Filterbank-Based Fingerprint Matching

Anil K. Jain, *Fellow, IEEE*, Salil Prabhakar, Lin Hong, and Sharath Pankanti

Abstract—With identity fraud in our society reaching unprecedented proportions and with an increasing emphasis on the emerging automatic personal identification applications, biometrics-based verification, especially fingerprint-based identification, is receiving a lot of attention. There are two major shortcomings of the traditional approaches to fingerprint representation. For a considerable fraction of population, the representations based on explicit detection of complete ridge structures in the fingerprint are difficult to extract automatically. The widely used minutiae-based representation does not utilize a significant component of the rich discriminatory information available in the fingerprints. Local ridge structures cannot be completely characterized by minutiae. Further, minutiae-based matching has difficulty in quickly matching two fingerprint images containing different number of unregistered minutiae points. The proposed filter-based algorithm uses a bank of Gabor filters to capture both local and global details in a fingerprint as a compact fixed length FingerCode. The fingerprint matching is based on the Euclidean distance between the two corresponding FingerCodes and hence is extremely fast. We are able to achieve a verification accuracy which is only marginally inferior to the best results of minutiae-based algorithms published in the open literature [1]. Our system performs better than a state-of-the-art minutiae-based system when the performance requirement of the application system does not demand a very low false acceptance rate. Finally, we show that the matching performance can be improved by combining the decisions of the matchers based on complementary (minutiae-based and filter-based) fingerprint information.

Index Terms—Biometrics, FingerCode, fingerprints, flow pattern, Gabor filters, matching, texture, verification.

I. INTRODUCTION

WITH THE advent of electronic banking, e-commerce, and smartcards and an increased emphasis on the privacy and security of information stored in various databases, *automatic* personal identification has become a very important topic. Accurate automatic personal identification is now needed in a wide range of civilian applications involving the use of passports, cellular telephones, automatic teller machines, and driver licenses. Traditional knowledge-based [password or personal identification number (PIN)] and token-based (passport, driver license, and ID card) identifications are prone to fraud because PIN's may be forgotten or guessed by an imposter and the tokens may be lost or stolen. As an example, Mastercard credit card fraud alone now amounts to more than 450 million U.S.

Fig. 1. Ridges and automatically detected minutiae points in a fingerprint image. The core is marked with a $\times$.

dollars annually [2]. *Biometrics*, which refers to identifying an individual based on his or her physiological or behavioral characteristics has the capability to reliably distinguish between an authorized person and an imposter.

A biometric system can be operated in two modes: 1) verification mode and 2) identification mode. A biometric system operating in the verification mode either accepts or rejects a user's claimed identity while a biometric system operating in the identification mode establishes the identity of the user without a claimed identity information. In this work, we have focused only on a biometric system operating in the verification mode.

Among all the biometrics (e.g., face, fingerprints, hand geometry, iris, retina, signature, voice print, facial thermogram, hand vein, gait, ear, odor, keystroke dynamics, etc. [2]), fingerprint-based identification is one of the most mature and proven technique.

A fingerprint is the pattern of ridges and valleys on the surface of the finger [3]. The uniqueness of a fingerprint can be determined by the overall pattern of ridges and valleys as well as the local ridge anomalies [a ridge bifurcation or a ridge ending, called minutiae points (see Fig. 1)]. Although the fingerprints possess the discriminatory information, designing a reliable automatic fingerprint matching algorithm is very challenging (see Fig. 2). As fingerprint sensors are becoming smaller and cheaper [4], automatic identification based on fingerprints is becoming an attractive alternative/complement to the traditional methods of identification. The critical factor in the widespread use of fingerprints is in satisfying the performance (e.g., matching speed and accuracy) requirements of the emerging civilian identification applications. Some of these applications (e.g., fingerprint-based smartcards) will also benefit from a compact representation of a fingerprint.
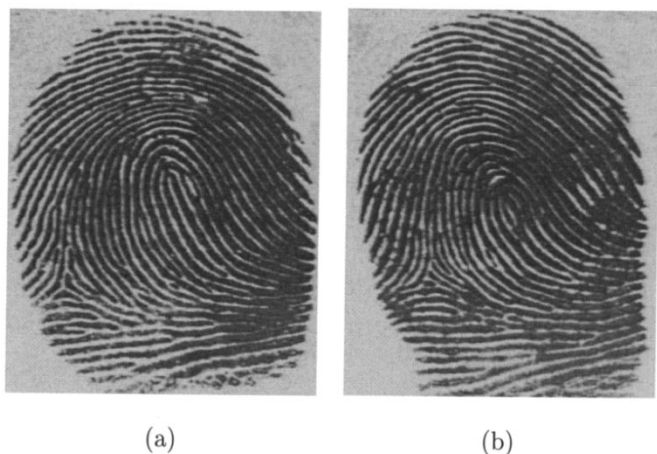
Fig. 2. Difficulty in fingerprint matching. (a) and (b) have the same global configuration but are images of two different fingers.

The popular fingerprint representation schemes have evolved from an intuitive system design tailored for fingerprint experts who visually match the fingerprints. These schemes are either based on predominantly local landmarks (e.g., minutiae-based fingerprint matching systems [1], [5]) or exclusively global information (fingerprint classification based on the Henry system [6]–[8]). The minutiae-based automatic identification techniques first locate the minutiae points and then match their relative placement in a given finger and the stored template [1]. A good quality fingerprint contains between 60 and 80 minutiae, but different fingerprints have different number of minutiae. The variable sized minutiae-based representation does not easily lend itself to indexing mechanisms. Further, typical graph-based [9]–[11], and point pattern-based [1], [12], [13] approaches to match minutiae from two fingerprints need to align the unregistered minutiae patterns of different sizes which makes them computationally expensive. Correlation-based techniques [14], [15] match the global patterns of ridges and valleys to determine if the ridges align. The global approach to fingerprint representation is typically used for indexing [6]–[8], and does not offer very good individual discrimination. Further, the indexing efficacy of existing global representations is poor due to a small number of categories that can be effectively identified and a highly skewed distribution of the population in each category. The natural proportion of fingerprints belonging to categories whorl (whorl and double loop put together), loop (right and left loop put together), and arch (arch and tented arch put together), is 0.279, 0.655, and 0.066, respectively. Both these approaches utilize representations which cannot be easily extracted from poor quality fingerprints.

The smooth flow pattern of ridges and valleys in a fingerprint can be viewed as an oriented texture field [16]. The image intensity surface in an ideal fingerprint image is comprised of ridges whose direction and height vary continuously, which constitutes an oriented texture. Most textured images contain a limited range of spatial frequencies, and mutually distinct textures differ significantly in their dominant frequencies [17]–[19]. Textured regions possessing different spatial frequency, orientation, or phase can be easily discriminated by decomposing the texture in several spatial frequency and orientation channels. For typical fingerprint images scanned at 500 dpi, there is a little variation in the spatial frequencies (inter-ridge distances) among different fingerprints. This implies that there is an optimal scale (spatial frequency) for analyzing the fingerprint texture. Every point in a fingerprint image is associated with a dominant local orientation and a local measure of coherence of the flow pattern. A symbolic description of a fingerprint image can be derived by computing the angle and coherence at each point in the image. Fingerprints can be identified by using quantitative measures associated with the flow pattern (oriented texture) as features.

It is desirable to explore representation schemes which combine global and local information in a fingerprint. We present a new representation for the fingerprints which yields a relatively short, fixed length code, called *FingerCode* [6] suitable for matching as well as storage on a smartcard. The matching reduces to finding the Euclidean distance between these FingerCodes and hence the matching is very fast and the representation is amenable to indexing. We utilize both the global flow of ridge and valley structures and the local ridge characteristics to generate a short fixed length code for the fingerprints while maintaining a high recognition accuracy.

The proposed scheme of feature extraction tessellates the region of interest of the given fingerprint image with respect to a reference point (Fig. 3). A feature vector is composed of an ordered enumeration of the features extracted from the (local) information contained in each subimage (sector) specified by the tessellation. Thus, the feature elements capture the local information and the ordered enumeration of the tessellation captures the invariant global relationships among the local patterns. The local discriminatory information in each sector needs to be decomposed into separate components. Gabor filterbanks are a well-known technique to capture useful information in specific bandpass channels as well as to decompose this information into biorthogonal components in terms of spatial frequencies. A feature vector, which we call FingerCode, is the collection of all the features (for every sector) in each filtered image. These features capture both the global pattern of ridges and valleys and the local characteristics. Matching is based on the Euclidean distance between the FingerCodes.

II. FILTER-BASED FEATURE EXTRACTION

It is desirable to obtain representations for fingerprints which are scale, translation, and rotation invariant. Scale invariance is not a significant problem since most fingerprint images could be scaled as per the dpi specification of the sensors. The rotation and translation invariance could be accomplished by establishing a reference frame based on the intrinsic fingerprint characteristics which are rotation and translation invariant. It is also possible to establish many frames of reference based upon several landmark structures in a fingerprint to obtain multiple representations. At the expense of additional processing and storage cost, the multiple representations offer robust matching performance when extraction algorithm fails to detect one or more frames of reference. In the proposed feature extraction scheme, translation is handled by a single reference point location during the feature extraction stage. The present implementation of feature extraction assumes that the fingerprints are vertically ori-
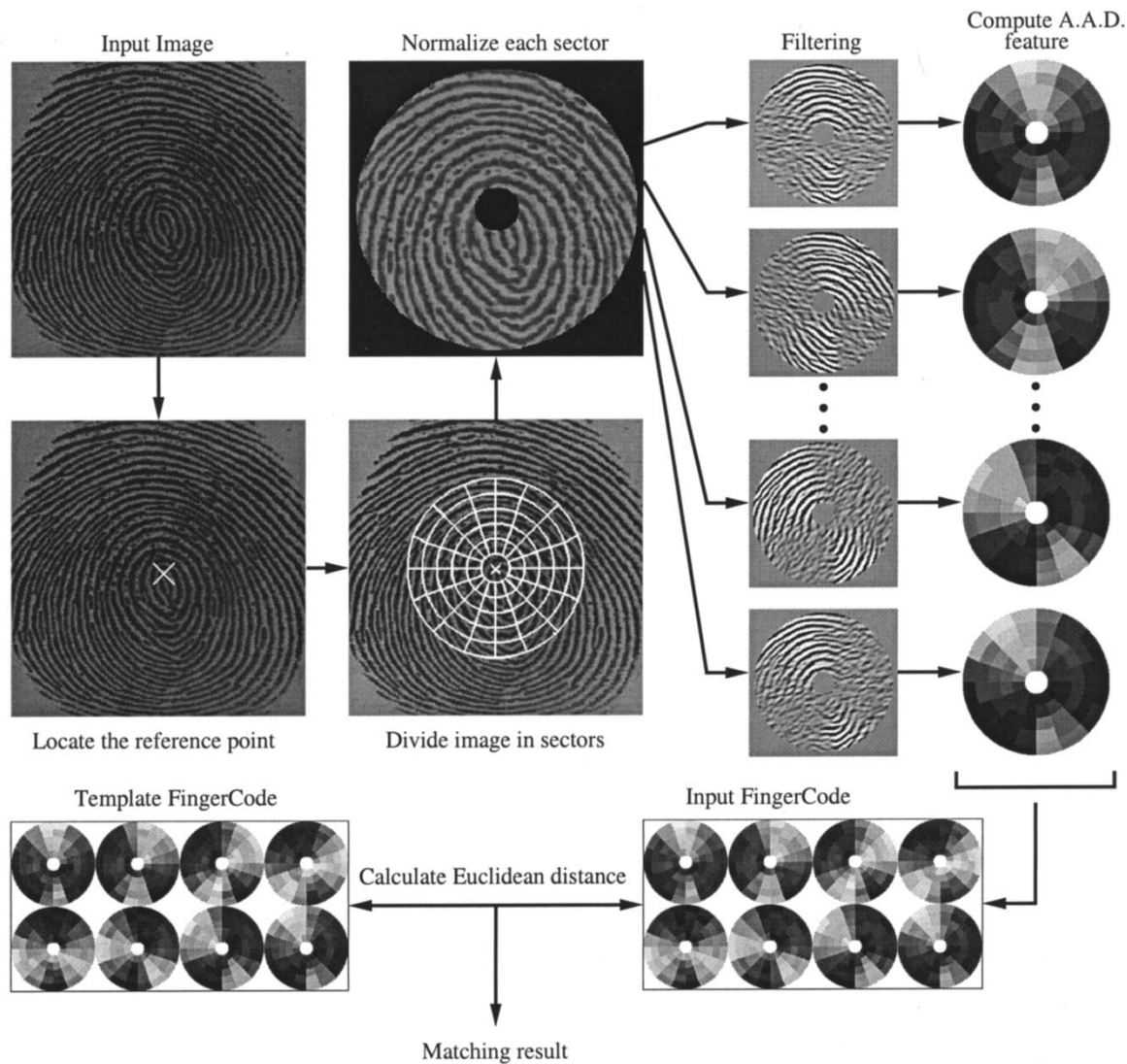
Fig. 3.   System diagram of our fingerprint authentication system.

ented. In reality, the fingerprints in our database are not exactly vertically oriented; the fingerprints may be oriented up to $\pm 45°$ away from the assumed vertical orientation. This image rotation is partially handled by a cyclic rotation of the feature values in the FingerCode in the matching stage; in future implementations, the image rotation will be correctly handled by automatically determining the fingerprint orientation from the image data. The current scheme of feature extraction tessellates the region of interest in the given fingerprint image with respect to the point of reference. The four main steps in our feature extraction algorithm are

1) determine a reference point and region of interest for the fingerprint image;
2) tessellate the region of interest around the reference point;
3) filter the region of interest in eight different directions using a bank of Gabor filters (eight directions are required to completely capture the local ridge characteristics in a fingerprint while only four directions are required to capture the global configuration [6]);

4) compute the average absolute deviation from the mean (AAD) of gray values in individual sectors in filtered images to define the feature vector or the FingerCode.

In the current implementation, we have used the AAD features which give slightly better performance than variance features [6] on both the MSU_DBI and NIST 9 databases. Although AAD features perform reasonably well, we believe that a significantly better performance can be achieved by using more discriminative features.

Let $I(x, y)$ denote the gray level at pixel (x, y) in an $M \times N$ fingerprint image and let (x_c, y_c) denote the reference point. The *region of interest* is defined as the collection of all the sectors S_i, where the ith sector S_i is computed in terms of parameters (r, θ) as follows:

$$S_i = \{(x, y) | b(T_i + 1) \leq r < b(T_i + 2),$$
$$\theta_i \leq \theta < \theta_{i+1}, 1 \leq x \leq N, 1 \leq y \leq M\} \quad (1)$$
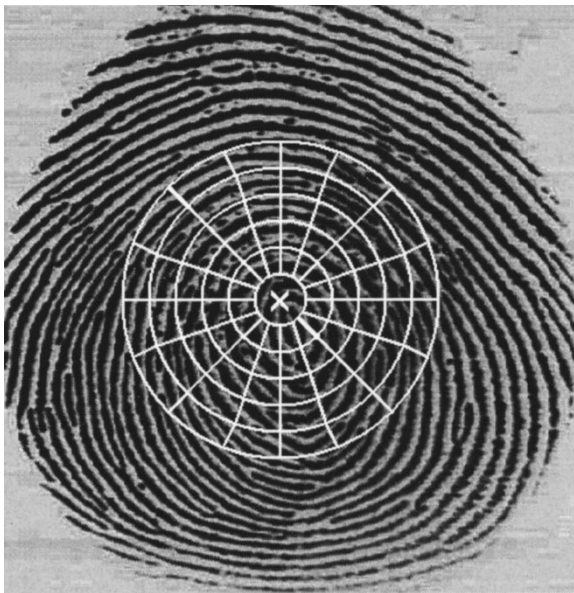
Fig. 4. Reference point (×), the region of interest, and 80 sectors superimposed on a fingerprint.

Concave ridges   Convex ridges

Fig. 5. Concave and convex ridges in a fingerprint image when the finger is positioned upright.

where

$$T_i = i \ div \ k \qquad (2)$$
$$\theta_i = (i \ \mathrm{mod} \ k) \times (2\pi/k) \qquad (3)$$
$$r = \sqrt{(x - x_c)^2 + (y - y_c)^2} \qquad (4)$$
$$\theta = \tan^{-1}\left((y - y_c)/(x - x_c)\right) \qquad (5)$$

b is the width of each band, k is the number of sectors considered in each band, and $i = 0 \cdots (B \times k - 1)$, where B is the number of concentric bands considered around the reference point for feature extraction. These parameters depends upon the image resolution and size. In our first experiment with MSU_DBI database (image size = 508×480 pixels, scanned at 500 dpi), we considered five concentric bands ($B = 5$) for feature extraction. Each band is 20-pixels wide ($b = 20$), and segmented into sixteen sectors ($k = 16$) (Fig. 4). A 20-pixel wide band captures an area spanning about one ridge and valley pair, on an average, in a 500 dpi fingerprint image. A band with a width of 20 pixels is necessary to capture a single minutia in a sector, allowing our low-level features to capture this local information. If the sector width is more than 20 pixels, then the local information may be modulated by more global information. The innermost band (circle) is not used for feature extraction because the flow field in a region around a very high curvature point (core) has poor coherence. Thus, absolute deviations of oriented Gabor responses to this region would be expected to be unreliable matching features. Thus, we have a total of $16 \times 5 = 80$ sectors (S_0 through S_{79}) and the region of interest is a circle of radius 120 pixels, centered at the reference point. Eighty features for each of the eight filtered images provide a total of 640 (80×8) features per fingerprint image. Each feature can be quantized into 256 values and requires 1 byte of storage, so the entire feature vector requires only 640 bytes of storage. In our second experiment with NIST 9 database (image size = 832×768 pixels, scanned at 500 dpi), we used seven concentric bands ($B = 7$), $b = 20$, and $k = 16$, giving us an 896 byte FingerCode.
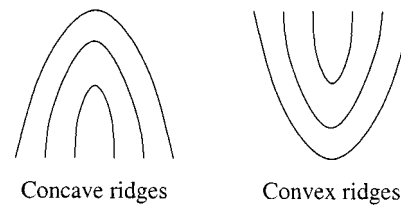
It is difficult to rely on feature extraction based on explicit detection of structural features in fingerprints, especially in poor quality images. Features based on statistical properties of images are likely to degrade gracefully with the image quality deterioration. For this study, we use grayscale variance-based features. The average absolute deviation of the gray levels from the mean value in an image sector is indicative of the overall ridge activity in that sector which we claim to be useful for fingerprint verification. As noted in Section IV, our matcher based on this simple statistical feature performs well and we expect to achieve significantly better accuracies with more discriminative attributes.

A. Reference Point Location

Fingerprints have many conspicuous landmark structures and a combination of them could be used for establishing a reference point. We define the reference point of a fingerprint as the point of maximum curvature of the concave ridges (see Fig. 5) in the fingerprint image.

Many previous approaches to determination of a reference point (x_c, y_c) critically relied on the local features like Poincaré index or some other similar properties of the orientation field. While these methods work well in good quality fingerprint images, they fail to correctly localize reference points in poor quality fingerprints with cracks and scars, dry skin, or poor ridge and valley contrast. Recently, Hong and Jain have attempted to judiciously combine the orientation field information with available ridge details in a fingerprint [8]. However, this method does not reliably handle poor quality fingerprints when the orientation field is very noisy and can be misled by poor structural cues in the presence of finger cracks.

In order that a reference point algorithm gracefully handle local noise in a poor quality fingerprint, the detection should necessarily consider a large neighborhood in the fingerprint. On the other hand, for an accurate localization of the reference point, the approach should be sensitive to the local variations in a small neighborhood. To meet these conflicting requirements of an accurate and reliable localization, we propose a new method of reference point determination based on multiple resolution analysis of the orientation fields. Our new method locates the reference point more precisely than the algorithm proposed by Hong and Jain [8].

Let us first define the *orientation field*, $\mathcal{O}$, for a fingerprint image. The orientation field, $\mathcal{O}$, is defined as a $P \times Q$ image, where $\mathcal{O}(i, j)$ represents the local ridge orientation at pixel (i, j). Local ridge orientation is usually specified for a block rather than at every pixel; an image is divided into a set of $w \times w$ nonoverlapping blocks and a single orientation is defined for
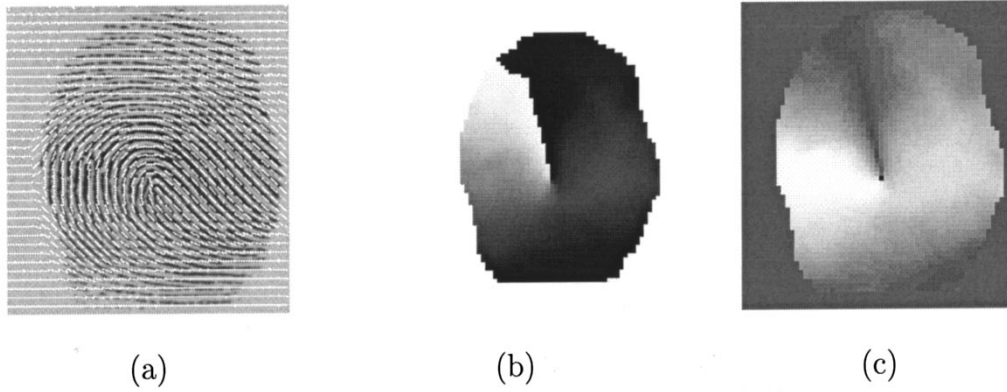
Fig. 6. Estimating the reference point. (a) Smoothed orientation field overlapped on the original image. (b) Orientation field ($w = 10$) shown as intensity distribution; the background has been segmented. (c) *sine* component of the orientation field; the darkest pixel marks the detected reference point. Images have been scaled for viewing.

each block [see Fig. 6(a) and (b)]. Note that there is an ambiguity of π in fingerprint orientation, i.e., local ridges oriented at $\pi/2$ and ridges oriented at $3\pi/2$ cannot be differentiated from each other. A number of methods have been developed to estimate the orientation field in a fingerprint [20]–[23]. The least mean square orientation estimation algorithm [33] has the following steps.

1) Divide $\mathcal{I}$, the input image, into nonoverlapping blocks of size $w \times w$.
2) Compute the gradients $\partial_x(i, j)$ and $\partial_y(i, j)$ at each pixel (i, j). Depending on the computational requirement, the gradient operator may vary from the simple *Sobel* operator to the more complex *Marr–Hildreth* operator [24].
3) Estimate the local orientation of each block centered at pixel (i, j) using the following equations [23]:

$$\mathcal{V}_x(i, j) = \sum_{u=i-w/2}^{i+w/2} \sum_{v=j-w/2}^{j+w/2} 2\partial_x(u, v)\partial_y(u, v) \quad (6)$$

$$\mathcal{V}_y(i, j) = \sum_{u=i-w/2}^{i+w/2} \sum_{v=j-w/2}^{j+w/2} (\partial_x^2(u, v) - \partial_y^2(u, v)) \quad (7)$$

$$\mathcal{O}(i, j) = \frac{1}{2} \tan^{-1}\left(\frac{\mathcal{V}_y(i, j)}{\mathcal{V}_x(i, j)}\right) \quad (8)$$

where $\mathcal{O}(i, j)$ is the least square estimate of the local ridge orientation at the block centered at pixel (i, j). Mathematically, it represents the direction that is orthogonal to the dominant direction of the *Fourier spectrum* of the $w \times w$ window.

A summary of our reference point location algorithm is presented below.

1) Estimate the orientation field $\mathcal{O}$ as described above using a window size of $w \times w$.
2) Smooth the orientation field in a local neighborhood. Let the smoothed orientation field be represented as $\mathcal{O}'$. In order to perform smoothing (low-pass filtering), the orientation image needs to be converted into a *continuous vector field*, which is defined as follows:

$$\Phi_x(i, j) = \cos(2\mathcal{O}(i, j)) \quad (9)$$

and

$$\Phi_y(i, j) = \sin(2\mathcal{O}(i, j)) \quad (10)$$

andwhere Φ_x and Φ_y, are the x and y components of the vector field, respectively. With the resulting vector field, the low-pass filtering can then be performed as follows:

$$\Phi_x'(i, j) = \sum_{u=-w_\Phi/2}^{w_\Phi/2} \sum_{v=-w_\Phi/2}^{w_\Phi/2} W(u, v) \cdot \Phi_x(i - uw, j - vw) \quad (11)$$

and

$$\Phi_y'(i, j) = \sum_{u=-w_\Phi/2}^{w_\Phi/2} \sum_{v=-w_\Phi/2}^{w_\Phi/2} W(u, v) \cdot \Phi_y(i - uw, j - vw) \quad (12)$$

where W is a two-dimensional low-pass filter with unit integral and $w_\Phi \times w_\Phi$ specifies the size of the filter. Note that the smoothing operation is performed at the block level. For our experiments, we used a 5×5 mean filter. The smoothed orientation field $\mathcal{O}'$ at (i, j) is computed as follows:

$$\mathcal{O}'(i, j) = \frac{1}{2} \tan^{-1}\left(\frac{\Phi_y'(i, j)}{\Phi_x'(i, j)}\right). \quad (13)$$

3) Compute $\mathcal{E}$, an image containing only the *sine* component of $\mathcal{O}'$ [see Fig. 6(c)]

$$\mathcal{E}(i, j) = \sin(\mathcal{O}'(i, j)). \quad (14)$$

4) Initialize $\mathcal{A}$, a label image used to indicate the reference point.
5) For each pixel (i, j) in $\mathcal{E}$, integrate pixel intensities (sine component of the orientation field) in regions R_I and R_{II} shown in Fig. 7 and assign the corresponding pixels in $\mathcal{A}$ the value of their difference

$$\mathcal{A}(i, j) = \sum_{R_I} \mathcal{E}(i, j) - \sum_{R_{II}} \mathcal{E}(i, j). \quad (15)$$

The regions R_I and R_{II} (see Fig. 7) were determined empirically by applying the reference point location algorithm over a large database. The geometry of regions R_I and R_{II} is designed to capture the maximum curvature
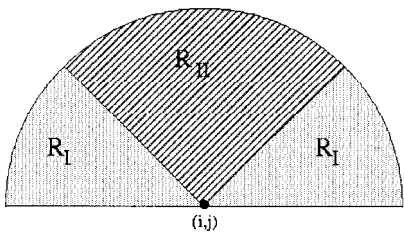
Fig. 7. Regions for integrating $\mathcal{E}$ pixel intensities for $\mathcal{A}(i, j)$.
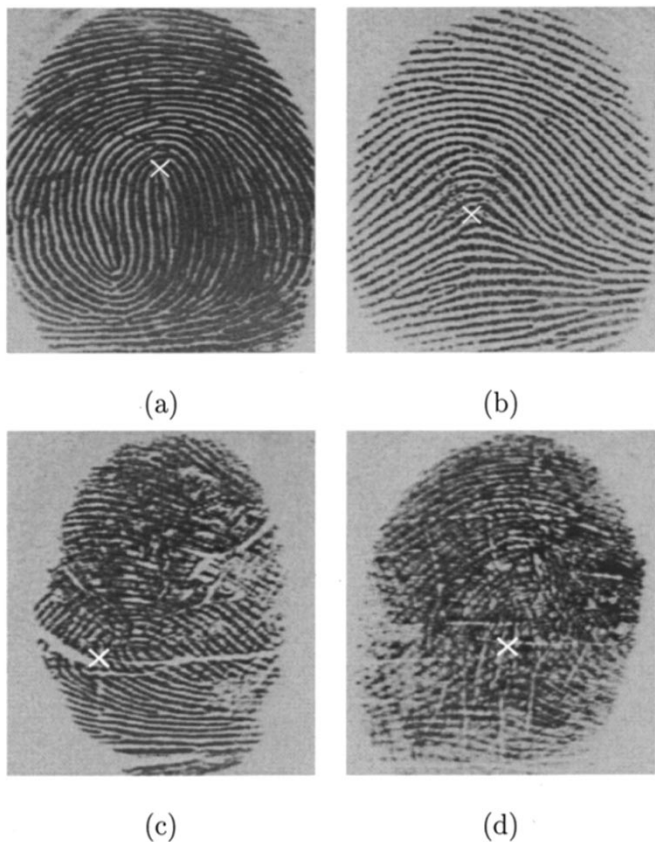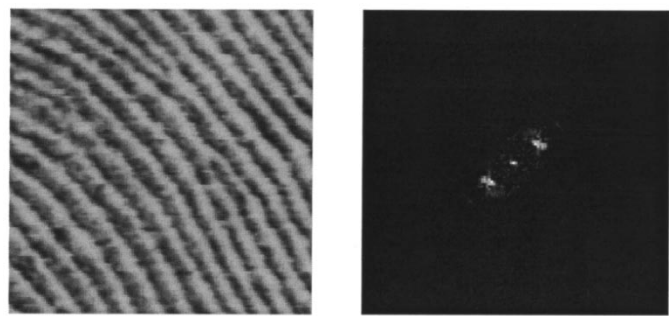

(a)   (b)

(c)   (d)

Fig. 8. Examples of the results of our reference point location algorithm. Our reference point location algorithm fails on very poor quality fingerprints.

in concave ridges (see Fig. 5). Although this successfully detects the reference point in most of the cases, including double loops [see Fig. 8(a)], the present implementation is not very precise and consistent for the arch type fingerprints.

6) Find the maximum value in $\mathcal{A}$ and assign its coordinate to the core, i.e., the reference point.

7) For a fixed number of times, repeat steps 1–6 by using a window size of $w' \times w'$, where $w' < w$ and restrict the search for the reference point in step 6 in a local neighborhood of the detected reference point. In our experiments, we used three iterations with $w = 15$, 10, and 5 pixels, respectively, and hence the precision of the detected reference point is 5 pixels.

Our representation scheme is able to tolerate the imprecision in the reference point estimates of up to 10 pixels (approxi-

(a) Ridges in a local region   (b) Fourier spectrum of (a)

Fig. 9. Fingerprints have well defined local frequency and orientation. (a) Ridges in a local region and (b) Fourier spectrum of (a).

mately 1 inter-ridge distance unit) away from its "true" location. Fig. 8 shows the results of our reference point location algorithm.

B. Filtering

Fingerprints have local parallel ridges and valleys, and well-defined local frequency and orientation (Fig. 9). Properly tuned Gabor filters [25], [26], can remove noise, preserve the true ridge and valley structures, and provide information contained in a particular orientation in the image. A minutia point can be viewed as an anomaly in locally parallel ridges and it is this information that we are attempting to capture using the Gabor filters.

Before filtering the fingerprint image, we normalize the region of interest in each sector separately to a constant mean and variance. Normalization is performed to remove the effects of sensor noise and gray level deformation due to finger pressure differences. Let $I(x, y)$ denote the gray value at pixel (x, y), M_i and V_i, the estimated mean and variance of sector S_i, respectively, and $N_i(x, y)$, the normalized gray-level value at pixel (x, y). For all the pixels in sector S_i, the normalized image is defined as

$$N_i(x, y)$$
$$= \begin{cases} M_0 + \sqrt{\dfrac{V_0 \times (I(x, y) - M_i)^2)}{V_i}}, & \text{if } I(x, y) > M_i \\ M_0 - \sqrt{\dfrac{V_0 \times (I(x, y) - M_i)^2)}{V_i}}, & \text{otherwise} \end{cases}$$

$$(16)$$

where M_0 and V_0 are the desired mean and variance values, respectively. Normalization is a pixel-wise operation which does not change the clarity of the ridge and valley structures. If normalization is performed on the entire image, then it cannot compensate for the intensity variations in different parts of the image due to the elastic nature of the finger. Separate normalization of each individual sector alleviates this problem. Fig. 10 shows an example of this normalization scheme. For our experiments, we set the values of both M_0 and V_0 to 100.
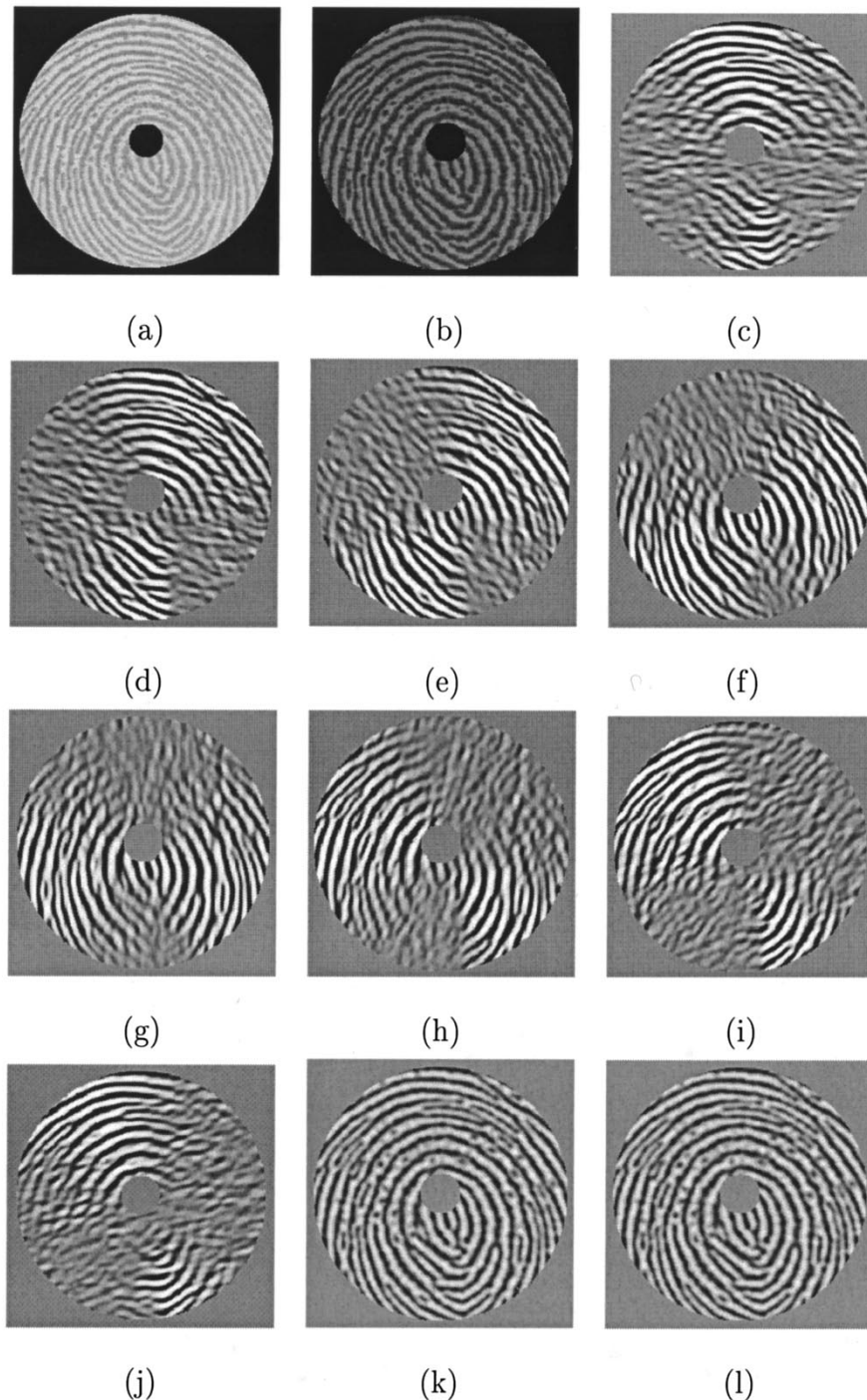
Fig. 10.   Normalized, filtered, and reconstructed fingerprint images: (a) area of interest, (b) normalized image, (c)–(j) $0°$, $22.5°$, $45°$, $67.5°$, $90°$, $112.5°$, $135°$, $157.5°$ filtered images, respectively, (k) reconstructed image with four filters, and (l) reconstructed image with eight filters. While four directions are sufficient to capture the global structure of the fingerprint, eight directions are required to capture the local characteristics.

An even symmetric Gabor filter has the following general form in the spatial domain:

$$G(x, y; f, \theta) = \exp\left\{\frac{-1}{2}\left[\frac{x'^2}{\delta_{x'}^2} + \frac{y'^2}{\delta_{y'}^2}\right]\right\}\cos(2\pi f x') \quad (17)$$

$$x' = x\sin\theta + y\cos\theta \quad (18)$$

$$y' = x\cos\theta - y\sin\theta \quad (19)$$

where f is the frequency of the sinusoidal plane wave along the direction θ from the x-axis, and $\delta_{x'}$ and $\delta_{y'}$ are the space constants of the Gaussian envelope along x' and y' axes, respectively. The spatial characteristics of Gabor filters can be seen in [6].

We perform the filtering in the spatial domain with a mask size of 33×33. However, to speed up the filtering process, we convolve a pixel only with those values in the filter mask whose absolute value is greater than 0.05. This speeds up the

convolution process significantly while maintaining the information content as the convolution with small values of the filter mask does not contribute significantly to the overall convolution. We also make use of the symmetry of the filter to speed up the convolution. However, convolution with Gabor filters is still the major contributor to the overall feature extraction time.

In our experiments, we set the filter frequency f to the average ridge frequency $(1/K)$, where K is the average inter-ridge distance. The average inter-ridge distance is approximately 10 pixels in a 500 dpi fingerprint image. If f is too large, spurious ridges are created in the filtered image whereas if f is too small, nearby ridges are merged into one. We have used eight different values for θ ($0°$, $22.5°$, $45°$, $67.5°$, $90°$, $112.5°$, $135°$, and $157.5°$) with respect to the x-axis. The normalized region of interest in a fingerprint image is convolved with each of these eight filters to produce a set of eight filtered images. A fingerprint convolved with a $0°$-oriented filter accentuates those ridges which are parallel to the x-axis and smoothes the ridges in the other directions. Filters tuned to other directions work in a similar way. These eight directional-sensitive filters capture most of the global ridge directionality information as well as the local ridge characteristics present in a fingerprint. We illustrate this through reconstructing a fingerprint image by adding together all the eight filtered images. The reconstructed image is very similar to the original image and only been slightly blurred (degraded) [Fig. 10(a)] due to lack of orthogonality among the filters. At least four directional filters are required to capture the entire global ridge information in a fingerprint [Fig. 10(k)], but eight directional filters are required to capture the local characteristics. So, while four directions are sufficient for classification [6], eight directions are needed for matching. Our empirical results support our claim, we could get better accuracy by using eight directions for matching as compared to only four directions. By capturing both the global and local information, the verification accuracy is improved although there is some redundancy among the eight filtered images. If $\delta_{x'}$ and $\delta_{y'}$ (standard deviations of the Gaussian envelope) values are too large, the filter is more robust to noise, but is more likely to smooth the image to the extent that the ridge and valley details in the fingerprint are lost. If $\delta_{x'}$ and $\delta_{y'}$ values are too small, the filter is not effective in removing the noise. The values for $\delta_{x'}$ and $\delta_{y'}$ were empirically determined and each is set to 4.0 (about half the average inter-ridge distance).

C. Feature Vector

Let $F_{i\theta}(x, y)$ be the θ-direction filtered image for sector S_i. Now, $\forall$ i $\in$ $\{0, 1, \cdots, 79\}$ and $\theta \in \{0°, 22.5°, 45°, 67.5°, 90°, 112.5°, 135°, 157.5°\}$, the feature value, $V_{i\theta}$, is the average absolute deviation from the mean defined as

$$V_{i\theta} = \frac{1}{n_i} \left(\sum_{n_i} |F_{i\theta}(x, y) - P_{i\theta}| \right) \qquad (20)$$

where n_i is the number of pixels in S_i and $P_{i\theta}$ is the mean of pixel values of $F_{i\theta}(x, y)$ in sector S_i. The average absolute deviation of each sector in each of the eight filtered images defines the components of our feature vector. Our empirical results show that using AAD features give slightly better performance

than variance features as used in [6]. The 640-dimensional feature vectors (FingerCodes) for fingerprint images of two different fingers from the MSU_DBI database are shown as gray level images with eight disks, each disk corresponding to one filtered image in Fig. 11. The gray level in a sector in a disk represents the feature value for that sector in the corresponding filtered image. Note that Fig. 11(c) and (d) appear to be visually similar as are Fig. 11(g) and (h), but the corresponding disks for two different fingers look very different.

III. MATCHING

Fingerprint matching is based on finding the Euclidean distance between the corresponding FingerCodes. The translation invariance in the FingerCode is established by the reference point. However, in our present implementation, features are not rotationally invariant. An approximate rotation invariance is achieved by cyclically rotating the features in the FingerCode itself. A single step cyclic rotation of the features in the FingerCode described by (21)–(23) corresponds to a feature vector which would be obtained if the image were rotated by $22.5°$. A rotation by R steps corresponds to a $R \times 22.5°$ rotation of the image. A positive rotation implies clockwise rotation while a negative rotation implies counterclockwise rotation. The FingerCode obtained after R steps of rotation is given by

$$V_{i\theta}^R = V_{i'\theta'} \qquad (21)$$
$$i' = (i + k + R) \bmod k + (i \text{ div } k) \times k \qquad (22)$$
$$\theta' = (\theta + 180° + 22.5° \times R) \bmod 180° \qquad (23)$$

where k (=16) is the number of sectors in a band, $i \in [0, 1, 2, \cdots 79]$, and $\theta \in [0°, 22.5°, 45°, 67.5°, 90°, 112.5°, 135°, 157.5°]$.

For each fingerprint in the database, we store five templates corresponding to the following five rotations of the corresponding FingerCode: $V_{i\theta}^{-2}$, $V_{i\theta}^{-1}$, $V_{i\theta}^{0}$, $V_{i\theta}^{1}$, and $V_{i\theta}^{2}$. The input FingerCode is matched with the five templates stored in the database to obtain five different matching scores. The minimum matching score corresponds to the best alignment of the input fingerprint with the database fingerprint. Since a single cyclic rotation of the features in the FingerCode corresponds to a rotation of $22.5°$ in the original image, we can only generate those representations of the fingerprint which are in steps of $22.5°$. Due to the nature of the tessellation, our features are invariant to only small perturbations that are within $\pm 11.25°$. Therefore, we generate another feature vector for each fingerprint during the time of registration which corresponds to a rotation of $11.25°$. The original image is rotated by an angle of $11.25°$ and its FingerCode is generated. Five templates corresponding to the various rotations of this FingerCode are also stored in the database. Thus, the database contains ten templates for each fingerprint. These ten templates correspond to all the rotations on the fingerprint image in steps of $11.25°$. As a result, we have generated FingerCodes for every $11.25°$ rotation of the fingerprint image. This takes care of the rotation while matching the input FingerCode with the stored templates. The final matching distance score is taken as the minimum of the ten scores, i.e., matching of the input FingerCode with each of the ten templates. This minimum score corresponds to the best alignment of the two fingerprints being matched. Since the
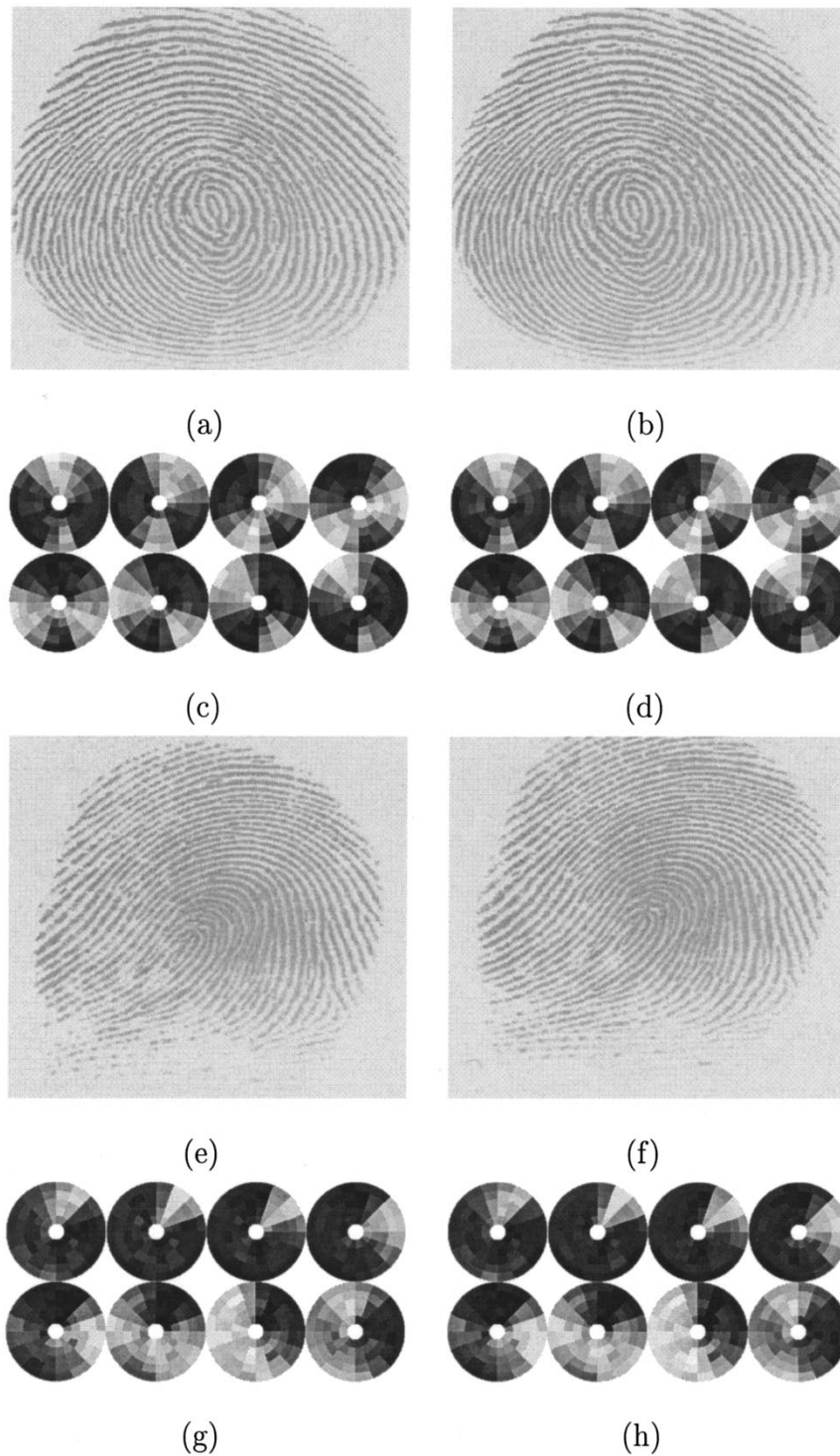
Fig. 11.   Examples of 640-dimensional feature vectors: (a) First impression of finger 1, (b) second impression of finger 1, (c) FingerCode of (a), (d) FingerCode of (b), (e) first impression of finger 2, (f) second impression of finger 2, and (g) FingerCode of (e), and (h) FingerCode of (f).

template generation for storage in the database is an off-line process, the verification time still depends on the time taken to generate a single template.

IV. EXPERIMENTAL RESULTS

We have collected fingerprint images from 167 nonhabituated cooperative subjects using a Digital Biometrics's optical sensor (image size $= 508 \times 480$ pixels) in our laboratory. The subjects mainly consisted of students at Michigan State University and their relatives and friends. Approximately 35% of the subjects were women; the age distribution of the subjects was as follows:

1) younger than 25 years: 46.5%;
2) between the ages of 25 and 50: 51%;
3) older than 50 years: 2.5%.

Each person was asked to provide fingerprint images for four fingers, namely, right index, right middle, left index, and left middle, in this order. This process was repeated to obtain a second impression of all the four fingers. This resulted in a total of 1336 ($167 \times 4 \times 2$) fingerprint images. The subjects were requested to provide their fingerprint images again after a period of 6 weeks. At this second time instant, the above procedure was repeated to collect another 1336 ($167 \times 4 \times 2$) fingerprint images. So, our database (called MSU_DBI) consists of a total of 2672 fingerprint images. The data acquisition process was carried out by several volunteers who are research assistants in the Pattern Recognition and Image Processing Laboratory at Michigan State University. A live feedback of the acquired image was provided and the volunteers guided the subjects in placing their fingers in the center of the sensor and in an upright position. Due to this assistance provided to the subjects, most of the fingerprints were reasonably well centered. Despite the supervised image acquisition, there is a significant intra-class deformation and up to $\pm45°$ deviation from the assumed vertical upright orientation in the acquired images. However, the images are not as noisy as the traditional inked fingerprints. The fingerprint images which were captured after a period of six weeks have significant nonlinear distortions due to finger pressure differences. This presents a challenge to our matching algorithm.

We also evaluated our system on 1800 images of the commercially available standard database NIST 9 (Vol. 1, CD. no. 1). NIST 9 (Vol. 1, CD no. 1) contains 1800 fingerprint images (image size = 832×768 pixels) from 900 different fingers. The complete NIST 9 fingerprint database contains 1350 mated fingerprint card pairs (13 500 fingerprint image pairs) that approximate a natural distribution of the National Crime and Information Center fingerprint classes. The database is divided into multiple volumes. Each volume has three compact discs (CD's). Each CD contains 900 images of card type 1 and 900 images of card type 2. Fingerprints on the card type 1 were scanned using a rolled method, and fingerprint on card type 2 were scanned using a live-scan method. The fingerprint images in the NIST 9 database are difficult compared to the live-scan fingerprint images because the two impressions from the same finger are captured using different methods (rolled and live-scan) and hence the two images have large discrepancy in their ridge structures. A large number of NIST 9 images are of much poorer quality than a typical live-scan fingerprint image and NIST 9 images often contain extraneous objects like handwritten characters and other artifacts common to inked fingerprints.

A total of 100 images (approximately 4% of the database) were rejected from the MSU_DBI database because of the following reasons: 1) the reference point was located at a corner of the image and therefore an appropriate region of interest could not be established and 2) the quality of the image was poor. See Fig. 12 for examples of images which were rejected. A total of 100 images (approximately 5.6% of the database) were rejected from the NIST 9 database based on the same criteria. Our quality checker algorithm estimates the dryness of the finger (or smudginess of the fingerprint image) and the extent to which the surface of the finger tip is imaged. The estimate of the dryness/smudginess is based on the variance of the grayscale in the captured image and a partial fingerprint is detected by tracking
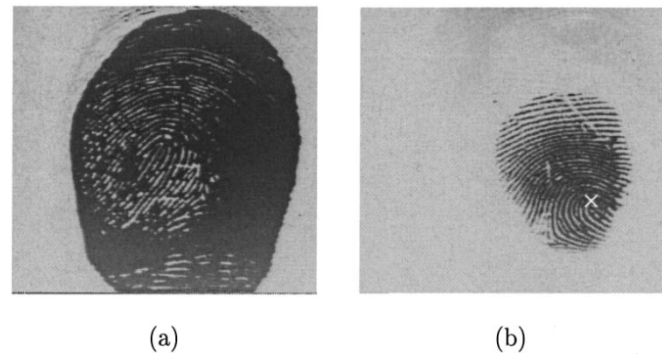

(a) (b)

Fig. 12. Examples of rejected images: (a) a poor quality image and (b) the reference point is (correctly) detected at a corner of the image and so an appropriate region of interest could not be established.

the ridge structure in the image. For algorithm development and parameter selection, an independent database of 250 impressions from ten different fingers was collected. The algorithm thus developed was tested on the MSU_DBI database. However, for NIST 9 database, we changed one parameter, i.e., the number of concentric circles considered around the reference point. We considered seven concentric circles around the reference point for NIST 9 database as compared to five circles for MSU_DBI database. This is because the images in the NIST 9 database are bigger in size and more information can be captured by considering more number of sectors. Consequently, the number of features for the NIST 9 database was 896 while there were only 640 features for images in the MSU_DBI database.

To establish the verification accuracy of our fingerprint representation and matching approach, each fingerprint image in the database is matched with all the other fingerprints in the database. A matching is labeled correct if the matched pair is from the same finger and incorrect, otherwise. None of the genuine (correct) matching scores was zero; the images from the same finger did not yield an identical FingerCode because of the rotation and inconsistency in reference point location. For the MSU_DBI database, a total of 6 586 922 matchings were performed. The probability distribution for genuine (correct) matches was estimated with 7472 matches and the imposter distribution was estimated with 6 579 450 matches. Fig. 13(a) shows the two distributions. For the NIST 9 database, a total of 722 419 matchings were performed and the genuine and imposter distributions were estimated with 1640 and 720 779 matching scores, respectively. Fig. 13(b) shows the imposter and genuine distributions for the NIST 9 database.

In a biometric system operating in a verification mode, there are four possible outcomes:

1) genuine acceptance;
2) imposter rejection;
3) genuine rejection (false reject);
4) imposter acceptance (false accept).

The first and the second outcomes are correct while the third and the fourth outcomes are errors. The performance of a biometric system is specified in terms of false accept rate (FAR). The decision scheme should establish a decision boundary which minimizes the false reject rate (FRR) for the specified FAR. If the Euclidean distance between two FingerCodes is
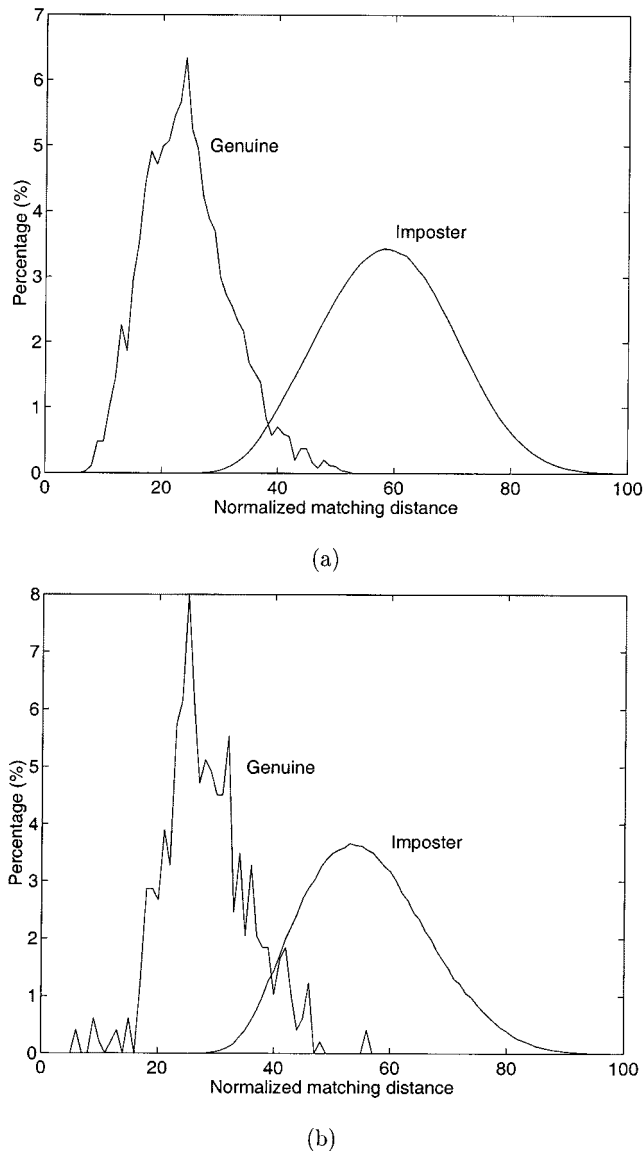
Threshold value	False Acceptance Rate (%)	False Reject Rate (%)
30	0.10	19.32
35	1.07	7.87
40	4.59	2.83

Fig. 13. Genuine and imposter distributions for the proposed verification scheme: (a) MSU_DBI database and (b) NIST 9 (Vol. 1, CD no. 1).

for all possible system operating points (i.e., matching distance threshold) and measures the overall performance of the system. Each point on the curve corresponds to a particular decision threshold. In the ideal case, both the error rates, i.e., FAR and FRR should be zero and the genuine distribution and imposter distribution should be disjoint. In such a case, the "ideal" ROC curve is a step function at the zero False Acceptance Rate. On the other extreme, if the genuine and imposter distributions are exactly the same, then the ROC is a line segment with a slope of $45°$ with an end point at zero False Acceptance Rate. In practice, the ROC curve behaves in between these two extremes. Fig. 14(a) and (b) compare the ROC's of a state-of-the-art minutiae-based matcher [1] with our filter-based matcher on the MSU_DBI and the NIST 9 databases, respectively. Since the ROC curve of the minutiae-based matcher is above the filter-based matcher, we conclude that our matcher does not perform as well as the state-of-the-art minutiae-based matcher on these databases.

Most of the false accepts in our system occur among the same "type" (class) of fingerprints; a whorl is confused with another whorl and not a loop. This confirms that the proposed approach captures more of the global information and hence is suitable for indexing as shown in [6]. An added advantage of an "independent" fingerprint representation such as one proposed here is that it captures discriminatory information that is complementary to the information used by commonly used minutiae-based fingerprint matchers. Consequently, the overall performance of fingerprint matching can be significantly improved by combining results of several matchers based on different representations. A combination of classifiers to improve the classification accuracy has been a common practice in building classifier systems [27], [28]. Many researchers have combined multiple biometrics (i.e., fingerprint and face) to improve the performance of a verification system [29], [30], but this involves the cost of additional sensors and inconvenience to the user in providing multiple cues. Jain *et al.* [31] have shown that matching accuracy can be improved by combining "independent" matchers. Fig. 14(a) and (b) show such an improvement in matching accuracy results by using the Neyman–Pearson [32] rule to combine scores obtained from the proposed filter-based and minutiae-based [1] matchers. The Neyman–Pearson rule used for this combination can be summarized as follows.

Let the scores from the two matching algorithms for matching the input fingerprint with the stored template be indicated by the random vectors X_1 and X_2, respectively. Let $p(X_1|w_G)$ and $p(X_2|w_G)$ be the genuine class-conditional density functions for X_1 and X_2, respectively. Similarly, let $p(X_1|w_I)$ and

less than a threshold, then the decision that "the two images come from the same finger" is made, otherwise a decision that "the two images come from different fingers" is made. There is a tradeoff between the two types of errors. If a higher threshold is chosen, the genuine rejection rate is lower but the false accept rate may be higher, and vice versa. Given a matching distance threshold, the genuine acceptance rate is the fraction of times the system correctly identifies two fingerprints representing the same finger. Similarly, false acceptance rate is the fraction of times the system incorrectly identifies two fingerprints representing the same finger. The given biometric application dictates the FAR and FRR requirements. For example, access to an ATM machine generally needs a small FRR, but access to a military installation requires a very small FAR. Different decision thresholds lead to different FAR and FRR (see Table I).

A receiver operating characteristic (ROC) curve is a plot of genuine acceptance rate (1-FRR) against false acceptance rate
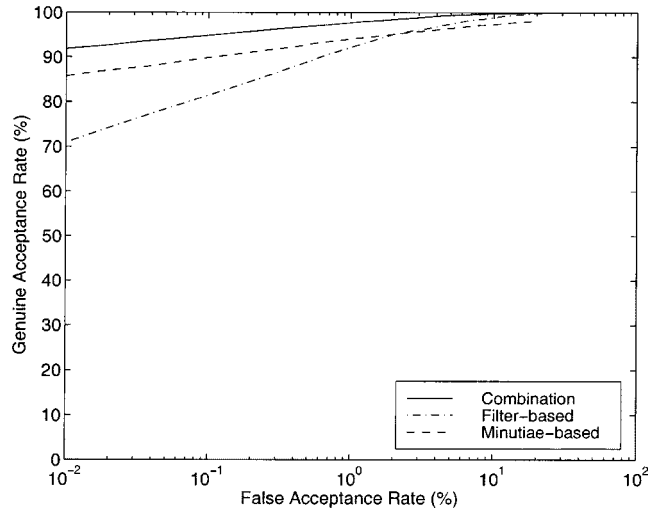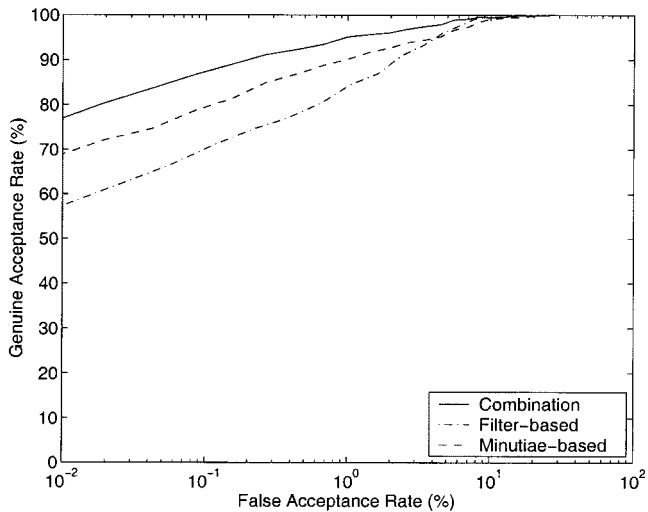
(a)

(b)

Fig. 14. ROC curves for several matchers: (a) MSU_DBI database and (b) NIST 9 (Vol. 1, CD no. 1).

$p(X_2|w_I)$ be the imposter class-conditional density functions for X_1 and X_2, respectively. Under the assumption that X_1 and X_2 are statistically independent, the joint class-conditional probability density functions of X_1 and X_2 are

$$p(X_1, X_2|w_G) = p(X_1|w_G)p(X_2|w_G), \qquad (24)$$
$$p(X_1, X_2|w_I) = p(X_1|w_I)p(X_2|w_I). \qquad (25)$$

Let R^2 denote the two-dimensional space spanned by (X_1, X_2); R_G^2 and R_I^2 denote the w_G-region and w_I-region, respectively, $(R_G^2 + R_I^2 = R^2)$; ϵ_0 denote the pre-specified FAR. According to the Neyman–Pearson rule, a given observation, $X^o = (X_1^o, X_2^o)$, is classified as

$$(X_1^o, X_2^o) \in \begin{cases} w_G, & \text{if } \dfrac{p(X_1^o, X_2^o|w_G)}{p(X_1^o, X_2^o|w_I)} > \lambda \\ w_I, & \text{otherwise} \end{cases} \qquad (26)$$

where λ is the minimum value that satisfies the following two conditions:

$$\lambda = \frac{p(X_1, X_2|w_G)}{p(X_1, X_2|w_I)} \qquad (27)$$

and

$$\epsilon_0 = \int_{R_G^2} p(X_1, X_2|w_I) \, dX_1 \, dX_2. \qquad (28)$$

For the MSU_DBI database, the training for the combination was performed using the scores from the first 83 subjects and the testing on the scores obtained by generating all the matching scores from the rest of the 84 users. For the NIST 9 database, the training was performed using the scores from the first 450 users and the testing on the scores obtained by generating all the matching scores from the rest of the 450 users. A two-dimensional plot of the scores from the two matchers for the MSU_DBI database is shown in Fig. 15. The performance improvement resulting from a combination of matchers is shown in Fig. 14(a) and (b).

V. SUMMARY AND CONCLUSIONS

We have developed a novel filter-based representation technique for fingerprint verification. The technique exploits both the local and global characteristics in a fingerprint image to verify an identity. Each fingerprint image is filtered in a number of directions and a fixed-length feature vector is extracted in the central region of the fingerprint. The feature vector (Finger-Code) is compact and requires only 640 (or 896, depending on image size) bytes. The matching stage computes the Euclidean distance between the template FingerCode and the input Finger-Code. On MSU_DBI database of 2672 fingerprints from 167 different subjects, four impressions per finger, we are able to achieve a verification accuracy which is only marginally inferior to the performance of a state-of-the-art minutiae-based fingerprint matcher. Our system, however, performs better than the minutiae-based system when the system performance requirements are less demanding on FAR (for example, in accessing ATM machines). Similar performance is obtained on the more challenging NIST 9 database. The minutiae-based technique is a very mature technology while the proposed filter-based approach is being developed and refined. We expect to improve the performance significantly by developing algorithms to overcome the following main shortcomings of our technique: 1) the reference point cannot be located accurately in noisy images and 2) the matching scheme is not able to tolerate large deformation in the ridge pattern due to finger pressure differences (see Fig. 16). About 99% of the total compute time for verification ($\sim$ 3 s on a SUN ULTRA 10) is taken by the convolution of the input image with eight Gabor filters.

The primary advantage of our approach is its computationally attractive matching/indexing capability. For instance, if the normalized (for orientation and size) FingerCodes of all the enrolled fingerprints are stored as templates, the identification effectively involves a "bit" comparison. As a result, the identification time would be relatively insensitive to the database size. Further, our approach to feature extraction and matching is more amenable to hardware implementation than, say, a string-based fingerprint matcher [1].
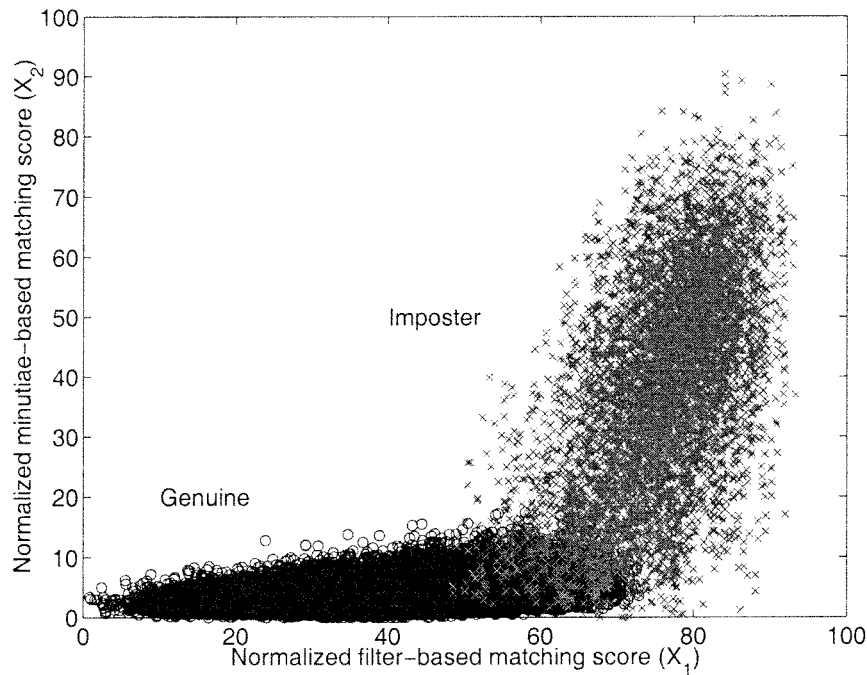
Fig. 15. Two-dimensional distribution of genuine and imposter scores from the proposed filter-based and the minutiae-based matchers for the MSU_DBI database. The matching distance obtained from the filter-based method was inverted (100-distance) to obtain a matching score for making the outputs of the two matchers consistent.
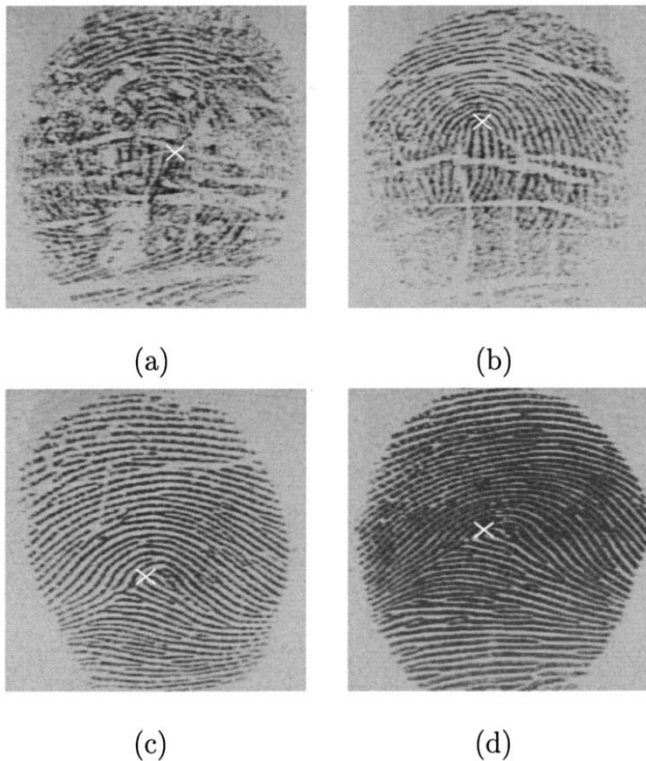

Fig. 16. Errors in filter-based matching; fingerprint images from the same finger which do not match: (a) and (b) do not match because of failure of reference location, and (c) and (d) do not match because of change in inter-ridge distance due to finger pressure difference.

There are a number of limitations of the initial implementation described in this paper. The representation and matching schemes assume that the reference point can be determined with a reasonable accuracy. A more realistic approach would consider a combination of frame determination methods and then verify the accuracy of the frame positioning by a consistency check among the results of several methods. As mentioned earlier, multiple frames could also be used to obtain multiple representations for a more robust performance. The current implementation requires that the entire region of interest be available and does not take into account occlusion or obliteration of a part of the fingerprint. This situation could be remedied by incorporation of "don't care" options for the components of the representation which do not correspond to fingerprint area. While the present approach tolerates small magnitudes of elastic distortion and local scaling (due to finger-pressure variations), it does not take care of significant nonlinear elastic distortion in the fingerprints. The inter-ridge densities in a fingerprint could be used to obtain a *canonical* representation to compensate for the large distortions due to shear and pressure variations caused by the contact of the finger with the sensing device. Further, indices composed from the attributes of the neighboring cells could also be used to compensate for the nonlinear distortions and inaccuracies in the determination of reference frame.

We are currently working on

1) handling nonlinear distortion;
2) refinements of initial strategies for feature extraction and matching;
3) indexing techniques based on the proposed representation.

REFERENCES

[1] A. K. Jain, L. Hong, S. Pankanti, and R. Bolle, "An identity authentication system using fingerprints," *Proc. IEEE*, vol. 85, pp. 1365–1388, Sept. 1997.

[2] A. K. Jain, R. M. Bolle, and S. Pankanti, Eds., *Biometrics: Personal Identification is a Networked Society.* Norwell, MA: Kluwer, 1999.

[3] H. C. Lee and R. E. Gaensslen, Eds., *Advances in Fingerprint Technology*, New York: Elsevier, 1991.

[4] L. O'Gorman, "Fingerprint verification," in *Biometrics: Personal Identification in a Networked Society*, A. K. Jain, R. Bolle, and S. Pankanti, Eds. Norwell, MA: Kluwer, 1999, pp. 43–64.

[5] D. Maio and D. Maltoni, "Direct gray-scale minutiae detection in fingerprints," *IEEE Trans. Pattern Anal. Machine Intell.*, vol. 19, pp. 27–40, Jan. 1997.

[6] A. K. Jain, S. Prabhakar, and L. Hong, "A multichannel approach to fingerprint classification," *IEEE Trans. Pattern Anal. Machine Intell.*, vol. 21, no. 4, pp. 348–359, 1999.

[7] G. T. Candela, P. J. Grother, C. I. Watson, R. A. Wilkinson, and C. L. Wilson, "PCASYS: A pattern-level classification automation system for fingerprints,", NIST Tech. Rep. NISTIR 5647, Aug. 1995.

[8] L. Hong and A. K. Jain, "Classification of fingerprint images," in *11th Scandinavian Conf. Image Analysis*, Kangerlussuaq, Greenland, June 7–11, 1999.

[9] M. Eshera and K. S. Fu, "A similarity measure between attributed relational graphs for image analysis," in *Proc. 7th Int. Conf. Pattern Recognition*, Montreal, P.Q., Canada, July 30–Aug. 3 1984.

[10] S. Gold and A. Rangarajan, "A graduated assignment algorithm for graph matching," *IEEE Trans. Pattern Anal. Machine Intell.*, vol. 18, no. 4, pp. 377–388, 1996.

[11] A. K. Hrechak and J. A. McHugh, "Automated fingerprint recognition using structural matching," *Pattern Recognit.*, vol. 23, pp. 893–904, 1990.

[12] A. Ranade and A. Rosenfeld, "Point pattern matching by relaxation," *Pattern Recognit.*, vol. 12, no. 2, pp. 269–275, 1993.

[13] J. Ton and A. K. Jain, "Registering landsat images by point matching," *IEEE Trans. Geosci. Remote Sensing*, vol. 27, pp. 642–651, May 1989.

[14] E. C. Driscoll, C. O. Martin, K. Ruby, J. J. Russel, and J. G. Watson, "Method and apparatus for verifying identity using image correlation," U.S. Patent 5 067 162, 1991.

[15] A. Sibbald, "Method and apparatus for fingerprint characterization and recognition using auto-correlation pattern," U.S. Patent 5 633 947, 1994.

[16] A. R. Rao and R. C. Jain, "Computerized flow field analysis: Oriented texture fields," *IEEE Trans. Pattern Anal. Machine Intell.*, vol. 14, pp. 693–709, July 1992.

[17] A. C. Bovik, M. Clark, and W. S. Geisler, "Multichannel texture analysis using localized spatial filters," *IEEE Trans. Pattern Anal. Machine Intell.*, vol. 12, pp. 55–73, Jan. 1990.

[18] J. Bigun, G. H. Granlund, and J. Wiklund, "Multidimensional orientation estimation with applications to texture analysis and optical flow," *IEEE Trans. Pattern Anal. Machine Intell.*, vol. 13, pp. 775–790, Aug. 1991.

[19] A. K. Jain and F. Farrokhnia, "Unsupervised texture segmentation using gabor filters," *Pattern Recognit.*, vol. 24, no. 12, pp. 1167–1186, 1991.

[20] M. Kass and A. Witkin, "Analyzing oriented patterns," *Comput. Vis. Graph. Image Processing*, vol. 37, no. 4, pp. 362–385, 1987.

[21] T. Chang, "Texture analysis of digitized fingerprints for singularity detection," in *Proc. 5th Int. Conf. Pattern Recognition*, 1980, pp. 478–480.

[22] M. Kawagoe and A. Tojo, "Fingerprint pattern classification," *Pattern Recognit.*, vol. 17, no. 3, pp. 295–303, 1984.

[23] A. R. Rao, *A Taxonomy for Texture Description and Identification*, New York: Springer-Verlag, 1990.

[24] D. Marr, *Vision.* San Francisco, CA: Freeman, 1982.

[25] J. G. Daugman, "High confidence recognition of persons by a test of statistical independence," *IEEE Trans. Pattern Anal. Machine Intell.*, vol. 15, no. 11, pp. 1148–1161, 1993.

[26] ——, "Uncertainty relation for resolution in space, spatial frequency, and orientation optimized by two-dimensional visual cortical filters," *J. Opt. Soc. Amer. A*, vol. 2, pp. 1160–1169, 1985.

[27] T. K. Ho, J. J. Hull, and S. N. Srihari, "On multiple classifier systems for pattern recognition," *IEEE Trans. Pattern Anal. Machine Intell.*, vol. 16, pp. 66–75, Jan. 1994.

[28] K. Woods, W. P. Kegelmeyer Jr, and K. Bowyer, "Combination of multiple classifiers using local accuracy estimates," *IEEE Trans. Pattern Anal. Machine Intell.*, vol. 19, pp. 405–410, Apr. 1997.

[29] E. S. Bigün, J. Bigün, B. Duc, and S. Fischer, "Expert conciliation for multi-modal person authentication systems by Bayesian statistics," in *Proc. 1st Int. Conf. Audio- Video-Based Biometric Person Authentication*, Crans-Montana, Switzerland, Mar. 1997, pp. 291–300.

[30] A. K. Jain, L. Hong, and Y. Kulkarni, "A multimodal biometric system using fingerprint, face, and speech," in *Proc. Int. Conf. Audio- Video-Based Biometric Person Authentication*, Washington, DC, Mar. 22–24, 1999, pp. 182–187.

[31] A. K. Jain, S. Prabhakar, and S. Chen, "Combining multiple matchers for a high security fingerprint verification system," *Pattern Recognit. Lett.*, vol. 20, no. 11–13, pp. 1371–1379, 1999.

[32] R. O. Duda and P. E. Hart, *Pattern Classification and Scene Analysis*, New York: Wiley, 1973.

[33] L. Hong, Y. Wan, and A. K. Jain, "Fingerprint image enhancement: Algorithm and performance evaluation," *IEEE Trans. Pattern Anal. Machine Intell.*, vol. 20, pp. 777–789, Aug. 1998.

Anil K. Jain (S'70–M'72–SM'86–F'91) is a University Distinguished Professor and Chair of the Department of Computer Science and Engineering at Michigan State University, East Lansing. His research interests include statistical pattern recognition, Markov random fields, texture analysis, neural networks, document image analysis, fingerprint matching, and 3-D object recognition. He is the co-author of *Algorithms for Clustering Data* (Englewood Cliffs, NJ: Prentice-Hall, 1988), edited the book *Real-Time Object Measurement and Classification* (Berlin, Germany: Springer-Verlag, 1988), and co-edited the books, *Analysis and Interpretation of Range Images* (Berlin, Germany: Springer-Verlag, 1989), *Markov Random Fields* (New York: Academic, 1992), *Artificial Neural Networks and Pattern Recognition* (Amsterdam, The Netherlands: Elsevier, 1993), *3D Object Recognition* (Amsterdam, The Netherlands: Elsevier, 1993), and *BIOMETRICS: Personal Identification in Networked Society* (Norwell, MA: Kluwer, 1998).

Dr. Jain received the Best Paper Awards in 1987 and 1991 and certificates for outstanding contributions in 1976, 1979, 1992, and 1997 from the Pattern Recognition Society. He also received the 1996 IEEE TRANSACTIONS ON NEURAL NETWORKS Outstanding Paper Award. He was the Editor-in-Chief of the IEEE TRANSACTIONS ON PATTERN ANALYSIS AND MACHINE INTELLIGENCE from 1990 to 1994. He is a Fellow of the IEEE and IAPR. He received a Fulbright Research Award in 1998.

Salil Prabhakar was born in Pilani, Rajasthan, India, in 1974. He received the B.Tech. degree in computer science and engineering from the Institute of Technology, Banaras Hindu University, Varanasi, India, in 1996. He is currently pursuing the Ph.D. degree at Michigan State University, East Lansing.

During 1996-1997, he was a Software Engineer with Tata Information Systems Ltd. (now IBM Global Services India Pvt. Ltd.), Bangalore, India. In the Fall of 1997, he joined the Department of Computer Science and Engineering, Michigan State University. His research interests include pattern recognition, image processing, computer vision, biometrics, network security, computer graphics, and multimedia systems.

Lin Hong received the B.S. and M.S. degrees in computer science from Sichuan University, China, in 1987 and 1990, respectively, and the Ph.D. degree in computer science from Michigan State University, East Lansing, in 1998.

He is currently with Visionics Corporation, Jersey City, NJ. His current research interests include multimedia, biometrics, data mining, pattern recognition, image processing, and computer vision application.

Sharath Pankanti is with the Exploratory Computer Vision and Intelligent Robotics Group, IBM T. J. Watson Research Center, Yorktown Heights, NY. He works on the Advanced Identification Solutions Project dealing with reliable and scalable identification systems. His research interests include biometrics, pattern recognition, computer vision, and human perception.