Available online at www.sciencedirect.com
SCIENCE DIRECT®
Pattern Recognition 38 (2005) 1672–1684
PATTERN RECOGNITION
THE JOURNAL OF THE PATTERN RECOGNITION SOCIETY
www.elsevier.com/locate/patcog

A minutia-based partial fingerprint recognition system

Tsai-Yang Jea*, Venu Govindaraju

Center for Unified Biometrics and Sensors, University at Buffalo, State University of New York, Amherst, NY 14228, USA

Received 20 October 2004; received in revised form 14 March 2005; accepted 14 March 2005

Abstract

Matching incomplete or partial fingerprints continues to be an important challenge today, despite the advances made in fingerprint identification techniques. While the introduction of compact silicon chip-based sensors that capture only part of the fingerprint has made this problem important from a commercial perspective, there is also considerable interest in processing partial and latent fingerprints obtained at crime scenes. When the partial print does not include structures such as core and delta, common matching methods based on alignment of singular structures fail. We present an approach that uses localized secondary features derived from relative minutiae information. A flow network-based matching technique is introduced to obtain one-to-one correspondence of secondary features. Our method balances the tradeoffs between maximizing the number of matches and minimizing total feature distance between query and reference fingerprints. A two-hidden-layer fully connected neural network is trained to generate the final similarity score based on minutiae matched in the overlapping areas. Since the minutia-based fingerprint representation is an ANSI-NIST standard [American National Standards Institute, New York, 1993], our approach has the advantage of being directly applicable to existing databases. We present results of testing on FVC2002's DB1 and DB2 databases.

Keywords: Partial fingerprint; Similarity score; Minimum cost flow; Minutia; Fingerprint matching

1. Introduction

Fingerprint matching based on minutia features is a well researched problem. During the last four decades, various algorithms have been proposed to match two minutia templates of fingerprints. Most of these algorithms assume that the two templates are approximately of the same size. This hypothesis is no longer valid. Miniaturization of fingerprint sensors has led to small sensing areas usually varying from 1" × 1" to 0.42" × 0.42". However, fingerprint scanners with a sensing area smaller than 0.5" × 0.7", which is considered to be the average fingerprint size [2], can only capture partial fingerprints.

Matching small (partial) fingerprints to full pre-enrolled images in the database has several problems: (i) the number of minutia points available in such prints is few, thus reducing its discriminating power; (ii) loss of singular points (core and delta) is likely and therefore, a robust algorithm independent of these singularities is required; and (iii) uncontrolled impression environments result in unspecified orientations of partial fingerprints, and distortions like elasticity and humidity are introduced due to characteristics of the human skin.

A minutiae-based fingerprint matching system usually returns the number of matched minutiae on both query and reference fingerprints and uses it to generate similarity scores. Generally, more matched minutiae yield higher similarity scores. That is when the number of minutiae on

* Corresponding author. Tel.: +1 716 645 6164; fax: +1 716 645 6176.
E-mail address: tjea@cedar.buffalo.edu (T.-Y. Jea).

0031-3203/$30.00
doi:10.1016/j.patcog.2005.03.016

both fingerprints is large we can confidently distinguish the genuine and imposter fingerprint using the number of matched minutiae. According to forensic guidelines, when two fingerprints have a minimum of 12 matched minutiae they are considered to have come from the same finger [3]. However, it is not reasonable to use an absolute number of matched minutiae alone in case of partial fingerprints. We must also consider the overlapped areas on both prints and the total distance between all the matched minutiae to obtain a similarity score.

In this paper, we discuss the various issues involved in the matching of such partial fingerprints. In Section 2, we outline our approach to partial fingerprint matching. A localized secondary feature matching is described. The secondary features are derived from minutiae features. It does not depend on global ridge structures (e.g. core and delta) making it suitable for matching partial fingerprints. The notion of dynamic tolerance area and generation of reliable similarity score is discussed. In Section 3, we briefly review the minimum-cost flow (MCF) problem in the context of fingerprint matching. This technique is applied to both secondary feature matching and the brute-force matching derived from minutiae features. Experimental results are presented in Section 4.

2. Partial fingerprint matching

Our system works on the minutiae-based representation of a fingerprint. Minutiae, in fingerprint context, are the various ridge discontinuities of a fingerprint. More than 100 different types of minutiae have been identified, among which ridge bifurcations and endings (Fig. 1) are the most widely used. Minutia-based representation of fingerprints is an ANSI-NIST standard [1,4] and contains only local information without relying on global information such as singular points or center of mass of fingerprints.

Matching two fingerprints (in minutiae-based representation) is to find the alignment and correspondences between minutiae on both prints. For matching regular sized fingerprint images, a brute-force matching, which examines all the possible solutions, is not feasible since the number of possible solutions increases exponentially with the number of feature points on the prints [3]. In order to increase the efficiency of matching process, other methods instead of brute-force matching must be applied. Intuitively, a pre-alignment method may obtain the alignment parameters of two fingerprints. Pre-alignment methods that depend on the global singular points [3,5] are not suitable for partial fingerprint matching. Other pre-alignment techniques [4,6] need to reprocess all the images thus they cannot be used on already existing databases.

There are two major types of features that are used in fingerprint matching: local and global features. Local features, such as the minutiae information and our secondary features, contain the information that is in a local area only and invariant with respect to global transformation. On the other hand, global features, such as number, type, and position of singularities, spatial relationship and geometrical attributes of ridge lines, size and shape of the fingerings, are characterized by the attributes that capture the global spatial relationships of a fingerprint [3]. Because of the nature of partial fingerprints, partial fingerprint matching requires a set of local features that do not depend on global singular structures. Moreover, localized features have the ability to tolerate more distortions. Kovács-Vajna [7] has shows that the geometric deformations on local areas can be more easily controlled than global deformations.

2.1. Secondary features

The secondary features are derived from minutiae information. We use the minutiae extraction techniques described in [8] with some modifications to remove the false minutiae on the edge of the fingerprint foreground to generate the minutiae for our system. The method first gets the image quality maps by checking the low contrast areas, low flow blocks, and high curve regions. And then, a binary

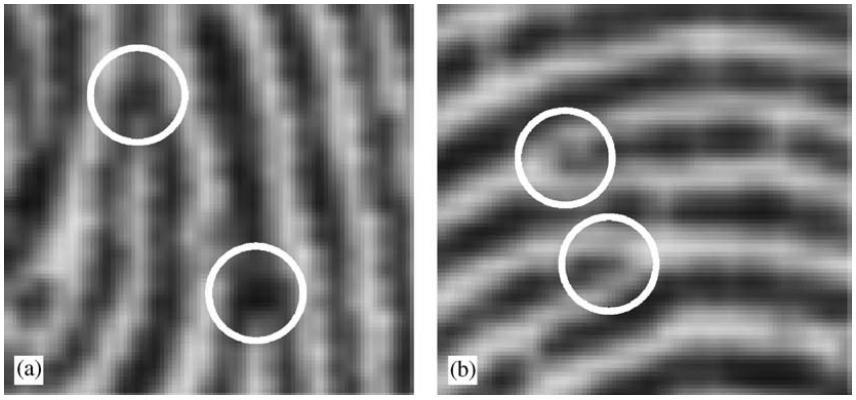

Fig. 1. (a) Ridge bifurcations, (b) ridge endings.

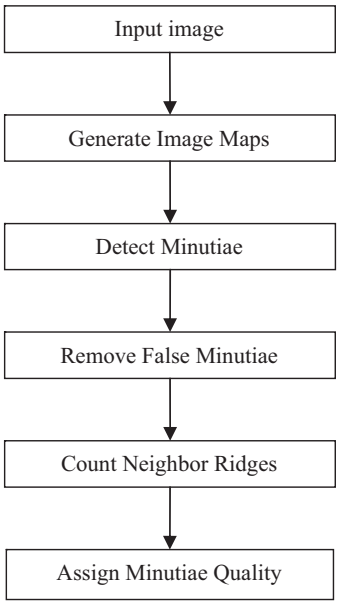

Fig. 2. Minutiae detection process described in [8].

representation of the fingerprint is constructed by applying a rotated grid on the ridge flows of the fingerprint. Minutiae are generated by comparing each pixel neighborhood with a family of minutiae templates. Finally, a series of heuristic rules is used to merge and filter out the spurious minutiae (Fig. 2).

Jiang and Yau [9] use relative distance, radial angle, and minutia orientation along with the ridge count and minutia type to generate the features for local matching. The secondary features that we use are similar but the minutiae type and ridge count elements are removed. Minutiae types are difficult to distinguish when impression pressure varies on different applications (Fig. 3). Furthermore, ridge count is not universally available and not all minutiae representations in existing databases contain this information.

We generate a five-element secondary feature vector (Fig. 4). For each minutiae M_i (x_i, y_i, θ_i) and its two

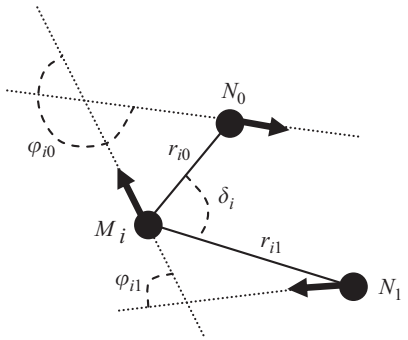

Fig. 4. Secondary feature of M_i. Where r_{i0} and r_{i1} are the Euclidean distances between central minutia M_i and its neighbors N_0 and N_1, respectively. φ_{ik} is the orientation difference between M_i and N_k where k is 0 or 1. δ_i represents the acute angle between the line segments $M_i N_0$ and $M_i N_1$.

nearest-neighbors N_0 $(x_{n0}, y_{n0}, \theta_{n0})$ and N_1 $(x_{n1}, y_{n1}, \theta_{n1})$, we construct a secondary feature vector S_i $(r_{i0}, r_{i1}, \varphi_{i0}, \varphi_{i1}, \delta_i)$ in which r_{i0} and r_{i1} are the Euclidean distances between the central minutia M_i and its neighbors N_0 and N_1 respectively. φ_{ik} is the orientation difference between M_i and N_k, where k is 0 or 1. δ_i represents the acute angle between the line segments $M_i N_0$ and $M_i N_1$. Note that N_0 and N_1 are the two nearest neighbors of the central minutia M_i and ordered not by their Euclidean distances but by satisfying the equation:

$$\left|\overrightarrow{N_0 M_i} \times \overrightarrow{N_1 M_i}\right| \geqslant 0.$$

N_0 is the first and N_1 is the second minutia that we encounter when we traverse the angle, $\angle N_0 M_i N_1$. This arrangement is again different from the feature vector proposed by Jiang and Yau [9], where the Euclidean distance to the central minutiae orders neighboring minutiae. However, this increases the chance of flipping the order of the neighboring minutiae.

2.2. Tolerance areas

Distortions are inevitable when mapping a three-dimensional fingertip onto a two-dimensional plane. These

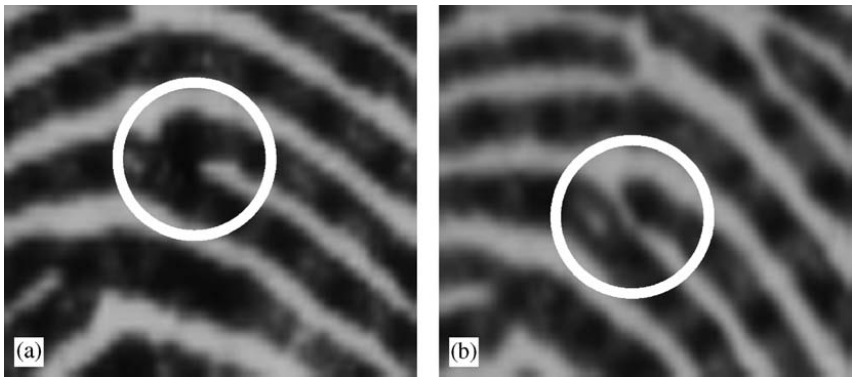

Fig. 3. The same minutiae extracted from two different impressions. In (a) it appears as a bifurcation but in (b) as a ridge ending.

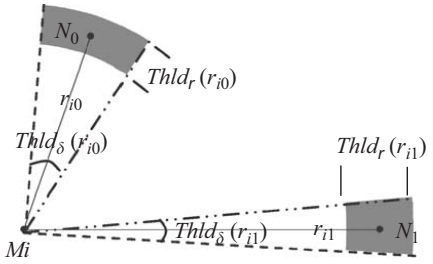

Fig. 5. Dynamic tolerance areas. The gray areas around the two neighbors, N_0 and N_1 of M_i, are decided according to the thresholds $Thld_r(r_{i0})$, $Thld_\delta(r_{i0})$, $Thld_r(r_{i1})$, and $Thld_\delta(r_{i1})$.

can be caused by vertical pressures, shear forces and varying impression conditions. As values of r_{i0} and r_{i1} increase, we observe that a secondary feature, S_i $(r_{i0}, r_{i1}, \varphi_{i0}, \varphi_{i1}, \delta_i)$, has larger distortions of φ_{i0}, φ_{i1} and δ_i. Kovács-Vajna [7] has demonstrated that small local deformations can result in a large global distortion. Thus, we make the assumption that the distortions of distance are less when the values of r_{i0} and r_{i1} are small. However, the distortions of the angle and orientation tend to be larger when r_{i0} and r_{i1} are small. Due to these factors, it is reasonable to adjust the tolerance areas according to the values of r_{i0} and r_{i1}.

A tolerance area is decided by three threshold functions $Thld_r(\cdot)$, $Thld_\delta(\cdot)$, and $Thld_\theta(\cdot)$. The distance thresholds (decided by $Thld_r(\cdot)$) should be more restrictive (smaller) when r_{i0} and r_{i1} are smaller and more flexible when r_{i0} and r_{i1} are larger. On the other hand, the thresholds on angles should be larger in order to allow large distortions when r_{i0} and r_{i1} are small, but smaller when r_{i0} and r_{i1} are large (Fig. 5). Since the thresholds change with the length of the line segment of central minutia and its neighbors, we use functions $Thld_r(\cdot)$ and $Thld_\delta(\cdot)$ instead of fixed numbers to represent the thresholds. Threshold $Thld_\theta(\cdot)$ is used for the orientation differences between the central minutiae and its neighbors. $Thld_\theta(\cdot)$ has the same characteristics as $Thld_\delta(\cdot)$ (not shown in Fig. 5).

In our implementation, the threshold functions are based on normalized feature distances. The normalization factors depend on the distance r_{i0} and r_{i1} from the central minutia. The normalization factor for radial distances (r) increases with r_{i0}, r_{i1}. The normalization factors for angular distance (δ) and orientation difference (θ) decrease with r_{i0}, r_{i1}. The normalized feature distances not only give us the ability to handle different types of feature distances directly but also reflect the dynamic tolerance areas as described above.

2.3. Feature matching

We use different matching schemes according to the number of minutiae on the query (I) and the reference (R) fingerprints (Fig. 6). A '*full*' fingerprint implies an image that is about 0.5" × 0.7" and usually leads to greater than α (a pre-defined threshold) minutiae. There are three matching scenarios: (1) both number of minutiae on I and R are less than α; (2) either I or R has number of minutiae less than α; and (3) both I and R contain more than α minutiae. In the first two cases, we have fewer minutiae on at least one fingerprint. Finding the two nearest neighbors to construct a secondary feature makes it difficult to discover a match when the fingerprint is small. In such cases, we match (by brute-force) all the feature points directly by examining all the possible solutions and finding the most matches. A brute-force matching technique tries all possible correspondences between the minutiae on query and reference fingerprints. This technique is usually very time-consuming. To make it practical, our system uses brute-force matching only when there are small numbers of minutiae, which commonly occurs when matching partial fingerprints. When, we have more than α minutiae, we use a secondary feature-based matching method [10] instead of the brute-force method to improve speed and accuracy.

2.3.1. The brute-force matching method works directly on the minutiae information. For each minutiae p_i (x_i, y_i, θ_i) on I and q_j (x_j, y_j, θ_j) on R, we take p_i and q_j as the matched reference points and find all the other matched minutiae in the polar coordinate system by converting the matching into a MCF (details are described later) problem to obtain the optimal pairing. In polar coordinates, a minutia m_i (x_i, y_i, θ_i) is represented as $(r_{i,k}, \Phi_{i,k}, \theta_{i,k})$, with respect to the original point m_k (x_k, y_k, θ_k). Where $(r_{i,k}, \Phi_{i,k})$ is in polar coordinates and $\theta_{i,k}$ is the orientation difference between θ_I and θ_k. For the given matched reference minutiae pair $p_{i'}$ and $q_{j'}$, we say minutiae p_i $(r_{i,i'}, \Phi_{i,i'}, \theta_{i,i'})$ matches q_j $(r_{j,j'}, \Phi_{j,j'}, \theta_{j,j'})$, if q_j is within the tolerance area of p_i. Thus for given threshold functions $Thld_r(\cdot)$, $Thld_\Phi(\cdot)$,and $Thld_\theta(\cdot)$, $|r_{i,i'} - r_{j,j'}| \leqslant Thld_r(r_{i,i'})$, $|\Phi_{i,i'} - \Phi_{j,j'}| \leqslant Thld_\Phi(\Phi_{i,i'})$ and $|\theta_{i,i'} - \theta_{j,j'}| \leqslant Thld_\theta(\theta_{i,i'})$. Note that the thresholds are not predefined values but are adjustable according to $r_{ii'}$ and $r_{jj'}$.

2.3.2. When both I and R have large numbers of minutiae (larger than α), brute-force matching is computationally expensive for real applications. In such cases, we match using secondary features. Given that the secondary features are localized and invariant to rotation we skip the pre-alignment stage.

Let S_i $(r_{i0}, r_{i1}, \varphi_{i0}, \varphi_{i1}, \delta_i)$ and S_j $(r_{j0}, r_{j1}, \varphi_{j0}, \varphi_{j1}, \delta_j)$ be the secondary features on the query and reference fingerprints, respectively. S_i matches S_j, if $|r_{i0} - r_{j0}| \leqslant Thld_r(r_{i0})$, $|r_{i1} - r_{j1}| \leqslant Thld_r(r_{i1})$, $|\varphi_{i0} - \varphi_{j0}| \leqslant Thld_\theta(\varphi_{i0})$, $|\varphi_{i1} - \varphi_{j1}| \leqslant Thld_\theta(\varphi_{i1})$, and $|\delta_i - \delta_j| \leqslant Thld_\delta(\delta_i)$. That is the two neighbors N_{i0} and N_{i1} of S_i fall in the corresponding tolerance areas of N_{j0} and N_{j1} of S_j. For every secondary feature in I, we find a list (candidate list) of possibly matched features in R. To find the one-to-one correspondence between the secondary features on I and T, we use MCF (described in Section 3) to obtain a candidate list of possibly matched secondary features.

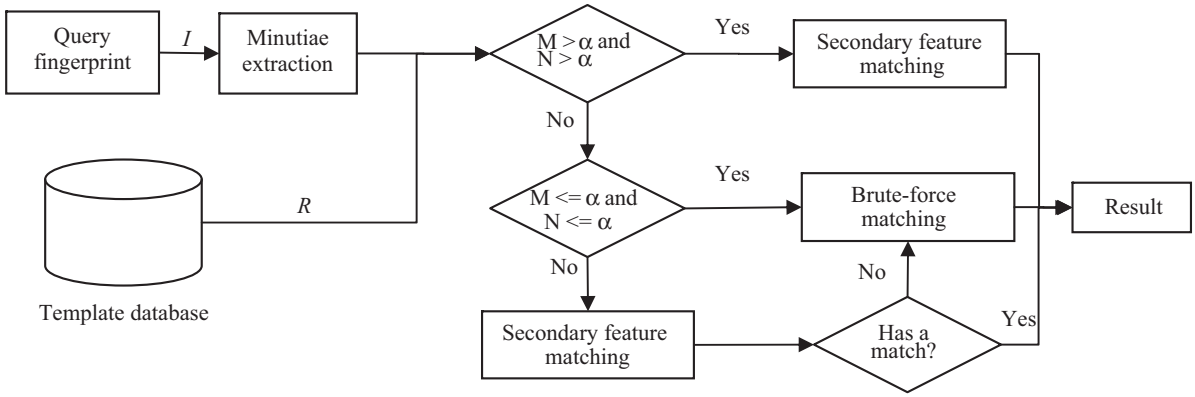

Fig. 6. Proposed fingerprint matching system. M and N are the number of minutiae on query and reference fingerprints. α is a pre-defined value.

Fig. 7. An example of two false matched secondary features. They are similar at the local structures but conflict with each other in global context (at very different locations with respect to the core and delta points).

There are cases where certain secondary features in the candidate list are in conflict with each other in the global context (Fig. 7). The secondary feature presents only the local structure around a minutia point. Considering two aligned fingerprint, a secondary feature locates right besides the core point on the first fingerprint can match to the secondary feature reside around the delta point of the second fingerprint, only because that they have similar local structures. To resolve the conflicts, a validation step is introduced by requiring that all matched feature pairs should have similar orientation differences if they come from the same finger. Jiang and Yau [9] estimate the orientation difference between features pairs using the best fit method. They make the best fit local feature structure pairs as reference points between two fingerprints. However, this may not always work since the best-matched feature is not necessarily correct. We perform the global structure validation and approximate the orientation difference between fingerprints by plotting a histogram where each bin is about 36°. The dominating bin and its neighbors are identified. The matched feature pairs in the other bins (not in the dominating bin and its neighbors) are removed from the candidate list.

Finally, according to the information derived from the matched secondary features, we convert the minutiae from the co-ordinate system of the reference fingerprint R into that of query fingerprint I and get the number of matched minutiae by applying the flow network method. We can choose the best-matched pairs in the dominating bin as our reference points or try the top N choices and get the largest matched number, say n, as the final result.

2.4. Similarity score calculation

Human experts make the final decision according to the forensic guidelines that state a minimum of 12 matched minutiae is required to consider the two fingerprints are from the same finger [3]. However, a minutiae-based automatic fingerprint recognition system cannot make decision using

Fig. 8. Examples of detected overlapped areas: (a) and (b) are from the same finger; (c) and (d) are from the same finger. The black solid points are the reference points and the areas inside the gray polygons are the detected overlapped areas. Note that the shape differences are caused by the inaccuracy of the coordinates translation and spatial distortions. The differences only happen on the points close to boundaries and would not bring too much impact to results.

an absolute value alone as a human expert. Unlike human experts having the access to all the information that a fingerprint image has, such as ridge flows, singular points and scars, etc., the minutiae-based automatic systems only have the information from the minutiae representation of fingerprints.

A traditional way to calculate the similarity scores for a minutiae-based system is $n^2/(size_I \times size_R)$. Where $size_I$ and $size_R$ represent the numbers of minutiae on query and reference fingerprints, and n is the number of matched minutiae on both prints. Bazen and Gerez [11] claim using $2n/(size_I + size_R)$ to compute the similarity scores will give better results. In our observation, we found both methods are unreliable, especially when matching fingerprints of different sizes. We propose to use the number of matched minutiae, the numbers of minutiae points on overlapping areas, and the average feature distances to calculate reliable similarity scores.

The convex hulls for a given reference point pair generates the overlapped areas of query and reference fingerprints. Let us denote the convex hull constructed from feature points on query fingerprint (I) as C_I. For every feature point on the reference fingerprint (R), if it falls inside C_I, we say it is in the overlapped area with I. Similarly, we would have a set

```
LET height_c as the height of the combined print
LET width_c as the width of the combined print
LET max_h as the maximum possible height
LET max_w as the maximum possible width
LET T_m as a integer-valued threshold

IF height_C > max_h OR width_C > max_w THEN
   Similarity_score = 0;
ELSE
   IF O_I < 5 THEN
      O_I = 5;
   END IF
   IF O_R < 5 THEN
      O_R = 5;
   END IF
   IF n >= T_m AND n > 3/5 O_I AND n > 3/5 O_R THEN
      Similarity_score = S_avg;
   ELSE
      Similarity_score = n^2×S_avg/(O_I×O_R);
      IF Similarity_score > 1.0 THEN
         Similarity_score = 1.0;
      END IF
   END IF
END IF
```

Fig. 9. Heuristic rule for calculation with number of matched feature points (n), numbers of feature points on overlapping areas (O_I, O_R), and the average score of all the matched features s_{avg}.

of feature points on I that fall in the overlapped area with R (Fig. 8). Thus, we can have the numbers (O_I and O_R) of feature points on overlapped areas of I and R. In our experiments, we found that it is difficult to find a good heuristic rule to combine all the information from $size_I, size_R, n, O_I, O_R$, and the average score of all the matched features s_{avg}, to compute the similarity score.

We propose two different methods, heuristic rule and neural network, to calculate the similarity scores with the extra information of number of feature points on the overlapping areas (O_I and O_R).

1. The heuristic rule is described in Fig. 9.
2. A two-hidden-layer fully connected neural network (Fig. 10) is trained to take the six values as input and return a similarity score between 0 and 1. Our experiments show about 1.21% and 0.68% improvement on minimum total error rate on the FVC 2002 DB1 and DB2 databases by simply using this similarity score calculation method.

A comparison of different similarity score calculations is shown in Table 1. The comparison shows our method gives higher scores of the genuine tests (the first case in Table 1) than the scores of imposter tests (the second row in Table 1).

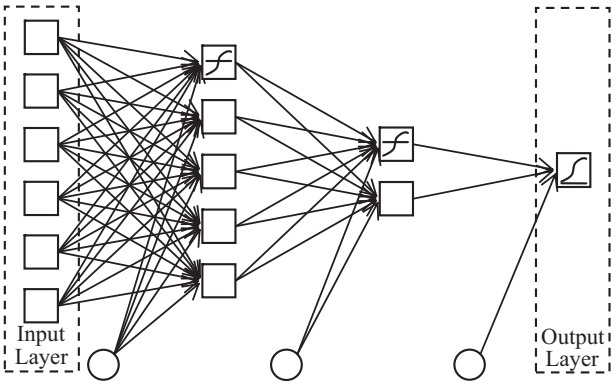

Fig. 10. The architecture of the neural network. The network has 2 hidden layers and 3 bias nodes link to each node. The input layer has 6 nodes. The hidden layers have 5 and 2 nodes with tangent sigmoid transfer function and logarithmic sigmoid transfer function, respectively. The output layer uses logarithmic sigmoid transfer function to generate the similarity score.

3. MCF

The matching of the minutiae feature points in fingerprints presents several unique challenges. Matching the feature points on two fingerprints is equivalent to finding the correspondences between the feature points (Fig. 11). The numbers of feature points on query and reference fingerprints are rarely equal and therefore not every feature point finds a matched feature point. Thus, obtaining an optimal pairing is not trivial even when two fingerprints are aligned.

The most important rule of matching feature points is to guarantee one feature point can match to at most one feature point. To comply with this constraint, one can mark the minutiae that have already been matched to avoid matching it twice or more. But, it is hard to find the optimal pairing of the feature points. For example, given that the feature point (m_1^2 in Fig. 12) of a query fingerprint, I, can fall within the tolerance area of more than one feature point of the template fingerprint, R, the best pairing is the configuration that can maximize the final number of matched minutia pairs (Fig. 12). A more sophisticated method should be used to obtain the optimum pairing [3].

In order to obtain the optimal pairing between feature points, various point pattern matching techniques can be applied. Families of point pattern matching methods have been studied in many pattern recognition and computer vision tasks. Relaxation methods [12] iteratively adjust the confidence level of the pairing until a certain acceptance criterion is satisfied. Energy minimization methods, such as genetic algorithms, find the optimal solutions by minimizing the energy functions associated with each solution. These methods are slow and unsuitable for real-time matching process. Tree pruning approaches usually require an equal number of points in both templates and no outliers which is difficult

Table 1
Comparison of different similarity score calculations

	n	$size_I$	$size_R$	$\frac{n^2}{size_I size_R}$ [11]	$\frac{2n}{size_I+size_R}$ [11]	Heuristic	NN
57_2 vs. 57_4	10	41	21	0.12	0.32	0.55	0.999996
23_1 vs. 45_1	14	39	45	0.13	0.35	0.18	0.005490

Results are calculated on the images from FVC2002 DB1 database. Images 57_2 and 57_4 are from the same finger but different impressions. Images 23_1 and 45_1 are from different fingers.

Fig. 11. Match two fingerprints is equivalent to find the corresponding links between feature points.

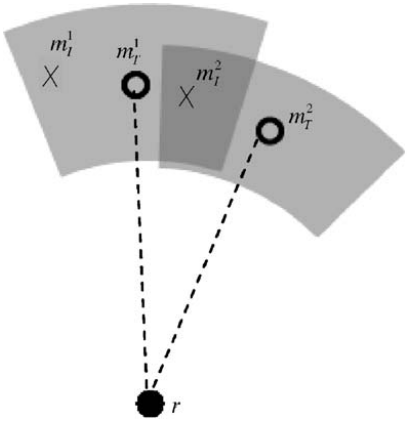

Fig. 12. Minutiae of I and R are aligned with respect to the reference point r. Minutiae from I are denoted by X's, and the minutiae form R are denoted as Os. If m_I^2 were matched with m_R^1, which is the closest minutia to m_I^2, then m_R^2 would stay unmatched.

to satisfy in fingerprint matching. Hough transform-based methods are also used [13]. We obtain alignment parameters by the secondary feature matching and then applying

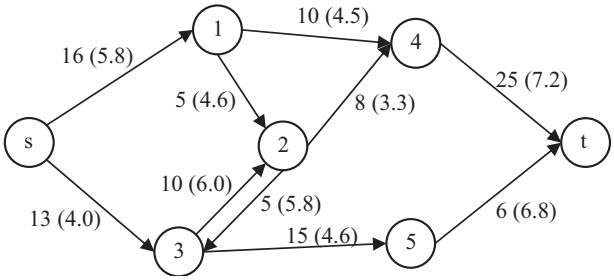

Fig. 13. An example of minimum cost flow problem. Our goal is to find a flow from s to t with maximum flow value and minimum cost. The numbers assigned to each edge is the capacity and its unit cost in parentheses.

the solutions from operational research techniques to find the one-to-one correspondence between feature points.

We use MCF technique (Fig. 13), to find the optimal pairing. The translation from fingerprint feature points matching to MCF is intuitive. MCF is the generalization of many network problems, such as the shortest path problem, maximum flow problem, transportation problem, transshipment

problem and maximum bipartite matching problem [14]. In this paper, we explore its use for fingerprint matching.

Fig. 13 gives a simple example of the MCF problem. Imagine a small with 7 cities. We want to transport as many supplies as possible from city s to city t. There are 5 cities on different toll-routes between s and t with varying rode widths (capacities) and tolls (costs). To solve the MCF problem, we must find the amount of supplies t can receive with a minimum cost. The MCF problem is defined as:

Given a directed graph $G = (V, E)$, where V and E are the sets of nodes and edges in G, with the source node s, sink node t, a real-valued capacity function $w(u, v)$ and a real-valued cost function $c(u, v)$ for all $u, v \in V$. A flow f in G is a real-valued function $f : V \times V \to R$ such that

$$\forall u,\ v \in V, \quad f(u, v) <= w(u, v) \quad \text{[Capacity constraint]},$$

$$\forall u,\ v \in V, \quad f(v, u) = -f(u, v) \quad \text{[Skew symmetry]},$$

$$\forall u \in V - \{s, t\}, \sum_{v \in V} f(u, v) = 0 \quad \text{[Flow conservation]}.$$

The value of the flow is $|f| = \sum_{v \in V} f(s, v)$, and $C(f) = \sum_{u,v \in V} c(u, v) f(u, v)$ is the cost of the flow. The capacity constraint simply implies that the net flow from one node to another must not exceed the given capacity. Skew symmetry states that the net flow from one node to another is the negative of the net flow in reverse direction. The property of flow conservation states that any node, which is not the source or sink, must have outgoing flow equal to incoming flow [14]. Different approaches have been proposed to effectively solve the MCF problem [15]. The objective is to find the maximum flow $|f|$ with the minimum cost $C(f)$.

MCF and its applications are extensively discussed in the publications of Ford and Fulkerson [16], Edmonds and Karp [17]. Besides the classical algorithms, Out-Of Kilter [16] and minimum-cost augmentation method [17], many other algorithms have been proposed to efficiently solve MFC problems, such as the methods of network simplex, cost-scaling, relaxation, and push-relabel [15,18]. The "Scaling & Canceling" algorithm proposed by Orlin et al. [19] can solve the problem in polynomial-time $(O(m(m + n\log^n)\log(nU)))$, where n is the number of nodes in the network, m is the number of edges and U is the upper bound of the value of capacity (which would be 1 in our case).

In fingerprint recognition, we are not only interested in the maximum number of matches but also in the minimum cost.

We translate our feature point matching problem into a MCF problem. Matching the feature points on two fingerprints is equivalent to finding the correspondences between the feature points (Fig. 11). Suppose we have two sets of feature points from different fingerprint images (I and R) and they are already aligned with respect to a pair of reference points in each image. We add one extra point (node), say the source s, into the set of I and add the point (node), say sink t, into the set of R. We also set up the links (edges) between nodes by obeying the following rules:

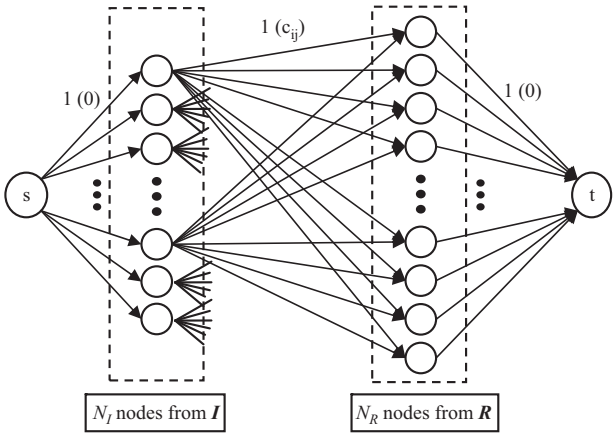

Fig. 14. Flow network representation of minutia matching problem. All the edges in the network have capacity 1. The edges between source node s and nodes from I have zero cost as well as the edges from R to sink node t. Costs of the edges between nodes from I and nodes from R are from the cost matrix c.

- There is one and only one link that connects s to every point in the first set.
- There is one and only one link that connects t to every point in the second set.
- There is no link between the points within the same set.
- There is exactly one link between every point in first set and every point in the second set.
- Every link is associated with a capacity and a cost.

The cost matrix $c(i, j) = dist(m_i, m'_j)$, where $1 \leqslant i \leqslant N_I$ and $1 \leqslant j \leqslant N_R$, represents the costs of the edges between I and R. N_I and N_R are the numbers of feature points on I and R, respectively. The function, $dist(a, b)$, is the distance measure of two feature points, a and b, on I and T, respectively. For efficiency purposes, we remove the edge between m_i and m'_j if m_i is not in the tolerance area of m'_j. There is a total of $N_I + N_T + 2$ (with the source s and sink t nodes) nodes in the network. In our application, the capacity on every edge is set to 1, and the costs associated with the edges that come from s and going to t are set to 0. The configuration of the fingerprint matching problem is shown in Fig. 14. The optimal flow value in this network is the number of matched feature points. Because the capacity of every edge is set to 1, there will be no two feature points on I that match with the same feature point of R and vice versa, thus, the one-to-one matching of feature points is guaranteed. Thus, solving the minimum cost flow problem of the generated flow network is equivalent to finding the maximum number of matched feature points (maximum flow) with the minimum total feature distance (minimum cost).

4. Results

Our system has been tested on fingerprint databases of FVC2002 [3]. The DB1 database contains 110 different fingers and 8 impressions of each finger yielding a total of 880 fingerprints (388 pixels × 374 pixels) at 500 dots-per-inch. The DB2 database has the same number of fingerprint images as DB1 but at different size and resolution (296 pixels × 560 pixels at 569 dpi). Each database has two different sets: A and B. Set A contains the fingerprint images from the first 100 fingers, while Set B has the images from the other 10 fingers. We use Set B of each database as our training set for matching parameters and then we perform the experiment on the fingerprints of Set A.

We followed the protocols of FVC2002 [3] to evaluate the FAR (False Accept Rate) and FRR (False Reject Rate) of our system. For FRR, the total number of genuine tests (with no rejection) is $(8 \times 7)/2 \times 100 = 2800$. For FAR, the total number of false acceptance tests (with no rejection) is $(100 \times 99)/2 = 4950$. On an Intel Pentium 4, 1.4 GHz machine, the average matching time for genuine tests is about 929 ms and 76 ms for false acceptance tests. The reasons for faster matching time in cases of false acceptance tests are due to the following: (i) the matching decision of impostor pairs are made earlier, right after the secondary feature matching. There is no need to match the corresponding minutiae. (ii) There are fewer overlapping minutiae. We used the first 1400 results of FRR and FAR testing for neural network training. The experimental results are shown in Figs. 15 and 16. We analyzed the reasons for the false non-matched cases and found that they are mostly caused by spurious minutiae or small overlapped areas. Problems like these make the secondary feature matching difficult but can possibly be solved by applying brute-force matching. However, knowing when to apply the brute-force matching still remains a challenge.

Fig. 17 demonstrates the performance of brute-force matching. The results suggest that there are improvements we can make (from 2.13% EER to 1.01% EER) for DB1. However, it also shows the limitation of the minutiae-based matching systems. Without improving the accuracy of minutiae extraction, the best performance possible is about 1.01% equal error rate.

In order to test the influence of the size of partial fingerprints, we generated two series of partial fingerprint databases with different sizes (in percentage) from the FVC 2002 DB1 data set. (i) The first data set was generated by considering the minutiae at random regions within the print. The region sizes considered were 10%, 15%, 20%, 25%, 30%, 35%, 40%, 45%, 50%, 55%, 60%, 65%, 70%, 75%, 80%, 85%, and 90% of the fingerprint foreground area. (ii) The second data set was generated by considering the central region of the fingerprint only. This is done to simulate conditions where physical guides are used to ensure that the finger is placed centrally on the sensor. We tested the system by matching the partial fingerprint templates of the second impressions against the first impression of every finger in DB1. The system test results for the first case are shown in Fig. 18 and Table 2. The results for the second condition are illustrated in Fig. 19 and Table 3. As one could expect, the performances of the partial fingerprints in the central areas are slightly better than those at random regions due to the relatively larger number of minutiae points. However, the system performance still drops dramatically when the image sizes are smaller than 60% (about 0.32" × 0.46") of the full prints.

5. Summary

Automated partial fingerprint identification is a problem that is not yet solved for generic applications. Proposed

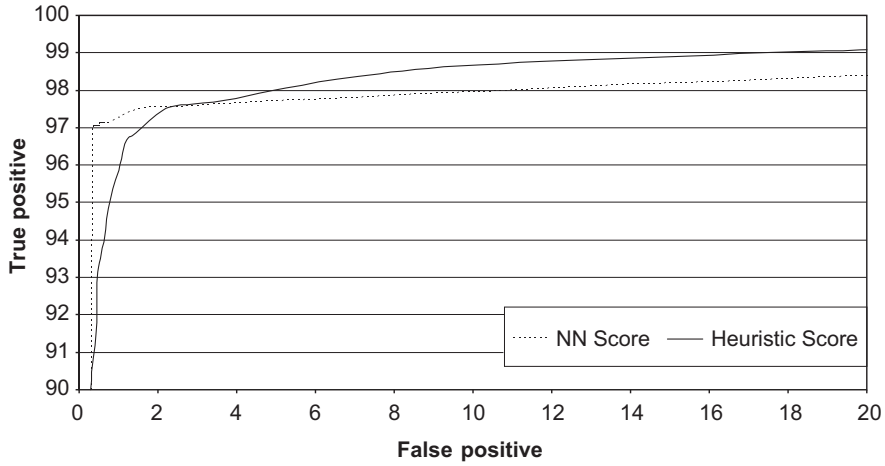

Fig. 15. ROC graph of system testing result on FVC2002 DB1 database. With heuristic rules for similarity scores, the system reaches the minimum total error rate at 4.53% (with FAR at 1.24% and FRR at 3.29%), and EER at 2.39%. With NN scores, the system reaches the minimum total error rate at 3.32% (with FAR at 0.39% and FRR at 2.93%), and EER at 2.13%.

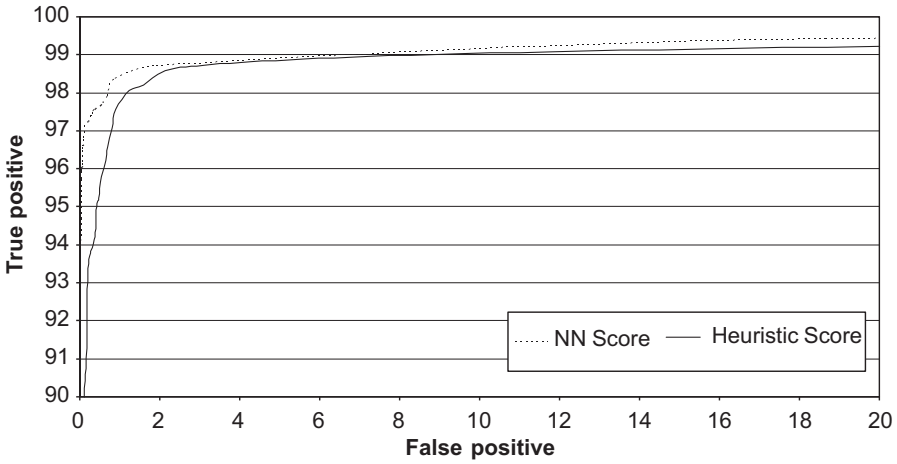

Fig. 16. ROC graph of system testing result on FVC2002 DB2 database. With heuristic rules for similarity scores, the system reaches the minimum total error rate at 3.17% (with FAR at 1.24% and FRR at 1.93%), and EER at 1.69%. With NN scores, the system reaches the minimum total error rate at 2.49% (with FAR at 0.85% and FRR at 1.64%), and EER at 1.57%.

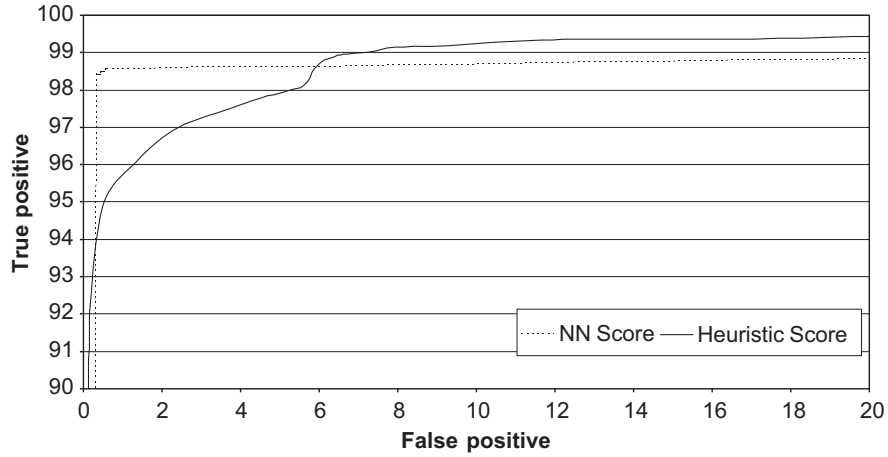

Fig. 17. ROC graph of the brute-force matching with the neural-network generated scores of FVC2002 DB1 database. The system reaches the minimum total error rate at 1.88% (with FAR at 0.31% and FRR at 1.57%), and EER at 1.01%. With heuristic rules for similarity scores, the system reaches the minimum total error rate at 5.27% (with FAR at 1.27% and FRR at 4.00%), and EER at 2.67%.

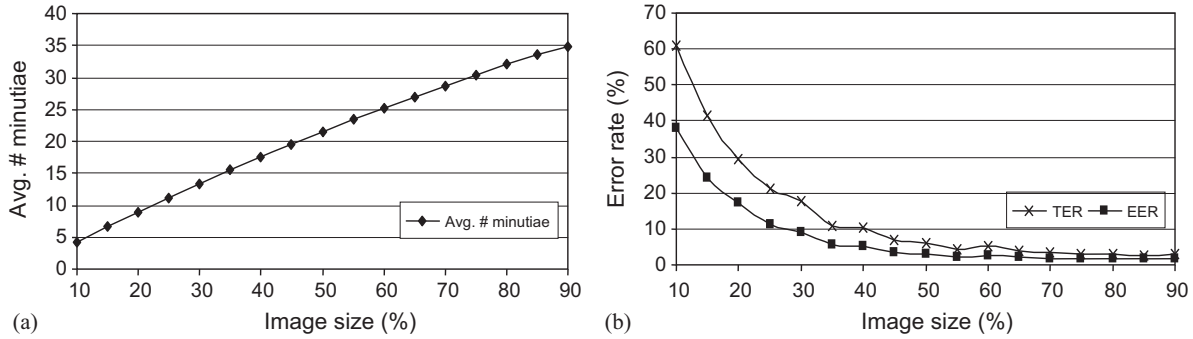

Fig. 18. (a) Shows the relation between the average number of minutiae and image sizes. (b) Shows the system performances vs. different-sized partial fingerprints at random positions.

Table 2
System performance with difference sized partial images at random positions

Size (%)	Avg. width	Avg. height	Avg. minu. num.	Min. total error rate (%)			EER (%)
				FAR	FRR	TER	
90	196.61	283.30	34.89	1.844871	1.090909	2.93578	1.71
80	185.32	267.07	32.00	1.988324	0.909091	2.897415	1.74
70	173.35	249.78	28.74	1.721435	1.818182	3.412844	1.77
60	160.42	231.22	25.28	2.675563	2.363636	5.039199	2.52
50	146.41	211.03	21.56	2.378649	3.636364	5.981651	3.17
40	130.90	188.70	17.49	2.905755	7.454545	10.3603	5.25
30	113.31	163.36	13.36	4.647206	13.09091	17.738115	9.12
20	92.42	133.30	8.84	9.611343	19.63636	29.247707	17.11
10	65.20	94.11	4.23	15.23937	45.63636	60.87573	38.21

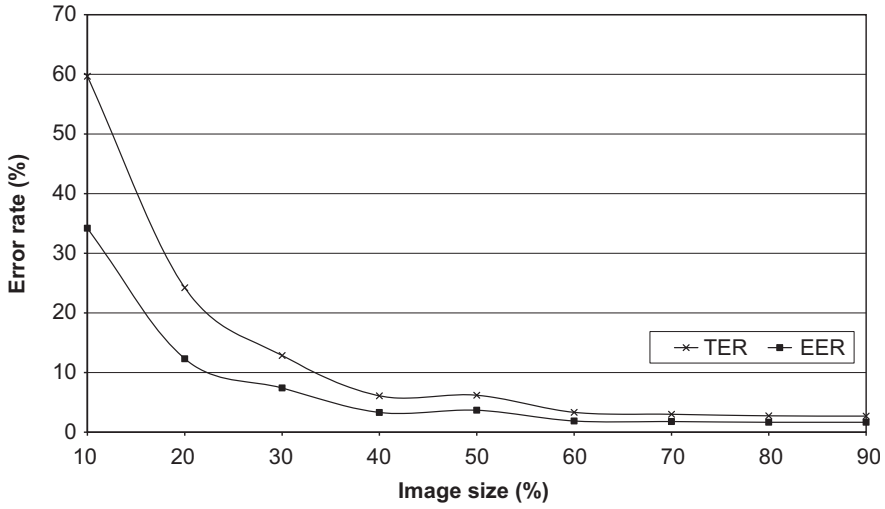

Fig. 19. The system performances vs. different-sized partial fingerprints at central positions.

Table 3
System performance with difference sized partial images around the center of the print

Size (%)	Avg. nimu. num.	Min. total error rate (%)			EER (%)
		FAR	FRR	TER	
90	36.24	1.784821	0.909091	2.693912	1.67
80	33.40	1.834862	0.909091	2.743953	1.68
70	30.33	0.266889	2.727273	2.994162	1.78
60	26.65	2.435363	0.909091	3.344454	1.88
50	22.90	2.552127	3.636364	6.188491	3.69
40	18.56	2.468724	3.636364	6.105088	3.32
30	14.47	3.761284	9.090909	12.852193	7.43
20	9.95	9.681188	14.545455	24.226643	12.31
10	5.02	16.948302	42.727273	59.675575	34.20

matching algorithms overcome the drawbacks of conventional approaches to partial fingerprint matching by using localized "secondary features" and a flow network based brute-force matching. The secondary features and matching algorithm have the following advantages: (i) secondary features are generated from minutiae, so can be easily adapted to existing applications; (ii) secondary features are invariant to orientations, overcoming one of the biggest challenges in partial fingerprint matching and (iii) localized features and dynamic tolerance areas provide the power to handle the spatial distortions. Solving the minutia matching problem by converting it into a minimum cost flow problem gives us an efficient way to find the optimal one-to-one correspondence between minutiae when the number of minutiae is not large. A convex hull-based method of estimating the overlapped areas of query and reference fingerprints is presented. Our experiments show that using a neural network for generating similarity scores improves accuracy. We obtained 1.21% and 0.68% improvements on minimum total error rates of FVC 2002 DB1 and DB2 databases, respectively.

References

[1] American National Standard for Information Systems, Data format for the interchange of fingerprint information, Doc# ANSI/NIST-CSL 1-1993, American National Standards Institute, New York, 1993.

[2] Biometrika inc., A technical evaluation of fingerprint scanners, http://www.biometrika.it/eng/wp_scfing.html, Monte Santo 21, 47100 Forli, Italy.

[3] D. Maltoni, D. Maio, A.K. Jain, S. Prabhakar, Handbook of Fingerprint Recognition, Springer, New York, 2003.

[4] A.K. Jain, L. Hong, S. Pankanti, R. Bolle, An identity-authentication system using fingerprints, Proc. IEEE 85 (9) (1997) 1365–1388.

[5] J.H. Wegstein, An automated fingerprint identification system, U.S. Government Publication, U.S. Department of Commerce, National Bureau of Standards, Washington, DC, 1982.

[6] X. Luo, J. Tian, Y. Wu, A minutia matching algorithm in fingerprint verification, Proceedings of the International Conference on Pattern Recognition (15th), vol. 4, 2000, pp. 833–836.

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

[8] M.D. Garris, C.I. Watson, R. Michael McCabe, C.L. Wilson, User's Guide to NIST Fingerprint Image Software (NFIS), NISTIR 6813, National Institute of Standards and Technology, Gaithersburg, MD, 2001.

[9] X. Jiang, W.-Y. Yau, Fingerprint minutiae matching based on the local and global structures, Internat. Conf. Pattern Recognition 2 (2000) 1038–1041.

[10] T. Jea, V.K. Chavan, V. Govindaraju, J.K. Schneider, Security and matching of partial fingerprint recognition systems, Proc. SPIE 5404 (2004) 39–50.

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

[12] A. Ranade, A. Rosenfeld, Point pattern matching by relaxation, Pattern Recognition 12 (2) (1993) 269–275.

[13] 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.

[14] T.H. Cormen, C.E. Leiserson, R.L. Rivest, Introduction to Algorithms, McGraw-Hill, New York, 1998.

[15] L. Jingping, Algorithms for minimum-cost flows, http://www.csd.uwo.ca/gradstudents/jliu36/min_cost.pdf, 2003.

[16] L.R. Ford, D.R. Fulkerson, Flows in Networks, Princeton University Press, Princeton, NJ, 1962.

[17] J. Edmonds, R. Karp, Theoretical improvements in algorithmic efficiency for network flow problems, J. ACM (1972) 248–264.

[18] A.V. Goldberg, An efficient implementation of a scaling minimum-cost flow algorithm, J. Algorithms 22 (1) (1997) 1–29.

[19] P.T. Sokkalingam, R.K. Ahuja, J.B. Orlin, New polynomial-time cycle-canceling algorithms for minimum-cost flows, Networks 36 (2000) 53–63.