

Fingerprint Matching with an Evolutionary Approach

W. Sheng, G. Howells, K. Harmer, M.C. Fairhurst, and F. Deravi

Department of Electronics,
University of Kent, Canterbury,
Kent, CT2 7NT, United Kingdom
{W. Sheng, K. Harmer, W. G. J. Howells, M. C. Fairhurst,
F. Deravi}@kent.ac.uk

Abstract. Minutiae point pattern matching is probably the most common approach to fingerprint verification. Although many minutiae point pattern matching algorithms have been proposed, reliable automatic fingerprint verification remains a challenging problem, both with respect to recovering the optimal alignment as well as to the construction of adequate matching function. In this paper, we develop an evolutionary approach for fingerprint matching by combining the use of the global search functionality of a genetic algorithm with a local improvement operator to search for the optimal global alignment between two minutiae sets. Further, we define a reliable matching function for fitness computation. The proposed approach was evaluated on two public domain collections of fingerprint images and compared with previous work. Experimental results show that our approach is reliable and practical for fingerprint verification, and outperforms the traditional genetic algorithm based method.

Keywords: Fingerprints, matching/verification, alignment, minutiae, genetic algorithms.

1 Introduction

Fingerprints are graphical ridge and valley patterns on the tips of human fingers. Owing to their uniqueness and permanence, the use of fingerprints is considered to be one of the most reliable methods of personal verification. Fingerprints are today among the most popularly used biometric modality in automatic verification systems. Due to the continuing needs of law enforcement and interest from the developers of civilian applications, automated fingerprint verification systems are becoming increasingly widespread and are being extensively researched by the pattern recognition and image processing communities. Although fingerprints possess much discriminatory information, and although significant progress in automating the verification process has been made, reliable automatic fingerprint verification is still a challenging problem [4].

The uniqueness of fingerprints has been well established [22], and can be determined by the overall pattern of ridges and valleys as well as the local ridge discontinuities termed “minutiae”. It is widely believed that the minutiae are the most discriminating and reliable features [17, 29] present in the fingerprints. For this reason, they are the most important and common features used in automatic fingerprint verification systems. The two prominent types of minutiae commonly used in automatic fingerprint verification are ridge ending and ridge bifurcation. The representation of a fingerprint pattern thus comprises all detected ridge endings and ridge bifurcations in a so called *minutiae set*.

By representing the minutiae set as a point pattern, the fingerprint verification problem can be viewed as a minutiae point pattern matching problem. Suppose a minutiae template set P is composed of m points, $P = \{p_1, p_2, \dots, p_m\}$, and a query minutiae set Q is composed of N points, $Q = \{q_1, q_2, \dots, q_n\}$. Each minutia is usually described by parameters (x, y, α) , where (x, y) are the pixel coordinates of the minutia with respect to the image frame and α is the orientation of the minutia, which is defined as the angle that the ridge associated with the minutia makes with the horizontal axis [17]. Typically, ridge orientations are restricted to the range $[0, \pi)$, so that directions θ and $\theta+\pi$ have the same orientation. It should be noted that most minutiae representation schemes do not distinguish endings from bifurcations since the type of a minutia can be easily interchanged by acquisition noise or pressure differences during acquisition. However, the orientation remains the same when this occurs. The alignment between a template and a query minutiae set can be simplified as an affine transformation $G(t_x, t_y, \theta, s)$ composed of four parameters: t_x, t_y, θ , and s , where t_x and t_y are the translations along the x and y directions, respectively, θ is the rotation angle and s is a scaling factor. Thus, the transformation $F(q_i) = (x'_i, y'_i, \alpha'_i)$ of a minutia $q_i = (x_i, y_i, \alpha_i)$ can be written as follows:

$$\begin{bmatrix} x'_i \\ y'_i \\ \alpha'_i \end{bmatrix} = s \begin{bmatrix} \cos \theta & -\sin \theta & 0 \\ \sin \theta & \cos \theta & 0 \\ 0 & 0 & 1/s \end{bmatrix} \begin{bmatrix} x_i \\ y_i \\ \alpha_i \end{bmatrix} + \begin{bmatrix} t_x \\ t_y \\ \theta \end{bmatrix}.$$

The fingerprint verification process can then be defined as the problem of finding the transformation between the template and query minutiae sets which can optimize a given matching function.

Due to the large number of possible translations, rotations and scalings, finding the best alignment between two point patterns is an extremely difficult problem. A number of algorithms [5, 9, 10, 12, 13, 14, 15, 23, 25, 26] have been proposed in the literature. A common technique for these algorithms is to use local features associated with minutiae and/or their spatial properties to reduce the exponential number of search paths. These methods typically recover the transformation by choosing a reference minutia or minutia group (in which minutiae are close to each other) from the template and the query fingerprint, respectively. The two sets of minutiae are then aligned according to local structures of the two references. We refer to these methods as single reference based approaches. This approach can guarantee satisfactory alignments of regions adjacent to the reference minutia or minutia group. However, alignments of regions far away from the reference minutia or minutia group are usually not so satisfactory. This is largely because the alignment which tends to be found is locally strong, yet poor in areas distant to the local structure it has matched. Naturally, researchers have explored the use of a size-changeable bounding box [10, 12, 13].

Another alternative approach is to find a transformation in order to globally align two sets of minutiae. This approach tends to evenly align two sets of minutiae and thus one can use a size-fixed bounding box to identify corresponding minutiae. Zhu et al. [28] have proposed a method which aligns two sets of minutiae based on multiple pairs of reference minutiae. This method depends highly on the initialization of the minutiae pairs. Since the global alignment is a computationally intractable problem, naturally inspired evolutionary optimization algorithms have recently been a source of interest for minutiae point pattern matching [16, 24]. Tan and Bhanu [24] proposed

the use of a traditional genetic algorithm [11] for fingerprint matching. Le et al. [16] employed the technique of fuzzy evolutionary programming to match two sets of minutiae. These methods try to identify the optimal or near optimal global alignment between two minutiae sets and their experimental results are promising. However, they may take a large amount of time to converge, mainly because these methods employ either the simple evolutionary algorithm or its variants, which may not be well suited to fine-tuning search in complex search spaces.

In this paper, we develop a novel evolution based algorithm for fingerprint matching that follows the scheme of global minutiae alignment. In contrast to previous methods, our proposed algorithm hybridizes a genetic algorithm (GA) with a local improvement operator. Further, we define a reliable and discriminating matching function for fitness computation by combining the globally matched minutiae pairs with the result of the minutiae's local feature similarity based on the product rule.

2 Evolution Based Algorithm for Fingerprint Matching

In this section, we present the details of our proposed algorithm, which combines the use of a global search via a genetic algorithm (GA) with a local improvement operator to search for the optimal global matching between two minutiae sets. Abstractly, the proposed algorithm (called EAFM) consists of selecting parents for reproduction, performing a modified arithmetic crossover with the parents, applying mutation to the offspring, running a local improvement operator on each offspring, and carrying out a competition replacement. The evolution is terminated when one of the following two stopping criteria is met: 1) the fitness value of the best individual has not changed for N generations or 2) the fitness value of the best individual exceeds a certain threshold value, which means the two fingerprints are verified as from the same finger. The output of the algorithm is the best solution encountered during the evolution. The algorithm may be expressed concisely as follows with the subsequent sections elaborating on the individual component phases:

- Step 1. Randomly initialize P sets of chromosomes, which encode alignment configurations, based on a real-parameter representation (see Section 2.1).
- Step 2. Calculate the fitness value according to the method described in Section 2.4 for each individual chromosome in the initial population.
- Step 3. Repeat the following sub-steps (a) to (e) until a stopping criterion is met.
 - a) Select the pairing individuals based on the k -fold tournament selection method [8]. This procedure is repeated until $P/2$ parent pairs are selected.
 - b) Generate intermediate offspring by applying a modified arithmetic crossover and then perform Gaussian mutation on the offspring (see Section 2.2).
 - c) Run the local improvement operator on the offspring and update the offspring (see Section 2.3).
 - d) Calculate the fitness value for each of the offspring according to the method described in Section 2.4.
 - e) Create a new generation of size P from the best individual of the previous generation and the best offspring that resulted from crossover, mutation and local improvement operations.
- Step 4. Provide the alignment configuration for the terminal population individual with the best fitness.

Algorithm 1. An evolution based algorithm for fingerprint matching

2.1 Representation and Initialization

Our representation for the individual chromosome consists of a vector of four real numbers, where the first two positions represent translations along the x and y directions respectively, the next position represents the rotation angle, and the last position represents the scaling factor. Each individual in the population is constructed by random assignment of a real number to each of the attributes of the solution. The initial values are constrained to be in the range of the attribute to which they are assigned (determined empirically from the experimental data sets) but are otherwise random.

2.2 Crossover and Mutation

Crossover is a probabilistic process that exchanges information between a pair of parents to generate two offspring. The arithmetic crossover technique [8] has been used as the reproduction operator in the EAFM. Traditional arithmetic crossover linearly combines two parent chromosomes to produce two new offspring according to the following equations:

$$Offspring1 = a * Parent1 + (1 - a) * Parent2, \quad (1)$$

$$Offspring2 = (1 - a) * Parent1 + a * Parent2, \quad (2)$$

where $a \in [0,1]$ is a random weighting factor. In our approach, we apply either equation (1) or (2) randomly to produce only one single offspring. This helps to save time in processing the other similar offspring. The crossover is performed on each paired parent.

After crossover, a low probability of Gaussian mutation is applied to the offspring. Gaussian mutation adds a unit Gaussian-distributed random value to the chosen attribute value. The new attribute value is clipped if it falls outside of the lower or upper bounds of that attribute.

2.3 Local Improvement Operation

GAs are able to escape from local optima by means of crossover and mutation operators. However, they are not well suited for fine-tuning structures which are close to optimal solutions [8], and this results in their exhibiting a large execution time. To improve the time efficiency, incorporation of local improvement operators into the regeneration step of GAs, which called hybrid GAs, is essential. Hybrid GAs have been shown to be very effective for many combinatorial optimization problems [1, 4, 18, 21, 27]. In this subsection we present the local improvement operator to effectively design a hybrid GA for minutiae point pattern matching.

The local improvement operator is inspired by the iterated closest point (ICP) algorithm [3]. The ICP is a widely used heuristic for the alignment of 3D geometric models. It utilizes the *nearest-neighbour* relationship to assign a binary correspondence at each step. This estimate of the correspondence is then used to refine the transformation, and vice versa. This iterative scheme is known to converge fast [6]. However, it is sensitive to its initial rotations and translations, and susceptible to local optima. Its performance degenerates quickly with outliers (point features exist in one point-set

that have no corresponding points in the other), which is common in fingerprint minutiae point pattern matching problem.

In order to improve the computational efficiency, we design a local improvement operator based on one iteration of the ICP to fine-tune new offspring during each generation, after the regeneration step. This operator is summarized below:

- Step 1. Extract the transformation information encoded in the individual solution. Apply the transformation to the query minutiae set.
- Step 2. Compute the closest point pairs between the two minutiae sets by estimating the Euclidean distance between the minutiae coordinates. Collect corresponding pairs if they satisfy the following geometric constraints: 1) the Euclidean distance does not exceed a certain value Δd and 2) the angular difference between their directions is less than a certain tolerance $\Delta\theta$ (see Section 4.2 for setting of the parameters Δd and $\Delta\theta$). To avoid a minutia being doubly used for pairing, we mark the minutiae that have already been paired.
- Step 3. Compute the new transformation, in the sense of minimizing the sum of squared Euclidean distance error among the collected corresponding pairs.
- Step 4. Update the individual solution with the new transformation.

Algorithm 2. A local improvement operator

2.4 Fitness Computation

The fitness of an individual indicates the degree of suitability of the solution it represents. Here, we introduce a local feature similarity to define a reliable and discriminating matching function to compute the fitness of individual solutions. The minutiae local feature proposed by Tico and Kuosmanen [25] has been used for designing such a matching function. In [25], each minutia defines a local structure, which is called a *minutia descriptor*. The minutia descriptor comprises information about the orientation field sampled in a circular pattern in a broad region around the minutia point. The circular pattern consists of L concentric circles of radii r_l , ($1 \leq l \leq L$), each one comprising K_l sampling points $p_{k,l}$, ($1 \leq k \leq K_l$), equally distributed along its circumference. Let $a = \{\alpha_{k,l}\}$ and $b = \{\beta_{k,l}\}$ be two minutia descriptors. The similarity function between a and b is defined as:

$$S(a, b) = (1/K) \sum_{l=1}^L \sum_{k=1}^{K_l} \exp(-16 * \Lambda(\alpha_{k,l}, \beta_{k,l})), \quad (3)$$

where $K = \sum_{l=1}^L K_l$, $\alpha_{k,l}$ and $\beta_{k,l}$ are the local ridge orientations estimated at $p_{k,l}$ for descriptors a and b , respectively, and $\Lambda(\alpha_{k,l}, \beta_{k,l})$ is the orientation distance between angles, which takes values between 0 and 1.

The minutia descriptors contain useful discriminatory information. Additionally, they can tolerate some deformation since they are formed from only a small area of the fingerprint. Thus, the similarity of minutia descriptors can be used to increase the reliability of the matching between two minutiae sets. Further, to make the matching function more discriminatory, we combine the globally matched minutiae pairs with the result of the minutiae's local feature similarity based on the product rule. Now the matching function can be defined as:

$$fitness = \frac{1}{n} \sum_{i=1}^n S(a_i, b_i) \times \frac{2n}{(q+t)}, \quad (4)$$

where t and q are the number of minutiae located inside the intersection of the two fingerprint images for template and query respectively. The number of matched pairs, n , is identified using the same geometric constraints as those described in Section 2.4.

For each individual, the transformation encoded within it is first extracted, then subsequently we apply the transformation onto the query minutiae set Q to obtain the set Q' comprising the transformed minutiae set. Given the minutiae sets P and Q' , the number of matched minutiae, n , is first computed. If n is less than a threshold F_n , then let the fitness of the individual be $f=-1/n$ ($1 \leq n \leq F_n$) or $f=-1$ ($n=0$). In this case, it makes no sense to evaluate the matching function. Otherwise, the fitness of the individual is defined according to equation (4), so that maximization of the fitness is equivalent to finding the maximum number of matched minutiae points with the maximum average local feature similarity.

3 Data Sets and Parameter Settings

In this section, we describe the data sets used in the experiments. This is followed by a description of the implementation parameter settings for the proposed algorithm. Two public domain collections of fingerprint images, labeled DB_1 and DB_3 , proposed in [32] as part of the FVC2002 Fingerprint Verification Competition were used in the experiments. The fingerprint images were captured using fingerprint scanners and contain a wide variety of fingerprint image qualities. Each of the two data collections comprises 800 fingerprint images captured at a resolution of 500dpi, from 100 fingers (eight impressions per finger). The minutiae information of the data sets is derived using the method described in [7].

The EAFM has a few parameters which need to be set. These include the GA parameters, the ranges used for population initialization and several thresholds. The values of GA parameters and ranges are determined experimentally on the above data sets. To establish these values, all other variables were held constant with only the one to be established changing, and five runs were completed for a wide range of values in each case. The results from each of the five runs were averaged and the best average was selected. Both the matching accuracy and efficiency of results were used in determining the values of variables. The crossover and mutation probabilities are set to be 0.9 and 0.02 respectively. Generally, we have found that a crossover rate of 0.8-0.95 with a mutation rate of 0.01-0.05 offer best results. The order of tournament selection controls how quickly a population is "taken over" by dominant individuals in the population [2]. We used a tournament order of three. The number of generations, N , used to terminate the evolution and the population size, p , are set to be 5 and 20, respectively. A larger value of either p or N may lead to a longer running time, however, with no significant improvement of the matching performance. The ranges of the rotation angle, scaling factor, translations along the x and y directions are set to be [-0.6, 0.6], [0.9, 1.1], [-150, 150] and [-150, 150], respectively. The threshold of F_n for fitness computation is set to be four. The minutiae matching threshold values of Δd and $\Delta\theta$ are set as 15 and 0.195, respectively, which were established in [22] for fingerprint images scanned at 500dpi resolution.

4 Experimental Results

In this section, we evaluate the EAFM and compare its performance with a traditional GA based fingerprint matching method described in [24]. All the results reported in this section were obtained with simulations on a PC with an IntelTM PentiumTM 4 running under WindowsXP Professional.

Before discussing the comparative results, we first briefly describe the method to be compared. The GA based fingerprint matching method was recently proposed by Tan and Bhanu [24]. In this method, a GA with the traditional roulette wheel selection, uniform crossover, and binary flip mutation was used with a binary code representation, which represents alignment parameters, for fingerprint matching. The fitness function is based on the local properties of each minutia-triplet.

We mainly concern the performance with respect to matching accuracy and efficiency. Therefore, we report the Equal Error Rate (EER) and average matching time estimated using the experimental protocol proposed in [19]. The EER, which is commonly used to summarize the accuracy of a matching system [20], is defined as the error rate where the system's false match rate equals its false non-match rate. Table 1 lists the results of the EER values and average matching time of the two methods over the two data sets introduced above.

Table 1. EERs and average matching times estimated on DB₁ and DB₃ for the two methods (EAFM and Tan et al.'s method)

Methods	DB ₁		DB ₃	
	EER (%)	Average matching time (s)	EER (%)	Average matching time (s)
EAFM	1.1	2.91	3.6	3.07
Tan et al.'s method [24]	1.5	6.74	4.2	7.26

The results in Table 1 show that the EAFM is a reliable and practical matching algorithm. Compared with Tan et al.'s method, the EAFM is able to achieve lower error rate in both data sets. The EER of Tan et al.'s method turns out to be 1.5% and 4.2%, while our algorithm returns about 1.1% and 3.6% on DB₁ and DB₃, respectively. Further, matching operations of our algorithm are faster than the Tan et al.'s method. For example, Tan et al.'s method needs 6.74 second on DB₁ while our method takes about 2.91 second on average. The improvement of matching efficiency of our proposed algorithm is mainly due to the use of local improvement operation.

5 Conclusion

In this paper, we have reported on the design and implementation of a novel evolution based algorithm for fingerprint matching which seeks to identify the optimal global matching between two minutiae sets. A GA based evolution with local improvement hybridization is at the heart of the proposed algorithm. This has been developed with the particular goal of improving the efficiency of identifying optimal, or near optimal,

global matching between two fingerprint minutiae sets. Another key aspect of the proposed algorithm is the use of a reliable matching function for fitness computation. The experimental results confirm that the EAFM is a reliable and practical matching algorithm. The algorithm can achieve accurate matching results faster than the traditional GA based global fingerprint matching method.

References

- Areibi, S., Yang, Z.: Effective memetic algorithms for VLSI design automation = genetic algorithms + local search + multi-level clustering. *Evolutionary Computation* 12(3), 327–353 (2004)
- Bäck, T., Kursawe, F.: Evolutionary algorithms for fuzzy logic: A brief overview. *Proc. Information Processing and Management of Uncertainty in Knowledge-Based Systems*, 659–664 (1994)
- Besl, P.J., McKay, N.D.: A method for registration of 3D shapes. *IEEE Trans. Pattern Anal. and Machine Intell.* 14, 239–256 (1992)
- Branke, J., Middendorf, M., Schneider, F.: Improved heuristics and a genetic algorithm for finding short supersequences. *OR Spektrum* 20(1), 39–45 (1998)
- Chen, X., Tian, J., Yang, X.: A new algorithm for distorted fingerprints matching based on normalized fuzzy similarity measure. *IEEE Transactions on Image Processing* 15(3), 767–776 (2006)
- Chui, H., Rangarajan, A.: A new point matching algorithm for nonrigid registration. *Comput. Vision and Image Und.* 89, 114–141 (2003)
- Garris, M.D., McCabe, R.M., Watson, C.I., Wilson, C.L.: User's guide to NIST fingerprint image software (NFIS). NISTIR 6813, National Institute of Standards and Technology, Gaithersburg, MD (2001)
- Goldberg, E.: *Genetic Algorithms in Search, Optimization, and Machine Learning*. Addison-Wesley, Reading, Mass (1989)
- He, Y., Tian, J., Li, L., Chen, H., Yang, X.: Fingerprint matching based on global comprehensive similarity. *IEEE Transactions on Pattern Analysis and Machine Intelligence* 28(6), 850–862 (2006)
- He, Y., Tian, J., Luo, X., Zhang, T.: Image enhancement and minutiae matching in fingerprint verification. *Pattern Recognition Letters* 24, 1349–1360 (2003)
- Holland, J.H.: *Adaptation in Natural and Artificial Systems*. University of Michigan Press, Ann Arbor (1975)
- Jain, K., Hong, L., Bolle, R.: On-line fingerprint verification. *IEEE Transactions on Pattern Analysis and Machine Intelligence* 19(4), 302–314 (1997)
- Jain, K., Hong, L., Pankanti, S., Bolle, R.: An identity-authentication system using fingerprints. *Proc. IEEE* 85(9), 1365–1388 (1997)
- Jeon, T.Y., Govindaraju, V.: A minutia-based partial fingerprint recognition system. *Pattern Recognition* 38, 1672–1684 (2004)
- Jiang, X., Yau, W.: Fingerprint minutiae matching based on the local and global structures. In: *Proc. 15th International Conference on Pattern Recognition*, pp. 1038–1041 (2000)
- Le, T.V., Cheung, K.Y., Nguyen, M.H.: A fingerprint recognizer using fuzzy evolutionary programming. In: *Proc. of 34th International Conference on System Sciences* (2001)
- Lee, H.C., Gaensslen, R.E.: *Advances in Fingerprint Technology*. Elsevier, New York (1991)
- Li, F., Morgan, R., Williams, D.: Hybrid genetic approaches to ramping rate constrained dynamic economic dispatch. *Electric Power Systems Research* 43(2), 97–103 (1997)

- Maio, D., Maltoni, R., Cappelli, J., Wayman, L., Jain, A.K.: FVC2002: second fingerprint verification competition. In: Proc. International Conference on Pattern Recognition, pp. 811–814 (2002)
- Maltoni, D., Maio, D., Jain, A.K., Prabhakar, S.: Handbook of Fingerprint Recognition. Springer, Heidelberg (2003)
- Merz, P., Freisleben, B.: Memetic algorithms and the fitness landscape of the graph bipartitioning problem. LNCS, pp. 765–774 (1998)
- Pankanti, S., Prabhakar, S., Jain, A.K.: On the individuality of fingerprints. IEEE Trans. Patt. Anal. Mach. Intell. 24(8), 1010–1025 (2002)
- Qi, J., Shi, Z., Zhao, X., Wang, Y.: A robust fingerprint matching method. In: 7th IEEE Workshops on Application of Computer Vision, pp. 105–110 (2005)
- Tan, X., Bhanu, B.: Fingerprint matching by genetic algorithms. Pattern Recognition 39(3), 465–477 (2006)
- Tico, M., Kuosmanen, P.: Fingerprint matching using an orientation-based minutia descriptor. IEEE Transactions on Pattern Analysis and Machine Intelligence 25(8), 1009–1014 (2003)
- Tong, X., Huang, J., Tang, X., Shi, D.: Fingerprint minutiae matching using the adjacent feature vector. Pattern Recognition Letters 26(9), 1337–1345 (2005)
- Whitley, D.: Modeling hybrid genetic algorithms. In: Winter, G., Periaux, J., Galan, M., Cuesta, P. (eds.) Genetic Algorithms in Engineering and Computer Science, pp. 191–201. John Wiley, Chichester (1995)
- Zhu, J., Yin, P., Zhang, G.M.: Fingerprint matching based on global alignment of multiple reference minutiae. Pattern Recognition 38(10), 1685–1694 (2005)
- The Science of Fingerprints: Classification and Uses. Federal Bureau of Investigation, Washington, DC (1984)