On-line Fingerprint Verification

Anil Jain and Lin Hong
Pattern Recognition and Image Processing Laboratory
Department of Computer Science
Michigan State University
East Lansing, MI 48824, USA
{jain,honglin}@cps.msu.edu

Abstract

We describe the design and implementation of an on-line fingerprint verification system which operates in two stages: (i) minutia extraction and (ii) minutia matching. An improved minutia extraction algorithm that is much faster and more accurate than our earlier algorithm [8] has been implemented. For minutia matching, an alignment-based elastic matching algorithm has been developed. This algorithm is capable of finding the correspondences between input minutiae and the stored template without resorting to exhaustive search and has the ability to adaptively compensate for the nonlinear deformations and inexact pose transformations between fingerprints. The system has been tested on two sets of fingerprint images captured with inkless scanners. The verification accuracy is found to be over 99% with a 15% reject rate. Typically, a complete fingerprint verification procedure takes, on an average, about 8 seconds on a SPARC 20 workstation. It meets the response time requirements of on-line verification with high accuracy.

1. Introduction

Fingerprint verification is one of the most reliable personal identification methods [2, 4]. It plays a very important role in forensic and civilian applications such as criminal identification, access control, and ATM card verification [4]. However, manual fingerprint verification is so tedious, time-consuming, and expensive that it is incapable of meeting today's increasing performance requirements. As a result, automatic fingerprint identification systems (AFIS) are in great demand [4].

In this paper, we will introduce an *on-line fingerprint verification system* which is capable of verifying identities of people in "real time". Such a system has great utility in a variety of personal identification and access control applications. The overall block diagram of our system is shown in Figure 1. It operates as follows: *(i) off-line phase*: a digital image of one fingerprint of a person to be verified is captured; a feature extraction algorithm is applied; minutiae are extracted and stored as a template for later use; *(ii) on-line phase*: the individual to be verified indicates his/her identity and places his/her finger on the inkless scanner; an digital image of his/her fingerprint is captured; minutiae are extracted from the captured image and fed to a matching algorithm, which matches it against the stored template.

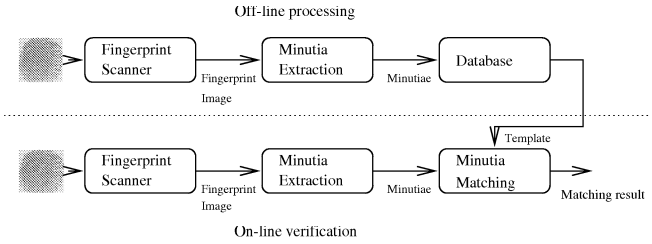


Figure 1. Overview of the system

In the following sections, we will describe in detail our on-line fingerprint verification system. Section 2 mainly discusses the feature extraction algorithm. Section 3 presents our minutia matching algorithm. Experimental results on fingerprint databases captured with inkless scanners are described in section 4. Section 5 contains the summary and discussion.

2. Minutia Extraction

We have implemented a minutia detection algorithm which is a modified version of the technique proposed by Ratha *et al.* [8]. The overall flowchart is depicted in Figure 2.

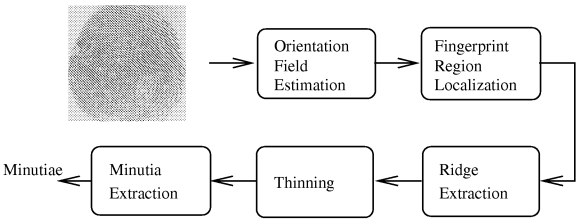


Figure 2. Flowchart of the minutia extraction algorithm

2.1. Estimation of Orientation Field

A new hierarchical implementation of Rao's algorithm [7] is used to estimate the orientation field of an input fingerprint image. It consists of the following two main steps:

1. Estimate the local orientation at each pixel using following formula:

$$\theta_o = \frac{1}{2} tan^{-1}\left(\frac{\sum_{i=1}^{W}\sum_{j=1}^{W} 2G_x(i,j)G_y(i,j)}{\sum_{i=1}^{W}\sum_{j=1}^{W}(G_x^2(i,j) - G_y^2(i,j))}\right), \quad (1)$$

   where W is the size of the local window; G_x and G_y are the gradient magnitudes in x and y directions, respectively.

2. Compute the *variance* of the orientation field in a local neighborhood at each pixel (i,j). If it is above a certain threshold T_v, then the local orientation at this pixel is re-estimated at a lower resolution level until it is above the threshold value.

Experimental results show that even in the presence of noise, smudges, and breaks in ridges, a fairly smooth orientation field estimate can be obtained with this algorithm. Before the ridge detection, a segmentation algorithm which bases its judgment on the local variance of grey levels is used to locate the fingerprint region in the image.

2.2. Ridge Detection

The most salient property corresponding to ridges in a fingerprint image is that grey level values on ridges attain their local maxima along the normal directions of local ridges. In our minutia detection algorithm, a fingerprint image is first convolved with the following two masks, $h_t(x,y;u,v)$ and $h_b(x,y;u,v)$ of size $W \times H$, which are capable of adaptively accentuating the local maximum grey level values along the normal direction of the local ridge directions:

$$h_t(x,y;u,v) = \begin{cases} \frac{-1}{\sqrt{2\pi}\delta} e^{\frac{-u}{\delta^2}}, & \text{if } u = l(v) - d, v \in \Omega \\ \frac{1}{\sqrt{2\pi}\delta} e^{\frac{-u}{\delta^2}}, & \text{if } u = l(v), v \in \Omega \\ 0, & \text{otherwise,} \end{cases} \quad (2)$$

$$h_b(x,y;u,v) = \begin{cases} \frac{-1}{\sqrt{2\pi}\delta} e^{\frac{-u}{\delta^2}}, & \text{if } u = l(v) + d, v \in \Omega \\ \frac{1}{\sqrt{2\pi}\delta} e^{\frac{-u}{\delta^2}}, & \text{if } u = l(v), v \in \Omega \\ 0, & \text{otherwise,} \end{cases} \quad (3)$$

$$l(v) = v\tan(\theta(x,y)), \quad (4)$$

$$d = \frac{H}{2\cos(\theta(x,y))}, \quad (5)$$

$$\Omega = H\left[\left|\frac{\sin(\theta(x,y))}{-2}\right|, \left|\frac{\sin(\theta(x,y))}{2}\right|\right], \quad (6)$$

where $\theta(x,y)$ represents the local ridge orientation at pixel (x,y). If *both* the grey level values at pixel (x,y) of the convolved images are larger than a certain threshold T_r, then pixel (x,y) is labeled as a ridge. By adapting the mask width to the width of the local ridge, this algorithm can efficiently locate the ridges in a fingerprint image. After the above steps, a thinning algorithm is applied to obtain a thinned 8-connected ridge map.

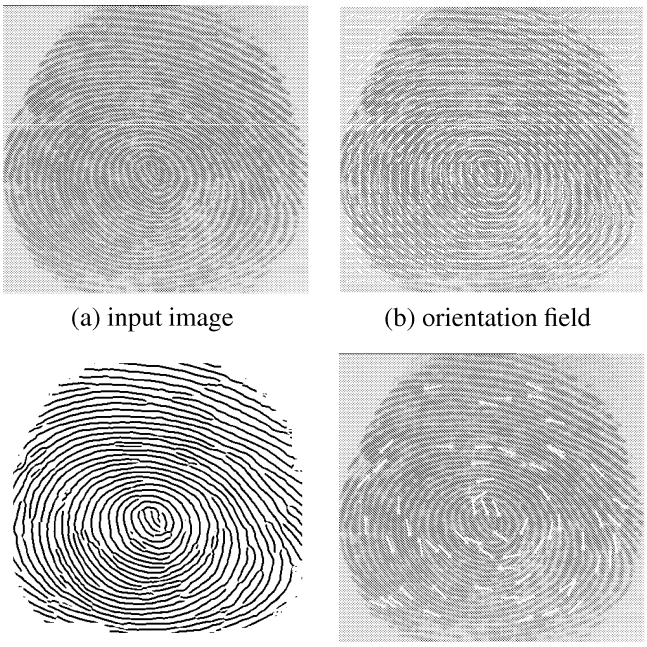
2.3. Minutia Detection

Without a loss of generality, we can assume that if a pixel is on a thinned ridge (8-connected), then it has a value 1, and 0 otherwise. Let $N_0, N_1, ..., N_7$ denote the 8 neighbors of a given pixel (x,y), then pixel (x,y) is a ridge ending if $\sum_{i=0}^{8} N_i = 1$ and a ridge bifurcation if $\sum_{i=0}^{8} N_i > 2$. However, the presence of undesired spikes and breaks present in a thinned ridge map may lead to many spurious minutiae being detected. Therefore, the following heuristics are used in preprocessing: (i) If a branch in a ridge map is orthogonal to the local ridge directions and its length is less than a specified threshold T_b, then it will be removed; (ii) If a break in a ridge is short enough and no other ridges pass through it, then it will be connected.

For each minutia, the following parameters are recorded: *(i) x-coordinate, (ii) y-coordinate, (iii) orientation, and (iv) the ridge on which it resides*. The recorded ridges are normalized by average local ridge distance along the normal direction of local ridges. Figure 3 shows the results of our minutia extraction algorithm on a fingerprint image captured with an inkless scanner.

3. Minutia Matching

Generally, an automatic fingerprint verification is achieved with minutia matching (point pattern matching) instead of a pixel-wise matching or a ridge pattern



(a) input image (b) orientation field

(d) ridge map (f) extracted minutiae

Figure 3. Results of the minutia extraction algorithm on a fingerprint image (640×480) captured with an inkless scanner.

matching of fingerprint images. Because a general point matching problem is essentially intractable, features associated with each point and their relative positions are widely used in the point pattern matching algorithms to reduce the exponential number of search paths [3, 1, 6, 9]. However, these algorithms are inherently slow and are unsuitable for an on-line fingerprint verification system. In our system, an alignment-based matching algorithm is implemented, which decomposes the minutia matching into two stages: (i) *alignment stage*, where transformations such as translation, rotation and scaling between an input pattern and a template are first estimated; the input patterns are then aligned with the template according to the estimated parameters; and (ii) *matching stage*, where both the input pattern and the template are converted to polygons in polar space and an elastic string matching algorithm is used to match the resulting polygons.

Let $P = ((x_1^P, y_1^P, \theta_1^P)^T, ..., (x_M^P, y_M^P, \theta_M^P)^T)$ and $Q = ((x_1^Q, y_1^Q, \theta_1^Q)^T, ..., (x_M^Q, y_N^Q, \theta_N^Q)^T)$ denote the M minutiae in the template and the N minutiae in the input image, respectively. The steps of our alignment-based matching algorithm are given below:

1. Match the ridge associated with each input minutia against the ridge associated with each template minutia and align the two patterns according to the matching result.

2. Convert the representations of template and input minutiae into the polar coordinate representations with respect to the corresponding minutia on which alignment is performed and represent them as two symbolic strings by concatenating each minutia in an increasing order of radial angles:

$$P_p = ((r_1^P, e_1^P, \theta_1^P)^T, ..., (r_M^P, e_M^P, \theta_M^P)^T) \quad (7)$$
$$Q_p = ((r_1^Q, e_1^Q, \theta_1^Q)^T, ..., (r_N^Q, e_N^Q, \theta_N^Q)^T), \quad (8)$$

   where r_*, e_*, and θ_* represent the corresponding radius, radial angle, and normalized minutia orientation with respect to the reference minutia, respectively.

3. Match the resulting strings P_p and Q_p with a modified dynamic-programming algorithm described below to find the 'edit distance' between P_p and Q_p.

4. Compute the matching score of the template and input minutiae as the minimum 'edit distance'.

$$M_{pq} = \frac{100 N_{pair}}{\max\{M, N\}}, \quad (9)$$

   where N_{pair} is the number of minutia pairs which fall within a given bounding box.

The maximum and minimum values of the matching score are 100 and 1, respectively. The former value indicates a perfect match, while the later value indicates no match at all.

3.1. Alignment of Point Patterns

It is well known that corresponding curve segments are capable of aligning two point patterns with high accuracy in the presence of noise and deformations. Each minutia in a fingerprint is associated with a ridge. A true alignment can be achieved by matching and aligning the corresponding ridges (see Figure 4). By matching the corresponding normalized ridges, the relative pose transformation between the input minutiae and the template can be estimated. With the estimated pose transformation, the input minutiae can then be translated and rotated to align the template minutiae.

3.2. Aligned Point Pattern Matching

If two identical point patterns are exactly aligned, each pair of corresponding points are completely overlapping. In such a case, a point pattern matching can be simply achieved by counting the number of overlapping pairs. However, in practice, such a situation is rarely encountered.

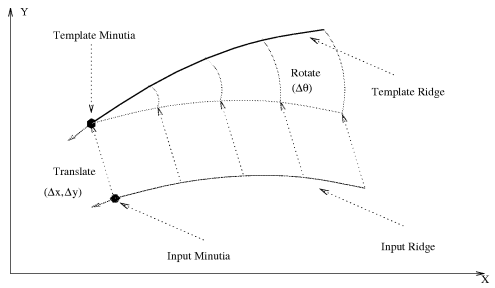


Figure 4. Alignment of the input ridge and the template ridge.

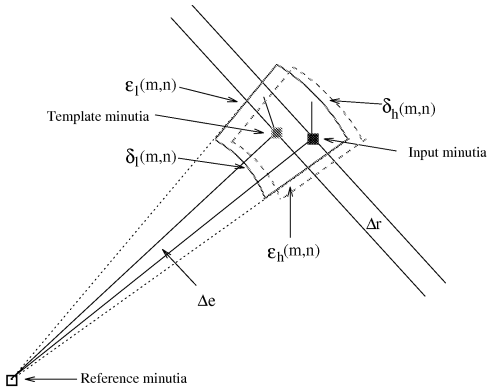


Figure 5. Bounding box and its adjustment.

The error in determining and localizing minutia hinders the alignment algorithm to recover the relative pose transformation exactly. Also, in the presence of nonlinear deformation of fingerprints which is an inherent nature of fingerprint impressions, it is impossible to exactly recover the position of each minutia with respect to its corresponding minutia in the template. Therefore, the aligned point pattern matching algorithm needs to be able to tolerate, to some extent, the deformations due to inexact extraction of minutia positions and nonlinear deformations.

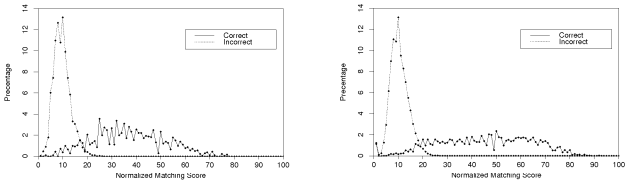
The symbolic string generated by concatenating points in an increasing order of radial angle uniquely represents a point pattern. A modified string matching algorithm which incorporates an elastic term is capable of achieving a certain type of tolerance. However, this algorithm can only tolerate, but not compensate, the adverse effect on the matching produced by the inexact localization of minutia and nonlinear deformations. Therefore, an adaptive mechanism is needed, which should be able to track the local nonlinear deformations and inexact alignment and try to alleviate them during the minimization process. In our string matching algorithm, this adaptation is achieved by adjusting the bounding box (Figure 5) when an inexact match is found during the matching process. It can be represented as follows:

$$\delta_l(m+1,n+1) = \delta_l(m,n) + \eta\Delta r_a, \quad (10)$$
$$\delta_h(m+1,n+1) = \delta_h(m,n) + \eta\Delta r_a, \quad (11)$$
$$\epsilon_l(m+1,n+1) = \epsilon_l(m,n) + \eta\Delta e_a, \quad (12)$$
$$\epsilon_h(m+1,n+1) = \epsilon_h(m,n) + \eta\Delta e_a, \quad (13)$$

where $\delta_l(m,n)$, $\delta_h(m,n)$, $\epsilon_l(m,n)$, and $\epsilon_h(m,n)$ specify the adaptive bounding box in radius and radial angle; Δr_a and Δe_a represent the current difference of input and template in radius and radial angle, respectively; η is the learning rate. In our system, the initial values of the bounding box are specified as follows: $\delta_l(0,0) = -8$; $\delta_h(0,0) = +8$; $\epsilon_l(0,0) = -4$; $\epsilon_h(0,0) = +4$. The learning rate, η, is 0.5.

4. Experimental Results

We have tested our system on two sets of fingerprint images. Set 1 contains 10 images (380 × 380) per finger from 18 individuals for a total of 180 images, which were captured with a scanner manufactured by Identix. Set 2 contains 10 images (640 × 480) per finger from 30 individuals for a total of 300 images, which were captured with a scanner manufactured by Digital Biometrics. The captured fingerprint images vary in quality. Approximately 90% are of satisfactory quality, while about 10% are of poor quality.



(a) Identix (b) Digital Biometrics

Figure 6. Distributions of correct and incorrect matching scores: vertical axis represents distribution of matching scores in percentage.

4.1. Matching

Each fingerprint in the test set was matched with the other fingerprints in the set. A matching was labeled correct if the matched fingerprint was among the 9 other fingerprints of the same individual, and incorrect otherwise. The distributions of matching scores are shown in Figure 6. This result indicates that our algorithm is capable of effectively differentiating fingerprints by setting an appropriate value of the matching threshold. Table 1 shows the recognition rates and reject rates with different threshold values

on the matching score. We have observed that the incorrect matchings occur mainly due to fingerprint images with poor quality.

Threshold Value	Recognition Rate (Set 1)	Reject Rate (Set 1)	Recognition Rate (Set 2)	Reject Rate (Set 2)
20	99.84%	11.23%	99.49%	8.67%
24	99.98%	16.50%	99.90%	14.06%
28	99.99%	25.22%	99.99%	18.85%
32	100%	27.72%	99.99%	22.08%

Table 1. Recognition and reject rates on test sets with different threshold values

4.2. Verification

In on-line verification, a user indicates his/her identity. Therefore, the system matches the input fingerprint image only to his/her stored template. To determine the verification accuracy of our system, we used each one of our database images as template and all the others as input fingerprints which need to be verified. An input fingerprint is matched against the template. If the matching score is below the threshold value of 25, then the input fingerprint is rejected. If the matching score exceeds 25, then a valid verification is established. With this scheme, based on 89,700 (300×299) verifications, a 99.99% verification rate can be achieved with a tolerable reject rate (15%).

In order for an on-line fingerprint verification system to be acceptable in practice, the response time of the system needs to be within a few seconds. Table 2 shows that our on-line fingerprint verification system does meet the response time requirement of on-line verification.

Minutia Extraction (second)	Minutia Matching (second)	Total (second)
5.35	2.55	7.90

Table 2. Average CPU time for minutia extraction and matching on a SPARC 20 workstation.

5. Conclusions

We have introduced an on-line fingerprint verification system. The implemented minutia extraction algorithm is accurate and fast in minutia extraction. The alignment-based elastic matching algorithm is capable of finding the correspondences between minutiae without resorting to exhaustive search. It provides a good performance, because it has the ability to adaptively compensate for the nonlinear deformations and inexact pose transformations between different fingerprints. Experimental results show that our system achieves excellent performance in a real on-line verification environment. It meets the response time requirements of on-line verification with high accuracy.

Based on the experimental results, we observe that the matching errors of our system mainly result from (i) incorrect minutiae and (ii) inaccurate alignment. A number of factors are detrimental to the correct location of minutia. Among them, poor image quality is the most serious one. We are currently investigating a number of image enhancement schemes.

References

[1] N. Ansari, M. H. Chen, and E. S. H. Hou, A Genetic Algorithm for Point Pattern Matching, Chapter 13 in *Dynamic, Genetic, and Chaotic Programming* by B. Souček and the IRIS Group. John Wiley & Sons, 1992.

[2] Federal Bureau of Investigation, The Science of Fingerprints: Classification and Uses, U.S. Government Printing Office, Washington, D. C. 1984.

[3] S. Gold and A. Rangarajan, A Graduated Assignment Algorithm for Graph Matching, *IEEE Transactions on PAMI,* Vol. 18, No. 4, pp. 377-388, 1996.

[4] Henry C. Lee and R. E. Gaensslen, editors, Advances in Fingerprint Technology, Elsevier, New York, 1991.

[5] B. Miller, Vital Signs of Identity, *IEEE Spectrum,* Vol. 31, No. 2, pp. 22-30, 1994.

[6] A. Ranade and A Rosenfeld, Point Pattern Matching by Relaxation, *Pattern Recognition*, Vol. 12, No. 2, pp. 269-275, 1993.

[7] A. Ravishankar Rao, A Taxonomy for Texture Description and Identification, Springer-Verlag, New York, 1990.

[8] N. Ratha, S. Chen, and A. K. Jain, Adaptive Flow Orientation Based Feature Extraction in Fingerprint Images, *Pattern Recognition*, Vol. 28, No. 11, pp. 1657-1672, 1995.

[9] J. Ton and A. K. Jain, Registering Landsat Images by Point Matching, *IEEE Transactions on Geoscience and Remote Sensing,* Vol. 27, No. 5, pp. 642-651, 1989.