Available online at www.sciencedirect.com
SCIENCE DIRECT
Pattern Recognition 39 (2006) 465–477
PATTERN RECOGNITION
THE JOURNAL OF THE PATTERN RECOGNITION SOCIETY
www.elsevier.com/locate/patcog

Fingerprint matching by genetic algorithms

Xuejun Tan*, Bir Bhanu

Center for Research in Intelligent System, University of California, Riverside, CA 92521, USA

Received 24 February 2004; accepted 6 September 2005

Abstract

Fingerprint matching is still a challenging problem for reliable person authentication because of the complex distortions involved in two impressions of the same finger. In this paper, we propose a fingerprint-matching approach based on genetic algorithms (GA), which tries to find the optimal transformation between two different fingerprints. In order to deal with low-quality fingerprint images, which introduce significant occlusion and clutter of minutiae features, we design a fitness function based on the local properties of each triplet of minutiae. The experimental results on National Institute of Standards and Technology fingerprint database, NIST-4, not only show that the proposed approach can achieve good performance even when a large portion of fingerprints in the database are of poor quality, but also show that the proposed approach is better than another approach, which is based on mean-squared error estimation.

Keywords: Fitness value; Corresponding triangles; Minutiae; Optimization; Fingerprint verification

1. Introduction

Fingerprint matching is one of the most promising methods among biometric recognition techniques and has been used for person authentication for a long time. Now, it is not only used by police for law enforcement, but also in commercial applications, such as access control and financial transactions. In terms of applications, there are two kinds of fingerprint recognition systems: verification and identification. In verification, the input is a query fingerprint and an identity (ID), the system verifies whether the ID is consistent with the fingerprint. The output is an answer of yes or no. In identification, the input is only a query fingerprint, the system tries to answer the question: Are there any fingerprints in the database that resemble the query fingerprint? The output is a short list of fingerprints. In this paper, we are dealing with the verification problem.

A fingerprint is formed by a group of curves. The most useful features, which include endpoints and bifurcations, are called minutiae. Fig. 1 shows examples of an endpoint and a bifurcation in a fingerprint image. In previous work (e.g. [30]), we present a learned templates-based algorithm for minutiae extraction. Templates are learned from examples by optimizing a criterion function using Lagrange's method. To detect the presence of minutiae in fingerprints, templates are applied with appropriate orientation to the binary fingerprints only at selected potential minutia locations.

Generally, the minutiae-based fingerprint verification is a kind of point-matching algorithm. However, the distortions between two sets of minutiae extracted from the different impressions of the same finger may include significant translation, rotation, scale, shear, local perturbation, occlusion and clutter, which make it difficult to find the corresponding minutiae reliably.

2. Related work and contribution

2.1. Related work

Generally, fingerprint-matching algorithms have two steps: (1) align the fingerprints and (2) find the correspondences between two fingerprints. The approach proposed

* Corresponding author.
E-mail addresses: xtan@cris.ucr.edu (X. Tan), bhanu@cris.ucr.edu (B. Bhanu).

0031-3203/$30.00
doi:10.1016/j.patcog.2005.09.005

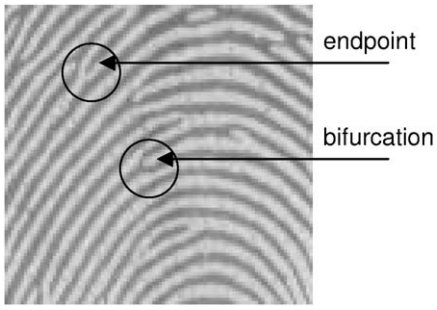

Fig. 1. Examples of minutiae.

by Jain et al. [2] is capable of compensating for some of the nonlinear deformations and finding the correspondences. However, since the ridges associated with the minutiae are used to estimate the alignment parameters, the size of the templates has to be large, which takes much memory and computation, otherwise, the alignment will be inaccurate. Jiang and Yau [3] use the local and global structures of minutiae in their approach. The local structure of a minutia describes a rotation and translation invariant feature of the minutia in its neighborhood, and the global structure tries to determine the uniqueness of a fingerprint. The problem with this technique is that it cannot compensate for real-world distortions of a 3D elastic finger. These distortions can be considered equivalent to a space variant scale distortion. Furthermore, the weight vector that is associated with each component of the feature vector, such as distances, directions, relative local orientations, etc., has to be empirically determined. Another prominent matching algorithm, which is proposed by Kovacs-Vajna [4], uses triangular matching to deal with the deformations of fingerprints. However, the final results of matching have to be validated by a dynamic time warping (DTW) algorithm. Without DTW for further verification, the results are not acceptable. In previous work (e.g. [5]), we have developed a fingerprint identification approach, which is based on the local optimization of the corresponding triangles to perform verification between two fingerprints.

Besides minutiae, researchers have also used other features for fingerprint matching. Saleh and Adhami [6] proposed an approach which transforms fingerprint images into a sequence of points in the angle-curvature domain. The matching between a query fingerprint and a template fingerprint is based on the least-squares error of the Euclidean distance between corresponding points in the angle-curve domain. Jain et al. [7] presented a filter-based algorithm, which uses a bank of Gabor filters to capture both local and global details in a fingerprint as a compact fixed length FingerCode. The authors reported that the FingerCode-based system performs better than a state-of-the-art minutiae-based system when the performance requirement of the application system does not demand a very low false acceptance rate.

The combinations of different kinds of features have also been used in fingerprint matching. Jain et al. [8] presented a hybrid-matching algorithm that uses both minutiae and texture information. Ceguerra and Koprinska [9] proposed an approach that uses matched minutiae as the reference axis to generate a shape signature for each fingerprint. The shape signature is then used to form a feature vector describing the fingerprint. A linear vector quantizer (LVQ) neural network is trained using the feature vectors to match fingerprints. Both approaches reported improvements in the matching results.

2.2. Contribution

In this paper, we use genetic algorithms (GA) to achieve a globally optimized solution for the transformation between two sets of minutiae extracted from two different fingerprints. The fitness function is based on the local properties of each triplet of minutiae, which include angles, triangle handedness, triangle direction, maximum side, minutiae density and ridges counts. The performance of our approach on the NIST-4 database, which has a large portion of fingerprints of poor quality, shows that our approach can tolerate highly nonlinear deformations. The comparison of the proposed approach with another approach based on mean-squared error estimation shows the advantage of GA-based verification.

3. Technical approach

In Ref. [10] we have provided an analysis of features (see Section 3.3.2), specifically the angles that are used as features. By using these features we are able to handle complex distortions encountered in fingerprint images. Thus, we can use a simple transformation consisting of scale, rotation and translation for matching between a template and a query fingerprint in this paper.

3.1. Fingerprint-matching problem

Suppose the sets of minutiae in the template and the query fingerprints are $\{(x_{n,1}, x_{n,2})\}$ and $\{(y_{m,1}, y_{m,2})\}$, respectively, where $n = 1, 2, 3, \ldots, N$, $m = 1, 2, 3, \ldots, M$, $(x_{n,1}, x_{n,2})$ and $(y_{m,1}, y_{m,2})$ are the coordinates of minutiae. The number of minutiae in the template and the query fingerprints are N and M, respectively. The transformation $Y_i = F(X_i)$ between $X_i(x_{i,1}, x_{i,2})$ and $Y_i(y_{i,1}, y_{i,2})$ can be simplified as

$$Y_i = s \cdot R \cdot X_i + T \tag{1}$$

where s is the scale factor,

$$R = \begin{bmatrix} \cos\theta & -\sin\theta \\ \sin\theta & \cos\theta \end{bmatrix},$$

θ is the angle of rotation between two fingerprints, and

$$T = \begin{bmatrix} t_x \\ t_y \end{bmatrix}$$

is the vector of translation. Thus, the matching problem can be defined as to find the optimized transformation, which can map as many as possible minutiae in the template fingerprint to the minutiae in the query fingerprint.

3.2. Selection of an optimization technique

We have reviewed many techniques commonly used for optimization to determine their usefulness for fingerprint recognition (e.g. [1]). The drawbacks of each of these methodologies are as follows:

- Exhaustive techniques (random walk, depth first, breadth first, enumerative): Able to find global maximum but computationally prohibitive because of the size of the search space and real-time considerations;
- Calculus-based techniques (gradient methods, solving systems of equations): No closed form mathematical representation of the objective function is available. Discontinuities and multimodal complexities are present in the objective function;
- Partial knowledge techniques (hill climbing, beam search, best first, branch and bound, dynamic programming, A*): Hill climbing is plagued by the foothill, plateau, and ridge problems. Beam, best first, and A* search techniques have no available measure of goal distance. Branch and bound requires too many search points while dynamic programming suffers from the curse of dimensionality and is computationally expensive;
- Knowledge-based techniques (production rule systems, heuristic methods): These systems have a limited domain of rule applicability, tend to be brittle, and are usually difficult to formulate. Further, the visual knowledge required by these systems may not be representable in knowledge-based formats;
- Hierarchical techniques: Generally, a coarse resolution is employed to find a narrow range of the solution, then using a fine resolution in the narrow range search the optimal solution. However, like hill-climbing techniques, hierarchical techniques cannot always find the optimal solution.

Genetic algorithms search from a population of individuals, which make them ideal candidates for parallel implementation and more efficient than exhaustive techniques. Since they use simple rules to generate new individuals, they do not require domain-specific knowledge or measures of a goal distance. The evolution of GA depends on the fitness function, which can be designed to avoid derivatives. Compared with other optimization algorithms, we believe GA is a better solution to this problem. And our experimental results showed that the performance of GA is better than a similar approach which uses mean-squared error as the optimization criteria.

3.3. Optimization based on GA

GA, introduced by Holland [11], provide an approach to learning that is loosely based on simulated evolution. The search for an appropriate hypothesis begins with a population of initial hypotheses. Members of the current population generate the new generation by means of selection, crossover and mutation, which are patterned after processes in biological evolution. At each step, hypotheses in the current population are evaluated by a fitness function, with the better fit hypotheses selected probabilistically for generating the next population. A detailed introduction to GA can be found in Ref. [12].

GA has been widely used for optimization. Johnson [13] considered the design of a switched beam linear array in which two beams with specified shapes are to be produced. GA-based optimizations are used for the design task. It is discovered that much better results are obtained by simultaneous, multi-objective optimization-based design using GA. Lam et al. [14] proposed the stability analysis of fuzzy model-based nonlinear control systems, and the design of nonlinear gains and feedback gains of the nonlinear controller using GA. The solution of the stability conditions are also determined by GA. An application example of stabilizing a cart-pole-typed inverted pendulum system was given to show the stability of the nonlinear controller. Other researches, which have used GA for optimization, can be found in Refs. [15–18].

GA has also been used in applications of object recognition. Bebis et al. [19] used GA to recognize 2D or 3D objects from 2D intensity images. The approach is model-based, while the recognition strategy lies on the theory of algebraic functions of views. Tsang [20] presented a GA-based technique for searching the best alignment between contours of near-planar objects. The method is more efficient and robust than the dominant point approaches. Similar idea has been used in Toet and Hajema [21]. Ozcan and Mohan [22] applied GA to the partial-shaped matching problem. The quality of matching is evaluated by a measure derived from attributed-shaped grammars. Kawaguchi and Nagao [23] and Zaki et al. [24] also used similar idea in their recognition systems. GA has also been used in handwriting-recognition systems, [25,26], and stereo matching of images [27].

Fig. 2 shows the block diagram of our approach. First, a feature extraction procedure, which is based on learned template introduced in Ref. [1], is applied to both the template and query fingerprints. The extracted features are the input to GA module, which is used to find the optimized transformation parameters according to the maximum fitness value. If the maximum fitness value is greater than the threshold, then the input fingerprints are from the same finger. Otherwise,

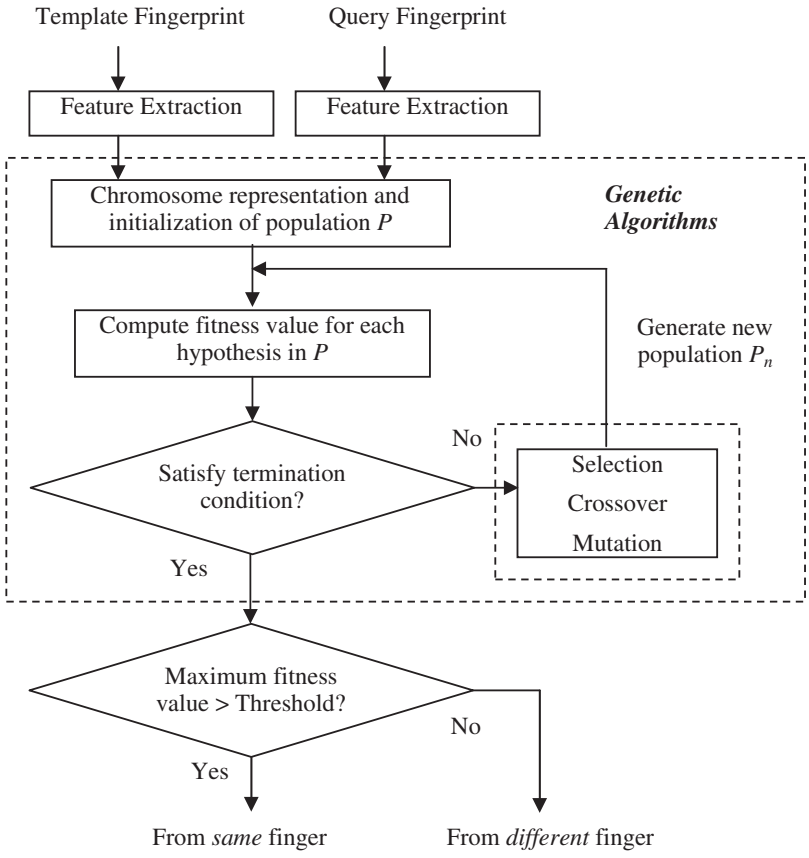

Fig. 2. Block diagram of GA-based approach for fingerprint matching.

they are not. In the following, we introduce our GA-based approach in detail.

3.3.1. Chromosome representation and initialization

As shown in Section 3.1, the parameters that need to be optimized are s, θ, t_x and t_y. According to our experimental results, which are explained in detail in Section 4.2, the ranges of these parameters (for NIST-4 database) are: $0.9 \leqslant s \leqslant 1.1$, $-30° \leqslant \theta \leqslant 30°$, $-128 \leqslant t_x \leqslant 128$, $-128 \leqslant t_y \leqslant 128$. The resolutions of these parameters are 0.01, 1°, 1 pixel and 1 pixel, respectively. Thus, the number of bits to represent s, θ, t_x and t_y are 5, 6, 8 and 8, respectively. The length of chromosome representation is 27 bits. The size of the entire search space is about $2^{27} \approx 1.34 \times 10^8$. The bit string is initialized randomly.

3.3.2. Fitness function

Fitness function is critical to the performance of GA. In our approach, fitness function is defined by a two-step process. During the first step, the optimized transformation is used to check the global consistency between two sets of minutiae. In the second step, local properties of the minutiae are used to verify the detailed matching.

(1) Step 1. Hypothesizing transformation: Suppose the optimized transformation is $\hat{F}_{\hat{e}}(\bullet)$, where $\hat{e}=(\hat{s}, \hat{\theta}, \hat{t}_1, \hat{t}_2)$ $\forall j$, $j = 1, 2, 3, \ldots, N$. Let

$$d_j = \min_k \left\{ \left| \hat{F}\left(\begin{bmatrix} x_{j,1} \\ x_{j,2} \end{bmatrix} \right) - \begin{bmatrix} y_{k,1} \\ y_{k,2} \end{bmatrix} \right| \right\}. \quad (2)$$

If d_j is less than a threshold T_d, then we define the points

$$\begin{bmatrix} x_{j,1} \\ x_{j,2} \end{bmatrix} \quad \text{and} \quad \begin{bmatrix} y_{j,1} \\ y_{j,2} \end{bmatrix}$$

are potential corresponding points. Fig. 3 shows the illustration of $\hat{F}_{\hat{e}}(\bullet)$. If n_c, the number of potential corresponding points based on $\hat{F}_{\hat{e}}(\bullet)$, is less than a threshold T_d, then let the fitness value for the transformation $\hat{F}_{\hat{e}}(\bullet)$ be $FV(\hat{F}_{\hat{e}}) = n_c$.

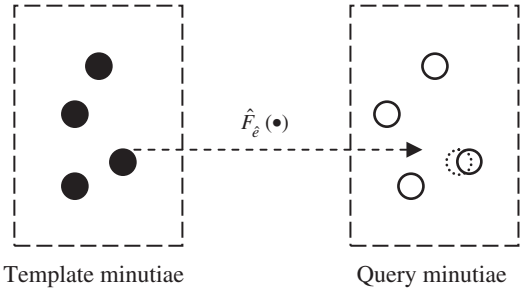

Fig. 3. Illustration of $\hat{F}_{\hat{e}}(\bullet)$.

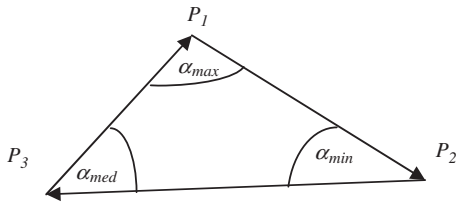

Fig. 4. Definition of feature points' labels.

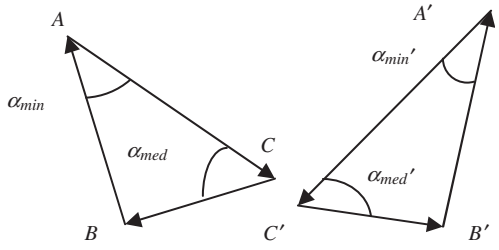

Fig. 5. $\Delta_{A'B'C'}$ and Δ_{ABC} have the same internal angles but different triangle handedness.

In this case, it does not make sense to further evaluate the matching. Otherwise, we check the local properties of the triplets of minutiae.

(2) Step 2. Detailed matching: Each noncolinear triplet of potential corresponding points can form a triangle. Germain et al. [28] used the triplets of minutiae in their identification procedure. The features they used as the indexing components are: the length of each side, the angles that the ridges make with respect to the *X*-axis of the reference frame, and the ridge count between each pair of vertices. In previous work for fingerprint indexing [10], the features we use associated with the triangles are: triangle's angles, handedness, type, direction, and maximum side. In this approach, the local properties associated with the triangle are described as the following. Note that, in order to reduce the effect of nonlinear deformation, not all the noncolinear triplets are used, e.g. the minimum length of the sides and the minimum angle in a triangle should be greater than certain thresholds.

- Angles $\alpha_{\min}$ and α_{med}: Suppose α_i are three angles in the triangle, $i = 1, 2, 3$. Let $\alpha_{\max} = \max\{\alpha_i\}$, $\alpha_{\min} = \min\{\alpha_i\}$, $\alpha_{med} = 180^\circ - \alpha_{\max} - \alpha_{\min}$, then the label of the triplets in this triangle is such that if the minutia is the vertex of angle $\alpha_{\max}$, we label this point as P_1; if the minutia is the vertex of angle $\alpha_{\min}$, we label it as P_2; the last minutia is labeled as P_3. Fig. 4 shows an example of this definition.
- Triangle handedness ϕ: Let $Z_i = x_i + jy_i$ be the complex number ($j = \sqrt{-1}$) corresponding to the coordinates (x_i, y_i) of point P_i, $i = 1, 2, 3$. Define $Z_{21} = Z_2 - Z_1$, $Z_{32} = Z_3 - Z_2$, and $Z_{13} = Z_1 - Z_3$. Let $\phi = sign(Z_{21} \times Z_{32})$, where *sign* is the signum function and $\times$ is the cross product of two complex numbers. Fig. 5 shows two triangles that have the same $\alpha_{\min}$ and α_{med} but different ϕ.

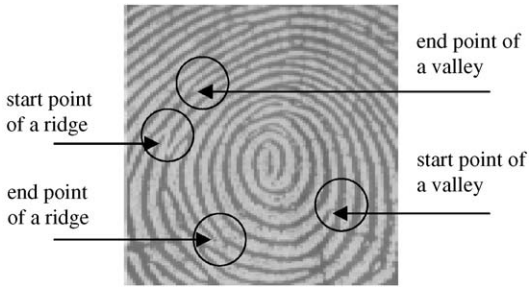

Fig. 6. Examples of minutiae with different v value.

- Triangle direction η: Search the minutia from top to bottom and left to right in the fingerprint, if the minutia is the start point of a ridge or valley, then $v = 1$, else $v = 0$. Let $\eta = 4v_1 + 2v_2 + v_3$, where v_i is the v value of point P_i, $i = 1, 2, 3$. Fig. 6 shows examples of minutiae with different v value.

Note that when two minutiae are on a horizontal ridge (which will seldom happen), this feature may not be robust as the distinction between start and end points can disappear with slight fingerprint rotations. This may happen whenever the ridge direction is approximately horizontal. In that case, the triangles that use this as an edge may effect ultimately a few pair of corresponding triangles. It does not have much effect on the final matching result since so many other minutiae features are used.

- Maximum side λ: Let $\lambda = \max\{L_i\}$, where $L_1 = |Z_{21}|$, $L_2 = |Z_{32}|$, and $L_3 = |Z_{13}|$.
- Minutiae density χ: In a local area (32×32 pixels) centered at the minutiae P_i, if there exists n_χ minutiae, then minutiae density for P_i is $\chi_i = n_\chi$. Minutiae density χ is a vector consisting of all χ_i's. Note that considering *elastic skin deformations* a circular window is a better choice for computing minutiae density. However, a rectangular window is computationally simpler and since we use relative minutiae density for matching within a threshold, there will not be much difference between using a circular or a rectangle window.
- Ridge counts ξ: Let ξ_1 , ξ_2 and ξ_3 be the ridge counts of sides P_1P_2, P_2P_3 and P_3P_1, respectively, then ξ is a vector consisting of all ξ_i's. Fig. 7 shows the examples of ridge count and minutiae density.

If two triangles from two different fingerprints satisfy the following criteria, then they are potential corresponding triangles, and the fitness value of the hypothesis $FV(\hat{F}) = n_t$, where n_t is the number of potential corresponding triangles. The criteria are:

$$|\alpha'_{\min} - \alpha''_{\min}| \leqslant T_{\alpha_{\min}},$$

$$|\alpha'_{med} - \alpha''_{med}| \leqslant T_{\alpha_{med}},$$

$$\phi' = \phi'',$$

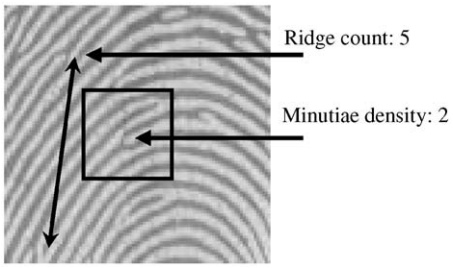

Fig. 7. Examples of ridge count and minutiae density.

$$\eta' = \eta'',$$

$$|\lambda' - \lambda''| \leqslant T_\lambda,$$

$$|\chi'_i - \chi''_i| \leqslant T_\chi, \quad i = 1, 2, 3,$$

$$|\xi'_i - \xi''_i| \leqslant T_\xi, \quad i = 1, 2, 3, \tag{3}$$

where $(\alpha'_{\min}, \alpha'_{med}, \phi', \eta', \lambda', \chi'_i, \xi'_i)$ and $(\alpha''_{\min}, \alpha''_{med}, \phi'', \eta'', \lambda'', \chi''_i, \xi''_i)$ are the local properties of the triangle in different fingerprints; $T_{\alpha_{\min}}$, $T_{\alpha_{med}}$, T_λ, T_χ, and T_ζ are thresholds to deal with the local distortions.

Thus, the fitness function is defined as

$$FV(\hat{F}) = \begin{cases} n_c, & \text{if } n_c < T_n, \\ n_t, & \text{if } n_c \geqslant T_n, \end{cases} \tag{4}$$

where n_c is the number of potential corresponding points, n_t is the number of potential corresponding triangles, and T_n is the threshold.

One of the challenges for fingerprint matching is to model the nonlinear/nonuniform distortions in finger's skin. We use a simplified model for matching. As long as using the simplified model we can find some matched features, we think we have a candidate transformation between a pair of fingerprints. We use local features to filter false matches. Thus, our approach can accommodate some nonlinear/nonuniform distortions.

3.3.3. Population generation

Suppose (a) the size of population P is N_p; (b) hypotheses of the transformation are $\hat{F}_i$, $i = 1, 2, 3 \ldots N_p$; (c) crossover rate is p_s; (d) mutation rate is p_m; (e) hypotheses are ordered in the descending order of their fitness values; (f) the fitness value of a hypothesis $\hat{F}_i$ is $FV(\hat{F}_i)$.

Then, a new generation P_n is generated by

- Selection: probabilistically select the first $p_s \times N_p$ hypotheses from P and add them to P_n.
- Crossover: probabilistically select $(1 - p_s) \times N_p/2$ pairs of hypotheses from P according to $Pr(\hat{F}_i)$. For each pair hypotheses, generate two children by applying the crossover operator and add them to P_n. The probability $Pr(\hat{F}_i)$ is defined by

$$Pr(\hat{F}_i) = \frac{FV(\hat{F}_i)}{\sum_{j=1}^{N_p} FV(\hat{F}_i)} \tag{5}$$

- Mutation: choose $p_m \times N_p$ hypotheses from P with uniform probability. For each hypothesis, invert one randomly selected bit.

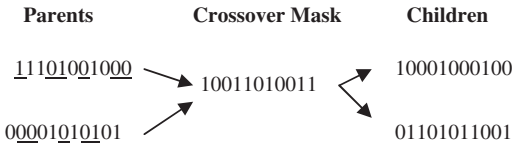

Fig. 8. An example for uniform crossover operations in GA.

Generally, crossover operators include single-point crossover, two-point crossover, and uniform crossover. In uniform crossover, the crossover mask is generated as a random bit string with each bit chosen at random and independent of the others. Fig. 8 shows an example of uniform crossover, which is used in our approach.

3.3.4. Termination conditions

Termination conditions decide the length of the learning time and possibly how good the solution is. The termination conditions we use are: (1) terminate GA if the maximum fitness value does not change in N_t generations; (2) Terminate GA if the fitness value of a matching is greater than 100, then it is a correct genuine matching and is unnecessary to continue the evolution.

3.3.5. Computation time reduction

One problem with GA is that the computation time of the evolution can be long. In order to reduce it, we use the termination conditions that are defined above. In addition, during the evolution, the first $p_s \times N_p$ hypotheses are selected and added to the next generation. Only mutation can change the fitness value of these hypotheses. If no mutation happens for a particular individual corresponding to any one of these hypotheses, then it is unnecessary to compute their fitness value again.

4. Experimental results

4.1. Database

The database we use in our experiments is the NIST Special Database 4 (NIST-4) [29], which is a publicly available fingerprint database. Since the fingerprints in NIST-4 are collected by an ink-based method, a large portion of the

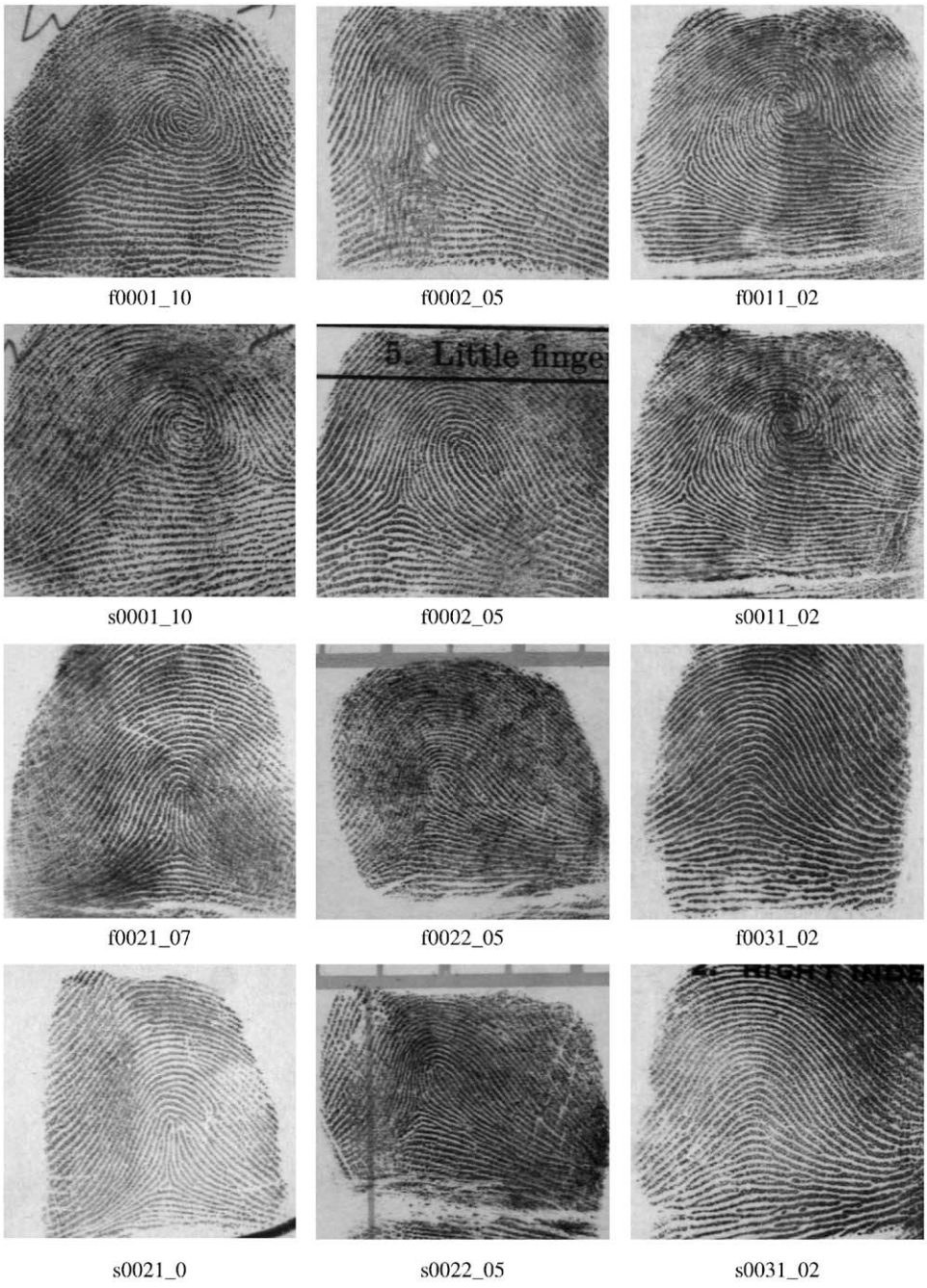

Fig. 9. Sample fingerprints in NIST-4 database.

fingerprints are of poor quality and contain certain other objects, such as characters and handwritten lines. The size of the fingerprint images is 480×512 pixels with a resolution of 500 DPI. NIST-4 contains 2000 pairs of fingerprints. Each pair is a different impression of the same finger. The fingerprint is coded as a f or s followed by 6 numbers, which means the fingerprint image is the first or second impression of certain finger. Some sample fingerprints are shown in Fig. 9. The features are extracted using the learned template approach described in Ref. [30].

4.2. *Estimation of data range for* GA

Parameters that need to be optimized are s, θ, t_x and t_y. The data range of these parameters are critical to the performance of the approach. On one side, if the data range is too small, we may miss the optimal solution. On the other side, if the data range is too large, the GA may take a much longer time to finish. The estimation of the data range is based on the experiments of the first 100 pairs of fingerprints of NIST-4.

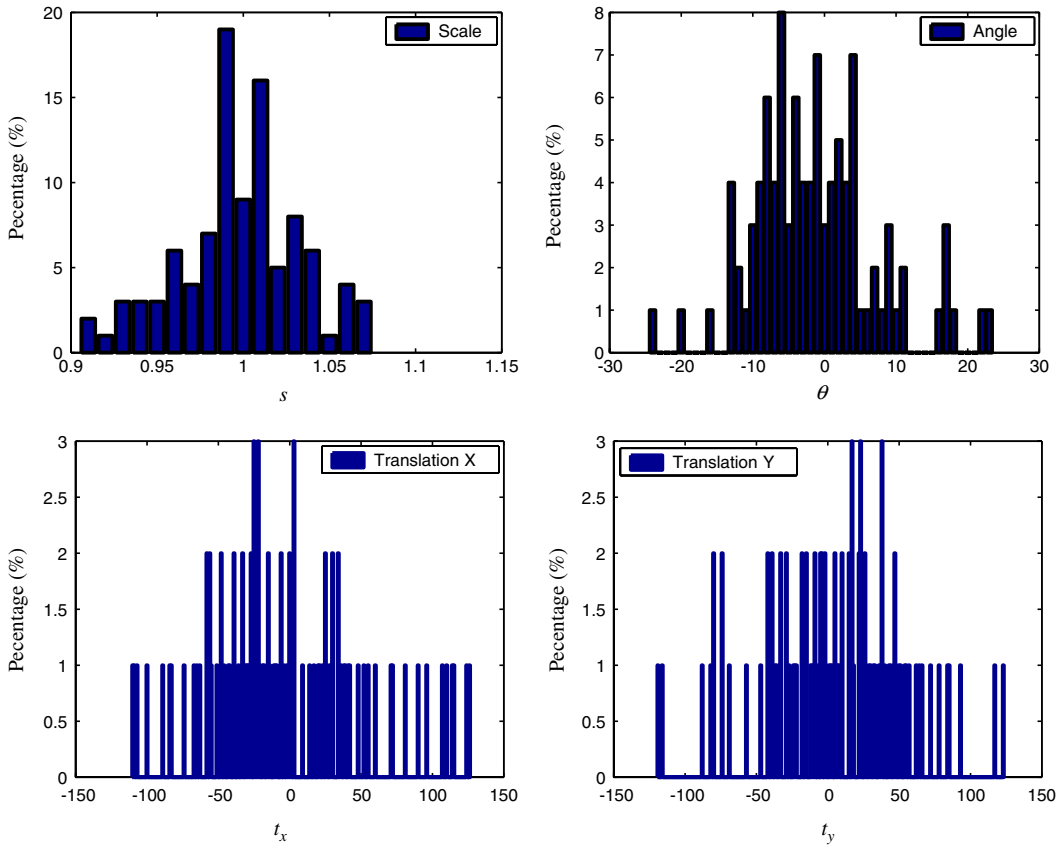

Fig. 10. Histograms of transformation parameters (first 100 pairs of fingerprints in NIST-4).

For the first fingerprint in each pair of fingerprints, we manually choose minutiae features, then find the correspondences in the second fingerprint. Using the pairs of correspondences of minutiae features in each pairs of fingerprint, we estimate the transformation parameters by mean-squared error. Fig. 10 shows the distributions of s, θ, t_x and t_y. The data range that we choose for these parameters are: $0.9 \leqslant s \leqslant 1.1$, $-30° \leqslant \theta \leqslant 30°$, $-128 \leqslant t_x \leqslant 128$, $-128 \leqslant t_y \leqslant 128$. Fig. 10 shows that the transformation parameters of the first 100 pairs of fingerprints are all within these data ranges.

4.3. Estimation of parameters for GA

Fig. 11 shows a triangle. Without loss of generality, we assume that one vertex, O, of the triangle is (0, 0), and it does not change under distortions. The analysis of angle α shows that (1) the minimum and the median angles $\alpha_{\min}$ and α_{med} in a triangle are more robust than the maximum angle $\alpha_{\max}$ under distortions; (2) 2° can accommodate the uncertainty of most distortions and keep the size of the search space as small as possible. Thus, we have $T_{\alpha_{\min}} = 2°$ and $T_{\alpha_{med}} = 2°$. We present a detailed analysis of threshold selection in Refs. [31] and [10].

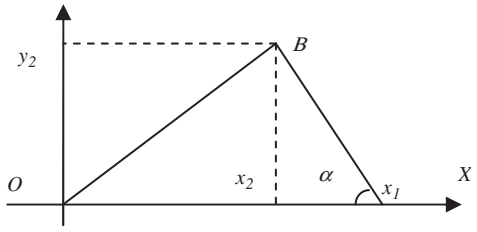

Fig. 11. Illustration of variables.

Table 1
Parameters

Parameters	Value	Parameters	Value	Parameters	Value
T_d	12	T_n	5	P_s	0.2
$T_{\alpha_{\min}}$	2°	$T_{\alpha_{med}}$	2°	P_m	0.005
T_λ	20	T_χ	2	N_t	15
T_ζ	2	N_p	100		

The parameters used in our approach are listed in Table 1. In the following, we provide comments on the parameters.

(1) s, θ, t_x and t_y—The ranges of these parameters are described in Section 4.2 and confirmed by Fig. 10, histograms of transformation parameters;

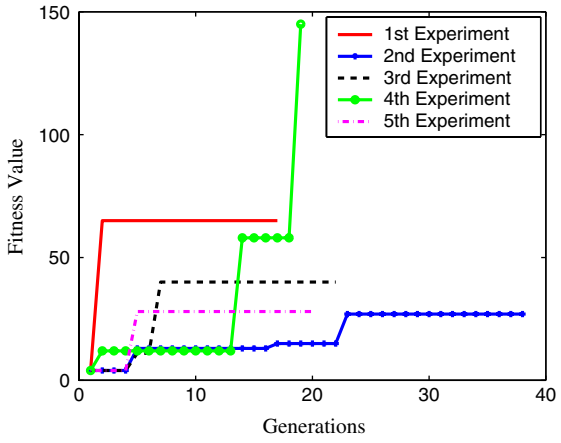

Fig. 12. Fitness value changes for matching between the first pair of fingerprints in Fig. 9 (f0001_10 and s0001_10).

(2) T_d and T_n—These are matching (see below Eq. (2)) and fitness (Eq. (4)) related thresholds. A higher value of T_d means a larger distortion in matching is allowed which will lead to increased computation. The value of T_n is estimated based on the distribution of matched triangles for correct and incorrect recognition [10];
(3) p_s, p_m, N_p—These are the standard crossover rate, the mutation rate, and the size of population parameters used in genetic algorithms [1,32]. They are based on the size of the search space, rate of convergence and diversity of the individuals in the population;
(4) N_t—It is the number of generations. We choose it from our observation of the evolution of GA. Fig. 12 shows an example of GA evolution process. If the fitness value does not change after N_t generations, then we stop the evolution;
(5) All the other thresholds listed in Table 1 are for choosing corresponding triangles [10]. We choose them based on the feature extraction results. The larger the thresholds, the more corresponding triangles will be found, which will require longer time by GA.

Note that for all the results reported in this paper all the parameters are fixed for the entire NIST-4 dataset.

4.4. Results

A total of 2000 matchings, between consistent pairs, are performed to estimate the distribution of genuine matching. We also perform 200,000 matchings between inconsistent pairs to estimate the distribution of imposter matching, where for each matching we randomly select two fingerprints from NIST-4 that are the impressions of different fingers. Fig. 12 shows five evolutions of the fitness value for the matching between the first pair of fingerprints (f0001_10 and s0001_10) shown in Fig. 9. Since we use two methods, explained in Section 3.3.4, for the computation time reduction, if the fitness value becomes greater than 100, then the evolution stops. We observe that the evolutions terminate within 40 generations. Considering the size of the search space is 2^{27}, the GA searches only a fraction of the search space. We repeat experiments of genuine matchings and imposter matchings five times, respectively. Fig. 13 shows four pairs of fingerprints, which are the different impressions of the *same* fingers, and their corresponding maximum fitness values. Fig. 14 shows four pairs of fingerprints, which are the different impressions of the *different* fingers, and their corresponding maximum fitness values. We observe that the more similar the two fingerprints, the larger the fitness value. Since the fingerprints in Fig. 14 are not from the same fin-

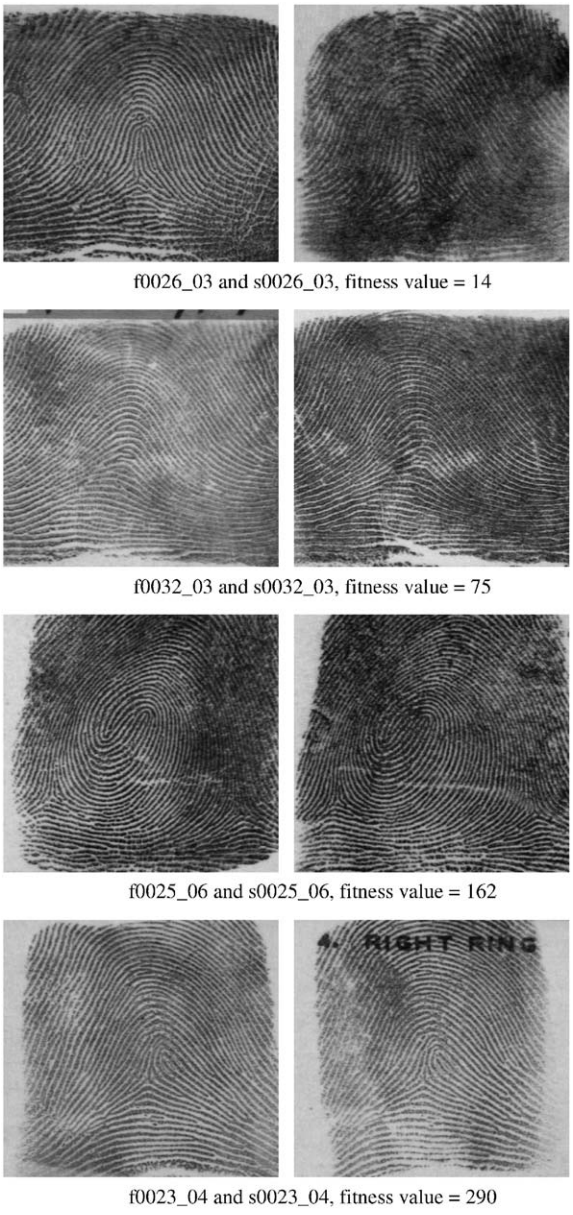

Fig. 13. Examples of genuine matching: maximum fitness value between two fingerprints which are from the *same* finger.

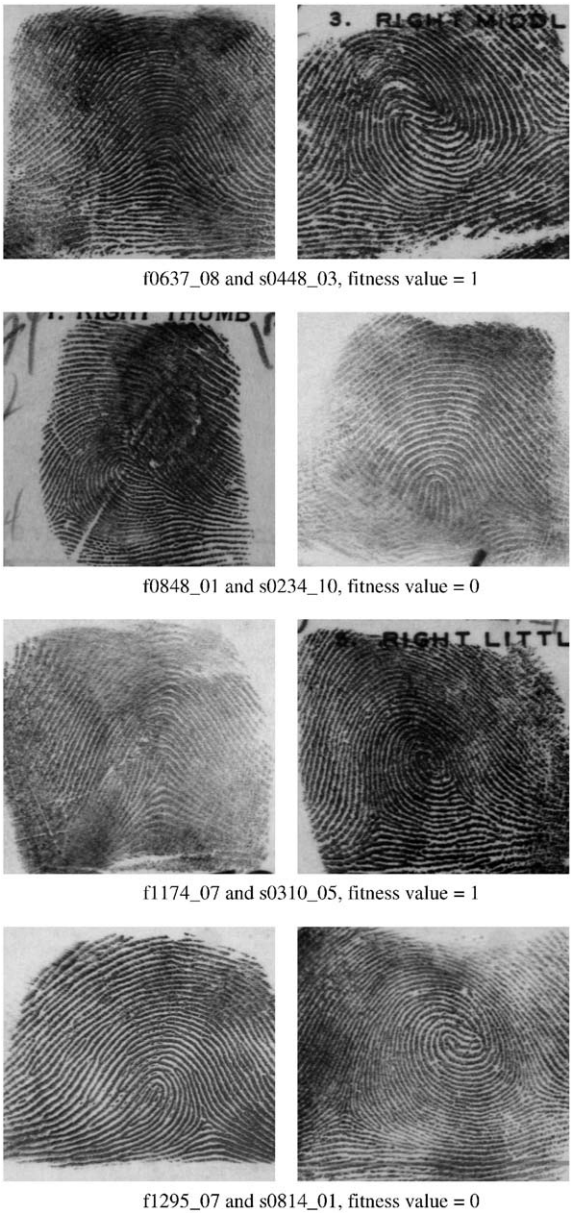

Fig. 14. Examples of imposter matching: maximum fitness value between two fingerprints which are from *different* fingers.

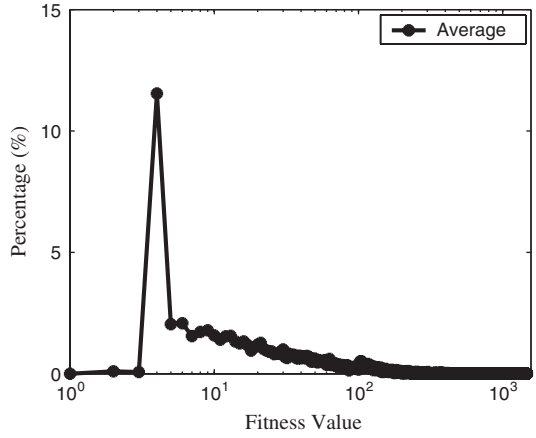

Fig. 15. PDF of fitness value for genuine matchings.

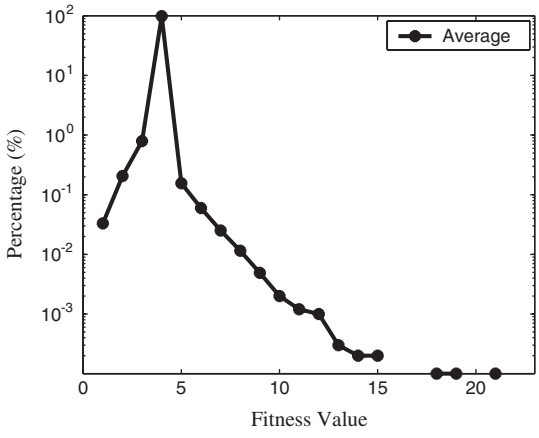

Fig. 16. PDF of fitness value for imposter matchings.

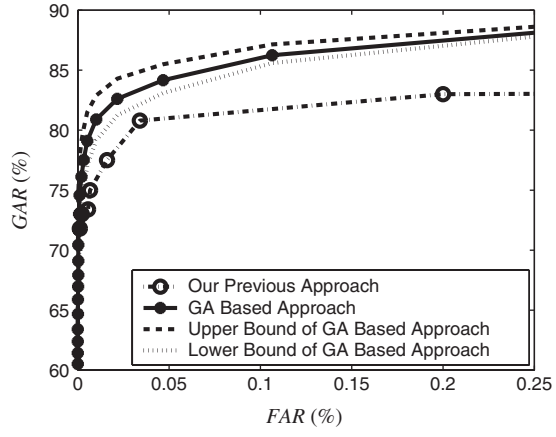

Fig. 17. Comparison of ROC curves of two approaches using entire NIST-4 (2000 pairs of fingerprints).

gers, their fitness values are small. Figs. 15 and 16 show the average probability distribution function (PDF) of 10 experiments for both distributions, respectively. Note that Figs. 15 and 16 are at different scales. On the average, 99.1% imposter matchings and 11.5% genuine matchings have a fitness value of 4.

Based on genuine and imposter distributions, the receiver operating characteristic (ROC) curve is defined as the plot of genuine acceptance rate (GAR) against false acceptance rate (FAR). Fig. 17 shows the comparison of the average ROC curve of our GA-based approach in this paper and the ROC curve reported in Ref. [5]. It also shows the lower and upper bounds of GA-based approach, which are estimated by repeating the experiments 10 times. The advantage of GA-based approach is about 3.0%. When FAR is small, i.e.

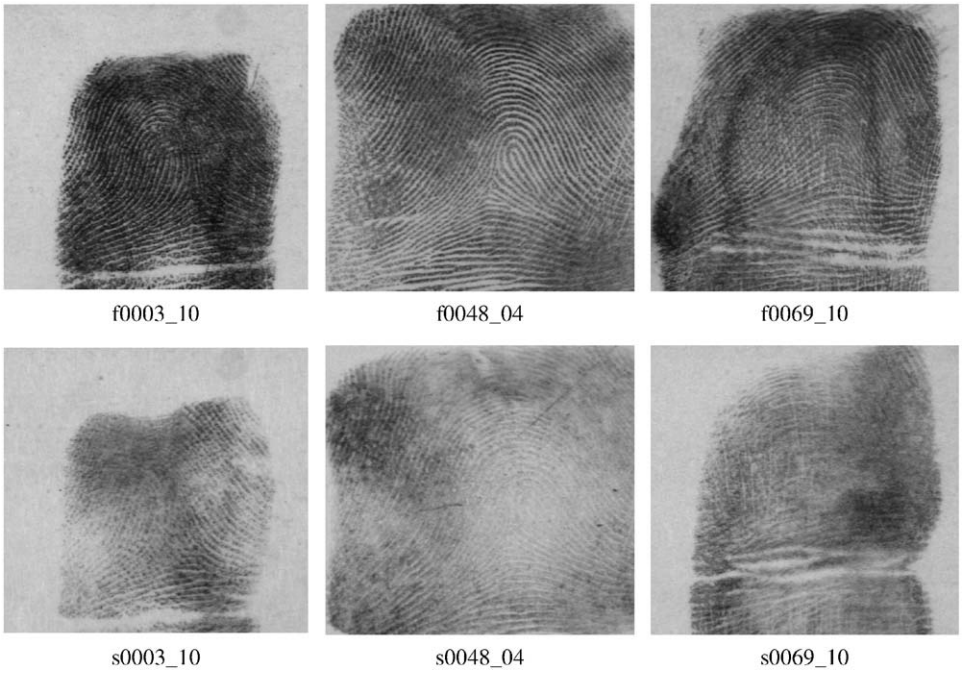

Fig. 18. Sample fingerprints with low fitness value in genuine matching.

less than 0.02%, the advantage is about 2.0%. One main reason is that this approach finds global transformation and uses of local properties to verify it, which is better than finding local transformation only. Note that we have shown the results on the entire NIST-4 database. Verification results on NIST-4 are reported in Ref. [4]. However, in Kovacs-Vajna [4] 6.0% data are rejected *manually* by the author because of bad quality. Without dynamic time warping (DTW) for the detailed verification, the FAR is 10.0%, which is unacceptable, although the GAR is 85.0%. With DTW, the GAR is 85% and 80%, and the corresponding FAR is 0.002% and 0.05%. It makes no sense for the use to compare the performance reported in Kovacs-Vajna [4] with DTW since the author has not used the entire database, and we do not know which of the fingerprints have been rejected manually.

Examining the results, we find that: (a) the low fitness values for most genuine matchings are due to the poor quality of fingerprints. There is not enough overlapped areas from which the feature extraction procedure can extract enough good minutiae; (b) the nonzero fitness value for most imposter matchings are due to the similar structures and clutter features in two different fingerprint images. Fig. 18 shows some low-quality fingerprint pairs. In each case, the maximum fitness value is 4 (obtained in different runs of experiments).

4.5. Effectiveness of selection and crossover

In order to demonstrate the effectiveness of selection and crossover operators, we compared the performance of the pure GA and that of other two variations of GA. These

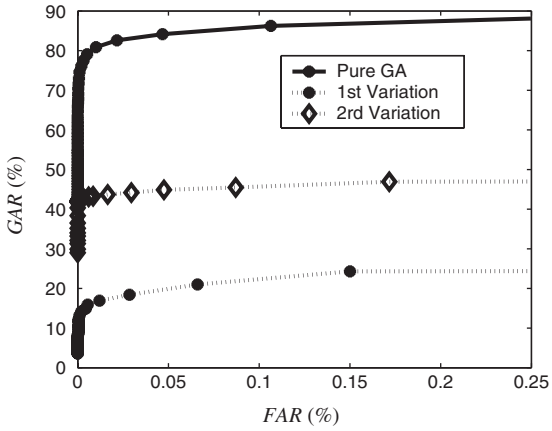

Fig. 19. Comparison of the average performance of pure GA and its two variations.

variations are:

- Instead of selecting hypotheses from parental population according to their fitness value, the first variation selects the hypotheses randomly for further evolution. The only restriction is that any hypothesis can only be selected once.
- The second variation simply skips the crossover. In order to generate the same number of children as the pure GA, the mutation rate of this variation is increased to 0.8, which is the same as the crossover rate.

We perform the same experiments, which are explained in Section 4.4, to test the performance of both variations. Fig. 19 shows the comparison of the average performance

of the pure GA and both variations. We observe that the performance of pure GA is much better than that of the other two variations. This demonstrates that the selection and crossover are critical to the success of GA.

4.6. Computation time

On a SUN Ultra II workstation, which has a 200 MHz cpu, the average computation time for a genuine matching and an imposter matching are 15 and 8 s, respectively. The difference of run-time between two kinds of matching is because the GA needs to check more detailed local properties of the fingerprints for the genuine matchings, while it does not need do this for imposter matchings. This run-time cannot satisfy the real-time requirement of the real-world applications. However, the computation time can be reduced by:

- Using a more powerful computer;
- Parallel computation on an appropriate hardware.

5. Conclusions

In this paper, we proposed a fingerprint-matching approach, which is based on GA to find the globally optimized transformation. The local properties of each triplets of minutiae are used to find potential corresponding triangles and tolerate reasonable distortions, including translation, rotation, scale, shear, local perturbation, occlusion and clutter. We achieve promising experimental results on the NIST-4 database, which has a large portion of poor-quality fingerprints. The comparison shows the advantage of the proposed approach.

References

[1] B. Bhanu, S. Lee, Genetic Learning for Adaptive Image Segmentation, Kluwer Academic Publishing, Dordrecht, 1994.
[2] A.K. Jain, L. Hong, S. Pankanti, R. Bolle, An identity-authentication system using fingerprints, Proc. IEEE 85 (9) (1997) 1364–1388.
[3] X. Jiang, W.Y. Yau, Fingerprint minutiae matching based on the local and global structures, Proceedings of the International Conference on Pattern Recognition, 2000, pp. 1038–1041.
[4] Z.M. Kovacs-Vajna, A fingerprint verification system based on triangular matching and dynamic time warping, IEEE Trans. Pattern Anal. Mach. Intel. 22 (11) (2000) 1266–1276.
[5] X. Tan, B. Bhanu, Robust fingerprint identification, Proceedings of the IEEE International Conference on Image Processing, September 2002.
[6] A.A. Saleh, R.R. Adhami, Curvature-based matching approach for automatic fingerprint identification, Proceedings of the Southeastern Symposium on System Theory, 2001, pp. 171–175.
[7] A.K. Jain, S. Prabhakar, L. Hong, S. Pankanti, Filterbank-based fingerprint matching, IEEE Trans. Image Process. 9 (5) (2000) 846–859.
[8] A.K. Jain, A. Ross, S. Prabhakar, Fingerprint matching using minutiae and texture features, Proceedings of the International Conference on Image Processing, vol. 3, 2001, pp. 282–285.
[9] A.V. Ceguerra, I. Koprinska, Integrating local and global features in automatic fingerprint verification, Proceedings of the International Conference on Pattern Recognition, vol. 3, 2002, pp. 347–350.
[10] B. Bhanu, X. Tan, Fingerprint indexing based on novel features of minutiae triplets, IEEE Trans. Pattern Recog. Anal. Mach. Intell. 25 (5) (2003) 616–622.
[11] J.H. Holland, Outline for a logical theory of adaptive systems, J. Assoc. Comput. Mach. 3 (1962) 297–314.
[12] T. Back, U. Hammel, H.P. Schwefel, Evolutionary computation: comments on the history and current state, IEEE Trans. Evol. Comput. 1 (1) (1997) 3–17.
[13] J.M. Johnson, Genetic algorithm design of a switchable shaped beam linear array with phase-only control, Proceedings of the IEEE Aerospace Conference, vol. 3, 1999, pp. 297–303.
[14] H.K. Lam, F.H. Leung, P.K.S. Tam, Design and stability analysis of fuzzy model-based nonlinear controller for nonlinear systems using genetic algorithm, IEEE Trans. Sys. Man Cybern.-Part B: Cybern. (2002).
[15] J.M. Johnson, Y. Rahmat-Samii, Genetic algorithm optimization for aerospace electromagnetic design and analysis, Proceedings of the IEEE Aerospace Applications Conference, vol. 1, 1996, pp. 87–102.
[16] C. Mandal, P.P. Chakrabarti, S. Ghose, GABIND: a GA approach to allocation and binding for the high-level synthesis of data paths, IEEE Trans. Very Large Scale Integration (VLSI) Systems 8 (6) (2000) 747–750.
[17] A.C.M. de Oliveira, L.A.N. Lorena, A constructive genetic algorithm for gate matrix layout problems, IEEE Trans. Comput. Aided Des. Integrated Circuits Syst. 21 (8) (2002) 969–974.
[18] B. Sareni, L. Krahenbuhl, A. Nicolas, Efficient genetic algorithms for solving hard constrained optimization problems, IEEE Trans. Magn. 36 (4) (2000) 1027–1030.
[19] G. Bebis, S. Louis, Y. Varol, A. Yfantis, Genetic object recognition using combinations of views, IEEE Trans. Evol. Comput. 6 (2) (2002) 132–146.
[20] P.W.M. Tsang, A genetic algorithm for aligning object shapes, Image Vision Comput. 15 (11) (1997) 819–831.
[21] A. Toet, W.P. Hajema, Genetic contour matching, Pattern Recog. Lett. 16 (1995) 849–856.
[22] E. Ozcan, C.K. Mohan, Partial shape matching using genetic algorithms, Pattern Recog. Lett. 18 (1997) 987–992.
[23] T. Kawaguchi, M. Nagao, Recognition of occluded objects by a genetic algorithm, Trans. IEICE D-II J82 (3) (1999) 350–360.
[24] M. Zaki, A. El-Ramsisi, R. Omran, A soft computing approach for recognition of occluded shapes, J. Syst. Software 51 (2000) 73–83.
[25] S.B. Cho, Pattern recognition with neural networks combined by genetic algorithm, Fuzzy Sets Systems 103 (2) (1999) 339–347.
[26] D.S. Yeung, Y.T. Cheng, H.S. Fong, F.L. Chung, Neocognitron based handwriting recognition system performance tuning using genetic algorithm, Proceedings of the IEEE International Conference on Systems, Man, and Cybernetics, vol. 5(5), 1998, pp. 4228–4233.
[27] H. Saito, M. Mori, Application of genetic algorithms to stereo matching of images, Pattern Recog. Lett. 16 (1995) 815–821.
[28] R.S. Germain, A. Califano, S. Colville, Fingerprint matching using transformation parameters clustering, IEEE Comput. Sci. Eng. Mag. 4 (4) (1997) 42–49.
[29] C.I. Watson, C.L. Wilson, NIST special database 4, fingerprint database, U.S. National Institute of Standards and Technology, 1992.
[30] B. Bhanu, X. Tan, Learned templates for feature extraction in fingerprint images, Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, vol. 2, 2001, pp. 591–596.
[31] B. Bhanu, X. Tan, A triplet based approach for indexing of fingerprint database for identification, Proceedings of the International Conference on Audio- and Video-Based Biometric Person Authentication, 2001, pp. 205–210.
[32] D.E. Goldberg, Genetic Algorithms in Search, Optimization and Machine Learning, Addison-Wesley, Reading, MA, 1989.

About the Author—Xuejun Tan received his B.S. in Automation from Tian Jin University, Tian Jin, China in 1995 and M.S. in Pattern Recognition and Artificial Intelligence from Institute of Automation, Chinese Academy of Sciences, Beijing, China in 1998. He received his Ph.D. degree in Electrical Engineering from the University of California, Riverside in June 2003. His research interests include Biometrics, Image Processing, Pattern Recognition and Machine Learning.

About the Author—Bir Bhanu received the S.M. and E.E. degrees in electrical engineering and computer science from the Massachusetts Institute of Technology, Cambridge, the Ph.D. degree in electrical engineering from the Image Processing Institute, University of Southern California, Los Angeles, and the M.B.A. degree from the University of California, Irvine.

Dr. Bhanu has been the founding Professor of Electrical Engineering and served its first Chair at the University of California, Riverside (UCR). He has been the Cooperative Professor of Computer Science and Engineering and Director of Visualization and Intelligent Systems Laboratory (VISLab) since 1991. Currently Dr. Bhanu also serves as the founding Director of an interdisciplinary Center for Research in Intelligent Systems (CRIS) at UCR. Previously, he was a Senior Honeywell Fellow at Honeywell Inc. in Minneapolis, MN. He has been on the faculty of the Department of Computer Science at the University of Utah, Salt Lake City, UT, and has worked at Ford Aerospace and Communications Corporation, CA, INRIA-France and IBM San Jose Research Laboratory, CA. He has been the principal investigator of various programs for DARPA, NASA, NSF, AFOSR, ARO and other agencies and industries in the areas of learning and vision, image understanding, pattern recognition, target recognition, biometrics, navigation, image databases, and machine vision applications. He is the co-author of books on *Evolutionary Synthesis of Pattern Recognition Systems* (Springer, 2005), *Computational Learning for Adaptive Computer Vision* (Springer, 2005), *Computational Algorithms for Fingerprint Recognition* (Kluwer, 2004), *Genetic Learning for Adaptive Image Segmentation* (Kluwer, 1994), and *Qualitative Motion Understanding* (Kluwer, 1992), and the co-editor of a book on *Computer Vision Beyond the Visible Spectrum*, (Springer, 2004). He has received two outstanding paper awards from the Pattern Recognition Society and has received industrial and university awards for research excellence, outstanding contributions and team efforts. He has been on the editorial board of various journals and has edited special issues of several IEEE transactions (PAMI, SMC, R&A, IP) and other journals. He holds 11 U.S. and international patents and over 230 reviewed technical publications in the areas of his interest. He has been General Chair for the IEEE Conference on Computer Vision and Pattern Recognition, IEEE Workshops on Applications of Computer Vision, IEEE Workshops on Learning in Computer Vision and Pattern Recognition; Chair for the DARPA Image Understanding Workshop and Program Chair for the IEEE Workshops on Computer Vision Beyond the Visible Spectrum.

Dr. Bhanu is a Fellow of IEEE, AAAS (American Association for the Advancement of Science), IAPR (International Association of Pattern Recognition) and SPIE (The International Society for Optical Engineering).