

Fingerprint Verification Using 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.

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 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. 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 in Jain et al. [4], 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 [5] used 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 [7], uses triangular matching to deal with the deformations of fingerprints. However, without Dynamic Time Warping (*DTW*) for further verification, the result is not acceptable. In our previous work [8], we have developed a fingerprint identification approach, which is based on the local optimization of the corresponding triangles to perform verification between two fingerprints.

2.2. Contribution

In this paper, we use Genetic Algorithms (*GA*) to achieve an 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 internal 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 our previous work [8] 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 $n = 1, 2, 3, \dots, N$, $m = 1, 2, 3, \dots, M$. The number of minutiae in the template and the query fingerprints are N and M , respectively. A transformation $Y_i = F(X_i)$ between Y_i and X_i can be written as:

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

where s is the scale factor, $R = \begin{bmatrix} \cos \theta & -\sin \theta \\ \sin \theta & \cos \theta \end{bmatrix}$, θ

is the angle of rotation between two fingerprints, and $T = [t_x, t_y]^T$ is the vector of translation.

Thus, the matching problem can be stated as to find the globally optimized transformation, which can map as many as possible minutiae in the template fingerprint to the minutiae in the query fingerprint.

3.2. Optimization based on GA

Genetic algorithms, introduced by Holland [3], provide an approach to learning that is loosely based on simulated evolution. The search for an appropriate hypothesis for a match begins with a population of initial hypotheses. Members of the current population generate the new generation by means of selection, crossover and mutation. 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 is in Back et al. [1].

GA are used in object recognition applications. Bebis et al. [2] used GA to recognize 2D or 3D objects from 2D intensity images. The approach is model-based, while the recognition strategy is based on the theory of algebraic functions of views. Tsang [9] 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 based approaches. Kawaguchi and Nagao [6] developed an algorithm to extract and locate partially visible objects from 2D images, in which GA are used to find the image fragment with the best match to the model fragment from the boundary of the image objects. Like other approaches, we used GA for optimization, but defined fitness function that exploits similarity of local structure.

3.2.1. 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 . The range of these parameters are: $0.9 \leq s \leq 1.1$, $-30^\circ \leq \theta \leq 30^\circ$, $-128 \leq t_x \leq 128$, $-128 \leq t_y \leq 128$. 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.

3.2.2. Fitness function.

1) **Step 1:** Suppose the optimized transformation is $\hat{F}(\hat{s}, \hat{\theta}, \hat{t}_1, \hat{t}_2)$, $\forall j, j = 1, 2, 3, \dots, N$. Let

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

If d_j is less than a threshold T_d , then we define the points $[x_{j,1}, x_{j,2}]^T$ and $[y_{k,1}, y_{k,2}]^T$ are potential corresponding points. If n_c , the number of potential corresponding points based on $\hat{F}(\hat{s}, \hat{\theta}, \hat{t}_1, \hat{t}_2)$, is less than a threshold T_n , then let the fitness value for the transformation $\hat{F}(\hat{s}, \hat{\theta}, \hat{t}_1, \hat{t}_2)$ be $FV(\hat{F}) = n_c$. In this case, it does not make sense to further evaluate the matching between two sets of minutiae. Otherwise, we check the local properties of the triplets of minutiae, which are defined in step 2.

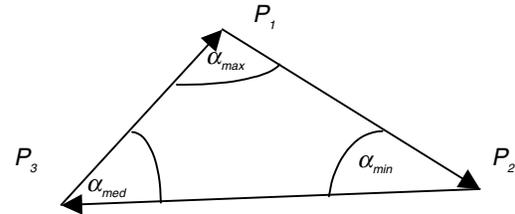


Figure 1. Definition of labels of feature points

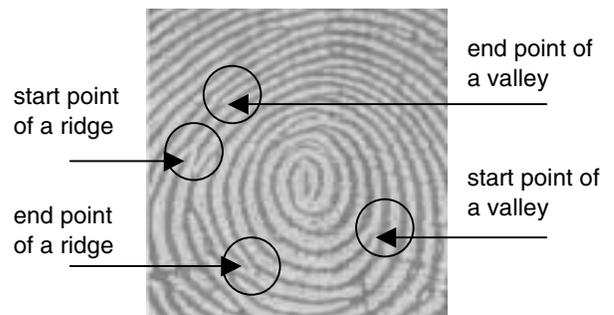


Figure 2. Examples of minutiae with different ν values

2) **Step 2:** Each noncolinear triplet of potential corresponding points can form a triangle in their respective query and template images. The local properties associated with a triangle include:

- **Angles α_{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 α_{max} , we label this point as P_1 ; if the minutia is the vertex of angle α_{min} , we label it as P_2 ; the last minutia is labeled as P_3 . Figure 1 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 $\phi = \text{sign}(Z_{21} \times Z_{32})$, where sign is the signum function and \times is the cross product of two complex numbers.

- **Triangle Direction η** : Search the minutia from top to bottom and left to right in the fingerprint images. 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 , $i = 1, 2, 3$. Figure 2 shows examples of minutiae with different v value.

- **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.

- **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.

If two triangles from two different fingerprints of the same finger 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:

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

$$\phi' = \phi'', \eta' = \eta'', \left| \lambda' - \lambda'' \right| \leq T_\lambda$$

$$\left| \chi'_i - \chi''_i \right| \leq T_\chi, \left| \xi'_i - \xi''_i \right| \leq T_\xi, i = 1, 2, 3 \quad (3)$$

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_ξ are thresholds to deal with the local distortions.

Thus, the fitness function is defined as:

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

3.2.3. Population generation. Suppose (a) the size of population P is N_p ; (b) hypotheses of the transformation

are \hat{F}_i , $i = 1, 2, 3, \dots, N_p$; (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 . Generally, crossover operators include single-point crossover, two-point crossover, and uniform crossover. For simplicity, we use uniform crossover in our experiments. The probability $Pr(\hat{F}_i)$ is defined by

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

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

3.2.4. 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) If the fitness value of a match is greater than 100, then the match is a correct genuine match, and we terminate the evolution.

3.2.5. 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 condition 2) that is 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 changes the fitness value of those hypotheses. If no mutation happens, it is unnecessary to compute the fitness value again.

4. Experimental results

4.1. Database and parameters

The database we use in our experiments is the NIST Special Database 4 ($NIST-4$) [10], 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 pairs of sample fingerprints can be found in Figure 4 and Figure 8. The parameters used are: $T_d = 12$, $T_n = 5$, $T_{a_{min}} = 2^\circ$, $T_{a_{med}} = 2^\circ$, $T_\lambda = 20$, $T_\chi = 2$, $T_\zeta = 2$, $p_s = 0.2$, $p_m = 0.005$, $N_p = 100$, $N_t = 15$.

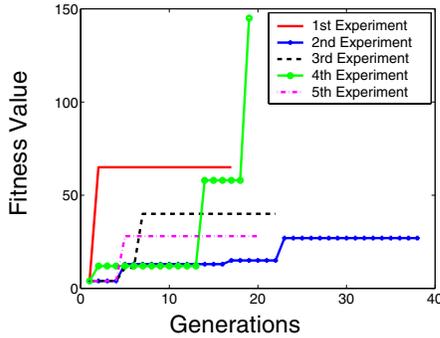


Figure 3. Fitness value changes for the matching between f0001 and s0001 in NIST-4

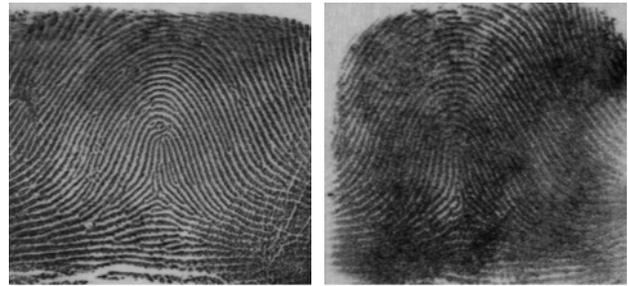
4.2. Results

A total of 2000 matchings, between consistent pairs, are performed to estimate the genuine distribution. We also perform 200,000 matchings between inconsistent pairs to estimate the imposter distribution, where for each matching we randomly select two fingerprints from NIST-4 that are the impressions of different fingers. Figure 3 shows five evolutions of the fitness value for the matching between f0001 and s0001 in NIST-4. Since we use two methods, explained in Section 3.2.5, for the reduction in computation time, if the fitness value jumps to be greater than 100, then the evolution will stop. We observe that the evolutions terminate within 40 generations. Considering the size of the search space is 2^{27} , the efficiency of GA is obvious. We repeat each experiment of genuine matching and imposter matching for five times. Figure 4 shows three pairs of fingerprints. We observe that the more similar the two fingerprints are, the larger the fitness value for their match is. Figure 5 and Figure 6 show the average PDF of five experiments for both distributions, respectively. Note that the fitness value is shown on logarithmic scale along different axis, and on the average, fitness value is 4 for 99.1% of imposter matchings.

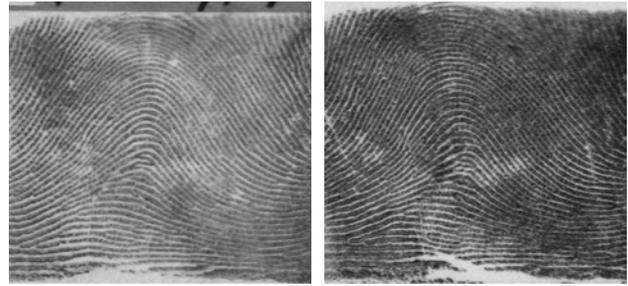
Based on genuine and imposter distributions, the Receiver operating Characteristic (ROC) curve is defined as the plot of Genuine Acceptance Rate (GAR) against False Acceptance Rate (FAR). Figure 7 shows the comparison of the average ROC curve of our GA based approach in this paper and the ROC curve reported in [8]. 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 GA attempt to find

optimal global transformation and use the local properties for verification it, which is better than using local optimization only.

Examining the results, we find that: a) the low fitness values for most genuine matchings are due to the poor quality of fingerprints. There are not enough common areas from the pair of fingerprints, from which feature extraction can extract enough good minutiae; b) the nonzero fitness value for most imposter matchings are due to similar structures and clutter features in two different fingerprint images. Figure 8 shows one pair of low quality fingerprints, whose maximum fitness value is 4.



f0026_03 and s0026_03, FV = 14, Correct Recognition



f0032_03 and s0032_03, FV = 75, Correct Recognition



f0025_06 and s0025_06, FV = 162, Correct Recognition

Figure 4. Maximum fitness value changes according to the similarity of fingerprints

4.3. 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. It does not satisfy the real-time requirement for some of the real-world applications. However, the computation time may be further reduced by parallel hardware.

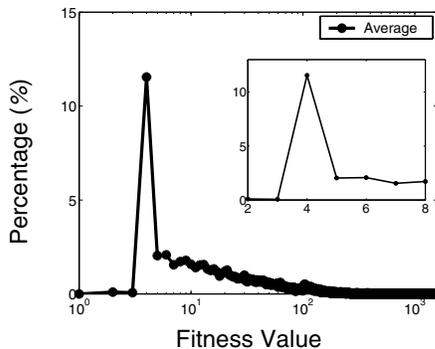


Figure 5. PDF of FV for genuine matchings (2000x5 matchings)

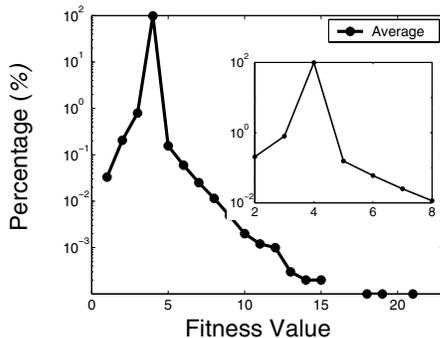


Figure 6. PDF of FV for imposter matchings (200000x5 matchings)

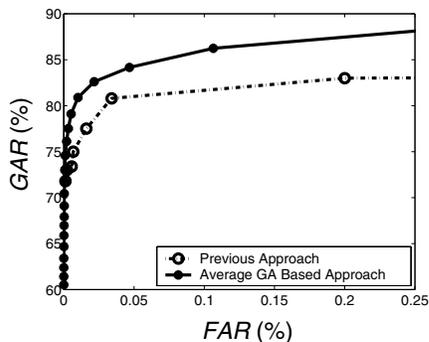


Figure 7. Comparison of two approaches using entire NIST-4 (2000 pairs fingerprints)

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 triplet 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. A comparison with a locally optimized approach with no global constraints shows the advantage of the proposed approach.

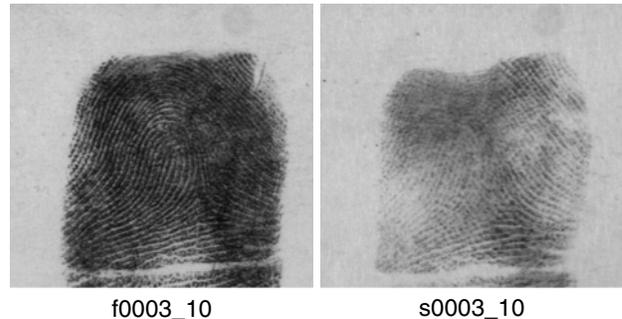


Figure 8. Sample fingerprints with low fitness value in genuine matching

Acknowledgment: This work is supported in part by grants from Sony, DiMI, I/O software and F496200210315. The contents and information do not necessarily reflect the positions or policies of the sponsors.

References

- [1] T. Back, U. Hammel and H.P. Schwefel, Evolutionary computation: comments on the history and current state. *IEEE Trans. Evolutionary Computation* 1(1), 1997, 3-17.
- [2] 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.
- [3] J. H. Holland, Outline for a logical theory of adaptive systems, *J. Assoc. Comput. Mach.*, vol.3, 1962, 297-314.
- [4] A.K. Jain, L. Hong, S. Pankanti, and R. Bolle, An identity-authentication system using fingerprints. *Proc. of the IEEE* 85(9), 1997, 1364-1388.
- [5] X. Jiang and W.Y. Yau, Fingerprint minutiae matching based on the local and global structures. *Proc. ICPR*, 2000, 1038-1041.
- [6] T. Kawaguchi and M. Nagao, Recognition of occluded objects by a genetic algorithm, *Trans. of IEICE D-II*, J82(3), 1999, 350-360.
- [7] 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.
- [8] X. Tan and B. Bhanu, Robust fingerprint identification, *Proc. IEEE Int. Conf. on Image Processing*, 2002.
- [9] P.W.M. Tsang, A genetic algorithm for aligning object shapes, *Image and Vision Comp.*, 15(11), 1997, 819-831.
- [10] C.I. Watson and C.L. Wilson, NIST special database 4, fingerprint database, U.S. National Institute of Standards and Technology, 1992.