondences in large-distorted fingerprints. First, minutiae similarity is measured by local features, and an assignment graph is constructed by local search. Then, the minutiae correspondences are established by a pseudo-greedy rule and local propagation, and the pheromone matrix is updated by the local and global update rules. Finally, the minutiae correspondences that maximize the matching score are selected as the matching result. To compensate resolution difference of fingerprint images captured from disparate sensors, a common resolution method is adopted. The proposed method is tested on FVC2004 DB1 and a FINGERPASS cross-matching database established by our lab. The experimental results demonstrate that the proposed algorithm can effectively improve the performance of large-distorted fingerprint matching, especially for those fingerprint images acquired from different modes of acquisition.

© 2011 Elsevier Ltd. All rights reserved.

1. Introduction

Fingerprints are graphical patterns of ridges and valleys on the skin surface of fingertips [25]. Due to its uniqueness, a fingerprint is considered to be one of the most reliable biometrics for personal verification. Widely known applications include the US-VIST program instituted by the Department of Homeland Security (DHS) and the IAFIS service developed by the Federal Bureau of Investigation (FBI) [30].

Fingerprint recognition has been studied for many years and numerous algorithms have been proposed to improve the performance of the automatic fingerprint identification system (AFIS). Among them, minutiae-based matching algorithms are the most popular approaches since they are widely believed that minutiae are the most discriminating and reliable features. However, there are still some challenging problems in minutiae-based fingerprint matching algorithms. Firstly, fingerprints may possess a large non-linear distortion caused by non-orthogonal finger pressure and the 3D to 2D mapping process [4]. Fig. 1 shows a pair of

fingerprints with large distortion from FVC2004 DB1. While the corresponding minutiae in the rectangle region are approximately overlapped, the maximal distance between corresponding minutiae in the elliptical region is more than 100 pixels [5]. Secondly, various types of fingerprint sensors in the biometric market introduce large variations into fingerprint resolution, distortion patterns and noise. Two fingerprints captured from disparate sensors are illustrated in Fig. 2, in which Fig. 2(a) is captured from a sweep mode sensor, while Fig. 2(b) is captured from a press mode sensor. Finally, minutia of a query fingerprint may fall within the tolerance area of more than one minutiae of the template fingerprint as shown in Fig. 3. Therefore, it is difficult to find the optimal pair since one minutia can match with at most one minutia in the minutiae pairing process. In addition, the effect between matching score computation and minutiae correspondences is unidirectional. There is no feedback to guide minutiae correspondences establishment. Due to these confounding factors, minutiae-based fingerprint matching is a complex combinatorial optimization problem. It is a crucial and challenging task to design a powerful matching algorithm for improving the performance of large-distorted fingerprint matching.

One way of tackling the above problems simultaneously is to use ant colony optimization (ACO), which is a paradigm for designing metaheuristic algorithms for combinatorial optimization problems.

* Corresponding author: Life Sciences Research Center, School of Life Sciences and Technology, Xidian University, Xi'an 710071, China.

E-mail address: tian@ieee.org (J. Tian).

¹ These authors contributed equally to this work.

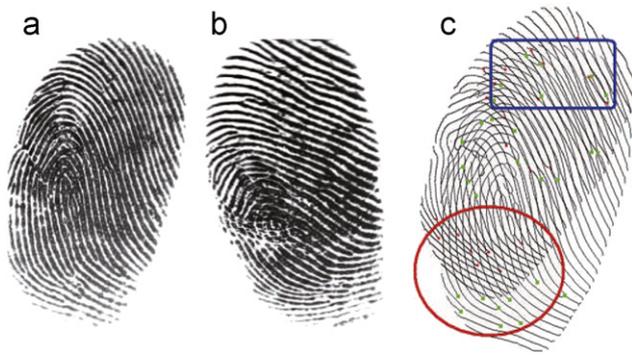


Fig. 1. An example of large distortion from FVC2004 DB1 [5]. (a) 102_3.tif; (b) 102_5.tif; and (c) the image is fingerprint 102_5 (after registration) added to 102_3. In the rectangular region, the corresponding minutiae are approximately overlapped. While in the elliptical region, the maximal vertical difference of corresponding minutiae is greater than 100 pixels.

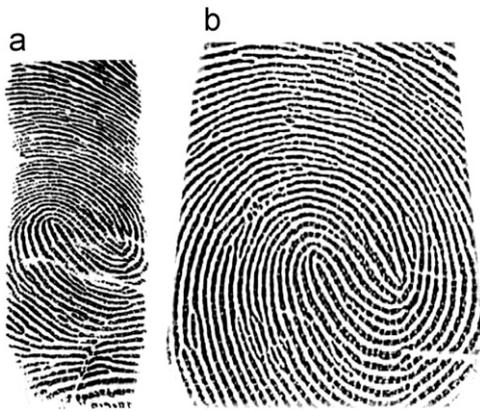


Fig. 2. Fingerprint images captured from different types of sensors. (a) Fingerprint image from the sweep sensor and (b) fingerprint image from the press sensor.

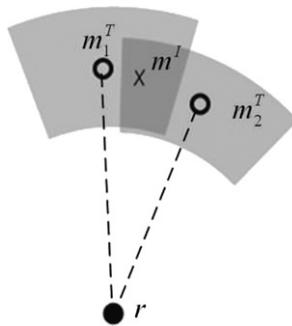


Fig. 3. Ambiguity of minutiae correspondences. r is the reference point, m_1^T and m_2^T are minutiae from the template fingerprint while m^I is a minutia from the input fingerprint.

The first ACO algorithm was developed by Colomi et al. [8], who successfully applied it to the traveling salesman problem (TSP) based on the path-finding abilities of real ants [11,10]. It simulates the behavior of ant colonies in nature as they forage for food and find the most efficient routes from their nests to food sources. While walking, ants deposit pheromones on the ground and follow the pheromones previously deposited by other ants. The essential trait of ACO algorithms is the combination of a priori information regarding the structure of a promising solution with a posteriori information regarding the structure of previously obtained good solutions [10].

In this paper, we propose a novel ACO algorithm for large-distorted fingerprint matching. Minutiae similarities are

measured by their orientation descriptor and local minutiae structure, and are viewed as heuristic values in ACO. Local minutiae matching is utilized to construct an assignment graph from which the artificial ants can find all minutiae correspondences. Pseudo-random proportional rule is adopted to select minutiae correspondences, and a new state transition rule is proposed for local propagation. For the fingerprint images captured by disparate sensors, a common resolution method is adopted to compensate for different resolutions. Experiments on FVC2004 DB1 and the FINGERPASS cross-matching database demonstrate that the proposed algorithm can effectively improve the performance of large-distorted fingerprint matching, especially for those fingerprint images acquired from different modes of acquisition.

The rest of the paper is organized as follows: Section 2 provides a review of previous attempts to tackle distortion in fingerprint images. Section 3 provides feature extraction and fingerprint representation. Section 4 describes the ACO-based minutiae pairing algorithm. The experimental results are reported in Section 5 and conclusions are drawn in Section 6.

2. Related work

Various minutiae-based fingerprint matching algorithms have been proposed to deal with distortion [25]. Due to translation, rotation and non-linear distortion, the correspondences between minutiae are very ambiguous. Researchers tried to attach local features to reduce the ambiguity. There are mainly three kinds of local features used in fingerprint verification. The ridge information associated with minutiae is the first kind of local feature introduced to select the reference minutiae pair [17]. Minor modifications of this strategy have been proposed by other researchers [15,21]. The second kind of local features is the local minutiae structural feature. Jiang and Yau [20], and Jea and Govindaraju [19] utilized k closest neighboring minutiae points to generate a fixed-length feature for each minutia and the similarities between minutiae were based on these features. Chen et al. [4] and Ratha et al. [27] adopted similar strategies by defining a feature vector which characterized the rotation and translation invariant relationship between a minutia and its neighbors circled within a radius. The third feature is the orientation features around minutia. The approaches presented in Refs. [33–35] used the orientation features for the minutiae similarity measure.

In the minutiae correspondences establishment process, classical methods aligned two minutiae sets by choosing a reference minutiae pair (one from the input fingerprint and the other from the template fingerprint) [17], and only translation and rotation were considered. In these methods, the corresponding minutiae located far away from the reference minutiae may have a larger position and direction difference than those adjacent to the reference minutiae. In order to tolerate distortion, Jain et al. [17] presented a fixed-size bounding box to match minutiae. Luo et al. [24] improved this algorithm by applying a changeable bounding box during the matching process which made it more robust to non-linear distortion. An alternative approach is to find a transformation to globally align the two minutiae sets. Zhu et al. [36] claimed that using multiple pairs of reference minutiae to estimate transformation gave better results. Tan and Bhanu [32] proposed a fingerprint matching approach based on genetic algorithm (GA), which tried to find the optimal transformation parameters (including scale, translation and rotation) between the two fingerprint images. Sheng et al. [31] developed a memetic fingerprint matching algorithm by introducing an efficient matching operation to produce an initial population and combining the

use of the global search functionality of GA with a local improvement operator to search for the optimal or near optimal global alignment. During this process, minutiae alignment intends to find the geometry transform of translation and rotation. However, both of these algorithms have not considered non-linear distortion. Kovacs-Vajna [22] proposed a method based on triangular matching and utilized dynamic time warping (DTW) to validate these minutiae correspondences founded by triangular matching. However, without DTW for further verification, the results are not acceptable [32]. Chikkerur et al. [7] developed a dual traversal algorithm called coupled BFS for consolidating all local neighborhood matches. In this method, only local neighborhoods were considered at each stage and a dynamic programming based optimization approach was employed to obtain the minutiae correspondences.

There are different attempts to deal with non-linear distortion in fingerprint images. Some of them focus on detecting distortion in the fingerprint capture process. Ratha and Bolle [26] proposed to measure the forces and torques on the scanner directly, while Dorai et al. [9] proposed to detect and estimate distortion occurring in fingerprint videos. However, both of these methods cannot deal with the captured fingerprint images. Bazen and Gerez [2] employed the thin-plate spline (TPS) model to describe non-linear distortion between two fingerprints. However, this method is highly dependent on the initialization of minutiae pairs, which are usually obtained by using rigid transformation. Ross et al. [28] proposed an average deformation model to cope with non-linear distortion. He et al. [14] proposed a global comprehensive similarity-based fingerprint matching algorithm, in which minutia-simplex, including a pair of minutiae as well as their associated textures, were employed to achieve fingerprint matching. Chen et al. [5] proposed a fuzzy feature based on a local triangle feature set to match the deformed fingerprint images.

Due to the development in the fingerprint market and fingerprint sensing technology, there are many fingerprint sensors on the market. Different sensor modes produce fingerprint images with different characteristics, which greatly challenge traditional matching algorithms. However, only a few works focused on the variations of fingerprints in cross-matching. Ross and Jain [29] discussed the problem of biometric sensor interoperability and presented a case study involving two different fingerprint sensors. Results confirmed that when the images being matched originated from two different sensors, then the performance of the matcher drastically deteriorated. Jang et al. [18] improved the interoperability of fingerprint recognition using resolution compensation based on sensor evaluation. However, it ignored different distortion patterns resulting from different capture modes and was only available for the press sensors. Ross and Nadgir [30] proposed a non-linear calibration scheme based on the TPS model to register a pair of fingerprint sensors. However, it is unable to compensate for the variations by an average model. The fingerprint distortion is affected by various factors (such as the press direction of the finger of the press sensor and moving speed of the sweep sensor).

3. Feature extraction and representation

For a gray-scale input fingerprint image captured at a resolution of d DPI (dots-per-inch), the orientation field is calculated by the approach proposed by Bazen and Gerez [1], the method described by Hong et al. [16] is used to enhance the image, and the thinned ridge map is then obtained. The thinned ridge map is post-processed by using Luo's method [23]. Local features are detected on the thinned ridge map and the orientation field. The set of local features is denoted as $M = \{(m_i, f_i)\}_{i=1}^N$, where

N denotes the number of detected minutiae, $m_i = (x_i, y_i, \theta_i, r_i)$ includes x, y coordinates, direction and reliability of the i th minutiae, respectively, f_i denotes the transform-invariant feature vector corresponding to the i th minutiae, which will be discussed in the next section. Minutiae coordinates and directions are detected by Hong's method [16]. It is difficult to reliably extract minutiae from the input fingerprint, especially from low-quality fingerprints. The performance of the matching algorithm highly depends on the quality of the fingerprint image and the reliability of minutiae. For each minutiae m_i , the method proposed by Feng [12] is used to classify it as reliable or unreliable. In this method, all the ridges associated with m_i or surrounding m_i are examined. For the configurations of these ridges of termination and bifurcation, we refer the authors to Ref. [12]. If the lengths of the ridges are all longer than the threshold, then m_i is regarded as a reliable one ($r_i=1$), otherwise it is an unreliable one ($r_i=0$). Since various resolution sensors are involved, we set the threshold as $12 \times d/500$ pixels in this work.

The segmentation of the fingerprint foreground plays an important role not only in the feature extraction but also in the minutiae similarity calculation. In this paper, the foreground of the fingerprint image is obtained by the approach proposed by Chen et al. [6]. We use its convex hull ($C = \{(x_i, y_i)\}_{i=1}^{N_c}$, where N_c is the number of vertices of the convex hull, (x_i, y_i) are the x and y coordinates of the i th vertex) to approach the fingerprint foreground. Then, the features of a fingerprint can be represented as $F = \{M, C, d\}$. For the fingerprint images captured from different resolutions, there are two overall resolution compensation schemes in the image level as well as the template level using a common resolution method and relative resolution method [18]. In this paper, we focus on minutiae correspondences establishment and then we adopt a common resolution method for simplicity. For each coordinate (x, y) from the minutiae set or convex hull, its transformed version (x_t, y_t) is calculated as follows:

$$x_t = d_0/d \times x \quad (1)$$

$$y_t = d_0/d \times y \quad (2)$$

where d_0 is the common resolution. Without introducing ambiguity, we use $F = \{M, C\}$ to denote the transformed features.

4. Minutiae matching by ACO

In this section we discuss the proposed minutiae matching algorithm in detail. The overview of the ACO-based minutiae matching algorithm is given in Fig. 4.

Algorithm: ACO for minutiae matching

Initialization:

Construct assignment graph,
Set parameters,
Initialize pheromone trails

While (termination conditions are not met)/* an iteration */

For each ant /* a step */

Construct a solution
Apply local pheromone trail updating rule

End

Apply global pheromone trail updating rule

End

Fig. 4. An overview of the proposed ACO algorithm for minutiae matching.

4.1. Problem formulation

Suppose that there are N^I minutiae in the input fingerprint feature set $F^I=(M^I,C^I)$ and there are N^T minutiae in the template fingerprint feature set $F^T=(M^T,C^T)$. By representing the minutiae as point patterns, minutiae matching can be viewed as a minutiae point pattern matching problem. Since the relative transformation between two fingerprints is unknown in advance, the correspondences between minutiae are very ambiguous and each minutia in the input fingerprint can be matched to any minutiae of the template fingerprint [12] and vice versa. Therefore, the matching function \mathcal{M} can be viewed as a binary function over the product set of M^I and M^T , which is denoted by,

$$\mathcal{M} : M^I \times M^T \rightarrow \{0,1\} \quad (3)$$

where \times denotes the direct product of the two sets and the function value 1 means matching and 0 is mismatching. Defined on the space of a Cartesian product set, the binary function can be regarded as a binary assignment matrix of N^I by N^T , i.e.

$$A = \begin{bmatrix} a_{11} & a_{12} & \dots & a_{1N^T} \\ a_{21} & a_{22} & \dots & a_{2N^T} \\ \vdots & \vdots & \vdots & \vdots \\ a_{N^I1} & a_{N^I2} & \dots & a_{N^IN^T} \end{bmatrix}, \quad a_{ij} \in \{0,1\} \quad (4)$$

where the elements a_{ij} correspond to the assignment of m_i^I to m_j^T . In the minutiae pairing process, one minutia can match with at most one minutia. Therefore, assignment A should be subject to the injective constraint, that is $\sum_{i=1}^{N^I} a_{ij} \leq 1$ and $\sum_{j=1}^{N^T} a_{ij} \leq 1$. The minutiae matching process can then be defined as the problem of finding the best correspondences between the input minutiae set and the template minutiae set, which optimizes a given objective function.

4.2. Minutiae similarity

Local features of a minutia describe the characteristics of the minutia in its neighborhood. These features indicate the probability that two minutiae should be matched and reduce the ambiguity between minutiae. In this paper, we combine local orientation and local minutiae structure to measure the similarity between minutiae [3].

The local orientation-based descriptor proposed by Tico and Kuosmanen [33] has been used to find potential matches. In this method, the descriptor consists of the orientation distances between the minutia and the sampling points around the minutia in a circular pattern. The circular pattern consists of L concentric circles of radii $r_l (1 \leq l \leq L)$ and K_l sampling points are equally sampled on the l th circle. They reported that the configuration $((r_0=27, K_0=10), (r_1=45, K_1=16), (r_2=63, K_2=22)$ and $(r_3=81, K_3=28))$ obtained the best performance on FVC2000 DB1 and DB2, in which fingerprint images were captured at a resolution of 500DPI. In this paper, we take cross-matching into account. Therefore, we modify the sampling radii using a linear function with respect to the fingerprint image resolution d , known as:

$$r'_l = r_l \times d/500 \quad (5)$$

Suppose that m_p^I is a minutia in the input fingerprint, m_q^T is a minutia in the template fingerprint, and $f_p^I = \{\alpha_{k,l}\}$ and $f_q^T = \{\beta_{k,l}\}$ are their corresponding transform-invariant feature vectors. The orientation similarity between these two feature vectors is calculated as

$$OS_{pq} = 1/76 \sum_{l=1}^L \sum_{k=1}^{K_l} s(A_1(\alpha_{k,l}, \beta_{k,l})) \quad (6)$$

where $A_1(\theta_1, \theta_2)$ is the orientation distance between θ_1 and θ_2 , and $s(x)$ denotes a similarity value with respect to the orientation difference x as follows:

$$s(x) = e^{-x/(\pi/16)} \quad (7)$$

The method proposed in our previous work is adopted to measure the local minutiae structural similarity [3]. The calculation of local minutiae structural similarity between m_p^I and m_q^T has two stages.

In stage (1), minutia m_p^I and its neighbors are mapped on the coordinate system of m_q^T . Let $N(m_p^I, r) = \{m_{p_i}^I\}_{i=1}^{n_p}$ denote the set of the neighboring minutiae circled m_p^I within r radius in input fingerprint, $N(m_q^T, r + \Delta r) = \{m_{q_j}^T\}_{j=1}^{n_q}$ denote the set of the neighboring minutiae circle m_q^T within $r + \Delta r$ radius in the template fingerprint and T_r represent the corresponding rigid transformation from m_p^I to m_q^T . Each minutia $m_{p_i}^I$ in $N(m_p^I, r)$ is mapped to $m_{p_i}^T$ using T_r . Then, the contribution of $m_{p_i}^I$ with respect to minutia m_q^T is computed as

$$C_{p_i} = \max_{m_{q_j}^T \in N(m_q^T, r + \Delta r)} f(D(m_{p_i}^I, m_{q_j}^T), d_1, d_2) \cdot f(|A_2(\theta_{p_i}^I, \theta_{q_j}^T)|, \theta_1, \theta_2) \quad (8)$$

where d_1 and d_2 are two distance thresholds, θ_1 and θ_2 are two direction distance thresholds, and function f , D and A_2 are defined as,

$$f(x, th_1, th_2) = \begin{cases} 1 & \text{if } x \leq th_1 \\ 0 & \text{if } x > th_2 \\ \frac{th_2 - x}{th_2 - th_1} & \text{otherwise} \end{cases} \quad (9)$$

$$D(m_p, m_q) = \sqrt{(x_p^I - x_q^T)^2 + (y_p^I - y_q^T)^2} \quad (10)$$

$$A_2(\theta_p^I, \theta_q^T) = \begin{cases} \theta_p^I - \theta_q^T & \text{if } |\theta_p^I - \theta_q^T| \leq \pi \\ \theta_p^I - \theta_q^T - 2\pi & \text{if } (\theta_p^I - \theta_q^T) > \pi \\ \theta_p^I - \theta_q^T + 2\pi & \text{otherwise} \end{cases} \quad (11)$$

If C_{p_i} is larger than 0, $m_{p_i}^I$ is regarded as a local matched minutia.

In stage (2), we define two other neighboring minutiae sets: $N(m_q^T, r)$ and $N(m_p^I, r + \Delta r)$. They are similar as in stage 1. We use the same symbol T_r to represent the relative rigid transformation from m_q^T to m_p^I . Each minutia $m_{q_j}^T \in N(m_q^T, r)$ is mapped to $m_{q_j}^I$ using T_r . The contribution of $m_{q_j}^T$ to the minutia m_q^T is calculated as follows:

$$C_{q_j} = \max_{m_{p_i}^I \in N(m_p^I, r + \Delta r)} f(D(m_{p_i}^I, m_{q_j}^T), d_1, d_2) \cdot f(|A_2(\theta_{p_i}^I, \theta_{q_j}^T)|, \theta_1, \theta_2) \quad (12)$$

The structural similarity between m_p^I and m_q^T is measured using the following formula:

$$MS_{pq} = \frac{1 + \sum_{m_{p_i}^I \in N(m_p^I, r)} C_{p_i}}{M_p + bias} \cdot \frac{1 + \sum_{m_{q_j}^T \in N(m_q^T, r)} C_{q_j}}{M_q + bias}, \quad (13)$$

where M_p and M_q represent the number of minutiae that should be matched [12] in $N(m_p, r)$ and $N(m_q, r)$, respectively, and $bias$ is a parameter ($bias = 2$ in our experiments). Minutia m is regarded as a minutia that should be matched if m is a local matched minutia or m' is a reliable minutia and m' is located inside the convex hull of the other fingerprint.

Two similarity functions are combined to measure the similarity between the minutiae pair by the product rule

$$S_{pq} = OS_{pq} \cdot MS_{pq} \quad (14)$$

Let $s = \{s_{pq}\}_{p=1, q=1}^{N^I, N^T}$ denote the set of similarity degrees between two minutiae sets. However, a minutia may exhibit a large similarity degree with more than one minutia. In order to identify the most distinguishable pairs of corresponding minutiae, the similarity degree set s is normalized by the method proposed by

Feng [12] as

$$ns_{pq} = \frac{s_{pq} \cdot (N^I + N^T - 1)}{\sum_{k=1}^{N^I} s_{kq} + \sum_{k=1}^{N^T} s_{pk} - s_{pq}} \quad (15)$$

4.3. Objective function

Objective function assigns an objective value to each feasible solution, which indicates the degree of suitability of the solution and guides the behavior of the ants. In this work, we use the matching score function as the objective function. Then, the minutiae matching problem can be viewed as a constraint maximization problem, and the goal is to find a globally optimal feasible solution A^* , that is, a maximum feasible solution for this maximization problem. In this paper, we adopt the following formula as the objective function to measure the matching status:

$$\begin{aligned} score &= \frac{2 \sum_{i=1}^{N^I} \sum_{j=1}^{N^T} a_{ij} s_{ij}}{CN^I + CN^T} \left(1 - \exp \left(- \sum_{i=1}^{N^I} \sum_{j=1}^{N^T} a_{ij} / \sigma \right) \right) \\ &= \frac{2 \sum_{k=1}^n s_{i_j k}}{CN^I + CN^T} (1 - \exp(-n/\sigma)) \end{aligned} \quad (16)$$

where n denotes the number of matched minutiae, $\{(i_k j_k)\}_{k=1}^n$ denotes the matched minutiae pair set, CN^I and CN^T denote the number of minutiae that should be matched for the input fingerprint and the template fingerprint, respectively, and σ is a control parameter. The average rotation and translation parameters between two fingerprint images are estimated by the matched minutiae pairs. It is then easy to obtain the values of CN^I and CN^T .

4.4. Assignment graph construction

When applying the ACO metaheuristic strategy to find the correspondences between minutiae, the most important step is to translate the minutiae matching problem into a graph from which the artificial ants can find solutions. An assignment graph $G^a=(V,E)$ is first constructed (totally $N^I \times N^T$ vertices ($V = \{v_{ij}\}_{i=1, j=1}^{N^I, N^T}$) are constructed in all). Each vertex in V corresponds to an assignment. For example, v_{ij} corresponds to an assignment of m_i^I to m_j^T . To reduce the search space and computational complexity, a local matching process is proposed to construct the connection among the graph vertexes. A directed edge (e_{ab}) exists from vertex $a = v_{i_0 j_0}$ to vertex $b = v_{i_1 j_1}$ if these two assignments satisfy the following three conditions:

- (1) $m_{i_1}^I$ is a neighbor of $m_{i_0}^I$ ($m_{i_1}^I \in N(m_{i_0}^I, r)$);
- (2) $m_{j_1}^T$ is a neighbor of $m_{j_0}^T$ ($m_{j_1}^T \in N(m_{j_0}^T, r)$);
- (3) $m_{i_1}^I$ and $m_{j_1}^T$ are matchable under the coordinate system of $m_{i_0}^I$ and $m_{j_0}^T$ (transform $m_{i_1}^I$ to $m_{i_1}^I$ using the rigid transformation from $m_{i_0}^I$ to $m_{j_0}^T$ and $f(D(m_{i_1}^I, m_{j_1}^T), d_1, d_2) \cdot f(A_2(m_{i_1}^I, m_{j_1}^T), \theta_1, \theta_2) > 0$).

Pheromone trails and heuristic values determine the probability distribution of the state transition, a key procedure in solution construction. In traditional approaches, each edge is initialized with a pheromone trail value and a heuristic value [10], which will lead to two arrays with a size of $N^I \times N^I \times N^T \times N^T$. The computational cost is very huge. In this paper, we aim to seek the optimal correspondences between the input minutiae set and the template minutiae set. Thus, each vertex such as $a = v_{ij}$ is associated with a pheromone trail τ_a and a heuristic value η_a . In the initialization, τ_a is initialized with a prefixed value τ_0 , and η_a is set as the normalized minutiae similarity (ns_{ij}) between the i th

minutia in the input fingerprint and the j th minutia in the template fingerprint.

4.5. Solution construction

In the proposed ACO-based minutiae matching algorithm, an artificial ant stands at a vertex and successively selects neighboring vertices to visit until no more vertices can be visited, because positions and directions of minutiae in a local region are less affected by non-linear distortion. In order to select an initial pair to start the solution construction conveniently, a virtual vertex v_0 is added to the assignment graph G^a . The vertices with the largest K heuristic value are inserted into the adjacency list of the virtual vertex. All of the ants start at the virtual vertex. The overview of the solution construction algorithm is given in Fig. 5. At the beginning of each iteration, the solution for ant k (S_k) is set as empty. For ant k positioned on vertex a , we first construct a set of candidate vertices $\mathcal{A}_k(a)$ which are allowed to be visited by ant k . A vertex $v = v_{ij}$ is deemed as $v \in \mathcal{A}_k(a)$ if it satisfies the following two conditions:

1. v has not been visited by ant k in this iteration;
2. There does not exist a vertex in S_k associated with the i th minutia in the input fingerprint and the j th minutia in the template fingerprint.

The state transition rule is as follows: ant k positioned at vertex a chooses vertex b to visit by applying the rule given in Eq. (17):

$$b = \begin{cases} \arg \max_{v \in \mathcal{A}_k(a)} \{[\tau_v] \cdot [\eta_v]^\beta\} & \text{if } q \leq q_0 \\ B & \text{otherwise} \end{cases} \quad (17)$$

where β is a parameter which determines the relative importance of the pheromone trail vs. heuristics ($\beta > 0$), q is a random number uniformly distributed in $[0,1]$, q_0 is a parameter ($0 \leq q_0 \leq 1$), which determines the relative importance of exploitation vs. exploration [10]. B is a random variable selected according to the probability distribution give in Eq. (18). If $q \leq q_0$ then the best vertex is chosen (exploitation). Otherwise, next vertex is selected according to the probabilistic distribution

Algorithm: construct solution for ant k

Initialization:
 Let Q represent a FIFO queue
 Let S_k be empty
 ENQUEUE(Q, v_0)

While (Q is not empty)
 $a = \text{DEQUEUE}(Q)$
 Construct $\mathcal{A}_k(a)$
While ($\mathcal{A}_k(a)$ is not empty)
 Select next vertex (e.g. $b = v_{i_1 j_1}$) according to (17)
 Check the compatibility of b with respect to S_k
 If b satisfies compatibility condition
 $S_k = S_k \cup b$
 ENQUEUE(Q, b)
 Remove the vertices associated with i_1 and j_1 from $\mathcal{A}_k(a)$
 Else
 Remove vertex b from $\mathcal{A}_k(a)$
 End if
End while
End while

Fig. 5. An overview of the solution construction algorithm.

(biased exploration):

$$p_k(a,v) = \begin{cases} \frac{\{\tau_v\} \cdot [\eta_v]^\beta}{\sum_{u \in \mathcal{A}_k(a)} \{\tau_u\} \cdot [\eta_u]^\beta} & \text{if } v \in \mathcal{A}_k(a) \\ 0 & \text{otherwise} \end{cases} \quad (18)$$

In this way, we favor the choice of vertices which have larger minutiae similarities and a great amount of pheromones.

For a selected vertex (e.g. $b = v_{i_1 j_1}$), an additional procedure is done to check its compatibility with respect to the selected vertices in S_k . If there exists a vertex $c = v_{i_2 j_2} \in S_k$ that satisfies $m_{i_1}^I \in N(m_{i_2}^I, r)$ and $m_{j_1}^T \in N(m_{j_2}^T, r)$, but b is not in the adjacency list of c , then we consider that b does not satisfy the compatibility condition and b is removed from $\mathcal{A}_k(a)$. Otherwise, $(m_{i_1}^I, m_{j_1}^T)$ is regarded as a matched minutiae pair and vertex b is appended to S_k . All of the vertices that are associated with the i_1 th minutia in F^I and the j_1 th minutia in F^T are removed from $\mathcal{A}_k(a)$. Therefore, the injectivity constraint and the compatibility among matched minutiae are satisfied simultaneously in this seeking process. In the case that there are more than one connected components of the assignment graph, this procedure is repeated until all connected components are considered. Finally, all minutiae correspondences found by the same ant are combined as a solution to calculate the objection function value.

4.6. Pheromone trail updating

The proposed algorithm has two kinds of pheromone trail updating rules, namely the local pheromone updating rule and the global pheromone updating rule [8]. The local pheromone updating rule is intended to avoid that a very strong edge being selected by all the ants. It is applied when an ant has constructed a solution. The pheromone on the vertex v chosen by the ants is updated based on the following rule:

$$\tau_v = (1 - \rho) \cdot \tau_v + \rho \cdot \tau_0 \quad (19)$$

where $0 < \rho < 1$ is a local pheromone decay parameter, and τ_0 is the initial pheromone.

On the other hand, the global updating rule intensifies the search in the neighborhood of the best solution. In this rule, only the best solution is used to globally modify the pheromone trail. Global pheromone trail updating is performed according to the following rule:

$$\tau_v = (1 - \alpha) \cdot \tau_v + \alpha \cdot \text{score}_{best} \quad (20)$$

where $0 < \alpha < 1$ is a global pheromone decay parameter, score_{best} is the largest score of the minutiae correspondences generated by ants since the beginning of the algorithm, and v is a vertex of the corresponding solution. Since at most $\min(N^I, N^T)$ pairs of minutiae are established by an ant, there are at most $\min(N^I, N^T)$ vertices needed to be updated in either local pheromone trail updating or global pheromone trail updating.

4.7. Termination conditions

Termination conditions decide the matching time and possibly how good the solution is. The termination conditions we used are: (1) Terminate ACO if the matching score is greater than the threshold, then it is regarded as a genuine match and it is unnecessary to continue the iteration; (2) Terminate ACO if the number of iteration exceeds a prefixed threshold.

5. Experimental results

In this section, we conduct a series of experiments on FVC2004DB1 and FINGERPASS cross-matching database to evaluate the performance of the ACO-based minutiae matching

algorithm. We first illustrate the database and protocol used in the performance evaluation and then present the matching results.

5.1. Databases and protocol

There are two kinds of databases used in the experiments. The first database is FVC2004 DB1 which contains 800 fingerprint images (100 different fingers, eight images for each finger). The fingerprint images of FVC2004 DB1 [13] were acquired through the optical sensor ‘‘CrossMatch V300’’. The size of the image is 640×480 pixels with a resolution of 500 DPI. In this database, the distortion between some fingerprints from the same finger is obvious. Fig. 1 has exemplified this condition. The second database is the FINGERPASS cross-matching database established by our laboratory, which can be downloaded from <http://www.fingerpass.net>. In this paper, we selected three different sub-databases, which were captured from URU4000B optical press sensor, UPEK TCRU2C capacitive press sensor and Authentec AES2501 sweep sensor, respectively, to conduct our evaluation. From now on, these sub-databases will be, respectively, referred to as URU, UPEK and AES. Table 1 summarizes the characteristics of each sensor and Fig. 6 shows some samples from this database. In each sub-database, there are 720×12 impressions captured from 720 fingers (12 impressions per finger).

It is difficult to evaluate the accuracy of minutiae correspondences derived from the proposed algorithm and other algorithms. The protocol proposed in the Fingerprint Verification Competition (FVC) is adopted to evaluate the overall performance of the proposed algorithm. In regular matching, each sample is matched against the remaining samples of the same finger for genuine test and the first sample of each finger is matched against

Table 1

Characteristics of the FINGERPASS cross-matching database.

Capture sensor	Capture mode	Image size	DPI
URU 4000B	Optical press	500*550	700
UPEK TCRU2C	Capacitive press	208*288	508
Authentec AES2501	Sweep	Unfixed	500

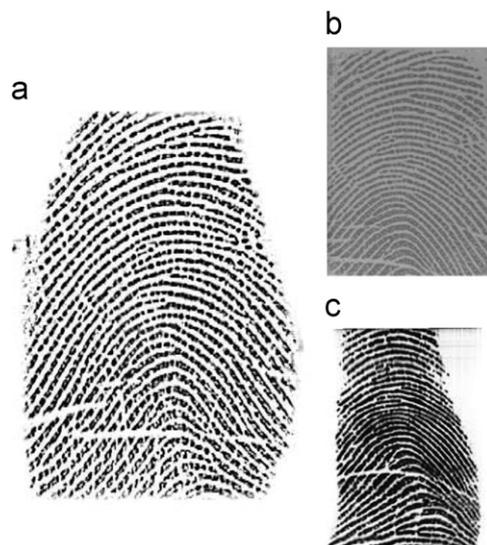


Fig. 6. Fingerprint samples of the same user of the FINGERPASS cross-matching database. (a) URU4000B; (b) UPEK TCRU2C; and (c) AES2501.

the first sample of the remaining fingers for imposter test. Hence, there are 2800 ($100 \times C_8^2$) genuine tests and 4950 (C_{100}^2) imposter tests for FVC2004 DB1, 47,520 ($720 \times C_{12}^2$) genuine tests and 258,840 (C_{720}^2) imposter tests for the FINGERPASS database. In cross-matching, the comparisons are conducted on two different sub-databases (i.e. database A and database B). Each sample in A is matched against all the samples of the same finger in B to compute the genuine test and the first sample of each finger in A is matched against the first sample of the remaining fingers in B to compute the imposter test. Therefore, there are 103,680 ($720 \times 12 \times 12$) genuine tests and 258,840 (C_{720}^2) imposter tests.

5.2. Overall performance of ACO

To validate the performance of our proposed algorithm (ACO), we have implemented two related algorithms (Algorithm Rigid and Algorithm CBFS) for comparison. Algorithm Rigid is based on the greedy matching approach proposed by Feng [12]. The similarity degree set s is first normalized and sorted in decreasing order. The top K minutiae pairs are used as the reference pair

candidates. For each of them, the following steps are adopted to calculate the matching score: (1) Two fingerprints are aligned using the relative translation and rotation between the reference pair; (2) the greedy matching algorithm proposed by Feng [12] is used to establish the correspondences between two minutiae sets; and (3) the score of this attempt is then calculated. The maximal score of these attempts is selected as the matching score. Algorithm CBFS is based on a dual graph traversal algorithm proposed by Chikkerur [7] for establishing minutiae correspondences. In their method, a local structure called K -plet was used to represent the fingerprint. The K -plet consisted of a central minutia and K other minutiae chosen from its local neighborhood. The matching algorithm was based on matching a local neighborhood and propagating the math to the K -plet of all of the minutiae in this neighborhood successively. Algorithm CBFS is modified from the publicly available code kindly provided by Chikkerur [7] by utilizing the same local minutiae neighbors with Algorithm ACO for local minutiae matching rather than K -plet. To make the comparisons meaningful, all these three algorithms use the same minutiae extraction, the same minutiae similarity and the same matching score calculation method.

Table 2 Results of different algorithms over FVC2004 DB1.

Algorithm	EER (%)	FMR100 (%)	FMR1000 (%)	ZeroFMR (%)
Rigid	3.20	5.36	11.39	18.50
CBFS	3.66	6.25	10.79	17.21
ACO	2.79	4.68	8.54	17.12
P101	2.72	3.86	9.25	13.43
P097	3.38	5.54	9.75	12.93

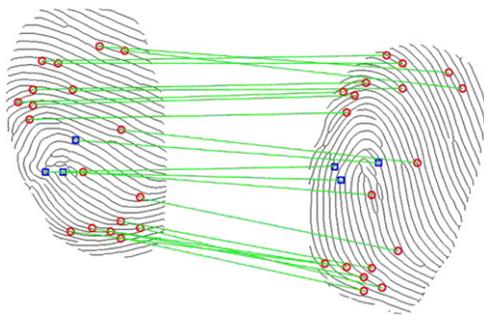


Fig. 7. Minutiae correspondences obtained by ACO. Red circles denote genuine minutiae correspondences while blue squares denote false minutiae correspondences. (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

5.2.1. Performance comparison on FVC2004 DB1

All three algorithms were conducted on FVC2004 DB1. The receiver operating characteristic (ROC) curves are plotted in Fig. 9 and their EER, FMR100, FMR1000 and ZeroFMR are reported in Table 2. From the results, we can easily obtain that Algorithm ACO performs the best, whereas Algorithm CBFS is the worst. Algorithm CBFS utilizes a dynamic programming approach based on the string alignment algorithm to match all

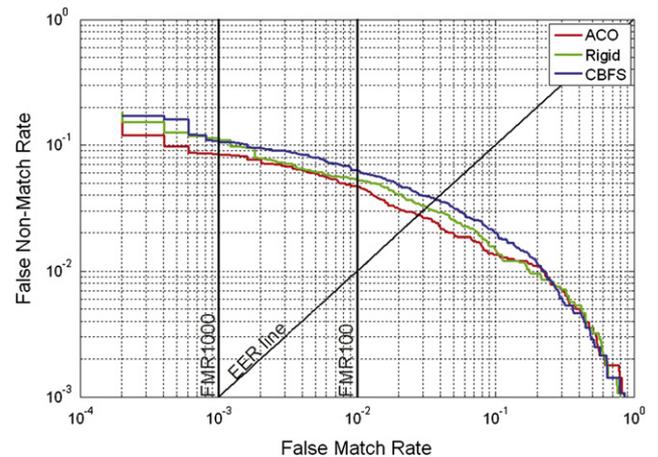


Fig. 9. ROC curve of different algorithms on FVC2004 DB1.

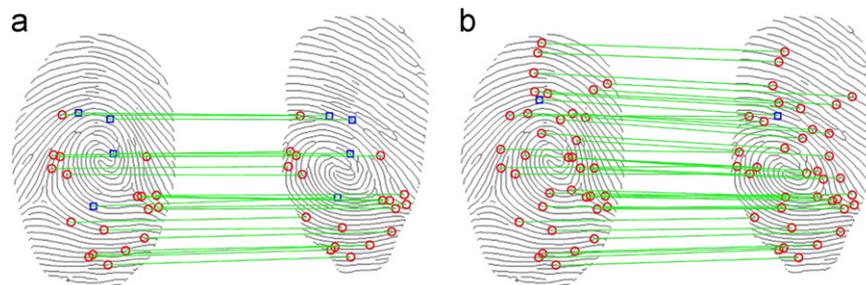


Fig. 8. Comparison of minutiae correspondences obtained by rigid registration and ACO. (a) Minutiae correspondences obtained by Rigid and (b) minutiae correspondences obtained by ACO. Red circles denote genuine minutiae correspondences while blue squares denote false minutiae correspondences (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

neighbors simultaneously. Its objective is to minimize the error of position and direction of the matched minutiae pairs. Therefore, when large distortion exists, minutiae may be wrongly paired. Since the local features are very powerful in discriminating minutiae and Algorithm CBFS does not take this into account during the matching process, the EER, FMR100, and FMR1000 of Algorithm CBFS are even higher than in Algorithm Rigid. Algorithm Rigid make use of minutiae similarity to

establish minutiae correspondences, however, rigid transformation (rotation and translation) is difficult to register large-distorted fingerprints (as shown in Fig. 1). Since no explicit alignment is required during the entire matching process and the local matching process is propagated to its neighbors by a pseudo-random proportional rule, ACO is able to find the best minutiae correspondences to maximize the objective function (16). Fig. 7 gives the minutiae correspondences of the two

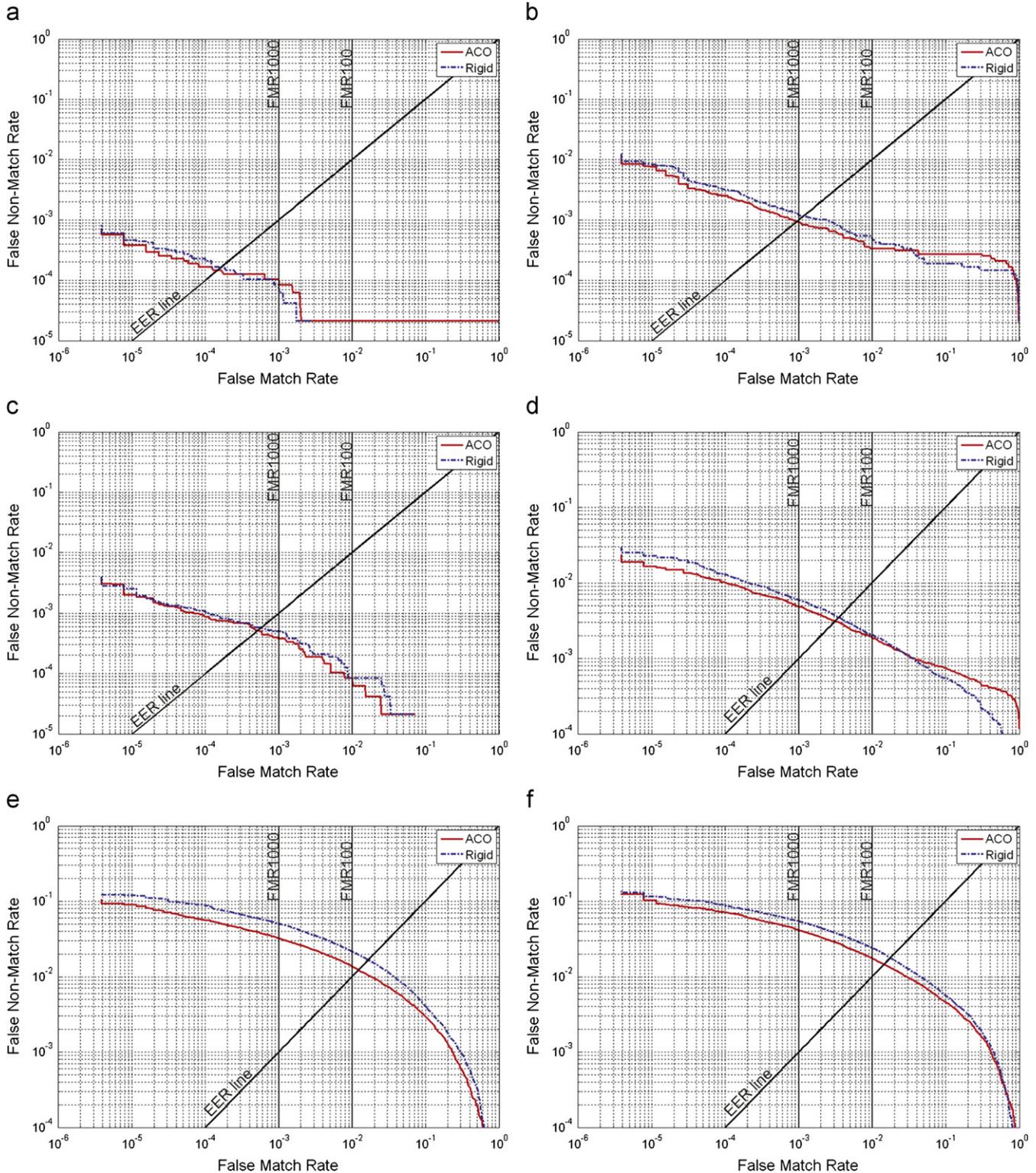


Fig. 10. ROC curves of Algorithm Rigid and Algorithm ACO over FINGERPASS database: (a) URU4000B; (b) UPEK; (c) AES2501; (d) URU4000B vs. UPEK; (e) URU4000B vs. AES2501 and (f) UPEK vs. AES2501.

fingerprints (Fig. 1) obtained by Algorithm ACO. Fig. 8 compares two matching results of Algorithm Rigid and Algorithm ACO, which confirms that Algorithm ACO is capable of matching minutiae between large-distorted fingerprints.

Table 3
The performance of regular matching based over three FINGERPASS databases.

	Method	EER (%)	FMR100 (%)	FMR1000 (%)	ZeroFMR (%)
URU	Rigid	0.017	0	0.006	0.082
	ACO	0.015	0.002	0.008	0.067
UPEK	Rigid	0.117	0.048	0.120	1.322
	ACO	0.093	0.034	0.091	1.031
AES	Rigid	0.058	0.008	0.057	0.396
	ACO	0.057	0.008	0.038	0.339

Table 4
The performance of cross-matching based over three FINGERPASS databases.

	Method	EER (%)	FMR100 (%)	FMR1000 (%)	ZeroFMR (%)
URU vs. UPEK	Rigid	0.357	0.204	0.595	2.929
	ACO	0.314	0.190	0.490	2.369
URU vs. AES	Rigid	1.666	2.162	5.050	12.231
	ACO	1.226	1.357	3.232	10.579
UPEK vs. AES	Rigid	1.783	2.408	5.451	13.584
	ACO	1.455	1.733	4.119	12.676



Fig. 11. Resized skeleton images of Fig. 2 using the common resolution method. (a) skeleton image of Fig. 2(a); (b) resized skeleton image of 2(b); and (c) rigid registration of (a) and (b).

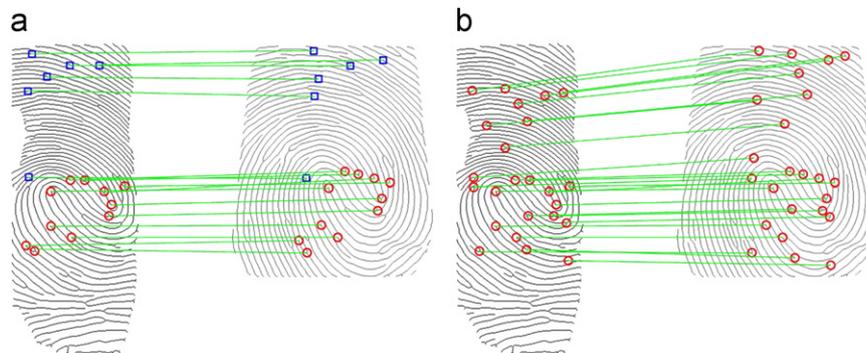


Fig. 12. Comparison of minutiae correspondences obtained by Algorithm Rigid and Algorithm ACO from cross-matching database. (a) Minutiae correspondences obtained by Algorithm Rigid and (b) minutiae correspondences obtained by Algorithm ACO. Red circles denote genuine minutiae correspondences while blue squares denote false minutiae correspondences. (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

Finally, we also compare our results with that of the algorithms called P101 and P097 in FVC2004 on DB1 in Table 2. P101 and P097 obtained second and third place, respectively, when ranked by EER. The detailed performances of FVC2004 algorithms can be seen from the web site [13]. According to the ranking rule in terms of EER in FVC2004, our algorithm rank in third place.

5.2.2. Performance comparison over FINGERPASS database

Three sets of regular matching (URU, UPEK and AES) and three sets of cross-matching experiments (URU vs. UPEK, URU vs. AES and UPE vs. AES) are conducted over the FINGERPASS database to evaluate the proposed algorithm. In each set of experiments, we compare Algorithm ACO and Algorithm Rigid. Fig. 10 compares ROC curves for three sets of regular matching and three sets of cross-matching. Table 3 and Table 4 summarize the matching performance for regular matching and cross-matching, respectively. Based on the results, the following conclusions can be drawn out: The quality of the fingerprints in the FINGERPASS database is adequate and the local features are discriminating, the EERs of Algorithm Rigid during regular matching conducted over URU, UPEK and AES are 0.017%, 0.117% and 0.058%, respectively. However, even though the performance over regular matching experimental sets is very good, the performance over cross-matching experimental sets is barely satisfactory. The EERs of Algorithm Rigid on URU vs. UPEK, URU vs. AES and UPEK vs. AES are 0.357%, 1.666% and 1.783%, respectively. The comparisons show that when fingerprints are captured from sensors of different types, the matching performance degrades, and it is worse than any one of the involved sensors. Specially, when two sensors are of different acquisition modes (such as URU and AES), the performance drastically deteriorates. The same conclusions can be obtained by analyzing the results of algorithm ACO.

Over URU vs. UPEK, URU vs. AES and UPEK vs. AES, the EERs of Algorithm ACO and Algorithm Rigid are 0.314% vs. 0.357%, 1.222% vs. 1.666% and 1.455% vs. 1.783%, respectively. Since URU and UPEK are both press mode sensors, rigid transformation is able to register fingerprints from these two sensors. Algorithm Rigid has good performance on this set, and Algorithm ACO has only 0.043% improvement. Algorithm ACO performs much better than Algorithm Rigid over the other two cross-matching experimental sets. The improvements are both larger than 0.32%.

Since fingerprint images of the sweep sensor are reconstructed from consecutive frames, distortion is introduced by finger movement during sweeping. Fig. 11 shows the resized skeleton fingerprint images of Fig. 2 based on the common resolution method. Compared with Fig. 11(b), the upper part of Fig. 11(a) is obviously compressed. When these two fingerprints are compared, rigid transformation is difficult to establish all the minutiae pairs. Fig. 12(a) and (b) shows the matching results of Algorithm Rigid and Algorithm ACO, respectively. The comparison demonstrates that

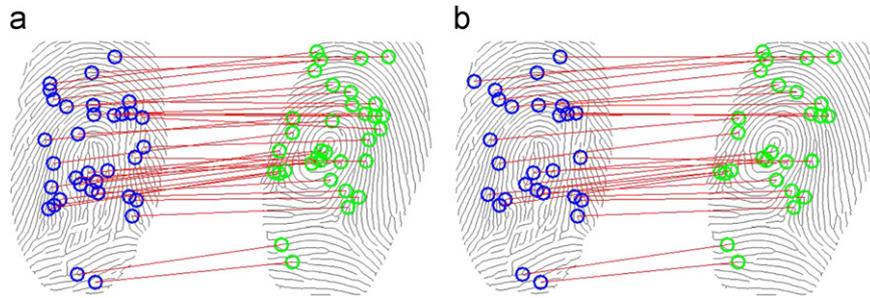


Fig. 13. Comparison of minutiae correspondences obtained by Algorithm Rigid and Algorithm ACO over fingerprints of low quality. (a) Minutiae correspondences obtained by Algorithm Rigid and (b) minutiae correspondences obtained by Algorithm ACO.

algorithm ACO is effective to find correct minutiae correspondences even when different distortion patterns are involved. When one of the compared fingerprints is of low quality and lots of spurious minutiae exist, Algorithm ACO may obtain less minutiae correspondences because of minutiae compatibility checking, which leads smaller matching score. Fig. 13 illustrates this case. That is the reason that FMR100 of Algorithm ACO is higher than Algorithm Rigid over URU (Table 3).

6. Conclusion and future work

Matching of fingerprint images with a large distortion or fingerprint images originating from two different sensors greatly challenges the traditional fingerprint matching algorithm. By representing the minutiae set as a point pattern, we propose a novel ant colony optimization algorithm, which is based on local descriptors and neighbor propagation, to find the minutiae correspondences. Experiments are conducted on FVC2004 DB1 and FINGERPASS database which are constructed by our lab. Results demonstrate that Algorithm ACO can effectively find the correct minutiae correspondences.

All experiments are conducted on the same PC with Intel Pentium 4 processor 3.4 GHz under Windows XP professional operating system. On FVC2004 DB1, URU, UPEK, AES, URU vs. UPEK, URU vs. AES and UPEK vs. AES, the average matching times of algorithm ACO and algorithm Rigid are 25.6 vs. 17.7 ms, 71 vs. 34.8 ms, 21.9 vs. 8 ms, 39.0 vs. 17.3 ms, 40.4 vs. 17.2 ms, 59.3 vs. 24.5 ms and 32.1 vs. 12.1 ms, respectively. The matching time of Algorithm ACO is nearly 2.3 times that of Algorithm Rigid. Most of the matching time of Algorithm Rigid is spent on minutiae similarity computation. In the future, we plan to improve the algorithm along the following two directions. The first direction is to speed up the algorithm by utilizing the global feature and ridge pattern. Since the proposed algorithm is able to find correct minutiae correspondences in large-distorted fingerprint images, the second direction is trying to use a non-linear distortion model, such as thin-plate spline or quadratic model, to account for the distortion between fingerprint images.

Acknowledgments

This paper is supported by the National Natural Science Foundation of China under Grant nos. 60902083, 60803151, 60875018, Beijing Natural Science Fund under Grant no. 4091004, and the Fundamental Research Funds for the Central Universities.

References

[1] A. Bazen, S. Gerez, Systematic methods for the computation of the directional fields and singular points of fingerprints, *IEEE Transactions on Pattern Analysis and Machine Intelligence* 24 (7) (2002) 905–919.

[2] A. Bazen, S. Gerez, Fingerprint matching by thin-plate spline modelling of elastic deformations, *Pattern Recognition* 36 (8) (2003) 1859–1867.

[3] K. Cao, X. Yang, X. Tao, P. Li, Y. Zang, J. Tian, Combining features for distorted fingerprint matching, *Journal of Network and Computer Applications* 33 (3) (2010) 258–267.

[4] X. Chen, J. Tian, X. Yang, A new algorithm for distorted fingerprints matching based on normalized fuzzy similarity measure, *IEEE Transactions on Image Processing* 15 (3) (2006) 767–776.

[5] X. Chen, J. Tian, X. Yang, Y. Zhang, An algorithm for distorted fingerprint matching based on local triangle feature set, *IEEE Transactions on Information Forensics and Security* 1 (2) (2006) 169–177.

[6] X.J. Chen, J. Tian, J.G. Cheng, X. Yang, Segmentation of fingerprint images using linear classifier, *EURASIP* 4 (2004) 480–494.

[7] S. Chikkerur, A. Cartwright, V. Govindaraju, K-plet and coupled BFS: a graph based fingerprint representation and matching algorithm, in: *Proceedings of 18th International Conference on Pattern Recognition 2006*, pp. 309–315.

[8] A. Colomi, M. Dorigo, V. Maniezzo, Distributed optimization by ant colonies, in: *European Conference on Artificial Life 1991*, pp. 134–142.

[9] C. Dorai, N.K. Ratha, R.M. Bolle, Detecting dynamic behavior in compressed fingerprint videos: distortion, *IEEE Computer Society Conference on Computer Vision and Pattern Recognition* 2 (2000) 2320.

[10] M. Dorigo, L. Gambardella, Ant colony system: a cooperative learning approach to the traveling salesman problem, *IEEE Transactions on Evolutionary Computation* 1 (1) (1997) 53–66.

[11] M. Dorigo, V. Maniezzo, A. Colomi, Ant system: optimization by a colony of cooperating agents, *IEEE Transactions on Systems, Man, and Cybernetics, Part B: Cybernetics* 26 (1) (1996) 29–41.

[12] J. Feng, Combining minutiae descriptors for fingerprint matching, *Pattern Recognition* 41 (1) (2008) 342–352.

[13] FVC2004, <<http://bias.csr.unibo.it/fvc2004/>>, 2004.

[14] Y. He, J. Tian, L. Li, H. Chen, X. Yang, Fingerprint matching based on global comprehensive similarity, *IEEE Transactions on Pattern Analysis and Machine Intelligence* 28 (6) (2006) 850–862.

[15] Y. He, J. Tian, X. Luo, T. Zhang, Image enhancement and minutiae matching in fingerprint verification, *Pattern Recognition Letter* 24 (9–10) (2003) 1349–1360.

[16] L. Hong, Y. Wan, A. Jain, Fingerprint image enhancement: algorithm and performance evaluation, *IEEE Transactions on Pattern Analysis and Machine Intelligence* 20 (8) (1998) 777–789.

[17] A. Jain, L. Hong, R. Bolle, On-line fingerprint verification, *IEEE Transactions on Pattern Analysis and Machine Intelligence* 19 (1997) 302–314.

[18] J. Jang, S.J. Elliott, H. Kim, On improving interoperability of fingerprint recognition using resolution compensation based on sensor evaluation, in: *Second International Conference on Advances in Biometrics, ICB2007*, pp. 455–463.

[19] T.-Y. Jea, V. Govindaraju, A minutiae-based partial fingerprint recognition system, *Pattern Recognition* 38 (2005) 1672–1684.

[20] X. Jiang, W.-Y. Yau, Fingerprint minutiae matching based on the local and global structures, *Proceedings of the 15th International Conference on Pattern Recognition*, vol. 2, 2000, pp. 1038–1041.

[21] Y. Jie, Y.Y. Fang, Z. Renjie, S. Qifa, Fingerprint minutiae matching algorithm for real time system, *Pattern Recognition* 39 (1) (2006) 143–146.

[22] Z. Kovács-Vajna, A fingerprint verification system based on triangular matching and dynamic time warping, *IEEE Transactions on Pattern Analysis and Machine Intelligence* 22 (11) (2000) 1266–1276.

[23] X. Luo, J. Tian, Knowledge based fingerprint image enhancement, *Proceedings of the 15th International Conference on Pattern Recognition*, vol. 4, 2000, pp. 783–786.

[24] X. Luo, J. Tian, Y. Wu, A minutia matching algorithm in fingerprint verification, *Proceedings of the 15th International Conference on Pattern Recognition*, vol. 4, 2000, pp. 833–836.

[25] D. Maltoni, D. Maio, A. Jain, S. Prabhakar, *Handbook of Fingerprint Recognition*, second ed., Springer, 2009.

[26] N. Ratha, R. Bolle, Effect of controlled image acquisition on fingerprint matching, *Proceedings of the Fourteenth International Conference on Pattern Recognition*, vol. 2, 1998, pp. 1659–1661.

- [27] N. Ratha, V. Pandit, R. Bolle, V. Vaish, Robust fingerprint authentication using local structural similarity, in: *The Fifth IEEE Workshop on Applications of Computer Vision* 2000, pp. 29–34.
- [28] A. Ross, S. Dass, A. Jain, A deformable model for fingerprint matching, *Pattern Recognition* 38 (1) (2005) 95–103.
- [29] A. Ross, A.K. Jain, Biometric sensor interoperability: a case study in fingerprints, in: *ECCV Workshop BioAW2004*, pp. 134–145.
- [30] A. Ross, R. Nadgir, A thin-plate spline calibration model for fingerprint sensor interoperability, *IEEE Transactions on Knowledge and Data Engineering* 20 (8) (2008) 1097–1110.
- [31] W. Sheng, G. Howells, M. Fairhurst, F. Deravi, A memetic fingerprint matching algorithm, *IEEE Transactions on Information Forensics and Security* 2 (3) (2007) 402–412.
- [32] X. Tan, B. Bhanu, Fingerprint matching by genetic algorithms, *Pattern Recognition* 39 (3) (2006) 465–477.
- [33] M. Tico, P. Kuosmanen, Fingerprint matching using an orientation-based minutia descriptor, *IEEE Transactions on Pattern Analysis and Machine Intelligence* 25 (8) (2003) 1009–1014.
- [34] X. Tong, J. Huang, X. Tang, D. Shi, Fingerprint minutiae matching using the adjacent feature vector, *Pattern Recognition Letter* 26 (9) (2005) 1337–1345.
- [35] X. Wang, J. Li, Y. Niu, Fingerprint matching using orientation codes and polylines, *Pattern Recognition* 40 (11) (2007) 3164–3177.
- [36] E. Zhu, J. Yin, G. Zhang, Fingerprint matching based on global alignment of multiple reference minutiae, *Pattern Recognition* 38 (10) (2005) 1685–1694.

Kai Cao received the Ph.D. degree from Key Laboratory of Complex Systems and Intelligence Science, Institute of Automation, Chinese Academy of Sciences, Beijing, China in 2010. He is currently a lecturer in School of Life Sciences and Technology, Xidian University, Xi'an, China. His research interests include pattern recognition, image processing, machine learning, and their applications in biometrics.

Xin Yang received the B.S., M.S., and Ph.D. degrees in intelligent instruments from Tianjin University, Tianjin, China, in 1994, 1997, and 2000, respectively. From 2001 to 2003, she was a Postdoctoral Fellow with the Biometric Research Group, Key Laboratory of Complex Systems and Intelligence Science, Institute of Automation, Chinese Academy of Sciences, Beijing, China. Since 2003, she has been an Associate Professor with the laboratory. Her research interests include bioinformatics, pattern recognition, etc.

Xinjian Chen received the Ph.D. degree (with honors) in 2006 from the Center for Biometrics and Security Research, Key Laboratory of Complex Systems and Intelligence Science, Institute of Automation, Chinese Academy of Sciences, Beijing, China. After graduation, he entered Microsoft Research Asia and researched on Handwriting Recognition. From January 2008 to October 2009, he was a Postdoctoral Fellow in the Medical Image Processing Group, Department of Radiology, University of Pennsylvania, Philadelphia. From October 2009, he has been a Postdoctoral Fellow in the Radiology and the Image Sciences Department, Clinical Center, National Institutes of Health, Bethesda, Maryland. His research interests include medical image processing, pattern recognition, machine learning, and their applications.

Yali Zang received the B.S. degree in biomedic engineering from Xi'an Jiaotong University, in 2008, and is currently pursuing the Ph.D. degree from the Institute of Automation, Chinese Academy of Sciences, Beijing, China. Her research interests include pattern recognition, machine learning, image processing, and their applications in biometrics.

Jimin Liang received the B.S., M.S., and Ph.D. degrees from Xidian University, Xi'an, China, in 1992, 1995, 2000, all majored in electronic engineering. In 1995, he joined Xidian University, where he is currently a Professor in the Life Science Research Center (LSRC), School of Life Sciences and Technology. During 2002, he was a Research Associate Professor in the Department of Electrical and Computer Engineering, University of Tennessee, Knoxville. His current research interests include image processing, image analysis, biometric recognition and biometric encryption. He became a member of IEEE since 2008.

Jie Tian received the Ph.D. degree (with honors) in artificial intelligence from the Institute of Automation, Chinese Academy of Sciences, Beijing, China, in 1992. From 1994 to 1996, he was a Postdoctoral Fellow with the Medical Image Processing Group, University of Pennsylvania, Philadelphia. Since 1997, he has been a Professor at the Institute of Automation, Chinese Academy of Sciences. Since 2007, he has also been the Chair Professor of the Cheung Kong Scholars at Xidian University, Xi'an, China. His current research interests include pattern recognition, machine learning, image processing and their applications in biometrics, etc.