Available online at www.sciencedirect.com

SCIENCE @ DIRECT®

Pattern Recognition 39 (2006) 2131–2140

PATTERN RECOGNITION
THE JOURNAL OF THE PATTERN RECOGNITION SOCIETY

www.elsevier.com/locate/patcog

Fingerprint matching using ridges

Jianjiang Feng*, Zhengyu Ouyang, Anni Cai

School of Telecommunication Engineering, Beijing University of Posts and Telecommunications, Box 113, Beijing 100876, PR China

Received 15 February 2006; accepted 11 May 2006

Abstract

Traditionally, fingerprint matching is minutia-based, which establishes the minutiae correspondences between two fingerprints. In this paper, a novel fingerprint matching algorithm is presented, which establishes both the ridge correspondences and the minutia correspondences between two fingerprints. First N initial substructure (including a minutia and adjacent ridges) pairs are found by a novel alignment method. Based on each of these substructure pairs, ridge matching is performed by incrementally matching ridges and minutiae, and then a matching score is computed. The maximum one of the N scores is used as the final matching score of two fingerprints. Preliminary results on FVC2002 databases show that ridge matching approach performs comparably with the minutia-based one.

Keywords: Fingerprint; Minutiae; Ridge matching; Curve matching; Alignment; Dynamic programming

1. Introduction

Fingerprints have been used in identification of individuals for many years because of the famous fact that each finger has a unique pattern. Many fingerprint identification and verification methods have been proposed, such as image correlation [1], graph matching [2], structural matching [3,4], and matching with transform features [5], and so on. Among them the most widely used one is methods based on point pattern matching [6,7].

However, fingerprint recognition is still a challenging problem due to the following difficulties:

(1) Low quality fingerprint images are often captured due to dry or wet skin, dirty or injured fingers, and non-uniform pressures. Feature detector operated on such images will miss genuine minutiae and introduce spurious minutiae.
(2) The solid-state sensors are increasingly used, which capture only a portion of a finger. This causes the deficiency of minutiae information. In such a case, it is difficult to make a reliable decision whether two fingerprints are from the same finger.
(3) The imaging process introduces elastic deformation in ridge pattern and minutiae locations of the fingerprint image. While large bounding box can be used during matching to tolerate it, the side-effect is that the false accept rate will increase.

In recent years, new representations of fingerprint image and new matching algorithms have been proposed to resolve the problems above. FingerCode representation and matching scheme introduced by Jain et al. [5], which captures the global and local features of fingerprints, is robust to low quality images and has an advantage of fixed-length feature vectors. Ross et al. [8] present a hybrid matcher that combines minutiae and texture features. Tico and Kuosmanen [9] introduce an orientation-based minutia descriptor to identify corresponding minutiae and compute the matching score. Kovács-Vajna [10] uses a triangular matching method to deal with the nonlinear deformation, which is based on the fact that local distortion is less than global distortion.

* Corresponding author. Tel.: +86 10 62283118x15; fax: +86 10 62285133.

E-mail addresses: fjianjiang@263.net (J. Feng), brian122@vip.sina.com (Z. Ouyang), annicai@bupt.edu.cn (A. Cai).

0031-3203/$30.00 © 2006 Pattern Recognition Society. Published by Elsevier Ltd. All rights reserved.
doi:10.1016/j.patcog.2006.05.001

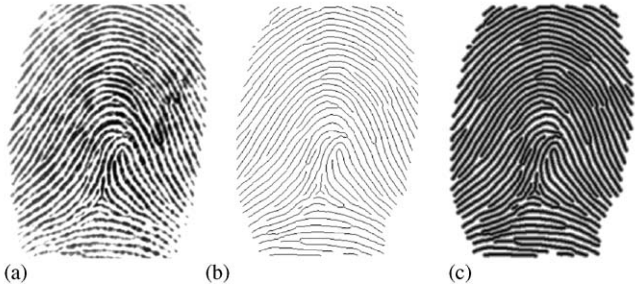


Fig. 1. (a) Intensity image, (b) ridge image, (c) synthesized image.

Bazen [11] estimates nonlinear distortion between two minutiae sets using thin-plate splines, and removes the distortion prior to the matching stage. All these methods perform reasonably well in certain circumstances; some methods do well for low quality fingerprints, some do well when nonlinear deformation exists, and others do well when the overlapped region of two fingerprint images is small.

In this paper, however, a novel approach based on ridges is proposed with the aim to try to solve all these problems. Ridge image, also called thinned image or skeleton image, is an intermediate image in many feature extraction algorithms. Since minutiae are generally thought of as enough to identify a person, ridge image is just used to extract minutiae from it. After that, the ridge image is discarded. However, we think ridge images have much more usages. In our opinion, ridge image has the following features:

(1) Ridge image is an effective representation of the fingerprint image. From a ridge image, we can synthesize an image similar to the enhanced version of the original fingerprint image. On the contrary, it is definitely impossible to do so from a minutia set. An example of a ridge image and its synthesized image is given in (Fig. 1). The synthesized image is generated using the following steps. To compute the distance transform of a ridge image (b), to replace the intensity value greater than a threshold T (e.g. 5) with T, and to scale the intensity range [0 T] to [0 255], then we can obtain a synthesized image (c) that looks like the enhanced version of the original fingerprint image (a).
(2) Ridge image is also a compact representation of the fingerprint image. Ridge images can be efficiently approximated by polygonal lines, so the size of a template file is small. We have conducted an experiment and found that for fingerprints in FVC2002 DB1 (388*374, 500 dpi), the average size of template files is 1.3 KBytes, which meets the storage requirement of "light" system in FVC2004 competition [12].
(3) Similar minutiae patterns do not mean similar ridge patterns. Actually from experiments, we observed that the ridge patterns of most different fingerprints which have similar minutiae patterns are significantly different.
(4) Unlike minutiae whose distribution on a fingerprint seems to be random, ridges cover the whole region of a fingerprint. As a result, with the reduction of the effective region of two fingerprints, the performance of the ridge-based system will not degrade dramatically.
(5) The topology information in ridge patterns is reliable (especially in the direction normal to ridges) and invariant to nonlinear distortion.

Ridges have been used for different purposes by some researchers. Ridges associated with corresponding minutiae are used to align two minutiae patterns by Jain [7]. Fingerprint classification algorithm in [13] is based on features extracted from ridge images. Ridges associated with corresponding minutiae are used to estimate the nonlinear distortion between two fingerprints [14]. Although ridges have been used in a number of aspects related to fingerprint recognition technology, to our knowledge, there have been no work published that reports matching two ridge images directly.

The algorithm proposed in this paper is novel on that it establishes both the ridge correspondences and the minutia correspondences between two fingerprints. The algorithm consists of three stages: preprocessing, alignment, and matching. In the preprocessing stage, ridges are extracted from the thinned image and sampled equidistantly, and relations between ridges and minutiae are established. In the alignment stage, a set of N initial substructure pairs is found using a novel approach. In the matching stage, for each of the N initial substructure pairs, ridge matching is performed to produce a matching score. Finally, the maximum of the N scores is used as the final matching score of the two fingerprints.

The idea underlying our alignment algorithm focuses on how to choose a reliable local feature pair as the base of matching. This is accomplished first by defining a substructure that contains as much local information (one minutia and several ridges) as possible, and secondly by finding the substructure pair which have the most consistent substructure pairs around.

In our matching algorithm, during the process of ridge matching, minutiae are also paired, and the matching score is computed according to both the matched minutiae and the matched ridges.

Many existing algorithms use a single global transformation to align two fingerprints [7,9]. Different from these algorithms, we apply different local transformations in different regions. The transformation is estimated using matched substructures, and applied to nearby ridges.

Experiments have been conducted on FVC2002 databases [15] and the preliminary results have demonstrated the validity of the proposed approach.

The rest of the paper is organized as follows. The next three sections are, respectively, devoted to the three stages of the algorithm, namely, preprocessing, alignment and matching. In Section 5, the experimental results and evaluation are presented. Finally in Section 6, summary and plans for future work are given.

2. Preprocessing

2.1. Feature extraction

Our matching algorithm is based on two types of features: minutiae and ridges. For a given grayscale fingerprint image, we obtain the ridge image (thinned image) and the minutiae set using an algorithm developed in our laboratory, which follows the common preprocessing steps in traditional minutiae matching algorithms. First the directional field is computed and the fingerprint is segmented into the foreground region and the background region. Then directional filtering is performed to improve the image quality, and the enhanced image is binarilized and thinned to produce the ridge image. At last minutiae are detected on the ridge image.

2.2. Ridge representation

Due to noise in fingerprint images and imperfectness of the feature extraction methods, there are complex structures, such as loops and bridges, in the obtained ridge image. In order to make the ridge structures simple and consequently the matching algorithm easier, an operation, called clear-up, is performed: (i) closed ridges are disconnected at an arbitrary point; (ii) ridges associated with bifurcations are split into three ridges; (iii) short ridges are removed. An example is given in Fig. 2 to illustrate the effect of this operation.

Each ridge in the cleared ridge image is sampled at a constant interval to obtain a short list of sampled points. The ridges in one fingerprint are then represented as $SP = \{\{(x_{i,j}, y_{i,j}, s_{i,j})\}_{j=1}^{nr_i}\}_{i=1}^{N_r}$, where $x_{i,j}$ and $y_{i,j}$ denotes the coordinates of the jth point of the ith ridge, N_r the number of ridges in the ridge image, nr_i the number of points of the ith ridge, $s_{i,j}$ the matching status of the point, $s_{i,j}$ being 0 represents the point has been matched; otherwise $s_{i,j}$ is set to the segment number of the sub-ridge that the point belongs to. The segment number uniquely identifies the sub-ridge in the ridge considered. For example, during the matching, the statuses of the points of a ridge are [2 2 2 2 0 0 0 0 0 0 0 0 1 1 1 1 1]. This means that, at that time, the middle portion of the ridge is matched and the sub-ridges 1 and 2 are unmatched.

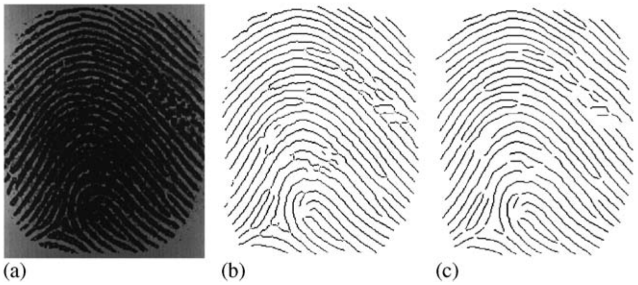


Fig. 2. (a) Intensity image (500 dpi), (b) ridge image before clear-up, (c) ridge image after clear-up and sampling (the sampling interval is 6 pixels, and the sampled points are connected).

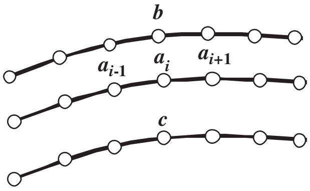


Fig. 3. b and c are the left and the right neighboring points of a_i, respectively.

2.3. Relations between ridges

Adjacent relations between ridges are important constraints for ridge matching. We use the relations between points of adjacent ridges to represent the relations between ridges since each ridge has been represented by a set of sampled points.

For a point, a_i, on a ridge, draw a line segment centered at a_i, normal to the local ridge direction, with a length of w_1 pixels. On each side of the ridge, the nearest point to a_i on the first ridge intersected by the line segment is regarded as a neighboring point of a_i. A neighboring point of a_i (if have) is classified as a left or right neighboring point by checking its position when moving along $(a_{i-1} \rightarrow a_i \rightarrow a_{i+1})$. An example is given in Fig. 3.

To record the neighborhood information of a point, the point representation $[x, y, s]$ is expanded to $[x, y, s, lr, lp, rr, rp]$, where lr and lp denote the ridge number and the point number of the left neighboring point respectively, rr and rp denote those of the right neighboring point. If a point has no neighboring point on either side, the corresponding element is set to -1.

3. Alignment

Since the pose of different impressions of the same finger can be different, the correspondences between features (minutiae, or sampled points of ridges) of two fingerprints are ambiguous. In such cases, the search for the best correspondence in the space of all possible correspondences will be time-consuming. The objective of the alignment process is to recover the transformation between two fingerprints by some means and to align them as well as possible. After that, the ambiguity of correspondence is reduced greatly and the search for the best correspondence will be less complex.

Alignment is generally done by finding the most similar substructure pairs [7,9] or by finding the substructure pairs most supported by other substructure pairs [17,18]. Such substructures can be minutiae subsets [17,18] or ridges associated with minutiae [7].

In this paper, we propose a novel approach to aligning two fingerprints. The substructure we used consists of one minutia and several associated ridges. The similarity of substructures and the support from other substructure pairs are

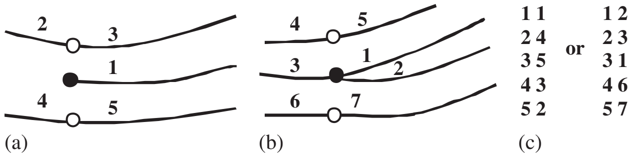


Fig. 4. (a) Substructure of a termination (solid circle represents the minutia, hollow circle represents the projection point, and the labels of ridges are given), (b) substructure of a bifurcation, (c) two possible correspondences between ridges when a termination is matched to a bifurcation.

combined to measure the goodness of substructure pairs. Since using only the best one for alignment may be too risky, top N substructure pairs can be used, respectively.

3.1. Substructure

In our approach, a substructure is constructed with the following way. First, each minutia is classified into termination or bifurcation by counting the number of ridges in a 3×3 window centered at this minutia. If the number is 1, the minutia is a termination; if the number is 3, it is a bifurcation. For a termination, in addition to the ridge that the termination belongs to, two more adjacent ridges are included (see Fig. 4(a)). For a bifurcation, in addition to the three ridges that the bifurcation belongs to, two more adjacent ridges are also included (see Fig. 4(b)). Each adjacent ridge is split into two sub-ridges by a neighboring point of the minutia, which is called the projection point of the minutia, and each sub-ridge is assigned a label.

The ridges of a substructure are labeled according to the order: first, the ridge that the minutia belongs to, and then the adjacent sub-ridges of the minutia. The three ridges that the bifurcation belongs to are labeled by checking the relative directions between the ridges. The adjacent sub-ridges of the minutia are labeled by checking the relative positions and the relative directions of the sub-ridges with respect to the minutia. Figs. 4(a) and (b) show examples for labeling of substructures of termination and bifurcation respectively.

The necessary condition for a match of two substructures is that their minutiae are of the same type and their corresponding ridges have the same labels, or their minutiae are of different types, but their ridge labels have one of the two possible correspondences shown in Fig. 4(c).

There are several advantages for choosing such substructures: (i) it contains all the topological information (minutia and ridges) in a local area; (ii) it allows different types of minutiae to be matched. This is desirable since the type of minutiae (termination and bifurcation) cannot be reliably detected during preprocessing; (iii) this helps to estimate deformation more accurately compared with based only on one ridge, when the ridge density of the two fingerprints differs.

The substructures contained in a fingerprint image are denoted as $AR = \{\{(r_{i,j}, sp_{i,j}, ep_{i,j})\}_{j=1}^{7}\}_{i=1}^{N_m}$, where $(r_{i,j}, sp_{i,j}, ep_{i,j})$ represent the ridge number, the point number of the starting point, and the point number of the endpoint of the jth labeled ridges in the ith substructure, and N_m the number of substructures (same as the number of minutiae) in the fingerprint image. The element with no corresponding ridge is set to -1. For easy check of topological relations among substructures, associated minutiae of ridges in a fingerprint image are recorded. Here associated minutiae of a ridge mean minutiae on the ridge or outside the ridge but in the substructures that contain portions of the ridge. The associated minutiae are denoted as $AM = \{\{(m_{i,j}, p_{i,j}, label_{i,j})\}_{j=1}^{n_i}\}_{i=1}^{N_r}$, where $(m_{i,j}, p_{i,j}, label_{i,j})$ represent the minutia number, the point number of the minutia (or the projection point when the minutia lies outside the ridge), and the label of the sub-ridge in the substructure for the jth associated minutia of the ith ridge.

3.2. Similarity between substructures

The objective of this step is not to compute a similarity degree between two substructures, but to give a binary decision if these two substructures are possibly matched.

For each pair of corresponding ridges, r_1 and r_2, which both contain more than four sampled points, the similarity degree of them is computed as follows. Let $p_{1,1}$ and $p_{2,1}$ be the starting points of r_1 and r_2, $p_{1,2}$ and $p_{2,2}$ be the endpoints of r_1 and r_2. Assume the shorter one of two ridges is r_1, $p_{1,2}$ is the ith point counting from $p_{1,1}$, and q is the ith point counting from $p_{2,1}$. Then the similarity degree between the two ridges is

$$sr = 1 - |d_1 - d_2| / \max(d_1, d_2), \qquad (1)$$

where d_1 denotes the Euclid distance between $p_{1,1}$ and $p_{1,2}$, and d_2 the Euclid distance between $p_{2,1}$ and q. Assume that the similarity degrees of n pairs of ridges are computed for a pair of substructures. These two substructures are regarded as possibly matched, if the following two conditions are both satisfied:

$$\begin{cases} sr_i > \sigma_1, \forall i \in [1, 2, \ldots, n], \\ (1/n)\sum_{i=1}^{n} sr_i > \sigma_2, \end{cases} \qquad (2)$$

where σ_1 and σ_2 are two thresholds, and $\sigma_1 < \sigma_2$.

3.3. Support from other substructure pairs

To compute the support of a substructure pair, which is possibly matched, from other substructure pairs, we perform a fast match for minutiae associated with each pair of corresponding ridges in these two substructures, and take the total number of matched minutiae as the support degree.

For each pair of corresponding ridges in the substructure pair $m_{1,1}$ and $m_{2,1}$, we search the two lists of associated minutiae for matched minutiae (see Fig. 5). Take a pair of ridges as example (the wider ridges in Fig. 5). Two associated minutiae, $m_{1,2}$ and $m_{2,2}$, are said to be matched if the lengths (in terms of the number of sampled points) of the

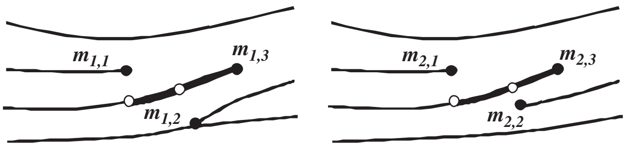


Fig. 5. Match associated minutiae of a pair of corresponding ridges (wider ridges).

two sub-ridges between two pairs of reference points (minutiae or projection points) differ less than 3, and the two sub-ridges are correspondent ridges in the substructure pair $m_{1,2}$ and $m_{2,2}$. In Fig. 5, in addition to $m_{1,2}$ and $m_{2,2}$ that are matched, $m_{1,3}$ and $m_{2,3}$ are also matched.

The substructure pairs with the top N degrees of support will be selected as the initial substructure pairs for matching.

3.4. Estimation of transformation between two substructures

While several researchers [11,14] have used thin plate spline to model the global deformation between fingerprints, we found affine transformation can approximate the local deformation between fingerprints well. In addition, estimation and application of affine transformation need less computation. Therefore, we choose affine transformation as permitted transformation and employ local transformations of matched substructure pairs to model nonlinear deformations in fingerprint images. The transformation parameters are estimated based on corresponding point sets of the substructure pair using the least square error technique.

4. Matching

Given an initial substructure pair obtained in the alignment stage, we will gradually match other ridges and minutiae. A list, *MRList*, is used to store matched ridges, and another one, *MMList*, is used to store matched minutiae. The algorithm repeatedly generates candidate ridge pairs (CRP) and matches them, until no more ridges can be matched. A CRP consists of the ridge numbers, segment numbers, and candidate point pairs (CPP) of the two ridges (or sub-ridges). A priority queue, *CRPQueue*, is used to store CRPs. The CRP with more CPPs will have higher priority value. The matching algorithm is described as follows.

(1) Generate CRPs for the initial substructure pair. Estimate the affine transformation between the two substructures. Push CRPs, the estimated transformation and priority values into *CRPQueue*.
(2) If *CRPQueue* is empty, go to step 9; otherwise, pop up the first CRP, *crp*, in *CRPQueue*.
(3) Check the validity of *crp*. If invalid, then generate new CRPs and push the new CRPs into *CRPQueue*, and then go to step 2.
(4) Match *crp* using dynamic programming matching algorithm. If there is no matched portion between these two ridges, go to step 2.
(5) Store the matched portion and set the statuses of points to 0. The remaining portions are considered as new sub-ridges and the statuses of points are set to the new segment number.
(6) Generate CRPs from the newly matched ridges, and push CRPs, the current transformation and priority values into *CRPQueue*.
(7) Search the newly matched ridges for matched minutiae. If matched minutiae are found, for each minutia (substructure) pair, generate CRPs, compute the transformation and priority values based on the newly matched substructures, and push them into *CRPQueue*.
(8) Go to step 2.
(9) Compute the matching score.

The above procedure is performed for each of N initial substructure pairs and the one with the highest matching score is chosen as the final result.

4.1. Generating CRPs

To reduce the ambiguity of correspondence, CRPs are always selected from ridges adjacent to already matched ridges or matched minutiae.

4.1.1. CRPs from matched minutiae

There are two cases in which we will generate CRPs from matched minutiae. The first case is when the matching process starts with an initial minutia pair. The second case is when new minutiae are matched during the matching process.

For a given pair of matched minutiae (substructures), the CRPs generated are the corresponding ridges in the substructure pair and the transformation is computed as described in Section 3. The CRPs generated from matched minutiae always have higher priority values than those generated from matched ridges, since local transformations are always estimated based on matched minutiae, and are better applied in nearer regions. We set the priority value of each CRP generated from matched minutiae to the number of CPPs plus a large positive number.

4.1.2. CRPs from matched ridges

Given a pair of matched ridges, $\{(SP_1(r_1, p_{1,i}), SP_2(r_2, p_{2,i}))\}_{i=1}^{N}$, and the transformation under which these two ridges are matched, we generate CRPs as follows. For each pair of matched points on these two ridges, $(SP_1(r_1, p_{1,i}), SP_2(r_2, p_{2,i}))$, a pair of neighboring points at the same side is regarded as a CPP, and the pair of ridges that this CPP belongs to is regarded as a CRP. Therefore, from a pair of matched ridges, we may get several CRPs and each CRP can contain several CPPs. The number of the CPPs contained in a CRP is used as the priority value of this CRP. The transformation of this CRP is the same as the transformation of the matched ridges from which this CRP is generated.

4.1.3. Validate a CRP

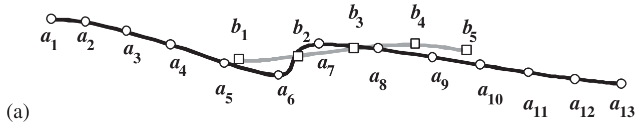
A ridge may be contained in multiple CRPs in *CRPQueue*. For example, a long ridge in one fingerprint may be paired with two short ridges in another fingerprint. During the ridge matching process, the status of points may change. Therefore, before matching a CRP, we check its validity by checking the status of each CPP of the CRP. If the status of any CPP is different from the segment number of this CRP, this CRP is invalid and we divide the original CRP into several new CRPs that have different segment numbers (note that if all CPPs of the original CRP have been matched, no new CRPs are generated). Then the new CRPs (if have), the transformation of the original CRP and priority values recomputed are pushed into *CRPQueue*.

4.2. Matching a CRP

The task of comparing two ridges is to find the longest common sub-curves. Curve matching has been formulated as a string matching problem in [16], where in order to make feature invariant with respect to rotation and translation, curvature at sampled points has been used as the characters. In this work, however, ridges have been aligned through local transformation, so that coordinates of sampled points are directly compared for matching. Two points whose Euclidean distance is less than a threshold are regarded as possible matching.

The matching procedure accepts a CRP and the corresponding transformation as inputs and proceeds as follows (see Fig. 6 for an example of matching two ridges).

(1) Take the two sub-ridges represented by the CRP, denoted as $\{a_i\}_{i=1}^{M}$ and $\{b_j\}_{j=1}^{N}$, and align them using the given transformation.
(2) A 2D array $T(i, j)$, $i = 1, \ldots, M$; $j = 1, \ldots, N$ is created, and its element $T(i, j)$ is equal to 0 or 1, which, respectively, means that the Euclidean distance between two points, a_i and b_j, is greater than or less than a threshold.
(3) Dynamic programming technique is used to find the longest 1 string in *T*, which satisfies the following conditions: (1) indices monotonously increase; (2) changes between adjacent indices should be less than 3.
(4) Array *T* is flipped horizontally, and the operation in step 3 is performed again to obtain another 1 string.
(5) If the longer 1 string is longer than 3, it is selected as the matching result; otherwise the two ridges have no matched portion.



	a_1	a_2	a_3	a_4	a_5	a_6	a_7	a_8	a_9	a_{10}	a_{11}	a_{12}	a_{13}
b_1	0	0	0	0	1	0	0	0	0	0	0	0	0
b_2	0	0	0	0	0	1	1	0	0	0	0	0	0
b_3	0	0	0	0	0	0	0	1	0	0	0	0	0
b_4	0	0	0	0	0	0	0	0	1	0	0	0	0
b_5	0	0	0	0	0	0	0	0	0	1	0	0	0

(b)

	a_1	a_2	a_3	a_4	a_5	a_6	a_7	a_8	a_9	a_{10}	a_{11}	a_{12}	a_{13}
Before	1	1	1	1	1	1	1	1	1	1	1	1	1
After	1	1	1	1	0	0	0	0	0	0	2	2	2

(c)

	b_1	b_2	b_3	b_4	b_5
Before	1	1	1	1	1
After	0	0	0	0	0

(d)

Fig. 6. Matching two ridges, *a* and *b*, by using DP matching. (a) Two aligned ridges with points labeled; (b) table *T* in which the continuous and longest 1 string is searched. The matching result is $\{(a_5, b_1), (a_6, b_2), (a_8, b_3), (a_9, b_4), (a_{10}, b_5)\}$; (c) the changes of the status of *a* before and after matching; and (d) the changes of the status of *b* before and after matching.

4.3. Detecting matched minutiae

When a pair of ridges is matched, we check the two pairs of endpoints of these two ridges for matched minutiae. First, select a pair of matched points on the two ridges as reference. If the two minutiae are possibly matched as described in Section 3.2, and neither of them is recorded in the matched minutiae list *MMList*, and the numbers of sampled points between the minutiae and the references differ less than 3 pixels, and the sub-ridges between the minutiae and the references are correspondent ridges in the substructure of these two minutiae, then these two minutiae are matched and stored in *MMList*.

4.4. Matching score

As two fingerprints may share a small overlapped region, in order to evaluate the similarity degree of them, we should consider only the overlapped region. Compared with the bounding box of minutia points used in [9], the convex hull of ridges is a more precise representation for border line of the fingerprints. In our implementation, the convex hull of ridges is approximated by the convex hull of sampled points of ridges. The overlapped region is the intersection of two convex hulls that are aligned using the initial substructure pair. The matching score can be computed according to the following formula:

$$\begin{aligned} score &= \lambda \cdot s_m + (1-\lambda) \cdot s_r \\ &= \lambda \frac{m_m^2}{M_m \cdot N_m} \min\left(1, \frac{m_m}{T_m}\right) \\ &\quad + (1-\lambda) \frac{m_r^2}{M_r \cdot N_r} \min\left(1, \frac{m_r}{T_r}\right), \end{aligned} \tag{3}$$

where λ $(0 \leqslant \lambda \leqslant 1)$ is used to weight the scores of matched minutiae and matched ridges, M_m and M_r are the numbers of minutiae and sampled points of the first fingerprint, and N_m and N_r are the numbers of minutiae and sampled points of the second fingerprint, m_m and m_r are the numbers of matched minutiae and matched points, T_m and T_r are the thresholds of the numbers of matched minutiae and matched points below which scores will be punished, respectively. Note that all the numbers in the formula are counted in the overlapped region.

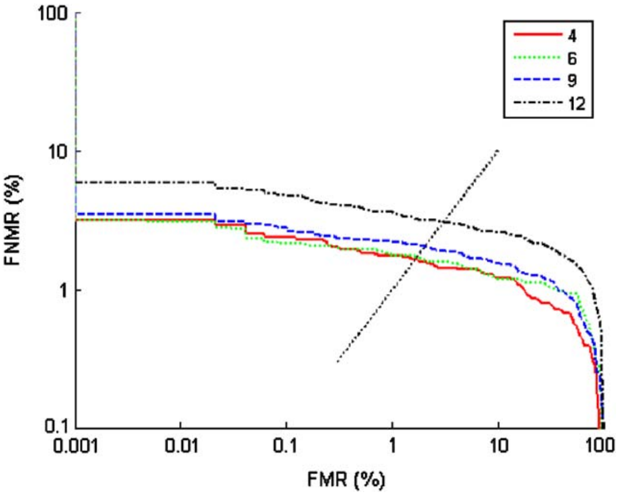


Fig. 7. ROC curves of the ridge-based algorithm with four sampling intervals on DB1.

Table 1
The EERs and the match times of ridge-based matching algorithm with four sampling intervals on DB1

	4 pixels	6 pixels	9 pixels	12 pixels
EER	0.016	0.017	0.020	0.031
Time (ms)	160	110	98	83

Table 2
Average numbers of minutiae, N_m, and average numbers of sampled points, N_r, on four training databases

	DB1	DB2	DB3	DB4
N_m	41	45	27	32
N_r	1472	1580	956	1361

5. Experimental results and analysis

Experiments have been conducted on four databases of FVC2002 [15]. The resolution of DB1, DB3, and DB4 is 500 dpi, and that of DB2 is 569 dpi. Each database consists of 800 fingerprint images from 100 fingers (eight impressions per finger). The performance evaluation protocol used in FVC2002 has been adopted. For genuine match, each impression of each finger is compared with other impressions of the same finger. The number of matches is $C_8^2 \times 100 = 2800$. For impostor match, the first impression of each finger is compared with the first impression of other fingers. The number of matches is $C_{100}^2 = 4950$.

Here we give the value of the parameters used during the matching process: the sampling interval (6 pixels on DB1, DB3 and DB4, 7 pixels on DB2), the number of initial substructure pairs (10), the threshold of distance between matched points (6 pixels on DB1, DB3 and DB4, 6.8 pixels on DB2), and the thresholds of similarity between substructures ($\sigma_1 = 0.7$, $\sigma_2 = 0.8$). Among these parameters, the sampling interval is crucial as it is the unit of several other parameters. We have tested our matching algorithm with different sampling intervals on DB1. The ROC (receiver operating characteristic) curves are plotted in Fig. 7, and EERs (equal error rates) and the matching times are given in Table 1. From the ROC curves and Table 1, we can see 6 pixels is a good choice with higher accuracy and moderate matching time. Larger sampling interval will make the performance decrease and smaller sampling interval will make the computation too time-consuming.

The values of the parameters used in the computation of the matching score are chosen as follows. λ is set to 0.5 on all databases to weight minutia information and ridge information equally. T_m is set to half of the average number of minutiae, and T_r half of the average number of sampled points on each database. The above parameter values are estimated using the training samples of FVC2002 and given in Table 2.

Two examples are given in Fig. 8 to demonstrate the effect of updating transformation based on matched minutiae during the matching process. The two fingerprints are from the same finger and there exists strong nonlinear deformation between them. If using only one global transformation, there are many ridges unmatched. If using matched minutiae to update the transformation, however, most minutiae and ridges in the overlapped region are matched. An example is given in Fig. 9 to illustrate the ability of our

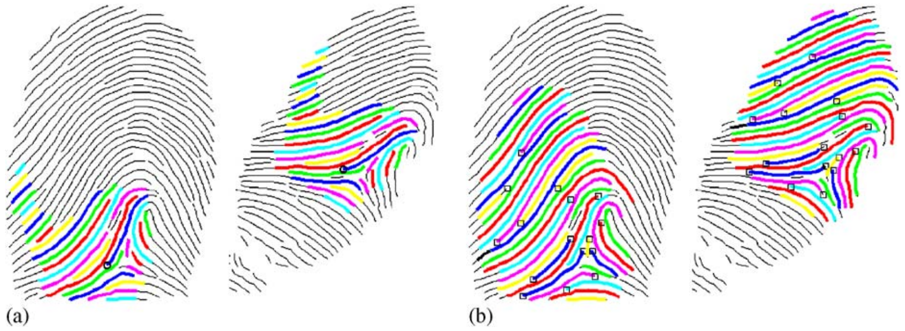


Fig. 8. (a) Matching result without updating transformation (initial minutiae are labeled, paired ridges are shown in the same color); (b) matching result with updating transformation (all matched minutiae are labeled). There exists strong nonlinear deformation between the two fingerprints from the same finger.

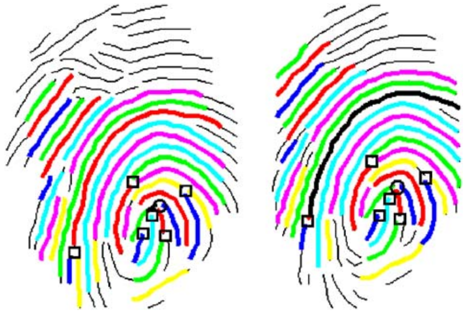


Fig. 9. Matching result of two poor quality fingerprints from the same finger.

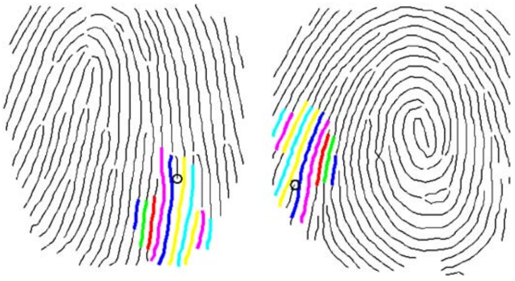


Fig. 10. Matching result of two fingerprints from different fingers.

algorithm to handle poor quality fingerprints from the same finger. From the figure, we can see that, except for the poor quality image region, the matching result is good. An example for impostor match is given in Fig. 10. From the figure, we can see that most of the ridges in the overlapped region are unmatched, and only one pair of minutiae is matched.

The algorithm presented in this paper is compared with a minutia-based matching algorithm previously developed in our lab, which is a tree search approach. In this algorithm, a pair of triplets that is most similar is first found and corresponding minutiae are regarded as matched. The nearest unmatched minutia of already matched minutiae in fingerprint 1 is assigned several candidate minutiae in fingerprint 2 that are still unmatched. Then for each path, the same procedure is performed until no more minutiae can be matched. Finally the path that contains the most matched minutiae is accepted as the best matching result, and a matching score is computed in terms of the percentage of matched minutiae over the total number of minutiae, the number of matched minutiae and distortion.

Table 3
EER (equal error rate) and match time of minutia-based and ridges-based matching algorithms on four database of FVC2002

		DB1	DB2	DB3	DB4
EER	Minutia	0.018	0.023	0.045	0.033
	Ridge	0.017	0.014	0.069	0.051
Time (ms)	Minutia	9	10	6	7
	Ridge	110	118	74	83

Equal error rates and the matching times of ridge-based method and that of minutia-based one on four databases are given in Table 3. The ROC curves of the two methods on four databases are plotted in Fig. 11. From the ROC curves, we can see that the ridge-based method is slightly better than the minutia-based one on DB1 and DB2, however, the latter is slightly better than the former on DB3 and DB4. Through the analysis of some examples of the ridge-based method on DB3 and DB4, we found genuine matches with low scores are mainly due to the following problems: (i) fail to find correct initial minutia pairs; (ii) transformation estimated based on low-quality minutia pair is inaccurate. Although the preliminary results are not very satisfactory, the ridge-based algorithm is promising and lots of works need to be done.

The above experiments were conducted on a PC with Pentium IV 2.2 GHz processor, running Windows 2000. The matching time of the proposed ridge-based matcher is about 90 ms and that of the minutia-based one is about 10 ms.

6. Summary and future work

We have proposed a fingerprint matching algorithm that matches both minutiae and ridges. A novel approach is used to find promising initial minutia pairs. For each initial

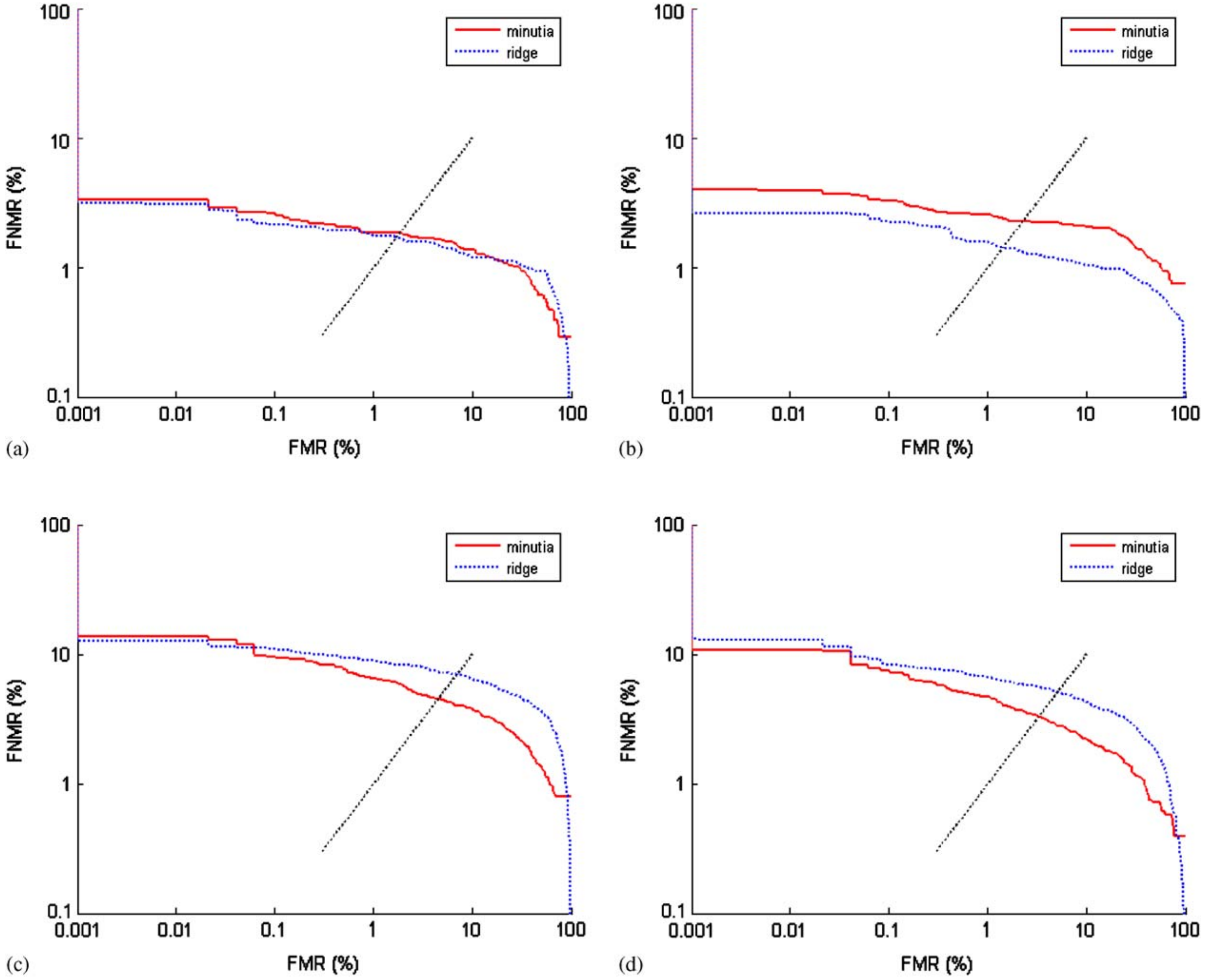


Fig. 11. ROC curves of minutia-based algorithm (solid line), ridge-based algorithm (dotted line) on: (a) DB1; (b) DB2; (c) DB3; and (d) DB4.

minutiae pair, a ridge matching process is performed, which incrementally matches the remaining minutiae and ridges. Both minutiae and ridges are taken into account when computing the matching scores. Experiment is done to demonstrate the ability of our algorithm to handle the nonlinear deformation and poor quality fingerprints. ROCs on FVC2002 databases show that the ridge-based approach performs comparably with the minutia-based approach.

While many aspects in the current algorithm should be improved, the following problems may be the most important ones:

(1) The alignment algorithm should be improved to handle fingerprints with fewer minutiae.
(2) The matching algorithm should be improved to handle low quality fingerprints and fingerprints with strong deformation.
(3) After matching, analyze the causes of the unmatched minutiae and give different punishment to different cases.
(4) A feature extraction algorithm should be specially designed for the ridge matching algorithm.
(5) Computational complexity should be decreased.

Acknowledgements

The authors would like to thank the anonymous referees for their valuable comments. This work is supported by National Natural Science Foundation of China under grant 60472069.

References

[1] C. Wilson, C. Watson, E. Paek, Effect of resolution and image quality on combined optical and neural network fingerprint matching, Pattern Recognition 33 (2) (2000) 317–331.
[2] D.K. Isenor, S.G. Zaky, Fingerprint identification using graph matching, Pattern Recognition 19 (2) (1986) 113–122.
[3] A.K. Hrechak, J.A. McHugh, Automatic fingerprint recognition using structural matching, Pattern Recognition 23 (8) (1990) 893–904.
[4] A. Wahab, S.H. Chin, E.C. Tan, Novel approach to automated fingerprint recognition, in: Proceedings of IEE Visual Image Signal Processing, 1998, pp. 160–166.

[5] A.K. Jain, S. Prabhakar, L. Hong, S. Pankanti, Filterbank-based fingerprint matching, IEEE Trans. Image Process. 9 (5) (2000) 846–859.

[6] N.K. Ratha, K. Karu, S. Chen, A.K. Jain, A real-time matching system for large fingerprint databases, IEEE Trans. Pattern Anal. Mach. Intell. 18 (8) (1996) 799–813.

[7] A.K. Jain, L. Hong, R. Bolle, On-line fingerprint verification, IEEE Trans. Pattern Anal. Mach. Intell. 19 (4) (1997) 302–313.

[8] A. Ross, A.K. Jain, J. Reisman, A hybrid fingerprint matcher, Pattern Recognition 36 (7) (2003) 1661–1673.

[9] M. Tico, P. Kuosmanen, Fingerprint matching using an orientation-based minutia descriptor, IEEE Trans. Pattern Anal. Mach. Intell. 25 (8) (2003) 1009–1014.

[10] Z.M. KovQacs-Vajna, A fingerprint verification system based on triangular matching and dynamic time warping, IEEE Trans. Pattern Anal. Mach. Intell. 22 (11) (2000) 1266–1276.

[11] A.M. Bazen, S. Gerez, Fingerprint matching by thin-plate spline modeling of elastic deformations, Pattern Recognition 36 (8) (2003) 1859–1867.

[12] The 3rd Fingerprint Verification Competition, http://bias.csr.unibo.it/fvc2004/.

[13] A. Senior, A hidden Markov model fingerprint classifier, in: Proceedings of the Thirty-First Asilomar Conference on Signals, Systems & Computers, 1997, pp. 306–310.

[14] A. Ross, Information Fusion in Fingerprint Authentication, Ph.D. Thesis, Michigan State University, 2003.

[15] D. Maio, D. Maltoni, R. Cappelli, J.L. Wayman, A.K. Jain, FVC2002: Second fingerprint verification competition, in: Proceedings of the International Conference on Pattern Recognition (ICPR), Quebec City, Canada, 2002, pp. 744–747.

[16] H.J. Wolfson, On curve matching, IEEE Trans. Pattern Anal. Mach. Intell. 12 (5) (1990) 483–489.

[17] S.H. Chang, F.H. Cheng, W.H. Hsu, G.Z. Wu, Fast algorithm for point pattern matching: invariant to translations, rotations and scale changes, Pattern Recognition 30 (2) (1997) 311–320.

[18] R. Udupa, G. Garg, P. Sharma, Fast and accurate fingerprint verification, in: Proceedings of the Third International Conference on Audio- and Video-Based Biometric Person Authentication, 2001, pp. 192–197.

About the Author—JIANJIANG FENG received the B.S. degree in Communication Engineering from Beijing University of Posts and Telecommunications (BUPT) in 2000. He is currently a Ph.D. student at BUPT. His research interests include fingerprint recognition and structural matching.

About the Author—ZHENGYU OUYANG received the B.S. degree in Communication Engineering from Beijing University of Posts and Telecommunications (BUPT) in 2003. He is currently a postgraduate at BUPT. His research interests include fingerprint recognition and statistical pattern recognition.

About the Author—ANNI CAI received the B.S. degree in radio engineering from Beijing University of Posts and Telecommunications (BUPT), Beijing, China, in 1965. She received the Ph.D. degree in electrical and computer engineering from University of California, Santa Barbara, CA, USA in 1988. Since 1965, she has been on the faculty of School of Telecommunication Engineering, BUPT, where she is presently a professor. She was a visiting scholar at California State University, Long Beach, CA, USA from 1980 to 1982. She has co-authored more than 50 journal papers and four books. Her current interests are in the fields of image and video processing, pattern recognition and multimedia telecommunications.