# Fingerprint Matching by Genetic Algorithms

Xuejun Tan and Bir Bhanu
Center for Research in Intelligent System
University of California, Riverside, CA 92521
*{xtan, bhanu}@cris.ucr.edu*

## 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 finds the optimal global 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 the 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 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 ques-

tion: 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. Figure 1 shows examples of an endpoint and a bifurcation in a fingerprint image. 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.

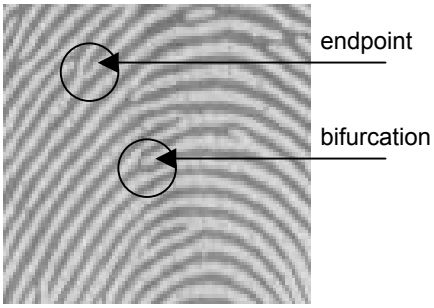



**Figure 1.** Examples of minutiae.

## 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 by Jain et al. [11] 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 [8] 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 can not 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 [15], 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 our previous work [23], 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 [20] 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 square error of the Euclidean distance between corresponding points in the angle-curve domain. Jain et al. [13] 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 kind of features have also been used in fingerprint matching. Jain et al. [12] presented a hybrid matching algorithm that uses both minutiae and texture information. Ceguerra and Koprinska [6] proposed an approach that uses matched minutiae as the reference axis to generate shape signature of the fingerprint. 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

### 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 $n = 1,2,3,\ldots,N$, $m = 1,2,3,\ldots,M$. 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:

$$Y_i = s \cdot R \cdot X_i + T \quad (1)$$

where $s$ is the scale factor, $R = \begin{bmatrix} \cos\theta & -\sin\theta \\ \sin\theta & \cos\theta \end{bmatrix}$, $\theta$ 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 [29]. 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 expensive computationally;

- ***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 is performed to achieve the optimal solution. However, like hill climbing techniques, hierarchical techniques can not always find the optimal solution.

Genetic algorithms are able to overcome the problems mentioned in the above. They 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 goal distance.

### 3.3 Optimization based on *GA*

Genetic algorithms (*GA*), introduced by Holland [7], 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 [3].

*GA* has been widely used for optimization. Johnson [9] 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. [16] 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, who have used *GA* for optimization, can be found in [10][17][18][21].

*GA* has also been used in applications of object recognition. Bebis et al. [4] 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 [25] 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 [24]. Ozcan and Mohan [19] applied *GA* to the partial shape matching problem. The quality of matching is evaluated by a measure derived from attributed shape grammars. Kawaguchi and Nagao [14] and Zaki et al. [28] also used similar idea in their recognition systems. *GA* has also been used in handwriting recognition systems, [5] and [27], and stereo matching of images [22].

Figure 2 shows the block diagram of our approach. First, a feature extraction procedure, which is based on learned template introduced in [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 corresponding 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, they are not. In the following, we introduce our *GA* based approach in detail.

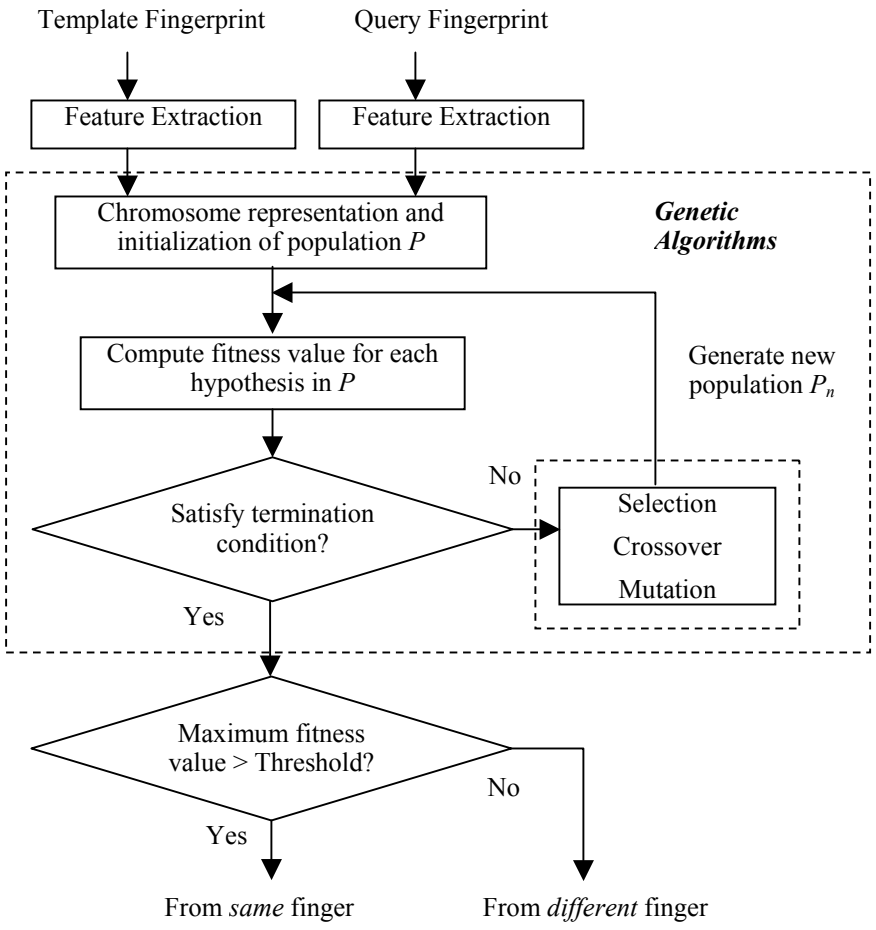



**Figure 2.** Block diagram of *GA* based approach for fingerprint matching.

### 3.3.1 Chromosome representation and initialization

As shown in Section 3.1, the parameters that need to be optimized are $s$, $\theta$, $t_x$ and $t_y$. According to our experimental results, which are explained in detail in Section 4.2, the ranges of these parame-

ters (for NIST-4 database) are: $0.9 \le s \le 1.1$, $-30^o \le \theta \le 30^o$, $-128 \le t_x \le 128$, $-128 \le t_y \le 128$. The resolutions of these parameters are *0.01*, $1^o$, *1* pixel and *1* pixel, respectively. Thus, the number of bits to represent $s$, $\theta$, $t_x$ 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**: 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,...,N,* and $N$ is the number of minutiae in the template. Let

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

If $d_j$ is less than a threshold $T_d$, then we define the points $\begin{bmatrix} x_{j,1} \\ x_{j,2} \end{bmatrix}$ and $\begin{bmatrix} y_{j,1} \\ y_{j,2} \end{bmatrix}$ are potential corresponding points. Figure 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 $T_n$, then let the fitness value for the transformation $\hat{F}_{\hat{e}}(\bullet)$ be $FV(\hat{F}_{\hat{e}}) = n_c$. In this case, it does not make sense to further evaluate the matching. Otherwise, we check the local properties of the triplets of minutiae.

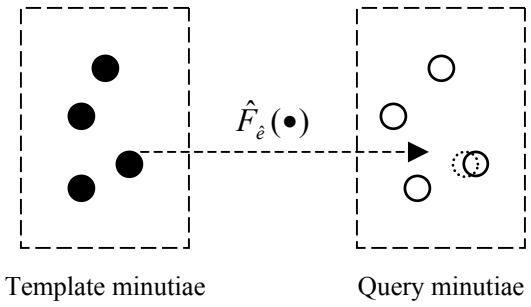



**Figure 3.** Illustration of $\hat{F}_{\hat{e}}(\bullet)$ .

**2) Step 2**: Each noncolinear triplet of potential corresponding points can form a triangle, the local properties associated with the triangle include:

- **Angles $\alpha_{min}$ and $\alpha_{med}$ :** Suppose $\alpha_i$ are three angles in the triangle, $i = 1, 2, 3$. Let $\alpha_{max} = max\{\alpha_i\}$, $\alpha_{min} = min\{\alpha_i\}$, $\alpha_{med} = 180° - \alpha_{max} - \alpha_{min}$ , 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 $P_3$. Figure 4 shows an example of this definition.

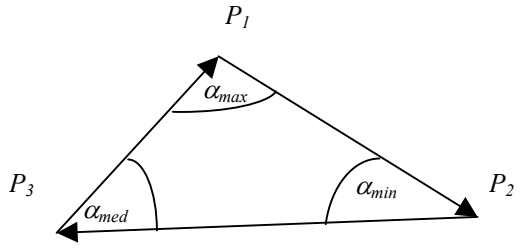



**Figure 4.** Definition of feature points' labels.

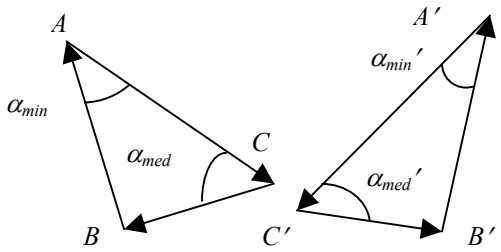



**Figure 5.** $\Delta_{A'B'C'}$ and $\Delta_{ABC}$ have the same internal angles but different triangle handedness.

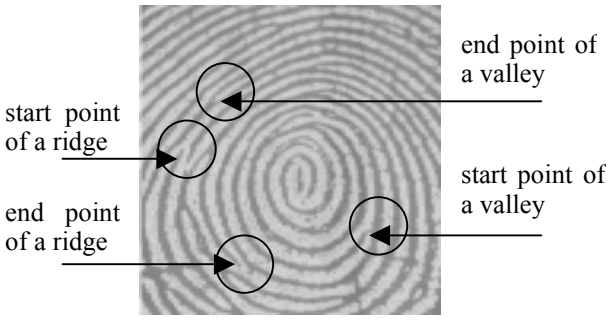



**Figure 6.** Examples of minutiae with different $\nu$ value.

- **Triangle handedness $\phi$:** 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 × is the cross product of two complex numbers. Figure 5 shows two triangles that have the same $\alpha_{min}$ and $\alpha_{med}$ but different $\phi$.

- **Triangle Direction $\eta$:** 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 $\nu = 1$, else $\nu = 0$. Let $\eta = 4\nu_1 + 2\nu_2 + \nu_3$ , where $\nu_i$ is the $\nu$ value of point $P_i$ , $i = 1,2,3$. Figure 6 shows examples of minutiae with different $\nu$ value.

- **Maximum Side $\lambda$:** Let $\lambda = max\{L_i\}$, where $L_1 = |Z_{21}|$, $L_2 = |Z_{32}|$, and $L_3 = |Z_{13}|$.

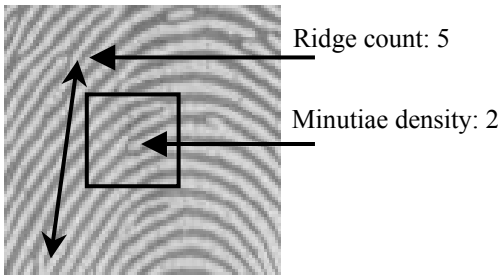



**Figure 7.** Examples of ridge count and minutiae density.

- **Minutiae Density χ:** In a local area (32×32 pixels) centered at the minutiae $P_i$, if there exists $n_\chi$ minutiae, then minutiae density for $P_i$ is $\chi_i = n_\chi$ . Minutiae density $\chi$ is a vector consisting of all $\chi_i$'s.

- **Ridge Counts ξ:** Let $\xi_1$ , $\xi_2$ and $\xi_3$ be the ridge counts of sides $P_1P_2$ , $P_2P_3$ and $P_3P_1$ , respectively, then $\xi$ is a vector consisting of all $\xi_i$'s. Figure 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:

$$\left|\alpha'_{min} - \alpha''_{min}\right| \leq T_{\alpha_{min}}$$

$$\left|\alpha'_{med} - \alpha''_{med}\right| \leq T_{\alpha_{med}}$$

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

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

$$\left|\lambda' - \lambda''\right| \leq T_\lambda$$

$$\left|\chi'_i - \chi''_i\right| \leq T_\chi \text{ , } i = 1,2,3$$

$$\left|\xi'_i - \xi''_i\right| \leq T_\xi \text{ , } i = 1,2,3 \qquad (3)$$

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

Thus, the fitness function is defined as:

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

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. Generally, the larger the $T_n$, the longer the evolution of *GA* takes.

### 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...*$N_p$; (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 $\frac{(1-p_s)\times N_p}{2}$ 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

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

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

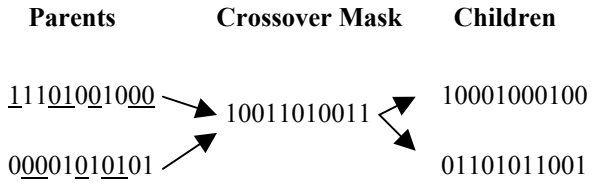



**Figure 8.** An example for uniform crossover operators in *GA*.

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. Figure 8 shows an examples 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 and parameters

The database we use in our experiments is the NIST Special Database 4 (*NIST-4*) [26], 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 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 Figure 9.

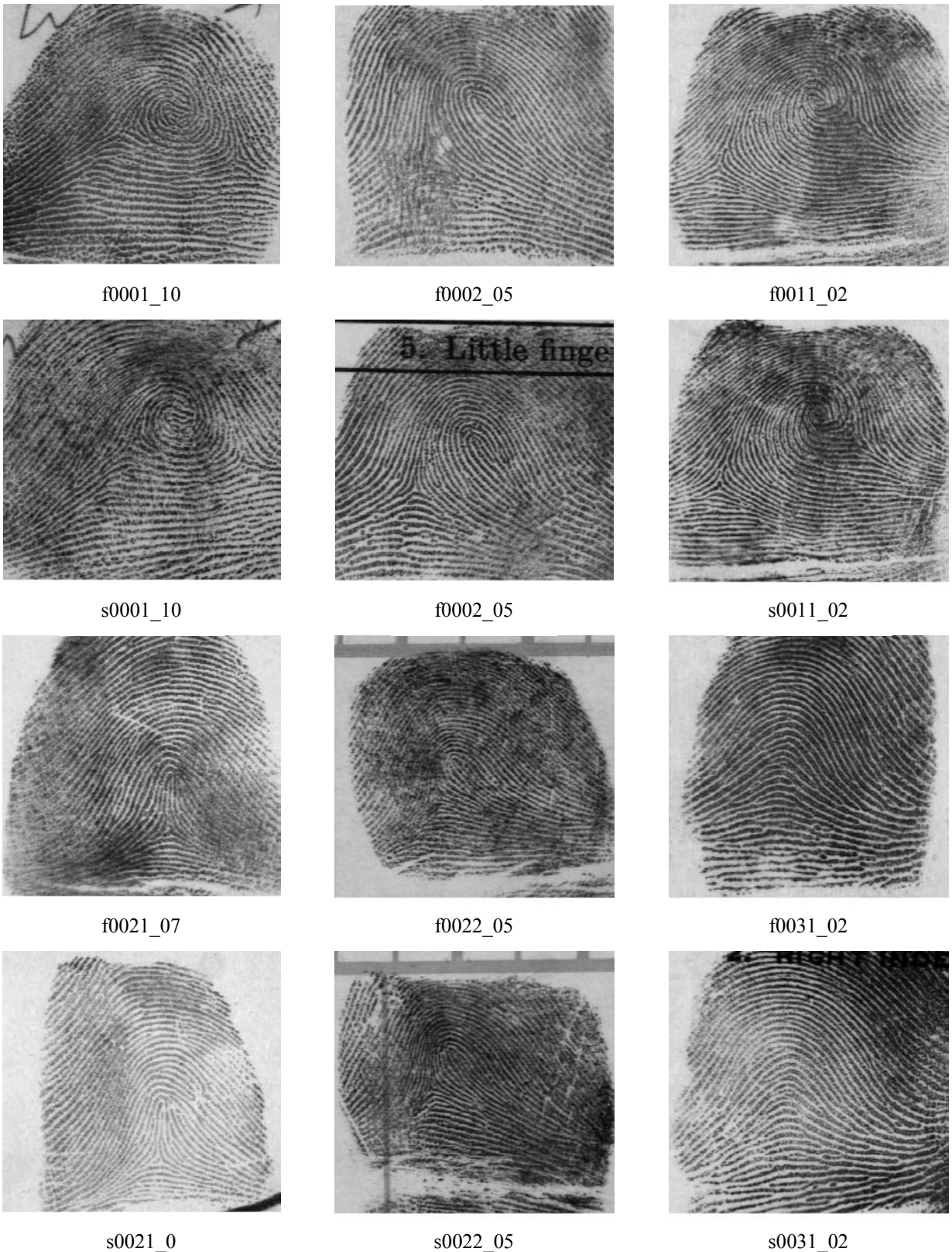



**Figure 9.** Sample fingerprints in *NIST-4* database.

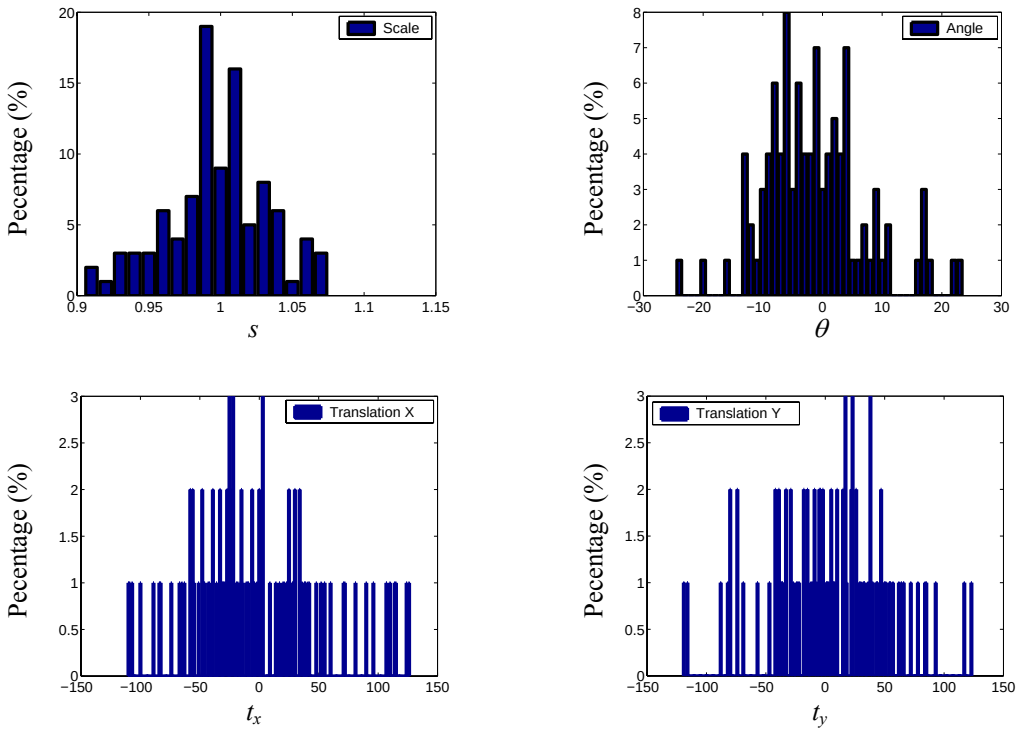



**Figure 10.** Histograms of transformation parameters (first 100 pairs of fingerprints in *NIST-4*).

### 4.2 Estimation of data range for *GA*

Parameters that need to be optimized are $s$, $\theta$, $t_x$ 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*.

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* [23]. Figure 10 shows the distributions of $s$, $\theta$, $t_x$ and $t_y$. The data range that we choose for

these parameters are: $0.9 \le s \le 1.1$, $-30^o \le \theta \le 30^o$, $-128 \le t_x \le 128$, $-128 \le t_y \le 128$. Figure 10 shows that the transformation parameters of the first 100 pairs of fingerprints are all within these data ranges.

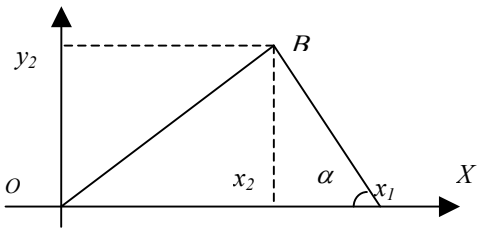



**Figure 11.** Illustration of variables.

### 4.3 Estimation of parameters for *GA*

Figure 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 $\alpha$ shows that 1) the minimum and the median angles $\alpha_{min}$ and $\alpha_{med}$ in a triangle are more robust than the maximum angle $\alpha_{max}$ under distortions; 2) 2º can accommodate the uncertainty of most distortions and keep the size of the search space as small as possible. Details of the analysis can be found in [2]. Thus, we have $T_{\alpha_{min}} = 2^o$ and $T_{\alpha_{med}} = 2^o$. Other parameters are decided based on experimental evaluations. The parameters used in our experiments are shown in Table 1.

**Table 1.** Parameters.

| Parameters | Value | Parameters | Value | Parameters | Value |
|---|---|---|---|---|---|
| $T_d$ | 12 | $T_n$ | 5 | $p_s$ | 0.2 |
| $T_{\alpha_{min}}$ | $2^o$ | $T_{\alpha_{med}}$ | $2^o$ | $p_m$ | 0.005 |
| $T_\lambda$ | 20 | $T_\chi$ | 2 | $N_t$ | 15 |
| $T_\zeta$ | 2 | $N_p$ | 100 | | |

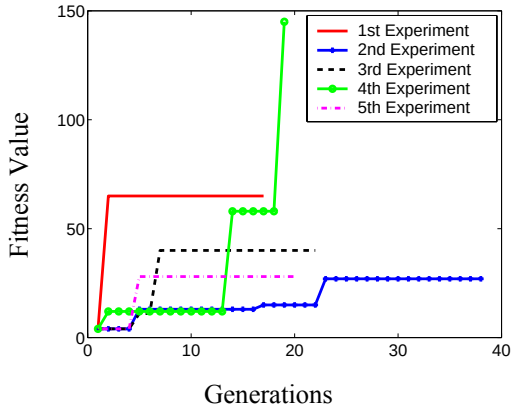
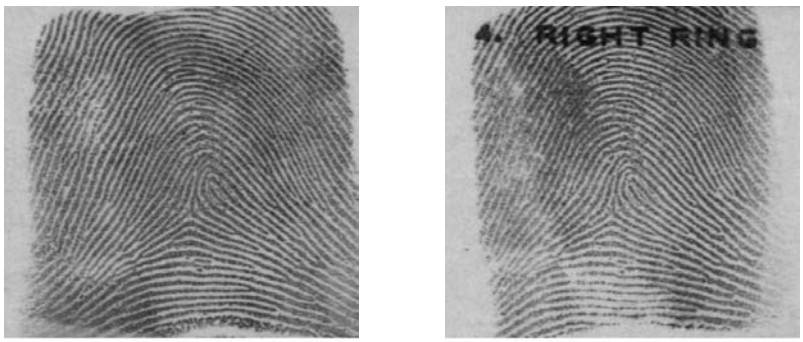



**Figure 12.** Fitness value changes for matching between the first pair of fingerprints in Figure 9 (f0001_10 and s0001_10).

f0026_03 and s0026_03, fitness value = 14

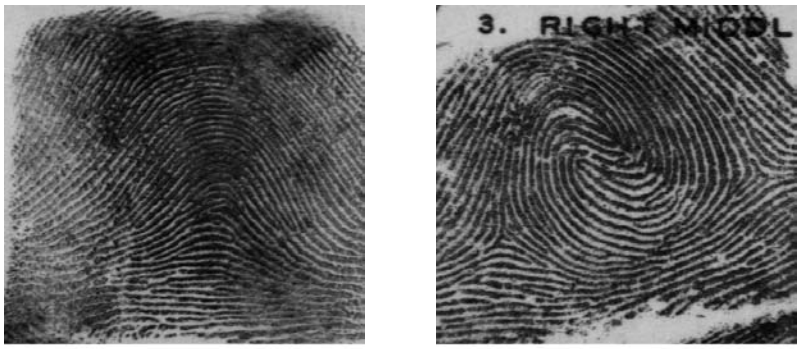
f0032_03 and s0032_03, fitness value = 75

f0025_06 and s0025_06, fitness value = 162




f0023_04 and s0023_04, fitness value = 290

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



f0637_08 and s0448_03, fitness value = 1

f0848_01 and s0234_10, fitness value = 0

f1174_07 and s0310_05, fitness value = 1

f1295_07 and s0814_01, fitness value = 0

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

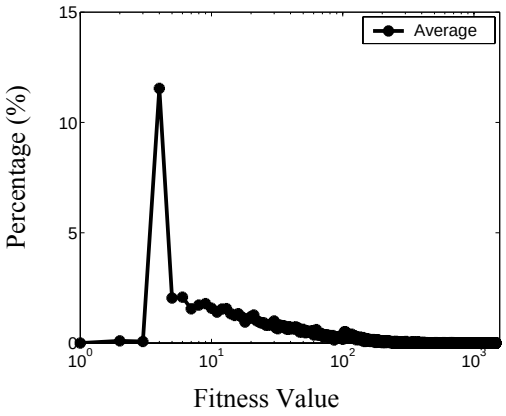



**Figure 15.** *PDF* of fitness value for genuine matchings.

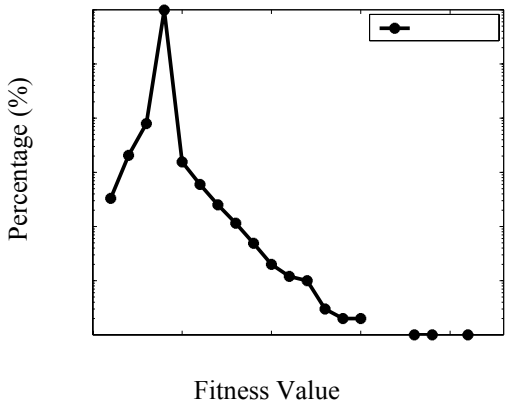



**Figure 16.** *PDF* of fitness value for imposter matchings.

### 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. Figure 12 shows five evolutions of the fitness value for the matching between the first pair of fingerprints (f0001_10 and s0001_10) shown in Figure 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. Figure 13 shows four pairs of fingerprints, which are the different impressions of the *same* fingers, and their corresponding maximum fitness values. Figure 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 Figure 14 are not from the same fingers, their fitness values are small. Figure 15 and Figure 16 show the average probability distribution function (*PDF)* of 10 experiments for both distributions, respectively. Note that the fitness value is shown on logarithmic scale along different axis, and on the average, 99.1% imposter matchings and 11.5% genuine matchings' have fitness value of 4.

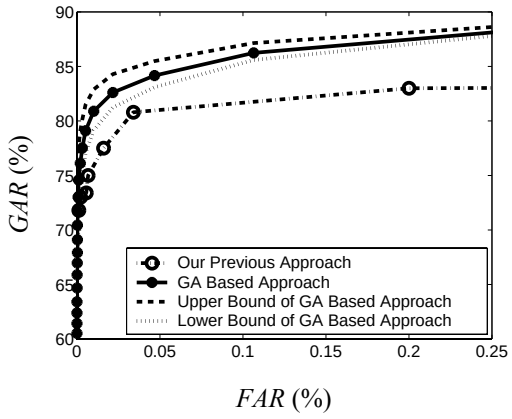



**Figure 17.** Comparison of *ROC* curves of two approaches using entire *NIST-4* (2000 pairs of fingerprints).

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*). Figure 17 shows the comparison of the average *ROC* curve of our *GA* based approach in this paper and the *ROC* curve reported in [23]. It also shows the lower and upper bounds of *GA* base

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. less than 0.02%, the advantage is about 2.0%. One main reason is that this approach finds global transformation and uses the 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 [15]. However, in [15] 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 us to compare the performance reported in [15] with *DTW* since the author has not used the entire database, and we do not know which of the fingerprints have been rejected manually.

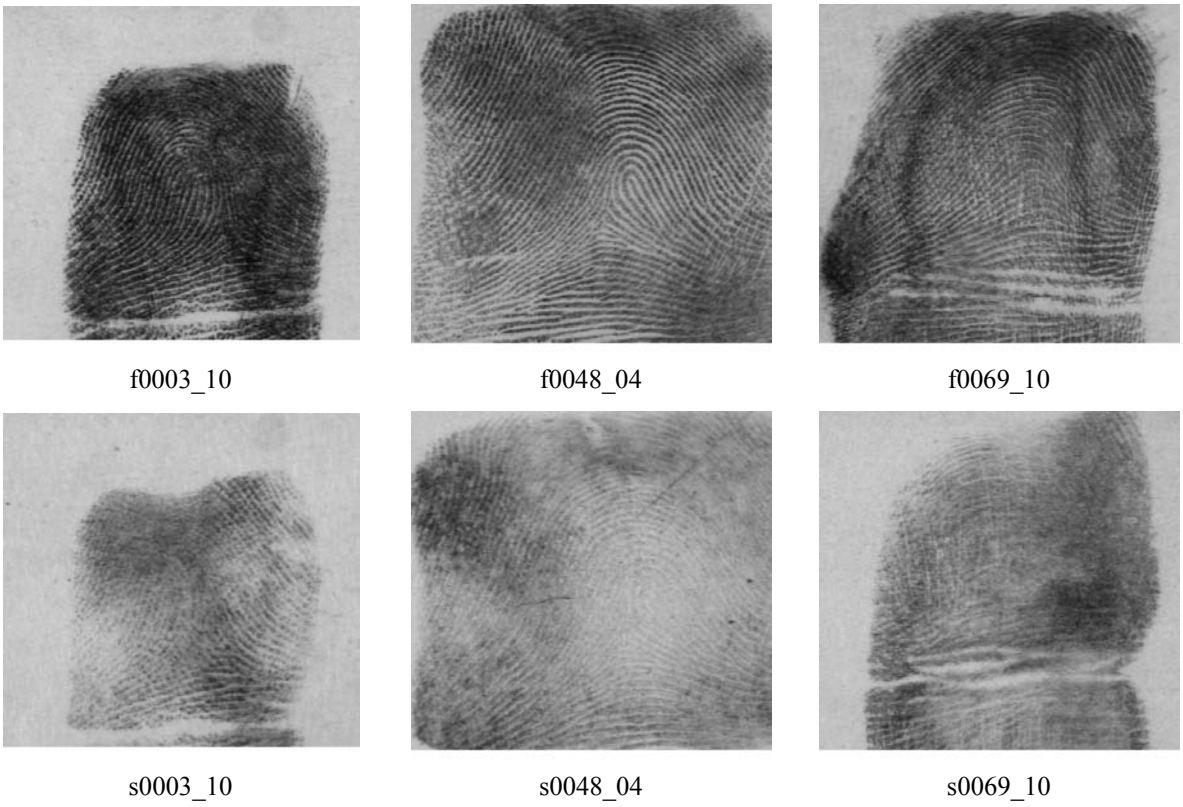



**Figure 18.** Sample fingerprints with low fitness value in genuine matching.

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. Figure 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 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. Figure 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*.

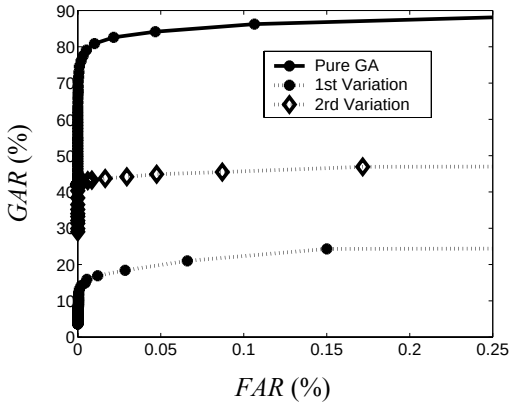



**Figure 19.** Comparison of the average performance of pure *GA* and its two variations.

### 4.6 Computation time

On a SUN Ultra II workstation, which has a 200MHZ cpu, the average computation time for a genuine matching and an imposter matching are 15 and 8 seconds, 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 can not satisfy the real-time requirement of the real-world applications. However, the computation time can be reduced by:

- Faster CPU (e.g. 3.0 GHZ CPU) is available in the PC market now, and it is supposed to have faster CPUs according to Moore's law;
- Parallel computation. Computation time may be reduced by parallel 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.

**Acknowledgment:** To be filled later.

**References**

[1] B. Bhanu and X. Tan, Learned templates for feature extraction in fingerprint images, *Proc. IEEE Conf. on Computer Vision and Pattern Recognition*, vol. 2, pp. 591-596, 2001.

[2] B. Bhanu and X. Tan, A triplet based approach for indexing of fingerprint database for identification, *Proc. Int. Conf. on Audio- and Video-Based Biometric Person Authentication*, pp. 205-210, 2001.

[3] T. Back, U. Hammel and H.P. Schwefel, Evolutionary computation: comments on the history and current state, *IEEE Trans. Evolutionay Computation 1(1)*, 1997, 3-17.

[4] G. Bebis, S. Louis, Y. Varol, and A. Yfantis, Genetic object recognition using combinations of views, *IEEE Trans. Evolutionary Computation*, 6(2), 2002, 132-146.

[5] S.B. Cho, Pattern recognition with neural networks combined by genetic algorithm, *Fuzzy Sets Systems*, 103(2), 1999, 339-347.

[6] A.V. Ceguerra and I. Koprinska**,** Integrating local and global features in automatic fingerprint verification, *Proc. International Conference on Pattern Recognition*, vol. 3, pp. 347-350, 2002.

[7] J. H. Holland, Outline for a logical theory of adaptive systems, *J. Assoc. Comput. Mach.*, vol.3, 1962, 297-314.

[8] X. Jiang and W.Y. Yau, Fingerprint minutiae matching based on the local and global structures, *Proc. International Conference on Pattern Recognition*, 1038-1041, 2000.

[9] J.M. Johnson, Genetic algorithm design of a switchable shaped beam linear array with phase-only control, *Proc. IEEE Aerospace Conference*, vol.3, pp. 297 –303, 1999.

[10] J.M. Johnson and Y. Rahmat-Samii, Genetic algorithm optimization for aerospace electromagnetic design and analysis, *Proc. IEEE Aerospace Applications Conference*, vol. 1, pp. 87-102, 1996.

[11] A.K. Jain, L. Hong, S. Pankanti, and R. Bolle, An identity-authentication system using fingerprints, *Proc. of the IEEE* 85(9), pp. 1364-1388, 1997.

[12] A.K. Jain, A. Ross and S. Prabhakar, Fingerprint matching using minutiae and texture features**,** *Proc. International Conference on Image Processing*, vol. 3, pp. 282-285, 2001.

[13] A.K. Jain, S. Prabhakar, L. Hong and S. Pankanti, Filterbank-based fingerprint matching, *IEEE Transactions on Image Processing*, 9(5), pp. 846-859, 2000.

[14] T. Kawaguchi and M. Nagao, Recognition of occluded objects by a genetic algorithm, *Trans. of IEICE D-II*, J82(3), 1999, 350-360.

[15] Z.M. Kovacs-Vajna, A fingerprint verification system based on triangular matching and dynamic time warping, *IEEE Trans. Pattern Analysis and Mach. Intel.* 22(11), 2000, 1266-1276.

[16] 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 Transactions on Systems, Man and Cybernetics - Part B: Cybernetics*, 2002.

[17] C. Mandal, P.P. Chakrabarti and S. Ghose, GABIND: a GA approach to allocation and binding for the high-level synthesis of data paths, *IEEE Transactions on Very Large Scale Integration (VLSI) Systems*, 8(6), pp. 747-750, 2000.

[18] A.C.M. de Oliveira and L.A.N. Lorena, A constructive genetic algorithm for gate matrix layout problems, *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, 21(8), pp. 969-974, 2002.

[19] E. Ozcan and C.K. Mohan, Partial shape matching using genetic algorithms, *Pattern Recognition Letters*, 18, 1997, 987-992.

[20] A.A. Saleh, R.R. Adhami, Curvature-based matching approach for automatic fingerprint identification, *Proc. Southeastern Symposium on System Theory*, pp. 171 –175, 2001.

[21] B. Sareni, L. Krahenbuhl and A. Nicolas, Efficient genetic algorithms for solving hard constrained optimization problems, *IEEE Transactions on Magnetics*, 36(4), pp. 1027-1030, 2000.

[22] H. Saito and M. Mori, Application of genetic algorithms to stereo matching of images, *Pattern Recognition Letters*, 16, 1995, 815-821.

[23] X. Tan and B. Bhanu, Robust fingerprint identification, *Proc. IEEE Int. Conf. on Image Processing*, Vol. 1, pp. 277-280, Sept. 2002.

[24] A. Toet and W.P. Hajema, Genetic contour matching, *Pattern Recognition Letters*, 16, 1995, 849-856.

[25] P.W.M. Tsang, A genetic algorithm for aligning object shapes, *Image and Vision Comp.*, 15(11), 1997, 819-831.

[26] C.I. Watson and C.L. Wilson, NIST special database 4, fingerprint database, U.S. National Institute of Standards and Technology, 1992.

[27] D.S. Yeung, Y.T. Cheng, H.S. Fong and F.L. Chung, Neocognitron based handwriting recognition system performance tuning using genetic algorithm, *Proc. IEEE Int. Conf. on Systems, Man, and Cyber.*, 5(5), 1998, 4228-4233.

[28] M. Zaki, A. El-Ramsisi and R. Omran, A soft computing approach for recognition of occluded shapes, *The Journal of Systems and Software*, 51, 2000, 73-83.

[29] B. Bhanu and S. Lee, Genetic learning for adaptive image segmentation, *Kluwer Academic Publishing*, 1994.