Identity Authentication Using Fingerprints

Lin Hong[1], Anil Jain[1], Sharath Pankanti[2], and Ruud Bolle[2]

[1]Department of Computer Science
Michigan State University
East Lansing, MI 48824
{honglin,jain}@cps.msu.edu

[2]Exploratory Computer Vision Group
IBM T. J. Watson Research Center
Yorktown Heights, NY 10598
{sharat,bolle}@watson.ibm.com

Abstract

We describe the design and implementation of an automatic identity authentication system which uses fingerprint to establish the identity of an individual. An improved minutia extraction algorithm that is much faster and more accurate than our earlier algorithm [12] has been implemented. An alignment-based elastic matching algorithm has been developed. This algorithm is capable of finding the correspondences between input minutia pattern and the stored template minutia pattern without resorting to exhaustive search and has the ability to adaptively compensate for the nonlinear deformations and inexact pose transformations between an input fingerprint and a template. The system has been tested on the MSU fingerprint database. A perfect authentication rate can be achieved with a 15% false reject rate on this data set. Typically, a complete authentication procedure takes, on an average, about 1.4 seconds on a Sun ULTRA 1 workstation.

Keywords: fingerprints, matching, authentication, minutia, orientation field, ridge extraction.

1. Introduction

Automatic identity authentication is becoming more and more important in our modern society [2, 3]. A number of automatic identity authentication techniques have been investigated, including blood vessel patterns in the retina or hand, fingerprint, hand geometry, iris, signature, and voiceprints [2]. Among them, fingerprint is one of the most reliable techniques [3, 5]. In this paper, we will introduce an *automatic identity authentication system* which is capable of authenticating the identity of an individual automatically using his/her fingerprints. Such a system has great utility in a variety of identity authentication applications such as access control and credit card verification [5].

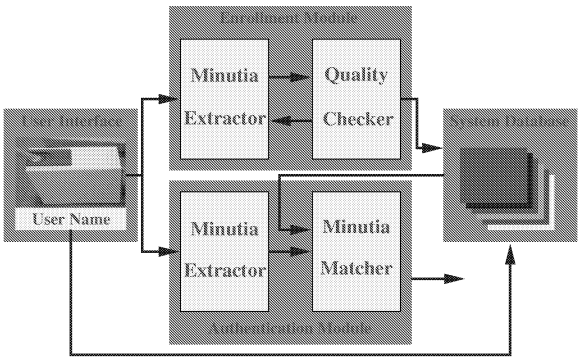


Figure 1. The system architecture of the automatic identity authentication system

It is widely known that a professional fingerprint examiner relies on minute details of ridge structures to match fingerprints [5, 3]. The topological structure of the minute details of ridge structures of a fingerprint is unique and invariant with aging and impression deformations [5, 3]. Eighteen different types of local ridge descriptions have been identified [5]. Among them, the two most prominent minutia details that are suitable for automatic detection from input fingerprint images are ridge endings and ridge bifurcations which are usually called minutiae (Figure 2). Therefore, in an automatic identity authentication system using fingerprint, the two most important components are: (i) minutia extraction which detects minutiae from input fingerprint images, and (ii) minutia pattern matching which matches two minutia patterns to establish the identity of an individual.

The system architecture of our automatic identity authentication system is shown in Figure 1. It consists of four components: (i) user interface, (ii) system database, (iii) enrollment module, and (iv) authentication module. The user interface provides a mechanism for a user to indicate his/her identity and input his/her fingerprint into the sys-

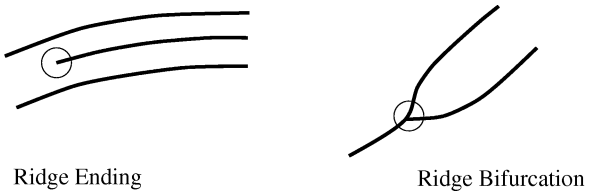


Figure 2. Ridge ending and ridge bifurcation

tem. The system database consists of a collection of records each of which corresponds to an authorized individual that has access to the system. Each record contains the following fields: (i) user name of the individual, and (ii) several minutia patterns of the individual's finger.

The task of enrollment module is to enroll individuals and their fingerprints into the system database. When the fingerprint images and the user name of an individual to be enrolled are fed to the enrollment module, a minutia extraction algorithm is first applied to the fingerprint images and the minutia patterns are extracted. A quality checking algorithm is used to ensure that the records in the system database only consist of minutia patterns of good quality [6]. This is important for the performance of our system. A fingerprint image of poor quality is enhanced to improve the clarity of ridge/valley structures and mask out all the regions that can not be reliably recovered [6]. The enhanced fingerprint image is fed to the minutia extractor. If at least 20 minutia points are recovered, then the fingerprint is accepted. Otherwise, it is rejected. Because the current quality checking algorithm is very slow [6], it is only used in the enrollment module.

The task of authentication module is to authenticate the identity of the individual who intends to access the system. The individual indicates his/her identity and places his/her finger on the fingerprint scanner; a digital image of his/her fingerprint is captured; minutia pattern is extracted from the captured fingerprint image and fed to a matching algorithm; the matching algorithm then matches it against the individual's minutia patterns stored in the system database to establish the identity. Figure 3 shows the graphic user interface (GUI) of our automatic identity authentication system.

In the following sections, we will describe in detail our automatic identity authentication system. Section 2 mainly discusses the feature extraction algorithm. Section 3 presents our minutia matching algorithm. Experimental results on the MSU fingerprint databases are described in Section 4. Section 5 contains the summary and discussion.

2. Minutia Extraction

Because fingerprint authentication is based on the topological structural matching of the minutia pattern, a reliable minutia extraction algorithm is critical to the performance

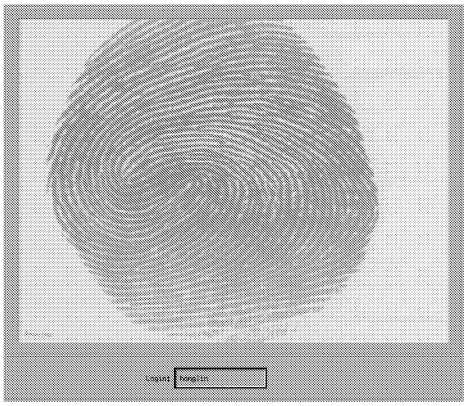


Figure 3. GUI of the authentication system

of an automatic identity authentication system. In our system, we have implemented a minutia extraction algorithm which is an improved version of the technique proposed in [12]. Experimental results show that this algorithm performs very well in operation. The overall flowchart of our minutia extraction algorithm is depicted in Figure 4.

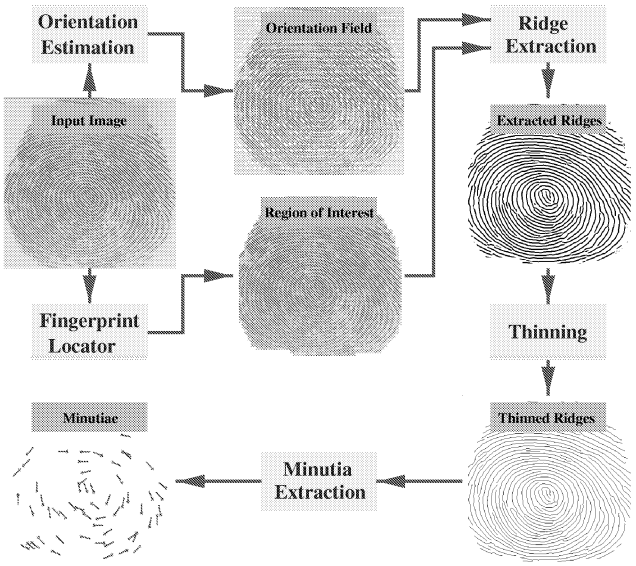


Figure 4. Flowchart of the minutia extraction algorithm

2.1. Estimation of Orientation Field

A new hierarchical orientation field estimation algorithm has been implemented 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 (i, j) using the approach in [11]:

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.

A segmentation algorithm which is based on the local certainty level of the orientation field is used to locate the region of interest within the input fingerprint image. The certainty level $C(i, j)$ of the orientation field at pixel (i, j) is defined as follows:

$$C(i,j) = \frac{1}{W \times W} \frac{(V_x(i,j)^2 + V_y(i,j)^2)}{V_e(i,j)}, \quad (1)$$

where

$$V_x(i,j) = \sum_{u=i-\frac{W}{2}}^{i+\frac{W}{2}} \sum_{v=j-\frac{W}{2}}^{j+\frac{W}{2}} 2G_x(u,v)G_y(u,v), \quad (2)$$

$$V_y(i,j) = \sum_{u=i-\frac{W}{2}}^{i+\frac{W}{2}} \sum_{v=j-\frac{W}{2}}^{j+\frac{W}{2}} (G_x^2(u,v) - G_y^2(u,v)), \quad (3)$$

$$V_e(i,j) = \sum_{u=i-\frac{W}{2}}^{i+\frac{W}{2}} \sum_{v=j-\frac{W}{2}}^{j+\frac{W}{2}} (G_x^2(u,v) + G_y^2(u,v)), \quad (4)$$

and W is the size of the local window; G_x and G_y are the gradient magnitudes in x and y directions, respectively. Physically, $C(i, j)$ reflects the coherence level of the ridge structures within the neighborhood of (i, j). For each pixel, if the certainty level of the orientation field is below a threshold T_s, then the pixel is marked as a background pixel. In our localization algorithm, we assume that there is only one fingerprint present 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:

$$h_t(x,y;u,v) = \begin{cases} \frac{-1}{\sqrt{2\pi}\delta} e^{\frac{-u}{\delta^2}}, & \text{if } u = l(v) - d, v \in \Omega \\ \frac{1}{\sqrt{2\pi}\delta} e^{\frac{-u}{\delta^2}}, & \text{if } u = l(v), v \in \Omega \\ 0, & \text{otherwise,} \end{cases} \quad (5)$$

$$h_b(x,y;u,v) = \begin{cases} \frac{-1}{\sqrt{2\pi}\delta} e^{\frac{-u}{\delta^2}}, & \text{if } u = l(v) + d, v \in \Omega \\ \frac{1}{\sqrt{2\pi}\delta} e^{\frac{-u}{\delta^2}}, & \text{if } u = l(v), v \in \Omega \\ 0, & \text{otherwise,} \end{cases} \quad (6)$$

$$l(v) = v \tan(\theta(x,y)), \quad (7)$$

$$d = \frac{H}{2\cos(\theta(x,y))}, \quad (8)$$

$$\Omega = H \left[\left| \frac{\sin(\theta(x,y))}{-2} \right|, \left| \frac{\sin(\theta(x,y))}{2} \right| \right], \quad (9)$$

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 ridges, 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.

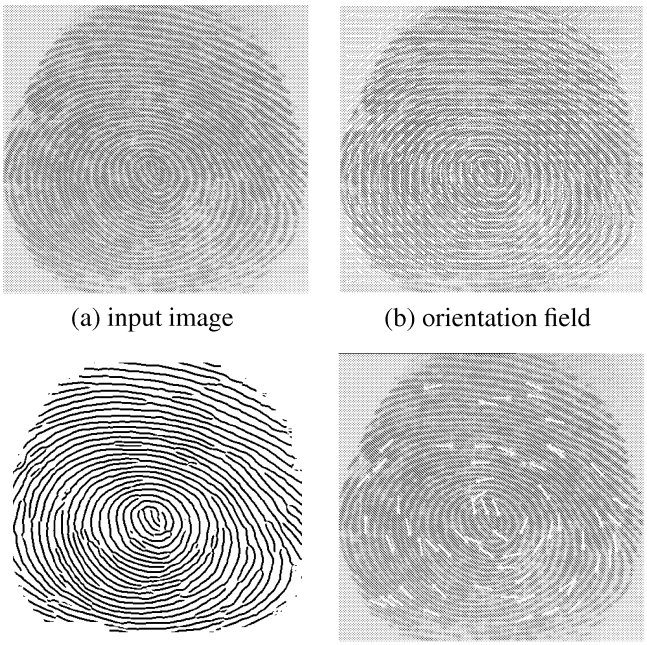
2.3. Minutia Detection

Without 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 which is defined as the local ridge orientation of the associated ridge, and (iv) the associated ridge. The recorded ridges are represented as one-dimensional discrete signals which are normalized by the average inter-ridge distance. These recorded ridges are used for alignment in the minutia matching algorithm. Figure 5 shows the results of our minutia extraction algorithm on a fingerprint image.

3. Minutia Matching

Generally, fingerprint matching is achieved with minutia matching (point pattern matching) instead of a pixel-wise



(a) input image (b) orientation field

(d) ridge map (f) extracted minutiae

Figure 5. Results of minutia extraction algorithm on a fingerprint image (640×480) captured with an inkless scanner.

matching or a ridge pattern matching of input fingerprint images. Because the general point matching problem is essentially intractable, features associated with each point and inter-point distances are widely used in the point pattern matching algorithms to reduce the exponential number of search paths [4, 1, 10, 8]. However, these algorithms are inherently slow and are unsuitable for an automatic identity authentication 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 parameters between an input minutia pattern and a template minutia pattern are first estimated; the input minutia pattern is aligned with the template minutia pattern according to the estimated parameters; and (ii) *matching stage*, where both the input minutia pattern and the template are converted to polygons in polar coordinates and an elastic string matching algorithm is used to match the resulting polygons.

Let $P = ((x_1^P, y_1^P, \theta_1^P), ..., (x_M^P, y_M^P, \theta_M^P))$ and $Q = ((x_1^Q, y_1^Q, \theta_1^Q), ..., (x_N^Q, y_N^Q, \theta_N^Q))$ 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. Estimate the translation and rotation parameters between the ridge associated with each input minutia and the ridge associated with each template minutia and align the two minutia patterns according to the estimated parameters.

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

$$P_p = ((r_1^P, e_1^P, \theta_1^P), ..., (r_M^P, e_M^P, \theta_M^P)) \quad (10)$$

$$Q_p = ((r_1^Q, e_1^Q, \theta_1^Q), ..., (r_N^Q, e_N^Q, \theta_N^Q)), \quad (11)$$

   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. Use the minimum edit distance between P_p and Q_p to establish the correspondence of the minutiae between P_p and Q_p. The matching score, S, is then defined as:

$$S = \frac{100 M_{PQ} M_{PQ}}{MN}, \quad (12)$$

   where M_{PQ} is the number of the minutiae which fall in the bounding boxes of template minutiae. The bounding box of a minutia specifies all the possible positions of the corresponding input minutia with respect to the template minutia.

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 [7]. 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 6). By matching the corresponding normalized ridges, the relative pose transformation between the input minutia pattern and the template minutia pattern can be estimated. With the estimated pose transformation, the input minutia pattern can then be translated and rotated to align the template minutia pattern.

3.2. Aligned Point Pattern Matching

If two identical point patterns are exactly aligned, each pair of corresponding points is completely overlapping. In such a case, a point pattern matching can be simply

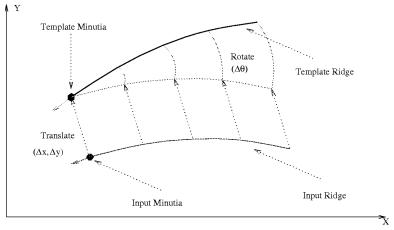


Figure 6. Alignment of the input ridge and the template ridge.

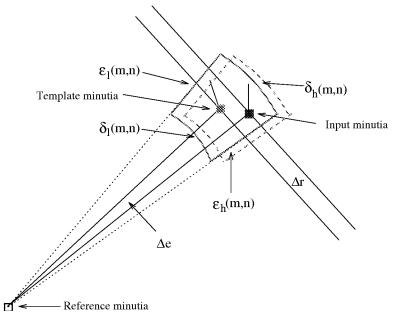


Figure 7. Bounding box and its adjustment.

achieved by counting the number of overlapping pairs. However, in practice, such a situation is rarely encountered. The error in determining and localizing minutia hinders the alignment algorithm to recover the relative pose transformation exactly. Also, due to the inherent nonlinear deformation of fingerprints, 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 with a linear prediction schema which is capable of predicting the new position of the bounding box (Figure 7) when an inexact match is encountered during the matching process. Figure 8 shows an example of applying the matching algorithm to a pair of minutia patterns.

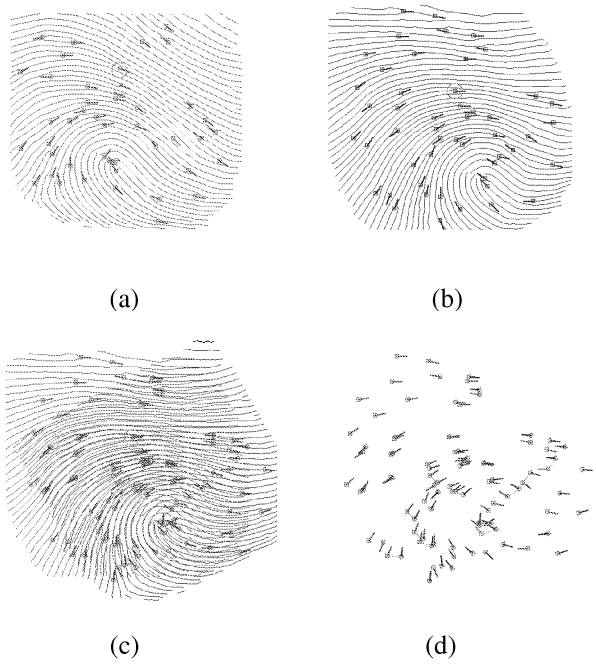


Figure 8. Matching result; (a) input minutia pattern; (b) template minutia pattern; (c) alignment result based on the minutiae marked with green circles; (d) matching result where template minutiae and their correspondences are connected by green lines.

4. Experimental Results

We have tested our system on the MSU fingerprint database. It contains 10 images (640×480) per finger from 70 individuals for a total of 700 fingerprint 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.

For each individual in the database, we select only 3 fingerprint images which pass the quality checking as the template minutia patterns for the individual and insert them into the system database. There are 6 individuals who can not be enrolled into the system database, because the quality of the captured fingerprints for them is too poor to pass the quality checking. The remaining 490 (70×7) fingerprint images are used as input fingerprints to test the performance of the system. An identity is established if at least one of the 3 matching scores is above a certain threshold value. Otherwise, the input fingerprint is rejected as an imposter. The false acceptance rates and false reject rates with different threshold values on the matching score are shown in

Table 1, which are obtained based on 31,360 (64 $\times$ 490) matches. We have observed that the incorrect matches and the false reject rates occur mainly due to fingerprint images of poor quality.

Threshold Value	False Acceptance Rate	False Reject Rate
7	0.07%	7.1%
8	0.02%	9.4%
9	0.01%	12.5%
10	0	14.3%

Table 1. False acceptance and false reject rates on test sets with different threshold values

In order for an automatic identity authentication system to be acceptable in practice, the response time of the system needs to be within a few seconds. Table 2 shows that our implemented system does meet the practical response time requirement.

Minutia Extraction (seconds)	Minutia Matching (seconds)	Total (seconds)
1.1	0.3	1.4

Table 2. Average CPU time for minutia extraction and matching on a Sun ULTRA 1 workstation.

5. Conclusions

We have proposed an automatic identity authentication system using fingerprint. In our system, the implemented feature extraction algorithm is accurate and fast in minutia extraction. The proposed alignment-based elastic matching algorithm is capable of finding the correspondences between minutiae without resorting to exhaustive search. It can achieve an excellent 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 performs very well in a real operational environment. It meets the response time requirements with a high accuracy.

Based on the experimental results, we observe that the matching errors of our system mainly result from (i) incorrect minutia extraction, 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 an image enhancement scheme which can be integrated into the authentication module without a large increase in the execution time of the authentication procedure.

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] J. G. Daugman, High Confidence Visual Recognition of Persons by a Test of Statistical Independence, *IEEE Transactions on PAMI,* Vol. 15, No. 11, pp. 1148-1161, 1993.

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

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

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

[6] L. Hong, A. K. Jain, S. Pankanti, and R. Bolle, Fingerprint Enhancement, *to appear in the Proc. IEEE Workshop on Applications of Computer Vision*, Sarasota, FL, 1996.

[7] D. P. Huttenlocher and S. Ullman, Object Recognition Using Alignment, *Proc. First Intern. Conf. Comput. Vision*, London, pp. 102-111, 1987.

[8] A. Jain and L. Hong, On-line Fingerprint Verification, *Proc. 13th ICPR*, Vienna, pp. 596-600, 1996.

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

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

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

[12] N. Ratha, K. Karu, S. Chen and A. K. Jain, A Real-time Matching System for Large Fingerprint Database, *IEEE Trans. on PAMI*, Vol. 18, No. 8, pp. 799-813, 1996.