



# Kent Academic Repository

Sheng, Weiguo, Howells, Gareth, Fairhurst, Michael and Deravi, Farzin (2007) *A memetic fingerprint matching algorithm*. IEEE Transactions on Information Forensics and Security, 2 (3). pp. 402-412. ISSN 1556-6013.

## Downloaded from

<https://kar.kent.ac.uk/2068/> The University of Kent's Academic Repository KAR

## The version of record is available from

<https://doi.org/10.1109/TIFS.2007.902681>

## This document version

UNSPECIFIED

## DOI for this version

## Licence for this version

UNSPECIFIED

## Additional information

## Versions of research works

### Versions of Record

If this version is the version of record, it is the same as the published version available on the publisher's web site. Cite as the published version.

### Author Accepted Manuscripts

If this document is identified as the Author Accepted Manuscript it is the version after peer review but before type setting, copy editing or publisher branding. Cite as Surname, Initial. (Year) 'Title of article'. To be published in *Title of Journal*, Volume and issue numbers [peer-reviewed accepted version]. Available at: DOI or URL (Accessed: date).

## Enquiries

If you have questions about this document contact [ResearchSupport@kent.ac.uk](mailto:ResearchSupport@kent.ac.uk). Please include the URL of the record in KAR. If you believe that your, or a third party's rights have been compromised through this document please see our [Take Down policy](https://www.kent.ac.uk/guides/kar-the-kent-academic-repository#policies) (available from <https://www.kent.ac.uk/guides/kar-the-kent-academic-repository#policies>).

# A Memetic Fingerprint Matching Algorithm

Weiguo Sheng, Gareth Howells, Michael Fairhurst, and Farzin Deravi

**Abstract**—Minutiae point pattern matching is the most common approach for fingerprint verification. Although many minutiae point pattern matching algorithms have been proposed, reliable automatic fingerprint verification remains as a challenging problem, both with respect to recovering the optimal alignment and the construction of an adequate matching function. In this paper, we develop a memetic fingerprint matching algorithm (MFMA) which aims to identify the optimal or near optimal global matching between two minutiae sets. Within the MFMA, we first introduce an efficient matching operation to produce an initial population of local alignment configurations by examining local features of minutiae. Then, we devise a hybrid evolutionary procedure by combining the use of the global search functionality of a genetic algorithm with a local improvement operator to search for the optimal or near optimal global alignment. Finally, we define a reliable matching function for fitness computation. The proposed algorithm was evaluated by means of a series of experiments conducted on the FVC2002 database and compared with previous work. Experimental results confirm that the MFMA is an effective and practical matching algorithm for fingerprint verification. The algorithm is faster and more accurate than a traditional genetic-algorithm-based method. It is also more accurate than a number of other methods implemented for comparison, though our method generally requires more computational time in performing fingerprint matching.

**Index Terms**—Alignment, fingerprints, genetic algorithms (GAs), matching, memetic algorithms, minutiae.

## I. INTRODUCTION

**F**INGERPRINTS 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 modalities 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 [8].

The uniqueness of fingerprints has been well established [40] and is determined by the overall pattern of ridges and valleys

as well as the local ridge discontinuities called “minutiae.” It is widely believed that the minutiae are the most discriminating and reliable features [29], [51] 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 used in automatic fingerprint verification are ridge endings and ridge bifurcations. A minutia detected in a fingerprint image can be characterized by a list of attributes that includes the minutia position, the minutia direction, and the type of minutia (ending or bifurcation). The representation of a fingerprint pattern thus comprises the attributes of all detected minutiae in a so-called minutiae set.

By representing the minutiae set as a point pattern, the fingerprint verification problem can be reduced to a minutiae point pattern matching problem. In the ideal case that two identical point patterns are exactly aligned with each other, fingerprint verification is a trivial task of counting the number of spatially overlapping pairs between the two minutiae sets. However, in practice, such a situation is generally not encountered. Due to variations that may occur between two minutiae sets extracted from different impressions of the same finger [23], determining whether they indeed represent the same finger can be an extremely difficult problem. The difficulty can be attributed to the following commonly encountered factors. First, no minutiae correspondence between the two minutiae sets is known beforehand. Second, both minutiae sets may suffer from false, missed, and displaced minutiae, caused by poor fingerprint image quality and imperfections in the minutiae extraction stage. Third, two fingerprints may be translated, rotated, and scaled with respect to each other. Fourth, fingers may exert an unevenly distributed pressure across the acquisition sensor resulting in local nonlinear deformations due to the elasticity of the skin. A fifth problem is that there may be only a small amount of overlap between the two fingerprints such that several minutiae are not “visible” in both instances. In the light of these complicating factors, the design of powerful matching algorithms is necessary to establish a realistic model of the variations to recover alignment configurations and to automatically obtain minutiae correspondences among minutiae sets of mated pairs. Generally, finding the best match between two point patterns can be challenging, even if minutiae are exactly located and no deformations exist between these two point patterns [22].

A way of tackling this problem is to use stochastic optimization schemes, among which is the genetic-algorithm (GA) approach. The GA, first developed by Holland [20], is biologically inspired and incorporates many mechanisms mimicking natural evolution. It has a great deal of potential in scientific and engineering optimization or search problems. Furthermore, the GA is naturally applicable to minutiae point pattern matching since it has an exponential search space. More recently, many hybrid

Manuscript received April 20, 2007; revised May 4, 2007. This work was supported by the U.K. Engineering and Physical Sciences Research Council (EP/C00793X/1). The associate editor coordinating the review of this manuscript and approving it for publication was Prof. Davide Maltoni.

The authors are with the Department of Electronics, University of Kent, Kent CT2 7NT, U.K. (e-mail: W.Sheng@kent.ac.uk; W.G.J.Howells@kent.ac.uk; M.C.Fairhurst@kent.ac.uk; F.Deravi@kent.ac.uk).

Digital Object Identifier 10.1109/TIFS.2007.902681

GA-local search methods, which incorporate local searches with traditional GAs, have been proposed and applied successfully to solve a wide variety of optimization problems [3], [8], [30], [36], [47]. These studies show that pure GAs are not well suited to fine-tuning the search in complex search spaces, and that hybridization with other techniques can greatly improve their efficiency. GAs that have been hybridized with local searches are also known as memetic algorithms (MAs). Since we are concerned here with a GA where the local search plays a significant role throughout the process, the term MA [35], [38], [39] is used in this paper.

In this work, we develop a memetic fingerprint matching algorithm (MFMA) that follows the scheme of a point pattern matching approach. The proposed algorithm can be divided into two stages: the first stage tries to efficiently produce an initial population of local alignment configurations, and the second stage proceeds by means of a hybrid evolutionary procedure which aims to identify the optimal or near optimal global matching between two minutiae sets. In the first stage, we introduce a local matching operation to obtain an initial population of alignment configurations by examining local features of minutiae which are rotation and translation invariant. After the initialization, a GA-based search procedure is then launched and tries to identify the optimal or near optimal global matching by generating hypothetical alignment configurations based on the GA's stochastic rules. In order to improve the matching efficiency, a local improvement operator is designed and hybridized with the GA search to fine-tune the solutions within the population. The fitness of solutions is computed using a reliable and discriminating matching function which is defined by combining the globally matched minutiae pairs with the result of the minutiae's local feature similarity based on the product rule. Thus, solving the fingerprint matching problem is equivalent to finding the maximum number of matched minutiae points with the maximum average local feature similarity.

The organization of the paper is as follows. In Section II, we formulate fingerprint verification as a minutiae point pattern matching problem and provide a review of previous minutiae point pattern matching methods. Then, in Section III, we present the details of our proposed algorithm—MFMA. Section IV describes the data sets employed in this work and this is followed by a discussion of the parameter settings of the MFMA. In the experiments described in Section V, we assess the performance of the MFMA. We conclude the paper with some final remarks and an outline of future directions in Section VI.

## II. PROBLEM FORMULATION AND RELATED WORK

Automatic fingerprint verification has been approached using a variety of different strategies, as reported in the literature [34]. Most existing fingerprint verification systems are based around a minutiae point pattern matching operation. This is a natural approach since a fingerprint's minutiae embody much of its individuality. In this section, we first formulate fingerprint verification as a minutiae point pattern matching problem and then provide a review of previous minutiae point pattern matching methods.

### A. Problem Formulation

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  $(x, y, \alpha)$ , where  $(x, y)$  are the pixel coordinates of the minutia with respect to the image frame and  $\alpha$  is the orientation of the minutia, which is defined as the angle that the ridge associated with the minutia makes with the horizontal axis [29]. Typically, ridge orientations are restricted to the range  $[0, \pi)$ , so that directions  $\theta$  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 minutia can be easily interchanged by acquisition noise or pressure differences during acquisition. However, the orientation remains the same when this occurs. By representing the minutiae as point patterns, fingerprint verification can be viewed as a minutiae point pattern matching problem. 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  $t_x$ ,  $t_y$ ,  $\theta$ , and  $s$ , where  $t_x$  and  $t_y$  are the translations along the  $x$  and  $y$  directions, respectively,  $\theta$  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:

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

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.

### B. Related Work

Due to the large number of possible translations, rotations, and scalings, aligning two minutiae point patterns is an extremely difficult problem. A number of algorithms 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.

Jain *et al.* [22], [23] use ridge information associated with minutiae as an aid for alignment. Minor modifications of this matching algorithm have been suggested by other researchers [11], [19], [26], [31]. The methods proposed in [12], [18], [24], and [25] use local structural features among several minutiae close to each other for alignment. Chen *et al.* [12] defined a feature vector which describes the relationship between a minutia and its neighbors circled within a radius  $r$ . Jiang and Yau [25] and Jea and Govindaraju [24] used features derived from minutia triplets. He *et al.* [18] built a minutia simplex that contains a pair of minutiae as well as their associated textures. These minutiae local feature representations may not be robust due to their reliance on the interdependency of minutiae, which can be missed or erroneously detected by a minutia extraction algorithm. The methods proposed by Hrechack [21], Wahab *et al.* [46], Kovács-Vajna [27], Germain *et al.* [16], as well as Tan [42] also use groups of minutiae to define local structural

features. These local structural features are directly used for verification or identification, which is performed based on the pairs of corresponding local structures that are found between a query fingerprint and a template fingerprint or template fingerprint database. However, the local structural feature is less distinct because it is determined only by a small subset of the minutiae. Fingerprints from different fingers may have many similar local structures and fingerprints from the same finger may only have a few similar structures due to the presence of spurious minutiae and the absence of genuine minutiae. Therefore, fingerprint matching/identification based only on local structural features is less reliable.

The approach presented in [41], [44], and [45] is based on information sampled around minutiae for alignment. Tico and Kuosmannen [44] built a minutia descriptor for each minutia, which consists of the original minutia point and a set of ridge orientation information. Similar to the minutia descriptor, Qi *et al.* [41] defined a feature vector for each minutia by integration of ridge orientations. Tong *et al.* [45] proposed an adjacent feature vector which consists of four adjacent relative orientations and six ridge counts of a minutia. In contrast to the local structural features proposed in [12], [18], [24], and [25], the representations proposed in these methods are independent of any other minutia detected in the fingerprint. Hence, they could be more robust to the erroneous outcomes of the minutia detection algorithm (i.e., missing and spurious minutiae). Since core points of fingerprints are common, they can also be used as an aid for fingerprint alignment. Zhang [48] and Chan *et al.* [10] have explored this possibility. However, it is impossible to always guarantee locating the core point precisely, and sometimes the core point cannot be detected at all due to poor image quality or only a partial finger image being obtained via the sensor.

The above 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 [19], [22], [23], [31].

Another alternative approach is to find a transformation in order to reduce the difference between two minutiae sets after alignment. That is to say, this alternative approach aims 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.* [50] have proposed a method which aligns two sets of minutiae based on multiple pairs of reference minutiae. This method is highly dependent 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 [28], [43] as well as for general point pattern matching [1], [2], [49]. Tan and Bhanu [43] proposed the use of a traditional GA for fingerprint matching. Le *et al.* [28] employed the technique of fuzzy evolutionary programming to match two sets of minutiae. Ansari *et al.* [2], Agrawal *et al.* [1] and Zhang *et al.* [49] presented GA-based methods for the general point pattern matching problem. 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 are not well suited to fine-tuning the search in complex search spaces.

### III. MEMETIC FINGERPRINT MATCHING ALGORITHM

In this paper, we develop a memetic fingerprint matching algorithm which aims to identify the optimal or near optimal global matching between two minutiae sets. In contrast to previous minutiae point pattern matching methods, our proposed algorithm combines the use of a global search via a GA with a local improvement operator. Additionally, we introduce an efficient local matching operation for population initialization by examining local features of minutiae which are rotation and translation invariant. Finally, we define a reliable and discriminating matching function to compute the fitness of individual solutions by combining the globally matched minutiae pairs with the result of the minutiae's local feature similarity based on the product rule.

The proposed algorithm can be divided into two stages: population initialization and hybrid evolution. The initialization stage aims efficiently to produce an initial population of local alignment configurations, which generally can satisfactorily align regions adjacent to the reference local structures. Using the initial population, GA-based hybrid evolution is then launched and tries to identify the optimal or near optimal global matching. The hybrid evolutionary procedure consists of selecting parents for reproduction, performing a modified arithmetic crossover with the parents, applying mutation to the offspring, running the local improvement operator on each offspring, and carrying out competition replacement. The evolution is terminated when one of the following two stopping criteria is met: 1) the fitness value of the best population individual has not changed for  $N$  generations or 2) the fitness value of the best population exceeds a certain threshold value, which means the two fingerprints are verified as having been from the same finger. The output of the algorithm is the best solution encountered during the evolution. The flow of the algorithm is as follows:

---

#### Algorithm 1. A memetic fingerprint matching algorithm.

---

- Step 1) Apply a local matching operation (see Section III-B) to initialize  $P$  sets of chromosomes, which encode alignment configurations, based on a real-parameter representation (see Section III-A).
- Step 2) Calculate the fitness value for each individual chromosome in the initial population (see Section III-E).

- Step 3) Repeat the following substeps (a)–(e) until a stopping criterion is met.
- Select the pairing individuals based on the  $k$ -fold tournament selection method [17]. This procedure is repeated until  $P/2$  parent pairs are selected.
  - Generate intermediate offspring by applying a modified arithmetic crossover and then perform Gaussian mutation on the offspring (see Section III-C).
  - Run the local improvement operator on the offspring (see Section III-D).
  - Calculate the fitness value for each of the offspring according to the method described in Section III-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.

In the following sections, we describe in more detail how the solutions are initially created, how they evolve during the optimization process, how the local improvement operator works, and how to measure the fitness of the solutions.

#### A. Representation

Our representation of the individual solution  $i = (t_{x,i}, t_{y,i}, \theta_i, s_i)$  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. In previous work based on using evolutionary algorithms for point pattern matching applications, the binary representation [17] is commonly used. However, in most cases, a binary representation suffers from the problems of redundancy and context insensitivity with traditional crossover and mutation operators [14]. In our approach, a real parameter representation is applied. In this context, Michalewicz [37] showed that a real parameter representation moves the problem closer to the problem representation and offers higher precision with more consistent results across replications.

#### B. Local Structure Matching and Population Initialization

The local feature of a minutia describes the characteristics of the minutia in its neighborhood. This information can be used to find potential matches in another minutiae set when the local feature is distinctive. Here, we introduce an efficient local matching operator to produce an initial population by examining local features of minutiae which are rotation and translation invariant. The resulting initial population can help reduce the search space for recovering the optimal or near optimal global transformation.

The minutiae local feature representation proposed by Tico and Kuosmanen [44] has been used for designing the local matching operator. In this representation, 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. The minutia descriptor is invariant to rotation and translation [44] and, hence, it can characterize the minutia location with respect to the fingerprint pattern regardless of the position and orientation of the finger on the acquisition sensor. Due to the fact that the orientation field is a relatively stable feature of fingerprint images, the minutia descriptor can be robust in regards to image quality and local changes in images. Let  $a = \{\alpha_{k,l}\}$  and  $b = \{\beta_{k,l}\}$  be two minutia descriptors. The similarity between  $a$  and  $b$  is computed as

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

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.

Using the minutia descriptor, we introduce a local matching operator as follows to produce an initial population. Suppose the two minutiae sets  $P_1$  and  $P_2$  to be matched are composed of  $m$  and  $n$  points ( $m \leq n$ ), respectively. The operator selects a random minutia descriptor from  $P_1$  and compares it to each minutia descriptor in  $P_2$ . The value of the similarity between the two minutia descriptors serves as a clue for identifying potential corresponding pairs. The minutia pair which generates the largest similarity value is considered to be correspondent. The transformation is calculated by translating and rotating the query fingerprint to align this correspondence and then using it to initialize an individual solution. The scaling factor is constructed by random assignment of a real number to the last attribute. The initial value is constrained to be in the predefined range  $\delta s$  but is otherwise random. Each individual in the population is constructed in this way, but by selecting a minutia descriptor in  $P_1$  that has not been considered before. In the case that  $m$  is less than the number of individuals within the population to be initialized, translation and rotation parameters of the remaining individuals are constructed within the predefined ranges of  $\delta t_x$ ,  $\delta t_y$ ,  $\delta \theta$ , respectively, in the same way as the scaling factor. All of these range values are determined empirically from the experimental data sets. Their values, however, are not critical for the initialization since the population is usually dominated by the solutions that are constructed by the local matching operator.

After each individual has been initialized, a population of local alignment parameters, which generally can satisfactorily align regions adjacent to the reference minutia descriptors, is obtained. Note that this population does not contain all appropriate transformations and that inclusion in this population does not necessarily indicate an appropriate transformation since the local matching phase for initialization does not consider global minutiae information, and is based only on the similarity of minutia descriptors. However, even if a population contains many inappropriate transformations (which lead to poor or wrong alignment in areas distant to the reference minutia descriptors), it is not expected to have a detrimental effect on the proposed evolutionary-based algorithm. This is because the

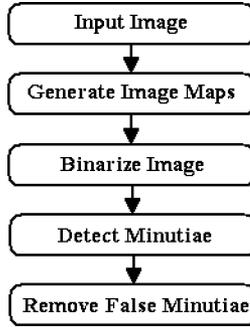


Fig. 1. Minutiae detection process described in [15].

inappropriate transformations will be unlikely to be selected for reproduction in the next generation. Using this initial population, a hybrid evolutionary procedure is then launched which tries to identify the optimal or near optimal global matching between the two minutiae sets. This local structure-based population initialization, together with the global structure-based evolution, is likely to provide a more solid basis for reliable fingerprint matching.

### C. Crossover and Mutation

Crossover is a probabilistic process that exchanges information between a pair of parents to generate two offspring. The arithmetic crossover technique [17] has been used as the reproduction operator in the MFMA. Traditional arithmetic crossover linearly combines two parent chromosomes to produce two new offspring according to the following equations:

$$\text{Offspring1} = a * \text{Parent1} + (1 - a) * \text{Parent2}, \quad (2)$$

$$\text{Offspring2} = (1 - a) * \text{Parent1} + a * \text{Parent2} \quad (3)$$

where  $a \in [0, 1)$  is a random weighting factor. In our approach, we apply either (2) or (3) randomly to produce only one single offspring. This helps to save time in processing the other similar offspring.

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 lower or upper bounds of that attribute.

### D. Local Improvement Operation

GAs can 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 [17], and this results in exhibiting a large execution time. To improve this situation, incorporation of local searches into the regeneration step of GAs is essential. GAs, which have been hybridized with local search techniques, are often called memetic algorithms (MAs) [35], [38], [39] and have been shown to be very effective for many combinatorial optimization problems [3], [8], [30], [36], [47]. In this subsection, we present a local improvement operator to effectively design an MA for the minutiae point pattern matching problem.

The local improvement operator is inspired by the iterated closest point (ICP) algorithm [6]. The ICP is a widely used heuristic for the alignment of 3-D geometric models. It utilizes

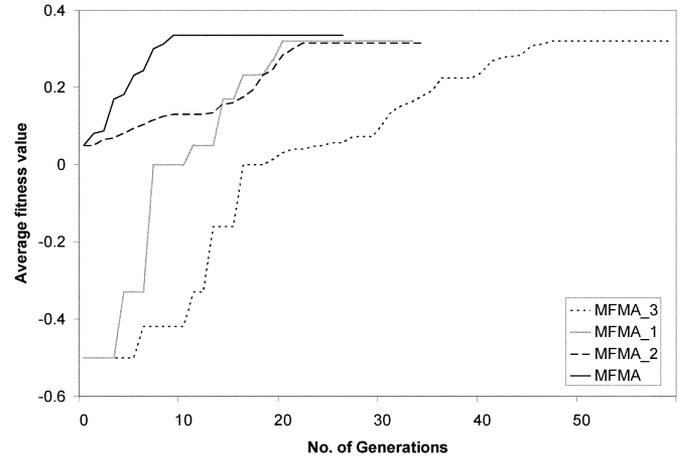


Fig. 2. Average fitness values of the best two individuals versus number of generations of the four algorithms corresponding to typical genuine matching.

the nearest-neighbor 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 [13], but it is sensitive to its initial rotations and translations, and susceptible to local optima. Its performance degenerates quickly with outliers (point features existing in one pointset that have no corresponding points in the other), which is common in the fingerprint minutiae point pattern matching problem.

In order to improve 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 re-generation step. This is achieved by first extracting the transformation information encoded in an offspring and applying the transformation to the query minutiae set. After that, we compute the closest point pairs between the two minutiae sets and collect corresponding pairs using the geometric constraints. (Note that by employing the geometric constraints, we are also trying to exclude the outliers for pairing). Finally, we calculate the new transformation and update the offspring. The operator is summarized as follows:

---

Algorithm 2. A local improvement operator.

---

- Step 1) Extract the transformation information encoded in an offspring. 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  $\Delta d$  and 2) the angular difference between their directions is less than a certain tolerance  $\Delta \theta$ . 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 offspring with the new transformation.

### E. Fitness Computation

The fitness of an individual indicates the degree of suitability of the solution it represents. A traditional way to calculate the matching scores for a minutiae point pattern matching system is  $n^2 / (\text{size}_t \times \text{size}_q)$ , where  $\text{size}_t$  and  $\text{size}_q$  represent the numbers of minutiae in template and query minutiae sets, respectively, and  $n$  is the number of matched minutiae in both sets. However, Bazen and Gerez [5] claimed that using the alternative  $2n / (\text{size}_q + \text{size}_t)$  to compute the matching scores will give better results. In our observation, we found that neither matching function may be reliable enough for fingerprint verification, especially when matching minutiae sets are extracted from low-quality fingerprint images. Here, we define a reliable and discriminating matching function by incorporating the similarity based on a local feature (i.e., minutia descriptor) to compute the fitness of individual solutions.

The local features of minutiae contain useful discriminatory information. Additionally, local features can tolerate some deformation since they are formed from only a small area of the fingerprint. Thus, the local feature similarity 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

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

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 by using the same geometric constraints as those described in Section III-D.

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 (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.

### IV. DATA SETS AND PARAMETER SETTINGS

In this section, we provide a description of the data sets used in the experiments. This is followed by a description of the implementation parameter settings for the proposed algorithm. The proposed algorithm has been evaluated using the publicly available FVC2002 database [33], which consists of four collections (labeled DB<sub>1</sub>, DB<sub>2</sub>, DB<sub>3</sub>, and DB<sub>4</sub>, respectively) of fingerprint images. These fingerprint images were collected using three different scanners and the SFinGE synthetic generator [9] and contain a wide variety of image qualities. Each of the four data collections comprises  $8 \times 100$  fingerprint images. The details of the fingerprint image data-collection process can be found in [33].

The minutiae information of the data sets is derived by using the method described in [15]. Fig. 1 lists the functional steps of this method. It first performs image enhancement to improve the contrast of ridges against valleys and then generates the image-quality maps by determining the directional flow of ridges in the image and detecting regions of low contrast, low ridge flow, and high curvature. Next, a binary representation of the fingerprint is constructed by applying a rotated grid on the ridge flows of the fingerprint. Following this, minutiae are generated by comparing each pixel neighborhood with a family of minutiae templates. Finally, a series of heuristic rules is used to merge and filter out the spurious minutiae.

The MFMA has a number of 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 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 offers the best results. The order of tournament selection controls how quickly a population is “taken over” by dominant individuals in the population [4]. 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  $N$  or  $p$  may lead to a longer running time, but without significant improvement of the matching performance. The ranges of  $\delta t_x$ ,  $\delta t_y$ ,  $\delta \theta$ , and  $\delta s$  are set to be  $[-150, 150]$ ,  $[-150, 150]$ ,  $[-0.6, 0.6]$  and  $[0.9, 1.1]$ , respectively, but their chosen values are not critical for our population initialization. The threshold of  $F_n$  for fitness computation is set to be four. The minutiae matching threshold values of  $\Delta d$  and  $\Delta \theta$  are set as 15 and 0.195, respectively, which were established in [40].

### V. EXPERIMENTS

In this section, we report a series of experiments performed over the FVC2002 database. We initially examine the significance of local matching and local improvement operations within the MFMA. Subsequently, we evaluate the MFMA with differing matching functions for fitness computation. Finally, we compare the performance of the MFMA with some previous matching methods. All of the results reported in this section were obtained with simulations on a PC with an Intel<sup>TM</sup> Pentium<sup>TM</sup> 4 processor at 2.1 GHz running under WindowsXP Professional.

First, experiments were conducted to examine the significance of local matching and local improvement operations within the MFMA. For this purpose, we examined and compared the MFMA with three variants—MFMA without local matching operation (MFMA\_1), MFMA without local improvement operation (MFMA\_2), and MFMA without either of the above operations (MFMA\_3). In the cases of MFMA\_1 and MFMA\_3, individuals within the initial population are randomly initialized using the method described in Section III-B.

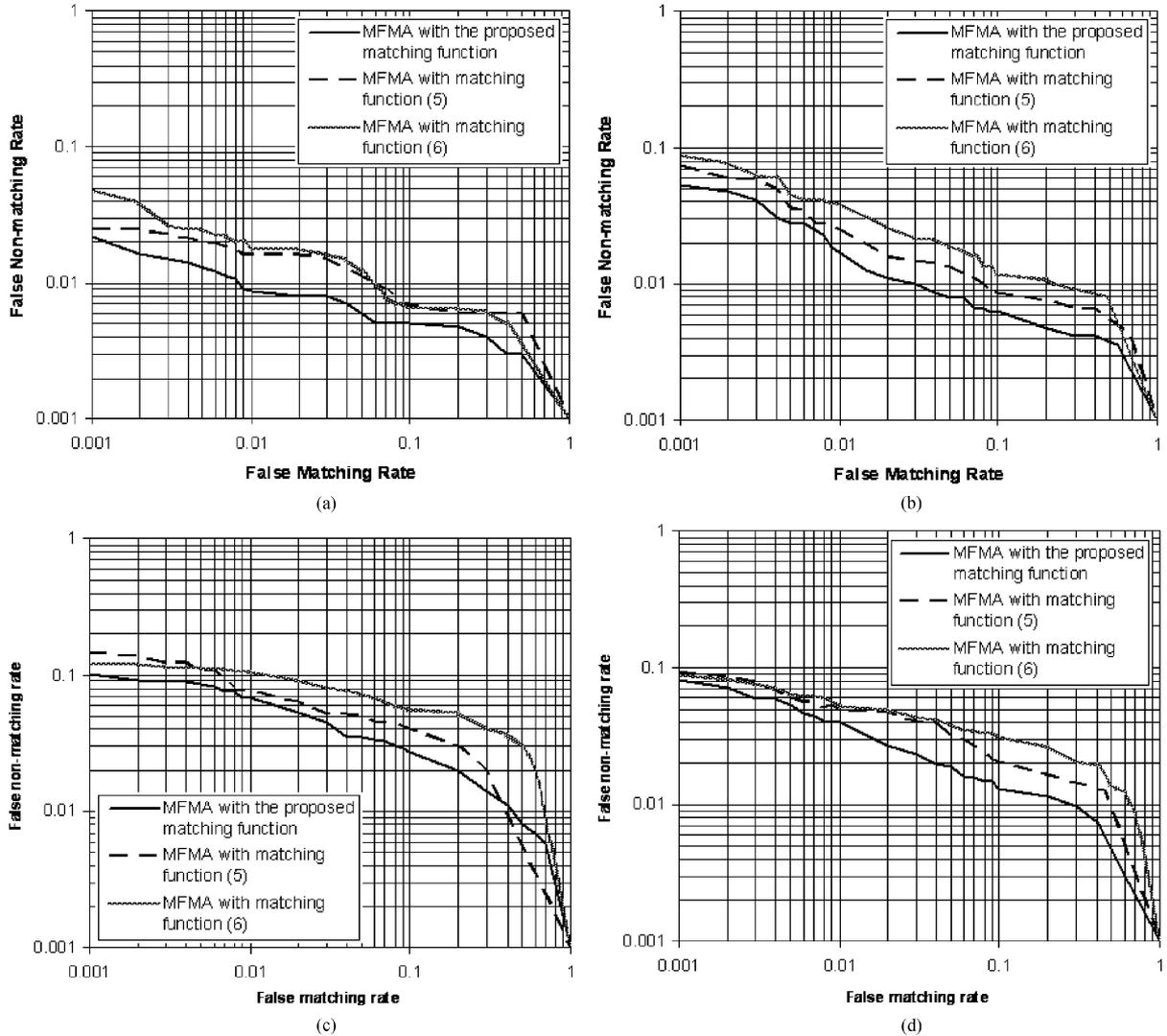


Fig. 3. ROC curves estimated on (a)  $DB_1$ , (b)  $DB_2$ , (c)  $DB_3$ , and (d)  $DB_4$  when different matching functions are used for fitness computation.

These algorithms were compared based on the same parameter settings. In order to investigate the convergence properties, the terminal condition for all four algorithms is that the fitness value of the best population individual has not changed for  $N$  generations with a relatively large parameter value  $N = 20$ .

Fig. 2 shows the fitness scores over generations corresponding to typical genuine matching. It can be observed that the convergence of MFMA\_3 is slow and each operation improves the convergence of the algorithm in different ways. MFMA\_2 shows that the local matching operation helps identify promising transformations. However, the algorithm becomes slow after the population initialization. MFMA\_1 shows that, compared with MFMA\_3, the local improvement operation speeds up the convergence during the evolution. However, it takes a significant number of generations to identify the promising transformations. By incorporating both operations, MFMA can significantly reduce the number of generations needed to identify the optimal or near optimal global alignment. In fact, this is the main reason for using these operations in the MFMA.

Next, we report experiments to evaluate the proposed matching function by comparing the MFMA with two alternative traditional matching functions

$$\frac{n^2}{(\text{size}_t \times \text{size}_q)} \quad (5)$$

and

$$\frac{2n}{(\text{size}_q + \text{size}_t)} \quad (6)$$

for fitness computation. We followed the experimental protocol proposed in [33] to evaluate the algorithms. Fig. 3 shows the receiver operating characteristic (ROC) curves obtained over the four data sets when using different matching functions for fitness computation. These results show that the MFMA with the proposed matching function outperforms the MFMA with traditional matching functions (5) and (6). The only difference among the three algorithms consists of the matching function used for fitness calculation. Consequently, the results reveal that the proposed matching function, by combining the globally

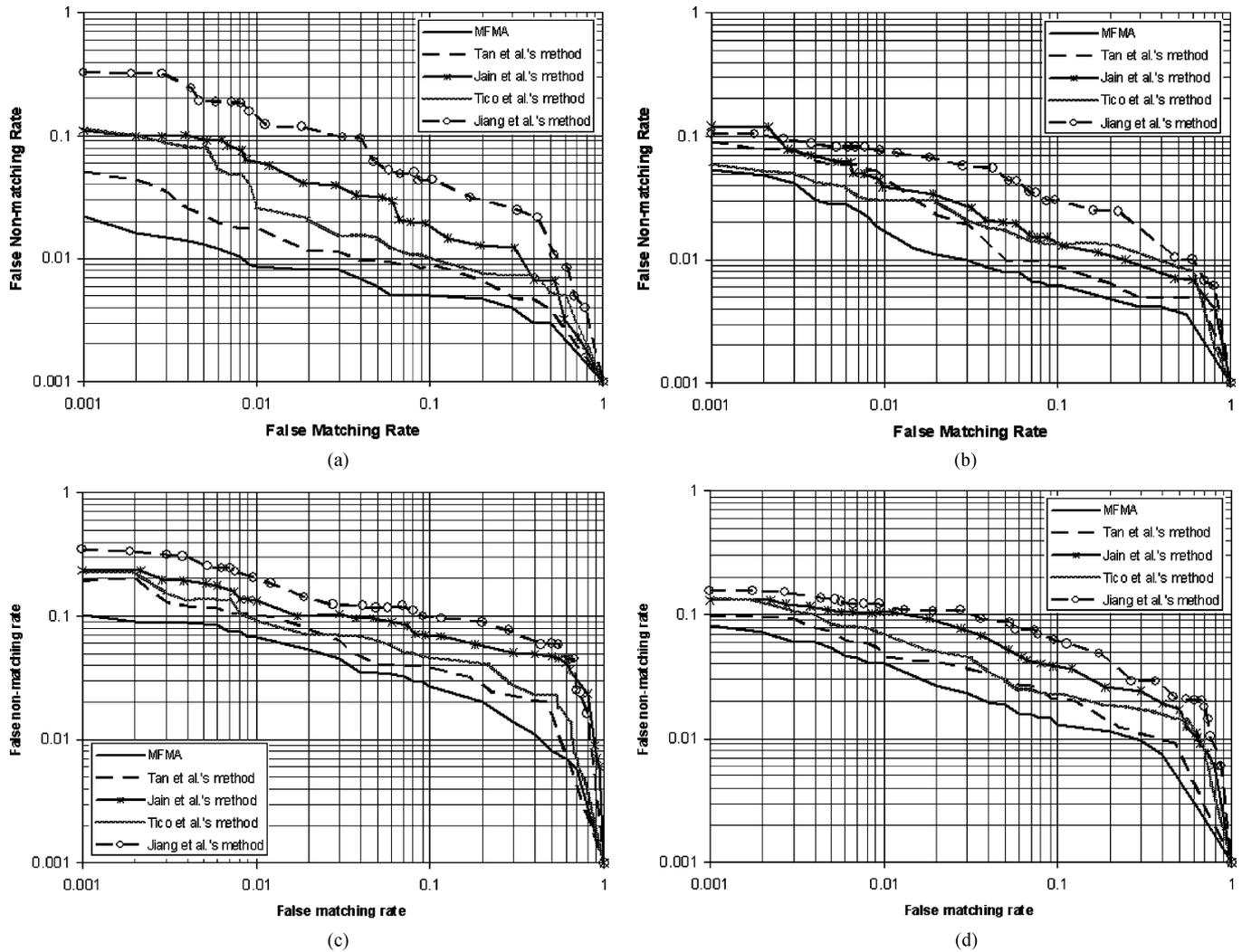


Fig. 4. ROC curves estimated on (a)  $DB_1$ , (b)  $DB_2$ , (c)  $DB_3$ , and (d)  $DB_4$  for Tan and Bhanu's method, Jain *et al.*'s method, Tico and Kuosmanen's method, Jiang and Yau's method and our proposed MFMA.

matched minutiae pairs with the result of the minutiae's local feature similarity based on the product rule, is able to more reliably decide whether the two input fingerprint impressions have been captured from the same finger.

Finally, we conduct a set of experiments to compare our algorithm with some previously reported work, a traditional genetic-algorithm-based method [43] and three single-reference-based methods [22], [25], [44]. Before discussing the comparative results, we first briefly describe the methods to be compared and their implementation details. The GA-based fingerprint matching method is recently proposed by Tan and Bhanu [43]. 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. This method also aims to find the optimal or near optimal global matching between two minutiae sets. For experiments on the FVC2002 database, the range of the parameters  $t_x$ ,  $t_y$ ,  $\theta$ , and  $s$  of the method is set to be identical to the  $\delta t_x$ ,  $\delta t_y$ ,  $\delta\theta$ , and  $\delta s$ , respectively, of the MFMA and all other parameter values are set to be the same as in [43].

The three single-reference-based methods [22], [25], [44] have been selected for comparison; each of which uses a different strategy to align fingerprints. Jain *et al.*'s method proposed in [22] is based on aligning ridges, Jiang and Yau's method [25] is based on the local structural features among several minutiae, and Tico and Kuosmanen's method is based on the minutia descriptor. For experiments on the FVC2002 database, parameter values of the three methods are specified or chosen according to the original papers with the best performance. The minutiae matching threshold values of  $\Delta d$  and  $\Delta\theta$ , which are not specified in Tico and Kuosmanen's method, are taken to be identical to those used in the MFMA. All five fingerprint matching algorithms are implemented using the C++ language. To make the comparisons more meaningful, the same minutiae extraction method as described in Section IV is used for all tested algorithms.

We compare the performance of the five methods with respect to the criteria of the matching accuracy and efficiency. Therefore, we report the ROC, equal error rate (EER), and average matching time estimated using the performance evaluation method proposed in [33] over the four data sets. The EER,

TABLE I  
EERS AND AVERAGE MATCHING TIMES ESTIMATED ON THE FOUR DATA SETS FOR FIVE DIFFERENT METHODS  
(TAN AND BHANU'S METHOD MFMA, JAIN'S METHOD, TICO'S METHOD, AND JIANG'S METHOD)

Methods	DB <sub>1</sub>		DB <sub>2</sub>		DB <sub>3</sub>		DB <sub>4</sub>	
	EER (%)	Matching time (s)						
<b>MFMA</b>	0.9	3.59	1.4	3.67	3.5	3.71	2.4	3.61
<b>Tan et al.'s method [43]</b>	1.5	7.94	2.1	8.12	4.2	8.46	3.4	7.95
<b>Jain et al.'s method [22]</b>	3.5	1.26	2.8	1.24	7.7	1.25	5.1	1.24
<b>Tico et al.'s method [44]</b>	2.2	1.27	2.4	1.28	5.7	1.29	3.8	1.26
<b>Jiang et al.'s method [25]</b>	5.4	1.24	4.6	1.22	9.8	1.23	7.2	1.24

which is commonly used to summarize the accuracy performance of a matching system, is defined as the error rate where the system's false match rate is equal to its false nonmatch rate. Fig. 4 shows the estimated ROC curves. Table I lists the results in terms of the EER values and average matching time.

Fig. 4 and Table I clearly show that, in comparison with Tan and Bhanu's method [43], the MFMA is able to achieve more accurate matching results in all four data sets. For Tan and Bhanu's method, the EER of the ROC turned out to be 1.5%, 2.1%, 4.2%, and 3.4% on the four data sets, respectively. By comparison, in our algorithm, the EER values are around 0.9%, 1.4%, 3.5%, and 2.4%, respectively. More important, matching operations for our algorithm are faster than Tan and Bhanu's method. For example, Tan and Bhanu's method needs 7.94 s on DB<sub>1</sub>, while our method takes about 3.59 s on average. The improvement in matching efficiency of our proposed algorithm is mainly due to the use of local matching and local improvement operations. In comparison with the single reference-based methods, the results reveal an interesting tradeoff between the matching accuracy and efficiency. The three single-reference-based methods are computationally more efficient, but our algorithm offers better matching accuracy performance, which can be observed from the ROC curves in Fig. 4 and EER values in Table I. The EERs, for example, on DB<sub>1</sub> of Jain *et al.*'s method, Tico and Kuosmanen's method and Jiang and Yau's method are 3.5%, 2.2%, and 5.4%, respectively, while our algorithm shows about 0.9%. It is also noted that both global alignment methods (MFMA and Tan and Bhanu's method) generally outperform the three single-reference-based methods in terms of matching accuracy. This result can be an indication that global alignment is able to recover the transformation between two fingerprint impressions more accurately than single reference-based alignment and, hence, global alignment-based methods could perform more accurate matching.

## VI. CONCLUSION

In this paper, we have reported on the design and implementation of a MFMA which seeks to identify the optimal or near optimal global matching between two minutiae sets. A GA-based evolution with a local matching startup routine and local improvement hybridization is at the heart of our proposed approach. 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

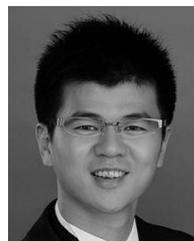
key aspect of the proposed algorithm is the use of a reliable matching function for fitness computation. To evaluate the proposed algorithm, we have conducted a series of experiments on the FVC2002 database and made comparisons with other reported work. Experimental results clearly show the effectiveness of the local matching and local improvement operations within the proposed algorithm. The results confirm that the MFMA is a reliable and practical matching algorithm. The algorithm can achieve accurate matching results faster than the traditional GA-based global fingerprint matching method, and is also more accurate in performing the matching operation than the single-reference-based methods implemented for comparison.

There are several directions in which the work may be extended further. Genetic algorithms are extremely easy to adapt to parallel computing and clustering environments [17]. As massively parallel computers become more common, parallel implementations of the proposed algorithm can be exploited. Each processor can be devoted to a pair of solutions because the algorithm's operations focus on them. As a result, the matching efficiency can be further improved as the entire population is processed in parallel. Additionally, there are fundamental limits to the accuracy that can be achieved when using rigid transformations. Therefore, another productive direction in which to extend the research is to include an operation that models nonrigid fingerprint deformations during the hybrid evolution search. This could be accomplished by incorporating, for example, the thin-plate spline model [5], which can model elastic distortions based on the locations and orientations of the matched minutiae pairs. Finally, the algorithm developed here can be suitably modified and tailored so that it is applicable to ridge pattern-based fingerprint alignment and/or matching, which is another challenging problem. In this regard, population initialization and local improvement operations based on the local ridge information may also be developed.

## REFERENCES

- [1] Agrawal, N. Ansari, and E. Hou, "Evolutionary programming for fast and robust point pattern matching," in *Proc. IEEE World Congr. Computational Intelligence*, 1994, pp. 1777–1782.
- [2] N. Ansari, M. Chen, and E. Hou, "A genetic algorithm for point pattern matching," in *Dynamic, Genetic, and Chaotic Programming*, B. Soucek, Ed. New York: Wiley, 1992, pp. 353–371.
- [3] S. Areibi and Z. Yang, "Effective memetic algorithms for VLSI design automation = genetic algorithms + local search + multi-level clustering," *Evol. Comput.*, vol. 12, no. 3, pp. 327–353, 2004.

- [4] T. Bäck and F. Kursawe, "Evolutionary algorithms for fuzzy logic: A brief overview," *Proc. Inf. Process. Management of Uncertainty in Knowledge-Based Systems*, pp. 659–664, 1994.
- [5] M. Bazen and S. H. Gerez, "Fingerprint matching by thin-plate spline modeling of elastic deformations," *Pattern Recognit.*, vol. 36, pp. 1859–1867, 2003.
- [6] P. J. Besl and N. D. McKay, "A method for registration of 3D shapes," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 14, no. 2, pp. 239–256, Feb. 1992.
- [7] B. Bhanu and X. Tan, *Computational Algorithms for Fingerprint Recognition*. Norwell, MA: Kluwer, 2003.
- [8] J. Branke, M. Middendorf, and F. Schneider, "Improved heuristics and a genetic algorithm for finding short supersequences," *OR Spektr.*, vol. 20, no. 1, pp. 39–45, 1998.
- [9] R. Cappelli, D. Maio, and D. Maltoni, "Synthetic fingerprint-database generation," presented at the 16th ICPR, Quebec City, QC, Canada, 2002.
- [10] K. C. Chan, Y. S. Moon, and P. S. Cheng, "Fast fingerprint verification using sub-regions of fingerprint images," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 14, no. 1, pp. 95–101, Jan. 2004.
- [11] S. Chen and A. K. Jain, "A fingerprint matching algorithm using dynamic programming," Tech. Rep. Dept. Comput. Sci. Eng., Michigan State Univ., East Lansing, MI, 2002.
- [12] X. Chen, J. Tian, and X. Yang, "A new algorithm for distorted fingerprints matching based on normalized fuzzy similarity measure," *IEEE Trans. Image Process.*, vol. 15, no. 3, pp. 767–776, Mar. 2006.
- [13] H. Chui and A. Rangarajan, "A new point matching algorithm for non-rigid registration," *Comput. Vis. Image Und.*, vol. 89, pp. 114–141, 2003.
- [14] E. Falkenauer, *Genetic Algorithms and Grouping Problems*. New York: Wiley, 1998.
- [15] M. D. Garris, R. M. McCabe, C. I. Watson, and C. L. Wilson, *User's Guide to NIST Fingerprint Image Software (NFIS)*, Nat. Inst. Standards Technol., Gaithersburg, MD, 2001, NISTIR 6813.
- [16] R. S. Germain, A. Califano, and S. Colville, "Fingerprint matching using transformation parameter clustering," in *Proc. IEEE Computational Science and Engineering*, 1997, vol. 4, no. 4, pp. 42–49.
- [17] E. Goldberg, *Genetic Algorithms in Search, Optimization, and Machine Learning*. Reading, MA: Addison-Wesley, 1989.
- [18] Y. He, J. Tian, L. Li, H. Chen, and X. Yang, "Fingerprint matching based on global comprehensive similarity," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 28, no. 6, pp. 850–862, Jun. 2006.
- [19] Y. He, J. Tian, X. Luo, and T. Zhang, "Image enhancement and minutiae matching in fingerprint verification," *Pattern Recognit. Lett.*, vol. 24, pp. 1349–1360, 2003.
- [20] J. H. Holland, *Adaptation in Natural and Artificial Systems*. Ann Arbor, MI: Univ. of Michigan Press, 1975.
- [21] K. Hrechack and J. A. McHugh, "Automatic fingerprint recognition using structural matching," *Pattern Recognit.*, vol. 23, no. 8, pp. 893–904, 1990.
- [22] K. Jain, L. Hong, and R. Bolle, "On-line fingerprint verification," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 19, no. 4, pp. 302–314, Apr. 1997.
- [23] K. Jain, L. Hong, S. Pankanti, and R. Bolle, "An identity-authentication system using fingerprints," *Proc. IEEE*, vol. 85, no. 9, pp. 1365–1388, Sep. 1997.
- [24] T. Y. Jea and V. Govindaraju, "A minutia-based partial fingerprint recognition system," *Pattern Recognit.*, vol. 38, pp. 1672–1684, 2004.
- [25] X. Jiang and W. Yau, "Fingerprint minutiae matching based on the local and global structures," in *Proc. 15th Int. Conf. Pattern Recognition*, 2000, pp. 1038–41.
- [26] Y. Jie, F. Yuan, R. Zhang, and Q. Song, "Fingerprint minutiae matching algorithm for real time system," *Pattern Recognit.*, vol. 39, pp. 143–146, 2006.
- [27] Z. M. Kovács-Vajna, "A fingerprint verification system based on triangular matching and dynamic time warping," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 22, no. 11, pp. 1266–1276, Nov. 2000.
- [28] T. V. Le, K. Y. Cheung, and M. H. Nguyen, "A fingerprint recognizer using fuzzy evolutionary programming," presented at the 34th Int. Conf. System Sciences, HI, 2001.
- [29] H. C. Lee and R. E. Gaensslen, Eds., *Advances in Fingerprint Technology*. New York, Elsevier, 1991.
- [30] F. Li, R. Morgan, and D. Williams, "Hybrid genetic approaches to ramping rate constrained dynamic economic dispatch," *Elect. Power Syst. Res.*, vol. 43, no. 2, pp. 97–103, 1997.
- [31] X. Luo, J. Tian, and Y. Wu, "A minutia matching algorithm in fingerprint verification," in *Proc. IEEE Int. Conf. Pattern Recognition*, 2000, pp. 4833–4836.
- [32] Maio, D. Maltoni, R. Cappelli, J. L. Wayman, and A. K. Jain, "FVC2000: Fingerprint verification competition," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 24, no. 3, pp. 402–412, Mar. 2002.
- [33] Maio, D. Maltoni, R. Cappelli, J. L. Wayman, and A. K. Jain, "FVC2002: Second fingerprint verification competition," in *Proc. Int. Conf. Pattern Recognition*, 2002, pp. 811–814.
- [34] D. Maltoni, D. Maio, A. K. Jain, and S. Prabhakar, *Handbook of Fingerprint Recognition*. Berlin, Germany: Springer Verlag, 2003.
- [35] P. Merz, "Advanced fitness landscape analysis and the performance of memetic algorithms," *Evol. Comput.*, vol. 12, no. 3, pp. 303–325, 2004.
- [36] P. Merz and B. Freisleben, "Memetic algorithms and the fitness landscape of the graph bi-partitioning problem," *Lecture Notes in Computer Science*, pp. 765–774, 1998.
- [37] Z. Michalewicz, *Genetic Algorithms + Data Structure = Evolution Programs*, 3rd ed. Berlin, Germany: Springer-Verlag, 1996.
- [38] P. Moscato, "On evolution, search, optimization, genetic algorithms and martial arts: toward memetic algorithms," Tech. Rep., Calif. Inst. Technol., Pasadena, CA, 1989.
- [39] P. Moscato, "Memetic algorithms: A short introduction," in *New Ideas in Optimization*, D. Corne, M. Dorigo, and F. Glover, Eds. New York: McGraw-Hill, 1999, pp. 219–234.
- [40] S. Pankanti, S. Prabhakar, and A. K. Jain, "On the individuality of fingerprints," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 24, no. 8, pp. 1010–1025, Aug. 2002.
- [41] J. Qi, Z. Shi, X. Zhao, and Y. Wang, "A robust fingerprint matching method," in *Proc. 7th IEEE Workshops Application of Computer Vision*, 2005, pp. 105–110.
- [42] X. Tan and B. Bhanu, "A robust two step approach for fingerprint identification," *Pattern Recognit. Lett.*, vol. 24, no. 13, pp. 2127–2134, 2003.
- [43] X. Tan and B. Bhanu, "Fingerprint matching by genetic algorithms," *Pattern Recognit.*, vol. 39, no. 3, pp. 465–477, 2006.
- [44] M. Tico and P. Kuosmanen, "Fingerprint matching using an orientation-based minutia descriptor," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 25, no. 8, pp. 1009–1014, Aug. 2003.
- [45] X. Tong, J. Huang, X. Tang, and D. Shi, "Fingerprint minutiae matching using the adjacent feature vector," *Pattern Recognit. Lett.*, vol. 26, no. 9, pp. 1337–1345, 2005.
- [46] Wahab, S. H. Chin, and E. C. Tan, "Novel approach to automated fingerprint recognition," *Proc. Inst. Elect. Eng., Vis. Image Signal Process.*, vol. 145, no. 3, pp. 160–166, 1998.
- [47] D. Whitley, "Modeling hybrid genetic algorithms," in *Genetic Algorithms in Engineering and Computer Science*, G. Winter, J. Periaux, M. Galan, and P. Cuesta, Eds. New York: Wiley, 1995, pp. 191–201.
- [48] W. Zhang and Y. Wang, "Core-based structure matching algorithm of fingerprint verification," in *Proc. ICPR*, 2002, pp. 70–74.
- [49] L. H. Zhang, W. L. Xu, and C. Chang, "Genetic algorithm for affine point pattern matching," *Pattern Recognit. Lett.*, vol. 24, pp. 9–19, 2003.
- [50] Zhu, J. P. Yin, and G. M. Zhang, "Fingerprint matching based on global alignment of multiple reference minutiae," *Pattern Recognit.*, vol. 38, no. 10, pp. 1685–1694, 2005.
- [51] *The science of fingerprints: Classification and uses*. Washington, DC, Federal Bureau of Investigation, 1984.



**Weiguo Sheng** received the M.Sc. degree in information technology from the University of Nottingham, Nottingham, U.K., in 2002 and the Ph.D. degree in computer science from Brunel University, Brunel, U.K., in 2005.

Currently, he is a Research Associate in the Department of Electronics at the University of Kent, Kent, U.K. His research interests include evolutionary computations, data clustering, and a heuristic search with a particular focus on biometrics and bioinformatics.



**Gareth Howells** received the Ph.D. degree in computer science from the University of Kent, Kent, U.K., in 1991.

Currently, he is a Lecturer in Electronic Engineering at the University of Kent, Kent, U.K. He was subsequently involved in substantial research into the formal verification of digital systems and the practical development and application of artificial neural systems. This work resulted in the development of a wholly declarative theorem proving tool for digital system verification and the development of a generic formal mathematical model describing the behavior of artificial neural systems. His research interests have also involved the practical application of formal mathematical logic to the design both of artificial neural networks and more general pattern classification techniques. Over the last decade, his research interests have moved into the biometric security field and he has been responsible for the design of a generalized document security system and substantial work on the investigation of techniques for the direct generation of encryption keys from biometric samples, removing the need for any form of biometric template storage.



**Michael Fairhurst** is Head of the Department of Electronics at the University of Kent, Kent, U.K. His research interests focus on computational architectures and algorithms for image analysis and classification, and applications including handwritten text reading and document processing, medical image analysis and, especially, security and biometrics. His current projects include work on multimodal biometrics, on assessing the vulnerability of biometric identification techniques, and on the analysis of handwriting, both for identification

purposes and to improve the effectiveness of automated processing for forensic applications. Biometric processing also underpins work which investigates document encryption linked to biometric data. He is also further developing work he pioneered at the University of Kent, which has established novel techniques for the assessment and monitoring of neurological conditions through the analysis of patients' writing and drawing abilities.

Professor Fairhurst sits on numerous conference, workshop, and government-sponsored committees, and on the editorial boards of several international journals. He has published many papers in scientific literature and is an elected Fellow of the International Association for Pattern Recognition (IAPR) in recognition of his contributions to the field.



**Farzin Deravi** (M'87) received the B.A. degree in engineering and economics from the University of Oxford, Oxford, U.K., in 1981, the M.Sc. degree in electronic engineering from Imperial College, University of London, London, U.K., in 1982, and the Ph.D. degree from the University of Wales, Swansea, U.K.

From 1983 to 1987, he was a Research Assistant with the University of Wales. In 1987, he joined the academic staff at the University of Wales, where he was active in teaching and research in the Department of Electrical and Electronic Engineering. In 1998, he joined the Department of Electronics at the University of Kent, Kent, U.K., where he is a Reader in information engineering. His current research interests include texture recognition, fractal coding, integrated audio-visual processing, and biometric systems.

Dr. Deravi is a member of the Institution of Engineering and Technology and the British Machine Vision Association. He was the Founding Chair of the IET Professional Network on Visual Information Engineering and is Editor-in-Chief of *IET Image Processing*. He serves on BSI, CEN, and ISO committees on biometric standardization.