

Combining minutiae descriptors for fingerprint matching

Jianjiang Feng*

Department of Computing, The Hong Kong Polytechnic University, Hung Hom, Kowloon, Hong Kong

Received 7 October 2006; received in revised form 15 March 2007; accepted 24 April 2007

Abstract

A novel minutiae-based fingerprint matching algorithm is proposed. A minutiae matching algorithm has to solve two problems: correspondence and similarity computation. For the correspondence problem, we assign each minutia two descriptors: texture-based and minutiae-based descriptors, and use an alignment-based greedy matching algorithm to establish the correspondences between minutiae. For the similarity computation, we extract a 17-D feature vector from the matching result, and convert the feature vector into a matching score using support vector classifier. The proposed algorithm is tested on FVC2002 databases and compared to all participants in FVC2002. According to equal error rate, the proposed algorithm ranks 1st on DB3, the most difficult database in FVC2002, and on the average ranks 2nd on all 4 databases. © 2007 Pattern Recognition Society. Published by Elsevier Ltd. All rights reserved.

Keywords: Descriptor; Minutiae matching; Matching score; Alignment; Fingerprint

1. Introduction

Fingerprint recognition has been widely used in both forensic and civilian applications. Compared with other biometrics features, fingerprint-based biometrics is the most proven technique and has the largest market shares. Although fingerprint recognition has been studied for many years and much progress has been made, the performance of even state-of-the-art matchers is still much lower than the expectations of people and theory estimation [1]. Therefore, much effort is still needed to improve both the performance and the speed of fingerprint recognition systems. The matching algorithm plays a key role in a fingerprint recognition system. In this paper, a novel fingerprint matching algorithm is proposed.

A fingerprint is the pattern of ridges and valleys on the surface of a fingertip. The endpoints and crossing points of ridges are called minutiae. It is a widely accepted assumption that the minutiae pattern of each finger is unique and does not change during one's life. When human fingerprint experts determine if two fingerprints are from the same finger, the matching degree between two minutiae pattern is one of the

most important factors. Thanks to the similarity to the way of human fingerprint experts and compactness of templates, the minutiae-based matching method is the most widely studied matching method. The algorithm proposed in this paper belongs to the minutiae-based matching method.

A minutiae-based matching algorithm has to solve two problems: correspondence and similarity computation. In the following two subsections, we review related work and outline our approach to each problem.

1.1. Correspondence problem

Most minutiae extraction algorithms attach a minutia with 3 features: x , y coordinates and direction. Since the relative transformation between two fingerprints is unknown in advance, the correspondence between minutiae is very ambiguous and each minutia of one fingerprint can be matched to any minutiae of the other fingerprint.

To reduce the ambiguity, additional distinguishing information can be attached to a minutia. Such additional information is generally called a descriptor. Descriptor-based matching and recognition have received a lot of interest in recent years. Shape context in Ref. [2] is a shape descriptor, which reflects the space distribution of other sampling points around sampling points. SIFT in Ref. [3] is a region descriptor, which captures the

* Tel.: +86 755 26035385.

E-mail address: fjianjiang@263.net.

gradient information around interesting points. An extensive performance evaluation of descriptors is performed in Ref. [4]. In the fingerprint recognition literature, a lot of minutiae matching algorithms fall into this category [5–11]. Among these algorithms, three types of information are most widely used, that is ridge [7,9], orientation [5,11] and minutia [6,8,10]. In our opinion, ridge-based descriptors are less discriminating than the other types, mainly because only one associated ridge is used in Refs. [7,9].

In this work, we propose two descriptors: texture-based descriptor, which captures the orientation and frequency information around a minutia, and minutiae-based descriptor, which reflects relationships between a minutia and nearby minutiae. Both descriptors are combined to further increase the distinctiveness of minutiae. We have conducted experiments to evaluate the performance of minutiae descriptors, and the results show that the combined descriptor is much more discriminating than any single descriptor.

Based on the proposed descriptors, an alignment-based greedy matching algorithm similar to [5] is used to establish the correspondences between minutiae. Experimental results show that, due to the discriminating ability of the proposed descriptors, such a simple matching algorithm is enough to establish correct correspondences.

1.2. Similarity computation

Based on the correspondences between minutiae, a matching score is computed to represent the similarity degree between two fingerprints. The matching score is generally normalized to [0, 1].

Most existing algorithms compute matching scores using manually designed formulas. There are two widely used formulas: the percentage of matched minutiae [7,12] and the percentage of matched minutiae in the common region [13]. The shortcoming of the first formula is that genuine matching score (GMS) may be very low due to missing minutiae, spurious minutiae, or small overlap region. The shortcoming of the second formula is that GMS may be low due to missing minutiae, spurious minutiae, and imposter matching score (IMS) may be very high when the number of minutiae in common region is small. The above shortcomings make it very difficult to distinguish genuine matches with imposter matches by using only the basic formulas. To solve this problem, some researchers modified the basic formulas by replacing the number of matched minutiae with the total similarity between matched minutiae descriptors [5]. Some researchers combine the similarity between minutiae pattern and that between orientation images [11], or that between ridge features [14].

Since it is difficult to distinguish genuine matches with imposter matches using only one or two features, in this work, we extract a 17-D feature vector from the matching result, which reflects the matching degree of minutiae, orientation image, frequency image, associated ridges of minutiae, and singular points. For such a high dimensional feature vector, it is quite difficult to design a suitable similarity formula

manually. Therefore, we resort to statistical method. By regarding fingerprint matching as a two-category classification problem (genuine match and imposter match), we train support vector classifier (SVC) and use it to compute similarity scores. Jea and Govindaraju [15] have used a similar method to compute matching scores. They extract six features from the matching result and compute matching scores by using neural network. Since their feature vectors only reflect the matching degree of minutiae, the discriminating ability of their feature vectors is less powerful than ours.

The proposed algorithm is tested on FVC2002 databases [16] and compared to all participants in FVC2002. According to equal error rate, the proposed algorithm ranks 1st on DB3, the most difficult database in FVC2002, and on the average ranks 2nd on all 4 databases.

The rest of the paper is organized as follows. In Section 2, feature extraction is described, including ridge and minutiae extraction, minutiae classification and feature images generation. In Section 3, texture-based and minutiae-based descriptors are proposed. In Section 4, the minutiae matching algorithm is presented. In Section 5, computation of matching score is discussed. Experimental results and analysis are given in Section 6. Finally in Section 7, we conclude the paper and suggest the future directions.

2. Feature extraction

In this section, we discuss how to extract the features used by the matching algorithm. First we give a brief description of the ridge and minutiae extraction algorithm. Then we discuss how to classify minutiae.

2.1. Ridge and minutiae extraction

Given a gray-scale fingerprint image, a series of steps, including orientation image computation [17], frequency image computation [7], directional filtering [17], local threshold-based binarization and thinning [18], are used to produce a skeleton image. From the skeleton image, minutiae are detected and ridges are extracted by tracing. Each minutia has four features: x coordinate, y coordinate, direction and type (termination or bifurcation). Ridges are represented as lists of points. To simplify the representation of ridges, ridges associated with a bifurcation are treated as three ridges and a closed ridge is broken at a randomly selected point. Singular points are extracted using an improved version of the Poincaré index method in Ref. [19]. Each singular point has four features: x coordinate, y coordinate, direction (only defined for core), and type (core or delta).

2.2. Minutiae classification

The purpose of minutiae classification is to classify a minutia as one of the three types: reliable minutia, unreliable minutia or spurious minutia. Most existing post-processing algorithms [20,21] classify a minutia as two types: genuine minutia or

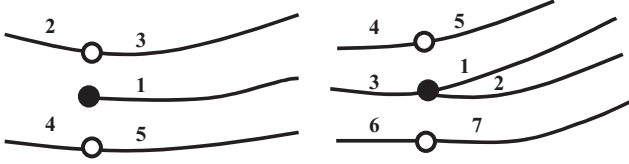


Fig. 1. Ridges to examine for a termination and a bifurcation.

spurious minutiae. Different from these algorithms, we classify genuine minutiae as reliable minutiae and unreliable minutiae further. Both reliable and unreliable minutiae will be used in the matching procedure. The difference between them is that, if unmatched, a reliable minutia will be punished more than an unreliable minutia (see more details in Sections 3.2 and 5).

To classify a minutia as genuine or spurious one, we adopt the rules in Refs. [20,21] to identify various types of spurious minutiae, such as blur, break, border, short ridge, bridge. To classify a genuine minutia as reliable or unreliable one, we use a strict condition that, for a reliable minutia, all associated ridges and surrounding ridges should be long enough (longer than 12 pixels in our experiments). For a termination, the lengths of five ridges are examined. For a bifurcation, the lengths of seven ridges are examined. The configurations of these ridges of termination and bifurcation are shown in Fig. 1.

3. Descriptors

In this section, we present texture-based descriptors, minutiae-based descriptors and the combination of them.

3.1. Texture-based descriptors

Texture-based descriptors consist of ridge orientation and frequency information at some sampling points around a minutia, and the sampling points are defined by using the same sampling structure as Ref. [5]. These sampling points are located on L circles centered at the minutia. Assume that the radius of the l th circle is r_l and there are K_l sampling points distributed equally on the l th circle (see Fig. 2). Using the minutia as origin and the direction of the minutia as the positive direction of x axis of polar coordinate system, the coordinate of the k th sampling point on the l th circle is defined as

$$\begin{cases} \rho_{l,k} = r_l, \\ \theta_{l,k} = \frac{2\pi k}{K_l}. \end{cases}$$

In our experiments, one set of parameters suggested in Ref. [5] is adopted, which is $((r_0 = 27, K_0 = 10), (r_1 = 45, K_1 = 16), (r_2 = 63, K_2 = 22), (r_3 = 81, K_3 = 28))$.

Assume that the angle of the minutia is α ($-\pi \leq \alpha < \pi$), the orientation at the sampling point is $\alpha_{l,k}$ ($-\pi/2 \leq \alpha_{l,k} < \pi/2$), and the frequency is $1/w_{l,k}$. The relative orientation of $\alpha_{l,k}$ with respect to α is computed as $\beta_{l,k} = \lambda(\alpha_{l,k} - \alpha)$, where

$$\lambda(\alpha) = \begin{cases} \alpha - \pi & \text{if } \alpha \geq \pi/2, \\ \alpha + \pi & \text{if } \alpha < -\pi/2, \\ \alpha & \text{otherwise.} \end{cases}$$

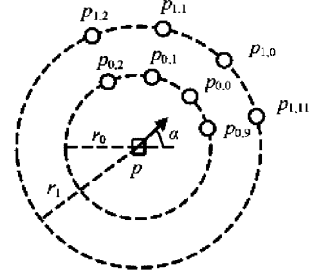


Fig. 2. Distribution of sampling points.

The texture-based descriptor of the minutia is represented as

$$D_o(p) = \{(\beta_{l,k}, w_{l,k})\}_{k=0}^{K_l-1}\}_{l=0}^{L-1},$$

which can be regarded as the combination of orientation-based descriptor $D_o(p)$ and frequency-based descriptor $D_f(p)$.

$$D_o(p) = \{(\beta_{l,k})\}_{k=0}^{K_l-1}\}_{l=0}^{L-1},$$

$$D_f(p) = \{(w_{l,k})\}_{k=0}^{K_l-1}\}_{l=0}^{L-1}.$$

Note that if a sampling point is located outside fingerprint, it is called an invalid sampling point and an invalid value (such as 1000) is assigned to the orientation and frequency of the point.

Let $\{(\alpha_{l,k}, v_{l,k})\}$ and $\{(\beta_{l,k}, w_{l,k})\}$ denote the texture-based descriptors of two minutiae p and q , the similarity between them is computed as the mean value of the similarity of texture between valid corresponding sampling points

$$S_t(p, q) = \text{mean}(s_t(p_{l,k}, q_{l,k})),$$

where both $p_{l,k}$ and $q_{l,k}$ are valid sampling points. It should be noted that, if the proportion of valid corresponding sampling points is less than 25%, the mean value is not reliable, so we set the similarity between two descriptors to 0.

To compute the similarity of texture between two sampling points, we first compute the similarity of orientation and the similarity of frequency as follows:

$$s_o(p_{l,k}, q_{l,k}) = e^{-|\lambda(\alpha_{l,k} - \beta_{l,k})|/(\pi/16)},$$

$$s_f(p_{l,k}, q_{l,k}) = e^{-|v_{l,k} - w_{l,k}|/3}.$$

Then the similarity of texture is defined as

$$s_t(p_{l,k}, q_{l,k}) = w \cdot s_o(p_{l,k}, q_{l,k}) + (1 - w) \cdot s_f(p_{l,k}, q_{l,k}),$$

where w is set to 0.5 in our experiments.

From the expressions above, we obtain

$$S_t(p, q) = w \cdot S_o(p, q) + (1 - w) \cdot S_f(p, q),$$

$$S_o(p, q) = \text{mean}(s_o(p_{l,k}, q_{l,k})),$$

$$S_f(p, q) = \text{mean}(s_f(p_{l,k}, q_{l,k})),$$

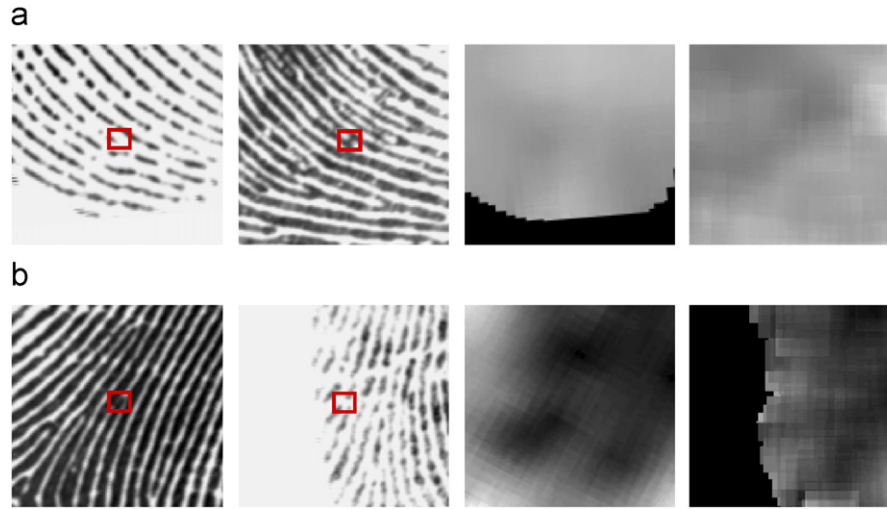


Fig. 3. The grayscale images and frequency images around two pairs of unmatched minutiae are aligned with respect to the central minutiae. Black regions in the frequency images represent background. Bright pixels in frequency images correspond to low frequencies. For the two minutiae in (a), the orientation images are similar, but the frequency images (3rd and 4th images on the first row) are dissimilar (the left one is brighter). For the two minutiae in (b), the orientation images are dissimilar, but the frequency images (3rd and 4th images on the second row) are similar.

where $S_o(p, q)$ and $S_f(p, q)$ represent the similarity of orientation-based descriptors and that of frequency-based descriptors, respectively.

In Fig. 3, the nearby regions of two pairs of unmatched minutiae are shown. If only orientation information is considered, the similarity degree between the two minutiae in (a) is high. If only frequency information is considered, the similarity degree between the two minutiae in (b) is high. However, if both orientation and frequency information are considered, these two pairs of minutiae can be easily inferred as unmatched. These two examples show that combining orientation-based and frequency-based descriptors can increase the discriminating ability of descriptors.

3.2. Minutiae-based descriptors

Local minutiae structures have been used by many researchers to increase the distinctiveness of minutiae. Two types of representation, fixed-length feature vectors and unfixed-length feature vectors, have been adopted by different researchers to describe local minutiae structures. In this paper, local minutiae structures are termed minutiae-based descriptors.

Fixed-length minutiae-based descriptors are used in Refs. [8,15,22,23]. Hrechak and McHugh [23] define an eight-dimensional feature vector, each dimension of which represents the number of certain type of minutiae in the nearby region of the central minutia. The problem with this descriptor is that it is difficult to detect various types of minutiae. Wahab et al. [22], Jiang and Yau [8], Jea and Govindaraju [15] sort the m nearest neighboring minutiae with respect to distance or angle in the polar coordinate system defined by the central minutia, and form the feature vector by relationships between neighboring minutiae and the central minutia. A problem with these

descriptors is that the order of neighboring minutiae is sensitive to many factors, such as occlusion, missing or spurious minutiae, inaccurate position or angle of minutia. However, an advantage of the fixed-length descriptors is that the similarity between two descriptors can be computed very fast.

Unfixed-length minutiae-based descriptors are used in Refs. [6,10]. In this case, descriptors consist of all minutiae whose distances from the central minutia are less than a threshold. The similarity between two descriptors is computed by a local minutiae matching algorithm. Although the time complexity of comparing two local minutiae patterns is higher than comparing two fixed-length feature vectors, the former is more flexible than the latter in dealing with many problems, such as occlusion, missing or spurious minutiae, inaccurate positions or angles of minutiae. The differences between two descriptors in Refs. [6,10] are that ridge count information is used in Ref. [10], and adaptive bounding boxes is used in Ref. [6]. The common problems with the descriptors in Refs. [6,10] is that they have not handled occlusion problems, and they are not robust enough to missing and spurious minutiae. When overlapped region is very small or many spurious minutiae occur, using the methods in Refs. [6,10], the similarity degrees between some matched minutiae will be very low.

In this paper, a simple but robust matching algorithm is used to compute the similarity of minutiae-based descriptors between two minutiae. Given a minutia p , its minutiae-based descriptor is defined as follows. A minutia p' is called a neighboring minutia of p , if the distance between p' and p is less than a predefined threshold r (60 pixels in our experiments). Let $N(p)$ denote the set of neighboring minutiae of p , and n_p denote the size of $N(p)$. The minutiae-based descriptor of p is defined as $D_m(p) = \{(x_i, y_i, \theta_i)\}_1^{n_p}$, where (x_i, y_i, θ_i) are the x , y coordinate and angle of the i th neighboring minutia.

Let $D_m(q) = \{(x_i, y_i, \theta_i)\}_1^{n_q}$ denote the descriptor of another minutia q . The similarity between $D_m(p)$ and $D_m(q)$ is defined

as

$$S_m(p, q) = \frac{m_p + 1}{M_p + 1} \cdot \frac{m_q + 1}{M_q + 1},$$

where m_p and m_q represent the number of matching minutiae of $N(p)$ and $N(q)$, respectively, and M_p and M_q represent the number of minutiae of $N(p)$ and $N(q)$ that should be matching, respectively. All terms plus 1 means that two central minutiae p and q are regarded as matching. Here m_p and m_q are different because we do not establish one-to-one correspondence.

m_p and M_p are computed using the following method, and m_q and M_q can be computed using the similar method. Let T represent the rigid transformation from p to q . The translation and rotation parameters of T are the relative x , y coordinate between two minutiae and the relative angle between two minutiae directions, respectively. Map each minutia p_i in $N(p)$ to p'_i using T , and check if there exist at least one minutiae q_j in $N(q)$ that can be matched to p'_i . If exist, p_i is regarded as a matching minutia and a minutia that should be matching. Otherwise, the following three conditions are examined, and if none of them is satisfied, p_i is regarded as a minutia that should be matching. The three cases below respond to three typical intra-class variations of local minutiae structure.

1. p_i is an unreliable minutia. An example is shown in Fig. 4(a) to illustrate this case.
2. p'_i is located at the occluded region of template fingerprint. An example is shown in Fig. 4(b) to illustrate this case.
3. The distance between p_i and p is greater than $0.8r$. In this case, it is unstable whether p_i belongs to $N(p)$. The example in Fig. 4(c) illustrates this case.

3.3. Combined descriptors

Since texture-based descriptors and minutiae-based descriptors capture contemporary information, we further improve the discriminating ability of descriptors by combining two descriptors using the product rule, $s_c = s_t \cdot s_m$, where s_c , s_t and s_m represent the similarity of combined, texture-based and minutiae-based descriptors, respectively. In Fig. 5, the nearby regions of two pairs of unmatched minutiae are shown. If only texture information is considered, the similarity degree between the two minutiae in (a) is high. If only neighboring minutiae are considered, the similarity degree between the two minutiae in (b) is high (although visually dissimilar, they are similar when neglecting intra-class variations as described in Section 3.2). However, if both texture and neighboring minutiae are considered, these two pairs of minutiae can be easily inferred as unmatched. These two examples show that combining texture-based and minutiae-based descriptors can increase the discriminating ability of descriptors.

4. Minutiae matching

Given two minutiae sets $\{p_i\}_{i=1,\dots,N_1}$ and $\{q_j\}_{j=1,\dots,N_2}$, and similarity degree between each pair of minutiae descriptors $\{s(i, j)\}_{i=1,\dots,N_1; j=1,\dots,N_2}$, an alignment-based greedy

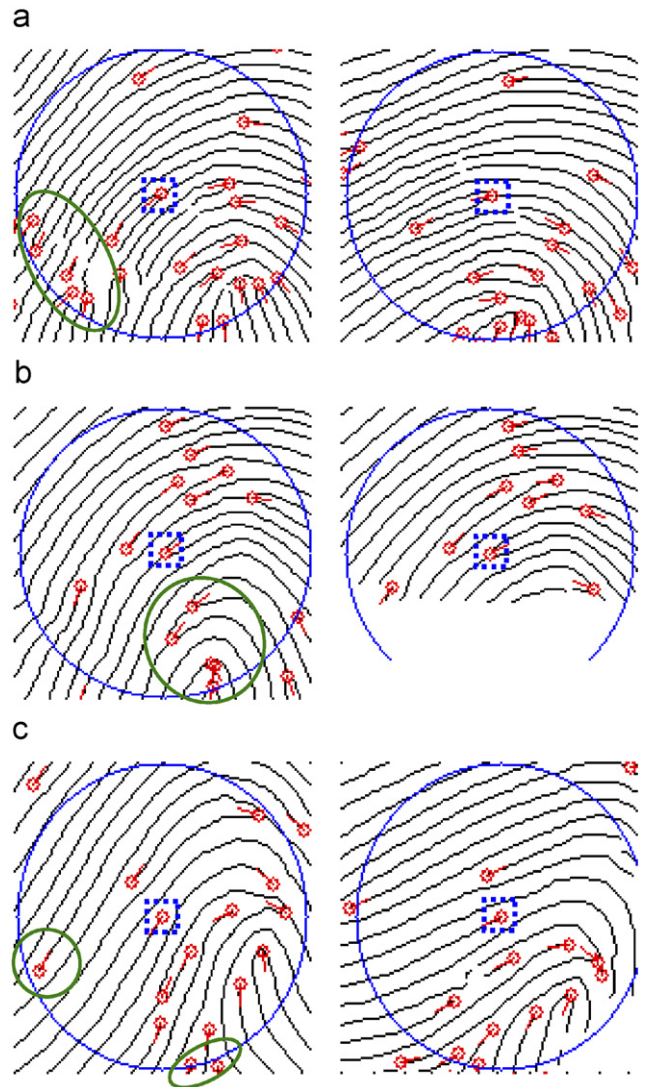


Fig. 4. Three pairs of matching minutiae and their neighboring minutiae are shown. The central minutiae are marked with blue boxes. Minutiae within blue circles are neighboring minutiae of the central minutiae. Those minutiae (marked with green circles) on the left column, which do not have matching minutiae, are not regarded as minutiae that should be matching. So the similarity degrees between three pairs of central minutiae are not affected by them. The three examples correspond to unreliable minutiae, occlusion and close to the border of circle, respectively.

matching algorithm is used to establish the correspondences between minutiae. We use a list $MP = \{(i_m, j_m)\}_{m=1,\dots,M}$ to record the correspondences, where (i_m, j_m) denotes that p_{i_m} corresponds to q_{j_m} . Starting from an empty correspondence set, the matching algorithm inserts a correspondence into MP till no more correspondences can be inserted.

To determine the order in which to insert correspondences, a normalized similarity degree s_n between two minutiae is defined based on similarity degree s :

$$s_n(i, j) = \frac{s(i, j) \cdot (N_1 + N_2 - 1)}{\sum_{k=1}^{N_1} s(k, j) + \sum_{k=1}^{N_2} s(i, k) - s(i, j)}.$$

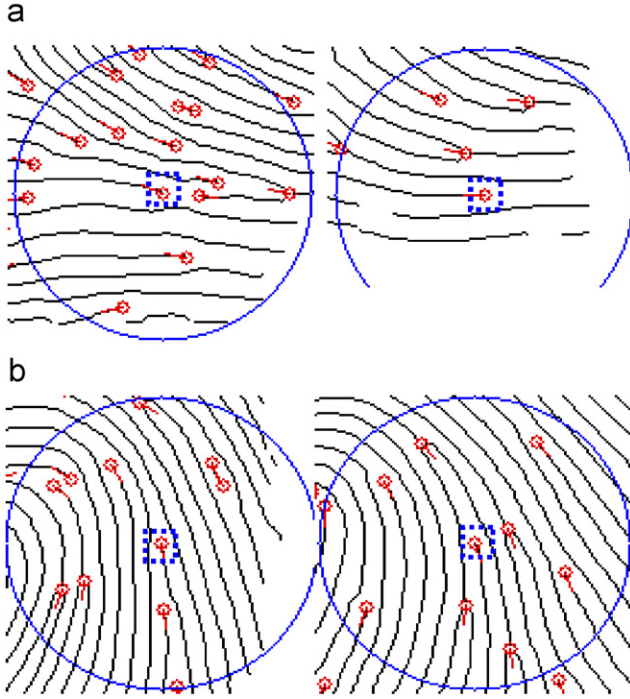


Fig. 5. The nearby regions of two pairs of unmatched minutiae are shown. For the two central minutiae in (a), the texture-based descriptors are similar, but the minutiae-based descriptors are dissimilar. For the two central minutiae in (b), the texture-based descriptors are dissimilar, but the minutiae-based descriptors are similar.

```

function GreedyMatch( $i_0, j_0$ )
  Initialize  $flag1$  and  $flag2$  with 0;
   $flag1[i_0] = 1$ ;
   $flag2[j_0] = 1$ ;
  for  $m = 1$  to  $N_1 \times N_2$ 
     $i = L(m).i$ ;  $j = L(m).j$ ;
    if ( $flag1[i]=0$ ) & ( $flag2[j]=0$ )
      & ( $p_i$  and  $q_j$  are matchable)
        Insert ( $i, j$ ) into  $MP$ ;
         $flag1[i] = 1$ ;
         $flag2[j] = 1$ ;
    endif
  endfor

```

Fig. 6. The pseudocode of greedy matching algorithm.

For the minutiae pair whose relative rotation is greater than a predefined threshold (100° in our experiments), s_n is set to 0. A list $L = \{(s_n(i, j), i, j)\}_{i=1, \dots, N_1; j=1, \dots, N_2}$ is used to store the normalized similarity degrees and indices of all minutiae pairs. Elements in L are sorted in decreasing order with respect to $s_n(i, j)$.

The first minutiae pair (i_1, j_1) in L is used as the initial minutiae pair, and two minutiae sets are aligned using the initial minutiae pair. A pair of minutiae is said to be matchable, if they

are close in position and the difference of direction is small. The pseudocode of greedy matching algorithm is given in Fig. 6. Two arrays $flag1$ and $flag2$ are used to mark minutiae that have been matched, in order that no minutia can be matched to more than one minutia.

As the initial pair is crucial to the matching algorithm, the minutiae pairs of the top N_a elements in L are used as the initial pairs, and for each of them, a matching attempt is made. Totally N_a attempts are performed, N_a scores are computed using the method in next section and the highest one is used as the matching score between two fingerprints.

5. Similarity score

Based on a set of matching minutiae pairs outputted by the matching algorithm described in Section 4, a matching score (between 0 and 1) will be computed to represent the similarity degree between two fingerprints. To this aim, a 17-D feature vector is computed from the matching results and converted into the matching score by a statistical classifier. In the following subsections, the definition of feature vector and the training procedure are described.

5.1. Feature vector

The feature vector consists of 17 elements that represent the matching status of minutiae, ridges, orientation images, frequency images and singular points of two fingerprints. Based on the initial matched minutiae pair, the rotation and translation parameters between two fingerprints are estimated. Using these parameters, for each minutia and singular point in each fingerprint, we compute the transformed minutia and singular point in another fingerprint. The orientation images and frequency images of two fingerprints are also aligned using these parameters. Because some features concern the similarity between minutiae triplets, we require that a matching result contains at least three pairs of matching minutiae. For a matching result with less than three pairs of matching minutiae, the matching score is set to 0. Each element of the feature vector is defined as follows.

The first four features have been used by many matching algorithms, but it should be noticed that our definition of the ratio of matching minutiae is different from traditional ones.

1. The number of matching minutiae pairs ($minuNum$).
2. The ratio ($minuRatio1$) of matching minutiae in the input fingerprint, which is defined as: $minuRatio1 = minuNum / M_1 = minuNum / (minuNum + n_1)$, where M_1 represents the number of the minutiae in the input fingerprint that should be matched, and n_1 represents the number of the unmatched minutiae in the input fingerprint that should be matched. To compute n_1 , we need to examine each unmatched minutia to determine whether it should be matched. If none of the following three conditions is satisfied by an unmatched minutia, it is regarded as an unmatched minutia that should be matched. (1) p_i is an unreliable minutia; (2) the mapping point of p_i is located

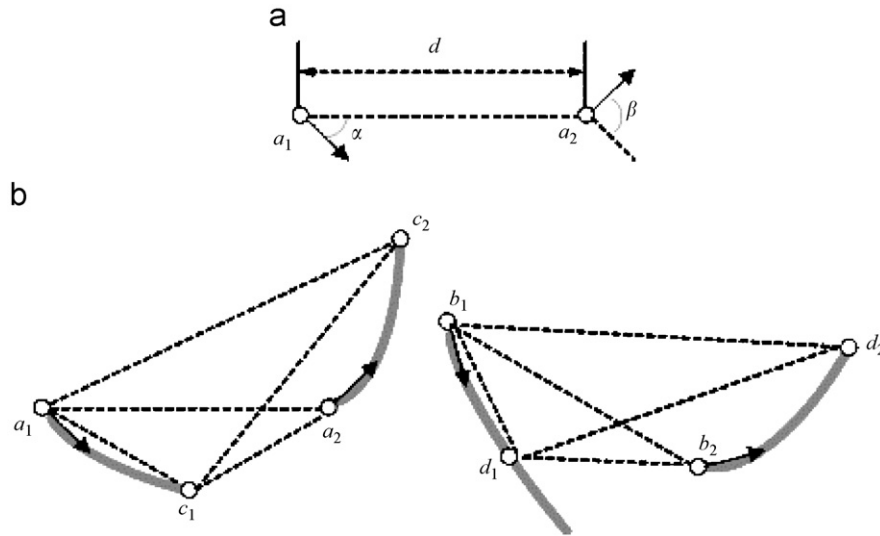


Fig. 7. Some features.

at the invalid region of the template fingerprint; (3) p_i can be matched to a minutia in the template fingerprint.

3. The ratio (*minuRatio2*) of matching minutiae in the template fingerprint, which is defined as: $\text{minuRatio2} = \text{minuNum}/M_2$, where M_2 represents the number of the minutiae in the template fingerprint that should be matched. M_2 can be computed using the same method as M_1 .
4. The mean similarity (*minuSimi*) between the descriptors of matching minutiae.

The following three features are concerned with two pairs of matching minutiae. The three attributes to compare are shown in Fig. 7(a): the distance d between two minutiae, the relative angle α between the line connecting two minutiae and one minutia, and the relative angle β between the angles of two minutiae.

5. The mean difference (*biLength*) of all corresponding d .
6. The mean difference (*biAngle1*) of all corresponding α .
7. The mean difference (*biAngle2*) of all corresponding β .

The following six features are concerned with three types of triangles. Triangles of the first type are those formed by three pairs of matching minutiae. Triangles of the second type are the triangle formed by two pairs of matching minutiae and the corresponding endpoints of the associated ridges of two minutiae ($a_1a_2c_1$ and $b_1b_2d_1$ in Fig. 7(b)). Triangles of the third type are the triangle formed by a pair of matching minutiae and the corresponding endpoints of the associated ridges of two pairs of matching minutiae ($a_1c_1c_2$ and $b_1d_1d_2$ in Fig. 7(b)). To reduce the time complexity, we only use triangles with each side longer than 16 pixels and shorter than 150 pixels.

8. The mean difference (*triArea1*) of the areas of all corresponding triangles of the first type.

9. The mean difference (*triAngle1*) of the internal angles all corresponding triangles of the first type.
10. The mean difference (*triArea2*) of the areas of all corresponding triangles of the second type.
11. The mean difference (*triAngle2*) of the internal angles all corresponding triangles of the second type.
12. The mean difference (*triArea3*) of the areas of all corresponding triangles of the third type.
13. The mean difference (*triAngle3*) of the internal angles all corresponding triangles of the third type.

The final four features are concerned with orientation images, frequency images and singular points.

14. The mean difference (*oriImage*) of aligned orientation images.
15. The mean difference (*freqImage*) of aligned frequency images.
16. The number (*matchSP*) of matching singular points. A pair of singular points is said to be matching if they are of the same type and are close in positions and directions (for core) after aligning according to the initial minutia pair.
17. The number (*missSP*) of the unmatched singular points that should be matched. We say that an unmatched singular point should be matched, if there is a singular point of the same type in the other fingerprint, or its mapping point is located at the foreground region of the other fingerprint.

5.2. Training

Fingerprint matching is regarded as a two-category classification problem. To perform training, we need to prepare a large training set that consists of lots of feature vectors of genuine matches and imposter matches. We use the matching algorithm described in Section 4 to match minutiae and compute feature vectors automatically. For a pair of fingerprints, 5 matching at-

tempts are made and 5 feature vectors are computed. Among them, the best feature vector is automatically selected. For genuine matches, the best one should correspond to correct matching result. For imposter matches, the best one should correspond to a “good” matching result. Since we do not have SVC before training SVC, the formula below is used to select the best feature vector,

$$S = \text{minuRatio1} \cdot \text{minuRatio2} \cdot \text{minuSimi}.$$

There are four databases in FVC2002. Each database has 110 fingers and each finger have 8 impressions. Each database is divided into a training set (10 fingers) and a test set (100 fingers). We have used the training set of each database to train a support vector classifier for each database. For each training set, totally $10 \times C_2^8 = 280$ genuine matches and $8 \times 8 \times C_2^{10} = 2880$ imposter matches can be performed. To balance the number of genuine match samples and that of imposter match samples, and to capture large variations in genuine matches, we generate another 9 matching results and 9 feature vectors from each genuine match.

Assume the number of matching pairs, M , of a genuine match is greater than 8. We generate 9 integral numbers $\{m_i\}$, ($2 < m_i < M$) randomly, which represent the numbers of matching pairs of 9 generated genuine matches. For the i th generated genuine matches, we randomly select m_i pairs from the matching pairs of the original genuine match. Then a new feature vector is computed based on this new matching result. When computing the feature vector, it should be noted that those original matching minutiae that have not been selected are removed from the original minutiae sets.

After the training set has been prepared, the training procedure is quite simple. We use LibSVM [24] to train a Support Vector Classifier with RBF kernel. Given a feature vector, the output of SVC is a real number in the range $[-5, 5]$, and is linearly normalized to a real number in the range $[0, 1]$.

6. Experimental results

Using FVC2002 databases, five sets of experiments are conducted to evaluate discriminating ability of descriptors, the correct rate of alignment, validity of greedy strategy, discriminating ability of each dimension of 17-D feature vector, and the overall performance of the matching algorithm.

6.1. Descriptors

The first set of experiments is aimed to evaluate the discriminating ability of the proposed descriptors. To conduct this set of experiments, we need a lot of matched minutiae and unmatched minutiae. To obtain matched minutiae, the minutiae matching algorithm proposed in Ref. [25] is used to output the matched minutiae for the 280 genuine matches in FVC2002 DB1_B, and then the matching results are checked and revised manually. Totally 2000 pairs of matched minutiae are selected. To collect unmatched minutiae, we select 2000 pairs of minutiae randomly from 20 pairs of different fingerprints. To make un-

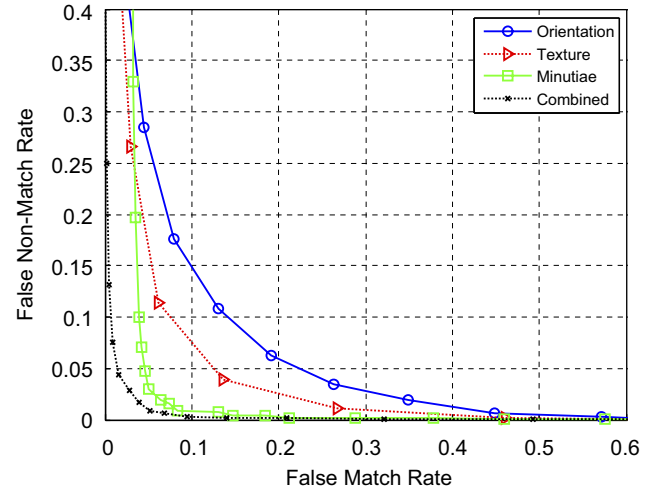


Fig. 8. The ROC curves of various minutiae descriptors, including orientation-based, texture-based, minutiae-based and combined descriptors.

matched minutiae not too easy, each pair of unmatched minutiae is required to have at least 25% common sampling points for orientation-based descriptors.

The similarity degrees of all matched minutiae and unmatched minutiae are computed. If the similarity degree between a pair of minutiae is higher than or equal to a threshold, they are inferred as a pair of matched minutiae; otherwise, they are inferred as a pair of unmatched minutiae. When the similarity degree between a pair of unmatched minutiae is higher than or equal to a threshold and inferred as a pair of matched minutiae, an error called false match occurs. When the similarity degree between a pair of matched minutiae is lower than a threshold and inferred as a pair of unmatched minutiae, an error called false non-match occurs. The ratio of false matches to all unmatched minutiae is called false match rate (FMR), and the ratio of false non-matches to all matched minutiae is called false non-match rate (FNMR). By changing the threshold, we obtain a ROC curve with false match rate as x -axis and false non-match rate as y -axis.

In Fig. 8, the ROC curves for orientation-based, texture-based, minutiae-based and combined descriptors are plotted. From this figure, we observe that texture-based descriptor is better than orientation-based one, and combined descriptor is better than both minutiae-based one and texture-based one.

6.2. Alignment

Alignment is a crucial step for the proposed algorithm, as misalignment of two fingerprints of the same finger certainly produces a false matching result. Although alignment-based algorithms are most widely adopted matching algorithm in fingerprint matching, few of these algorithms have reported the performances of alignment.

In our algorithm, two fingerprints are aligned using the top n most similar minutiae pair. If none of the n pairs is correct, a misalignment occurs. We test the texture-based, minutiae-based

Table 1

When top n minutiae pairs are outputted, the numbers of misalignment for three descriptors

	Top 1	Top 2	Top 3	Top 4
Texture	3	1	0	0
Minutiae	2	1	0	0
Combined	0	0	0	0

Table 2

The percentages of the ranks of matched pairs among candidate pairs for three types of descriptors

	1(%)	2(%)	3(%)
Texture	96.9	1.5	1.6
Minutiae	98.0	1.1	0.9
Combined	99.1	0.9	0

and combined descriptors using 280 pairs of mated fingerprints on DB1_B. Alignment result is compared to the ground truth matching results described in Section 6.1. The numbers of misalignment of three descriptors are given in Table 1. It can be concluded that alignment based on combined descriptors is very reliable.

6.3. Validity of greedy strategy

After two fingerprints are aligned, matching minutiae pairs are found according to the decreasing order of normalized similarity. Such a greedy strategy is valid only when matched minutiae pair ranks first among all candidate pairs. The candidate pairs of a matched pair (p_i, q_j) include (p_i, q_j) , $\{(p_i, q_n) | q_n \in N(q_j)\}$ and $\{(p_m, q_j) | p_m \in N(p_i)\}$, where $N(p)$ denotes the set of neighboring minutiae of minutia p . To validate the greedy strategy for different descriptors, we select 1000 pairs of matched minutiae from the ground truth matching results described in Section 6.1, and compute the ranks of matched pairs among candidate pairs. The percentages of rank 1, 2 and 3 of matched pairs for three descriptors are given in Table 2. From the table, we can observe that using combined descriptors, 99.1% of all matched pairs ranks first, so greedy strategy is valid.

6.4. Feature vector

One of the contributions of this paper is that a 17-D feature vector is proposed to describe the matching degree of fingerprints. To compare the discriminating ability of different features, we perform 280 genuine matches and 2880 imposter matches on FVC2002 DB1_B, and output corresponding feature vectors. Since the feature vector is not computed when the number of matching minutiae is less than 3, we obtain feature vectors of 279 genuine matches and 1845 imposter matches. For all these samples, each of the 17 dimensions is normalized to a real number between 0 and 1. For those dimensions that genuine matches should be smaller than imposter matches generally, such as *biLength*, *triAngle1*, *oriImage*, *missSP*, etc.,

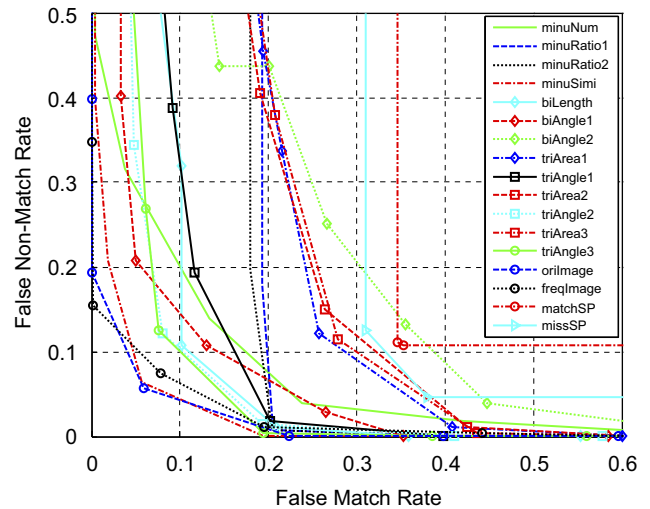


Fig. 9. ROC curves of the 17 features.

the feature x is replaced by $1 - x$. Then each feature is used to classify a sample as a genuine match or an imposter match by comparing it to a threshold. By computing FMR and FNMR under a set of thresholds, we plot the ROC curve of each of the 17 features in Fig. 9. We also sort the 17 features in the increasing order of EER (see Table 3). From Fig. 9 and Table 3, following observations can be made.

1. Because minutiae descriptors capture both texture information and nearby minutiae information, *minuSimi* produces the lowest EER.
2. Both *oriImage* and *freqImage* produce low EERs.
3. *minuNum* is the most widely used feature in the literature, but its EER is not very low. This is not surprising, since we have excluded all samples with less than 3 matching minutiae pairs, and most of them are from imposter matches.
4. *minuRatio1* and *minuRatio2* are also widely used features, but their EERs are not very low. This is not surprising, since they are always used together, not singly.
5. Among the three types of triangles, those formed by minutiae and ridges are slightly better than that formed by three minutiae.
6. *matchSP* and *missSP* can produce low FNMR when FMR is above 30%.
7. Although no any single feature can produce very low EER ($< 1\%$), in next subsection, we will show a much lower EER can be obtained by submitting all these features to a SVC.

6.5. Overall performance

The core idea of the proposed algorithm is about descriptors and scoring. We have compared the performances of three combinations of descriptors and methods of scoring on DB3_A, including orientation-based descriptors and formula (the formula is similar to that in Ref. [5]), orientation-based descriptors and SVC, and combined descriptors and SVC. The three

Table 3
EERs of 17 features (sorted in the increasing order)

	<i>minuSimi</i>	<i>oriImage</i>	<i>freqImage</i>	<i>triAngle3</i>	<i>triAngle2</i>	<i>biLength</i>	<i>biAngle1</i>	<i>triAngle1</i>	<i>minuNum</i>	<i>minuRatio2</i>	<i>minuRatio1</i>	<i>triArea1</i>	<i>triArea2</i>	<i>triArea3</i>	<i>biAngle2</i>	<i>missSP</i>	<i>matchSP</i>
EER(%)	5.6	5.7	7.7	8.9	9.2	10.8	10.8	10.8	12.5	13.3	17.9	19.3	22.8	23.9	24.4	25.1	34.6

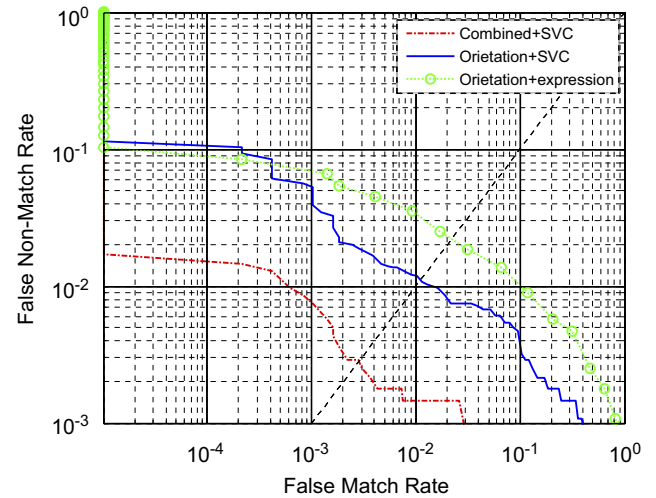


Fig. 10. ROC curves of different methods on DB3_A.

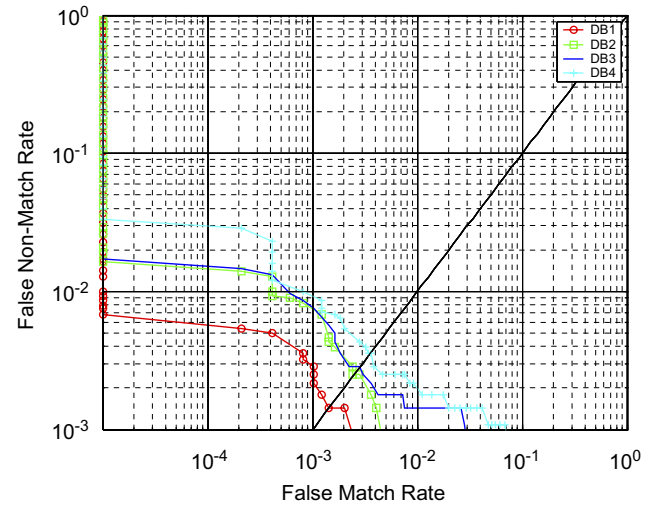


Fig. 11. ROC curves of our algorithm on four databases of FVC2002.

ROC curves are shown in Fig. 10. We can observe that SVC-based scoring is superior to formula-based one, and combined descriptor is superior to orientation-based one.

We have tested the proposed algorithm (combined descriptor and SVC-based scoring) on four test databases of FVC2002. ROC curves of our algorithm on four databases are given in Fig. 11. Equal error rates (EER) of our algorithm and the best two algorithms PA15 and PA27 on four databases are given in Table 4. These data show that our algorithm is comparative to the best ones in FVC2002.

Experiments are conducted on a PC with Pentium IV 2.1 GHz. The matching algorithm consists of two parts: comparing minutiae descriptors and the matching process. For texture-based descriptor, the average time of the first part is less than 10 ms. For minutiae-based descriptor, the average time of the first part is about 40 ms. The second part is independent of the descriptors adopted, and the average time is less than 10 ms.

Table 4
Equal error rates on four databases of two best algorithms and ours

	DB1(%)	DB2(%)	DB3(%)	DB4(%)	Average(%)
PA15	0.10	0.17	0.37	0.10	0.19
PA27	0.25	0.14	0.72	0.21	0.33
ours	0.14	0.25	0.28	0.36	0.26

7. Conclusions

A descriptor-based minutiae matching algorithm is proposed in this paper. Different from most existing matching algorithms, the proposed algorithm puts more emphasis on minutiae descriptors and the computation of matching scores. Combining texture information and neighboring minutiae, we obtain a descriptor with high discriminating ability. Based on this descriptor, a simple alignment-based greedy matching algorithm is used to establish the correspondences between minutiae. A 17-D feature vector is computed from the matching result and converted to a matching score by a support vector classifier. Experimental results on FVC2002 show that the proposed algorithm is comparative to the best algorithms in the competition.

The current algorithm can be improved along two directions. The first direction is to improve the accuracy by designing more complex matching strategy. The second direction is to speed up the algorithm, and to study how the performance changes while some information is not used, like ridges, orientation images or frequency images.

Acknowledgments

The author would like to thank the reviewer for the valuable comments and Marius Tico for the useful discussions. This work is partly supported by National Natural Science Foundation of China under grant 60472069.

References

- [1] S. Pankanti, S. Prabhakar, A.K. Jain, On the individuality of fingerprints, *IEEE Trans. Pattern Anal. Mach. Intell.* 24 (8) (2002) 1010–1025.
- [2] S. Belongie, J. Malik, J. Puzicha, Shape matching and object recognition using shape contexts, *IEEE Trans. Pattern Anal. Mach. Intell.* 24 (4) (2002) 509–522.
- [3] D.G. Lowe, Distinctive image features from scale-invariant keypoints, *Int. J. Comput. Vision* 60 (2) (2004) 91–110.
- [4] K. Mikolajczyk, C. Schmid, A performance evaluation of local descriptors, *IEEE Trans. Pattern Anal. Mach. Intell.* 27 (10) (2005) 1615–1630.
- [5] M. Tico, P. Kuosmanen, Fingerprint matching using an orientation-based minutia descriptor, *IEEE Trans. Pattern Anal. Mach. Intell.* 25 (8) (2003) 1009–1014.
- [6] X. Chen, J. Tian, X. Yang, A new algorithm for distorted fingerprints matching based on normalized fuzzy similarity measure, *IEEE Trans. Image Process.* 15 (3) (2006) 767–776.
- [7] A.K. Jain, L. Hong, R.M. Bolle, On-line fingerprint verification, *IEEE Trans. Pattern Anal. Mach. Intell.* 19 (4) (1997) 302–314.
- [8] X. Jiang, W.-Y. Yau, Fingerprint minutiae matching based on the local and global structures, in: *Proceedings of the International Conference on Pattern Recognition, Barcelona, 2000*, pp. 1038–1041.
- [9] X. Luo, J. Tian, Y. Wu, A minutiae matching algorithm in fingerprint verification, in: *Proceedings of the International Conference on Pattern Recognition, Barcelona, 2000*, pp. 833–836.
- [10] N.K. Ratha, R.M. Bolle, V.D. Pandit, V. Vaish, Robust fingerprint authentication using local structural similarity, in: *Fifth IEEE Workshop on Applications of Computer Vision, 2000*, pp. 29–34.
- [11] J. Qi, Y. Wang, A robust fingerprint matching method, *Pattern Recognition* 38 (10) (2005) 1665–1671.
- [12] A.M. Bazen, S.H. Gerez, Fingerprint matching by thin-plate spline modelling of elastic deformations, *Pattern Recognition* 36 (8) (2003) 1859–1867.
- [13] N.K. Ratha, K. Karu, S. Chen, A. Jain, Real-time matching system for large fingerprint databases, *IEEE Trans. Pattern Anal. Mach. Intell.* 18 (8) (1996) 799–813.
- [14] A. Ross, A.K. Jain, J. Reisman, A hybrid fingerprint matcher, *Pattern Recognition* 36 (7) (2003) 1661–1673.
- [15] T.Y. Jea, V. Govindaraju, A minutia-based partial fingerprint recognition system, *Pattern Recognition* 38 (10) (2005) 1672–1684.
- [16] FVC2002, Second international fingerprint verification competition, (<http://bias.csr.unibo.it/fvc2002/>).
- [17] L. O’Gorman, J.V. Nickerson, An approach to fingerprint filter design, *Pattern Recognition* 22 (1) (1989) 29–38.
- [18] Z. Guo, R.W. Hall, Parallel thinning with two-subiteration algorithms, *Commun. ACM* 32 (3) (1989) 359–373.
- [19] A.K. Jain, S. Prabhakar, L. Hong, A multichannel approach to fingerprint classification, *IEEE Trans. Pattern Anal. Mach. Intell.* 21 (4) (1999) 348–359.
- [20] Q. Xiao, H. Raafat, Fingerprint image postprocessing: combined statistical and structural approach, *Pattern Recognition* 24 (10) (1991) 985–992.
- [21] Z. Bian, D. Zhang, W. Shu, Knowledge-based fingerprint post-processing, *Int. J. Pattern Recognition Artif. Intell.* 16 (1) (2002) 53–67.
- [22] A. Wahab, S.H. Chin, E.C. Tan, Novel approach to automated fingerprint recognition, *IEEE Proc. Vision, Image and Signal Processing* 145 (3) (1998) 160–166.
- [23] A.K. Hrechak, J.A. McHugh, Automated fingerprint recognition using structural matching, *Pattern Recognition* 23 (8) (1990) 893–904.
- [24] C.-C. Chang, C.-J. Lin, LIBSVM: a library for support vector machines, (<http://www.csie.ntu.edu.tw/~cjlin/libsvm/>).
- [25] J. Feng, A. Cai, Fingerprint representation and matching in ridge coordinate system, in: *Proceedings of the International Conference on Pattern Recognition, Hong Kong, 2006*, pp. 485–488.

About the Author—JIANJIANG FENG received his B.S. degree and Ph.D. degree in Communication Engineering from the Beijing University of Posts & Telecommunication in 2000 and 2006, respectively. Currently he is a research associate with the Hong Kong Polytechnic University. His research interests include fingerprint recognition and structural matching.