From Template to Image: Reconstructing Fingerprints from Minutiae Points

Arun Ross, *Member*, *IEEE*, Jidnya Shah, and Anil K. Jain, *Fellow*, *IEEE*

Abstract—Most fingerprint-based biometric systems store the minutiae template of a user in the database. It has been traditionally assumed that the minutiae template of a user does not reveal any information about the original fingerprint. In this paper, we challenge this notion and show that three levels of information about the parent fingerprint can be elicited from the minutiae template alone, viz., 1) the orientation field information, 2) the class or type information, and 3) the friction ridge structure. The orientation estimation algorithm determines the direction of local ridges using the evidence of minutiae triplets. The estimated orientation field, along with the given minutiae distribution, is then used to predict the class of the fingerprint. Finally, the ridge structure of the parent fingerprint is generated using streamlines that are based on the estimated orientation field. Line Integral Convolution is used to impart texture to the ensuing ridges, resulting in a ridge map resembling the parent fingerprint. The salient feature of this noniterative method to generate ridges is its ability to preserve the minutiae at specified locations in the reconstructed ridge map. Experiments using a commercial fingerprint matcher suggest that the reconstructed ridge structure bears close resemblance to the parent fingerprint.

Index Terms—Fingerprints, minutiae, templates, security, fingerprint reconstruction, line integral convolution, streamlines.

1 Introduction

A fingerprint is an oriented texture pattern consisting of ridges and valleys present on the tip of an individual's finger. The ridges exhibit various types of imperfections, called minutiae (minor details in fingerprints). Among a total of 150 different minutiae types [24], the ridge ending and ridge bifurcation are the most stable points in a fingerprint. The distribution of these points in a fingerprint has been observed to be unique across individuals. In fact, this distribution is claimed to be unique to each finger of an individual. Thus, most automatic fingerprint authentication systems do not store the raw fingerprint image of a user in its entirety during enrollment. Rather, a template consisting of a set of salient features (e.g., singular points, such as core and delta, and ridge anomalies, such as minutiae) from the fingerprint image is stored in the database. Since the template, by definition, is a compact description of the biometric sample, it is not expected to reveal significant information about the original data. Therefore, template-generation algorithms have been traditionally assumed to be "one-way" algorithms. However, Hill [14] designed a technique to determine the fingerprint structure from the minutiae template alone. He assumed that a fingerprint template stores the coordinates of the core and delta points (if present) along with the minutiae points. His technique utilized the location of the singular points to derive the orientation map of the fingerprint based on the method proposed in [31]. This orientation map was then used by a heuristic line drawing algorithm to generate a sequence of splines passing through the minutiae points. Hill demonstrated his reconstruction scheme on a small database of 25 arch type fingerprints. His scheme also predicted the shape of the fingerprint (i.e., its class) using a neural network classifier consisting of 23 input neurons, 13 hidden neurons, and four output neurons (corresponding to the four major fingerprint classes, viz., **A**, **L**, **R**, and **W**). However, the classification performance was observed to be rather low (an error rate of 28.9 percent on a small set of 242 fingerprints). Similarly, in the face domain, Adler [2] demonstrated that a face image could be reconstructed from face templates using a "hill climbing attack."

The information elicited from a biometric template can be categorized into three distinct levels based on the complexity of the reconstructed information/image [12]: 1) feature reconstruction: an image that is sufficient to fool a biometric system but *unrepresentative* of the underlying physical characteristics to the naked eye, 2) generic image recreation: an image *resembling* the actual physical trait, and 3) total image recreation: an image *identical* to the original physical trait.

In this paper, we concern ourselves with systematically addressing the following question: How much information does the minutiae template reveal about the original fingerprint image?[1] The benefits of such a study are two fold: 1) It helps in understanding the vulnerability of decrypted fingerprint templates to a masquerade attack (generating a

- A. Ross is with the Lane Department of Computer Science and Electrical Engineering, West Virginia University, PO Box 6109, Morgantown, WV 26506. E-mail: arun.ross@mail.wvu.edu.
- J. Shah is with L-1 Identity Solutions Corporate Research Center, One Exchange Place, Jersey City, NJ 07302. E-mail: jidnya@gmail.com.
- A.K. Jain is with the Department of Computer Science and Engineering, Michigan State University, 3115 Engineering Building, East Lansing, MI 48824. E-mail: jain@cse.msu.edu.

Manuscript received 1 Feb. 2006; revised 7 June 2006; accepted 9 June 2006; published online 18 Jan. 2007.
Recommended for acceptance by S. Prabhakar, J. Kittler, D. Maltoni, L. O'Gorman, and T. Tan.
For information on obtaining reprints of this article, please send e-mail to: tpami@computer.org and reference IEEECS Log Number TPAMISI-0109-0206.
Digital Object Identifier no. 10.1109/TPAMI.2007.1018.

[1] The minutiae template is assumed to be decrypted.

0162-8828/07/$25.00 © 2007 IEEE Published by the IEEE Computer Society

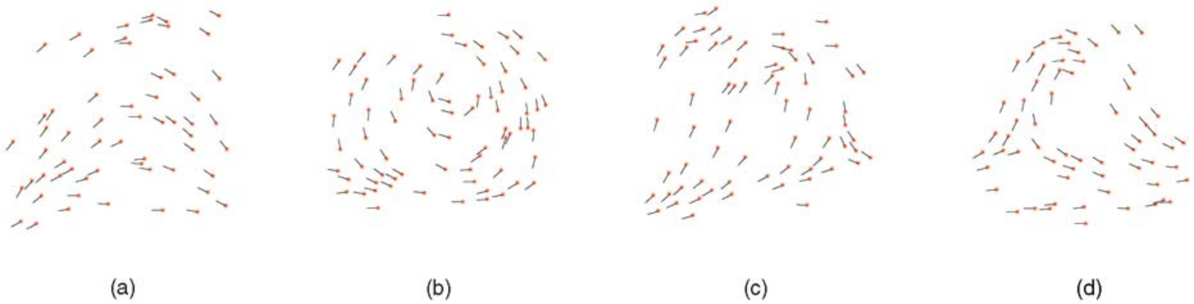

Fig. 1. Minutiae plots of four major fingerprint classes: (a) Arch $(\mathbf{A})$. (b) Whorl $(\mathbf{W})$. (c) Left Loop $(\mathbf{L})$. (d) Right Loop $(\mathbf{R})$. Lines associated with the minutiae depict their orientation.

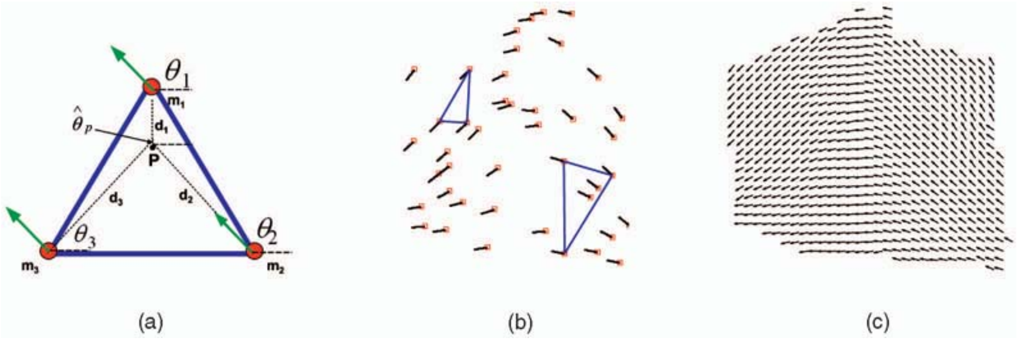

Fig. 2. Deducing the orientation field from minutiae distribution. (a) A single minutiae triplet. (b) Forming triplets across the minutiae distribution. (c) Estimated orientation field using minutiae triplet information.

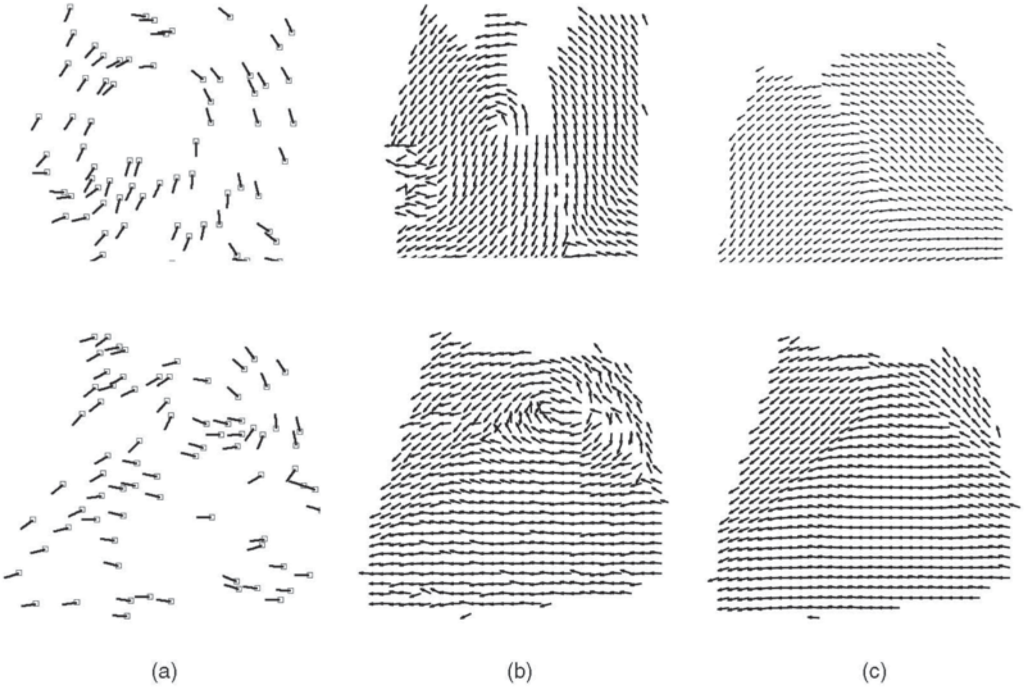

Fig. 3. Comparing the estimated orientation map $\hat{\boldsymbol{\theta}}$ with the true map $\boldsymbol{\theta}$ for two different fingerprints. (a) Minutiae template. (b) True orientation map. (c) Estimated orientation map. Due to the absence of valid triplets in certain regions, the estimated orientation map cannot characterize these regions, e.g., the concentric pattern in whorls.

physical artifact from the stored template and using it for spoofing fingerprint devices) and 2) it provides insight into the individuality of fingerprints as assessed using the minutiae distribution. Specifically, we attempt to extract three levels of information about the original fingerprint from its minutiae template, viz., the ridge orientation, class, and ridge structure of the original fingerprint. Our approach is significantly different from the one proposed by Hill [14] both in its formulation as well as its scope. While Hill's orientation field estimation was based on singular point information, our method relies on minutiae information alone. Thus, it obviates the need for the presence of singular point information in the

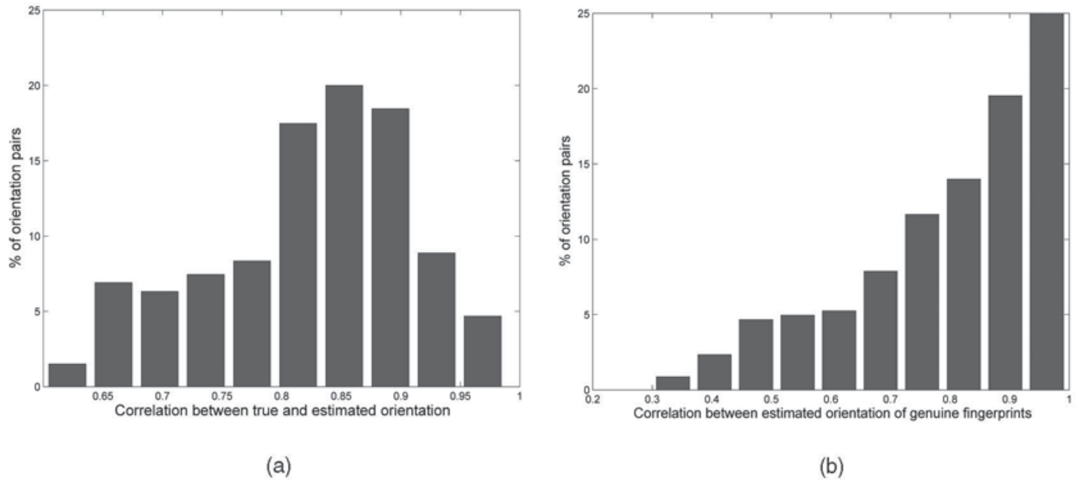

Fig. 4. Validating the estimated orientation map. (a) Correlation between $\hat{\boldsymbol{\theta}}$ and $\boldsymbol{\theta}$ as observed on the NIST 4 database. (b) Correlation between $\hat{\boldsymbol{\theta}}$ of 400 genuine fingerprint pairs from the $\mathbf{L}$ class in the NIST 4 database.

template. We assume the *simplest* characterization of a minutiae template, i.e., the location, (x, y), and the orientation, θ, of component minutiae points.

The rest of the paper is organized as follows: The algorithm for orientation estimation is described in Section 2. The minutiae-based classification algorithm is explained in Section 3. Section 4 presents the algorithm for fingerprint reconstruction. Finally, Section 5 draws some concluding remarks and discusses possible future work in this area.

2 Estimating Ridge Orientation

The orientation of a minutia is an indication of the local ridge direction since the fingerprint is a smoothly changing oriented texture pattern. Fig. 1 shows the minutiae plots of four major classes of fingerprints viz., Arch ($\mathbf{A}$), Whorl ($\mathbf{W}$), Left loop ($\mathbf{L}$), and Right loop ($\mathbf{R}$). These plots clearly suggest the possibility of deducing the direction of local ridges by examining the orientation of minutiae points in that local region. Therefore, by observing the orientation of a group of neighboring minutiae, one can "interpolate" the underlying local ridge information. The proposed algorithm utilizes a set of three minutiae points (minutiae triplet) to predict the orientation of a triangular fingerprint region defined by the triplet. In our formulation, as stated earlier, a minutia point is represented as $(\mathrm{x}, \mathrm{y}, \theta)$, with (x, y) being its spatial location and θ its orientation.[2] The algorithm for generating the orientation map has three main stages: 1) triplet generation, 2) orientation estimation, and 3) averaging orientation map.

1. **Triplet generation**. Consider a minutiae template, $\mathcal{M}$, of a fingerprint containing N minutiae points given by $\mathcal{M} = \{m_1, m_2, \cdots, m_N\}$, where $m_i = (x_i, y_i, \theta_i)$. A set of three minutiae points, $\{m_i\}_{i=1,2,3}$, characterized by a triangle with sides $\{L_i\}_{i=1,2,3}$ and interior angles $\{\phi_i\}_{i=1,2,3}$, is said to constitute a "valid" triplet, T, if the following conditions hold:

   a. $L_{min} \leq L_i \leq L_{max}$, $\forall\ i = 1, 2, 3$. This ensures that the perimeter of the triangle traverses a compact region, thus avoiding the large global variability observed in the fingerprints of most classes.
   b. $\theta_{dif} \leq \theta_{tol}$, where $\theta_{dif} = \max_{i=1,2,3}(\theta_i - \theta_{med})$ and θ_{med} is the median of $\{\theta_i\}_{i=1,2,3}$. This ensures that the orientations of component minutiae points in the triplet are within a small tolerance interval θ_{tol}.
   c. $\phi_i > \phi_{min}$, $\forall\ i = 1, 2, 3$. This ensures that "narrow" triangles subtending a very small area are avoided.

In our experiments using the NIST-4 database, the following parameter values were used: $L_{min} = 20$, $L_{max} = 300$, $\theta_{tol} = 30^\circ$, and $\phi_{min} = 20^\circ$.

In a fingerprint image, minutiae tend to appear in clusters [21]. For instance, the regions near the core and delta have dense minutiae activity. In such cases, a triplet may reside inside the triangular region of another triplet or may overlap with it. Rather than consolidating the orientation information estimated

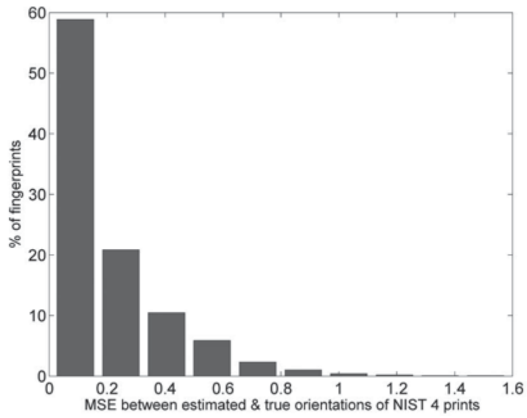

Fig. 5. Validating the estimated orientation map. MSE between $\hat{\boldsymbol{\theta}}$ and $\boldsymbol{\theta}$ as observed on the NIST 4 database.

[2] For orientation prediction, we do not make a distinction between opposing angles, i.e., both 30° and 210° orientations are assumed to be the same. The range of θ is [90°, 270°].

TABLE 1
A Summary of Different Fingerprint Classification Techniques

Authors	Features used
Candela et al. 1995 [5], Hong et al. 1999 [15], Chong et al. 1997 [10]	Ridge line flow
Cappelli et al. 1999 [6], Senior 2001 [30], Wilson et al. 1994 [34]	Orientation image
Karu and Jain 1996 [20], Cho et al. 2000 [9]	Singularities
Jain et al. [17] 1999, Yao et al. 2001 [35]	Gabor filters
Ross et al. 2005 [27]	Minutiae points

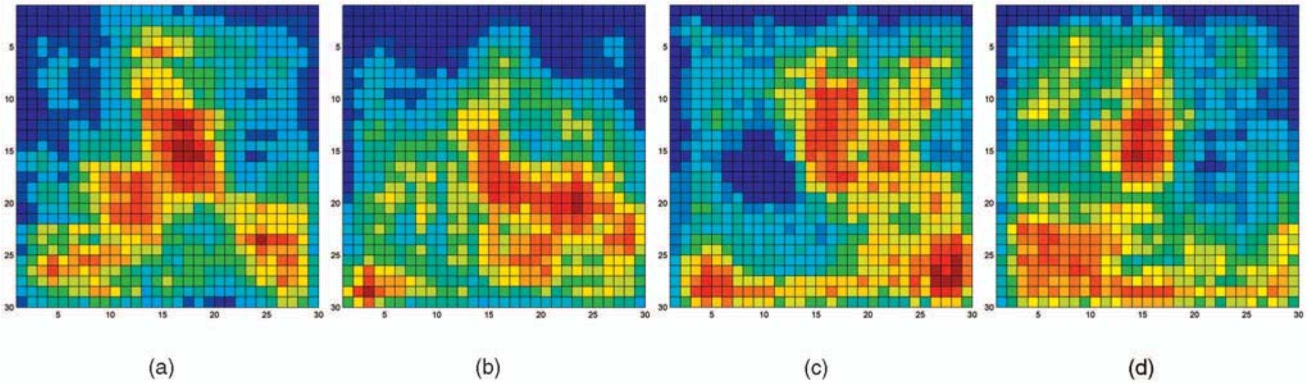

Fig. 6. Minutiae density associated with four different classes of fingerprints: (a) $\mathbf{A}$, (b) $\mathbf{W}$, (c) $\mathbf{L}$, and (d) $\mathbf{R}$. These plots were generated using 30 images per class. Blue indicates a low density region, while red indicates a high density region.

by multiple triplets, we utilize the information presented only by a good quality triplet. The quality, Q, of each selected triplet is measured by examining the average length of the sides of the triangle and the orientations of component minutiae points and is computed as

$$Q = (L_{max} - L_{avg})w_1 + \left(\frac{\theta_{tol} - \theta_{diff}}{\theta_{diff}} L_{max}\right)w_2. \quad (1)$$

Here, L_{avg} is the average length of the sides of the triplet, θ_{diff} is the maximum pairwise difference between the three minutiae orientations, and w_1 and w_2 are the weights associated with each term ($w_1 = 0.4$ and $w_2 = 0.6$, in our experiments). This ensures that a triplet containing minutiae of similar orientations and covering a relatively small area is assigned a higher Q value. A good quality triplet is expected to result in a better estimation of the orientation of the underlying ridges. Here, the orientations are averaged over a block (13×13 in our experiments). Hence, for a 512×512 image, a 39×39 orientation map, $\hat{\boldsymbol{\theta}}$ is generated. Certain points in this grid may not contain any orientation information due to the nonavailability of "valid" triplets in those regions.

2. **Orientation estimation**. Consider a pixel $P(x, y)$ located inside the triangular region defined by a triplet. Let $d_i = \text{dist}\{m_i, P\}$, $i = 1, 2, 3$ be the Euclidean distances of this pixel from all the ith vertex. The orientation of the pixel P, $\hat{\boldsymbol{\theta}}_P$, is then estimated as

$$\hat{\boldsymbol{\theta}}_P = \frac{d_3 d_2}{(d_3 d_2 + d_1 d_3 + d_1 d_2)}\theta_1 + \frac{d_1 d_3}{(d_3 d_2 + d_1 d_3 + d_1 d_2)}\theta_2 + \frac{d_1 d_2}{(d_3 d_2 + d_1 d_3 + d_1 d_2)}\theta_3.$$

   The angle θ_1 (θ_3) corresponds to the orientation of the vertex that is nearest to (farthest from) the pixel $P(x, y)$. Thus, the orientation at $P(x, y)$ is estimated as a weighted sum of all the three orientations with a higher weight assigned to the orientation of the closest vertex. The result of the generated orientation map is shown in Fig. 2c.

3. **Averaging orientation map**. To obtain a smooth transition in orientations, the estimated orientation map is convolved with a 3×3 local averaging filter.

2.1 Validating the Estimated Orientation Map $\hat{\boldsymbol{\theta}}$

Visually, we observe that the estimated orientation map is fairly consistent with the underlying ridge flow of the original fingerprint. In order to verify the accuracy of the algorithm, we compare the true orientation map with the estimated one using a correlation measure. The true orientation map, $\boldsymbol{\theta}$, can be computed using various techniques as described in [23]. We use the least mean square orientation approach [26] for estimating the true orientation maps of fingerprint images from the NIST 4 database. The true and estimated orientation maps ($\boldsymbol{\theta}$ and $\hat{\boldsymbol{\theta}}$) for two different fingerprints are shown in Fig. 3.

To determine the similarity between the true and estimated orientation maps, we compute the Pearson's correlation coefficient, $r(\boldsymbol{\theta}, \hat{\boldsymbol{\theta}})$. Fig. 5a shows the histogram of correlation coefficients computed for the NIST 4 database (2,000 fingerprint pairs), indicating that the estimated ridge orientations are correlated with the underlying true ridge flow. About 79 about of the orientation pairs have correlation values greater than 0.75. In order to present a context to this histogram, another histogram of correlation coefficients is generated by comparing the estimated orientation maps of 400 genuine fingerprint pairs in the NIST 4 database (after accounting for the translation and rotation parameters

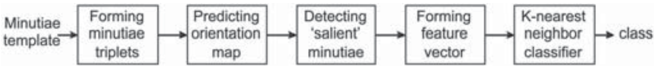
Minutiae template
Forming minutiae triplets
Predicting orientation map
Detecting 'salient' minutiae
Forming feature vector
K-nearest neighbor classifier
class

Fig. 7. Estimating fingerprint class based on minutiae.

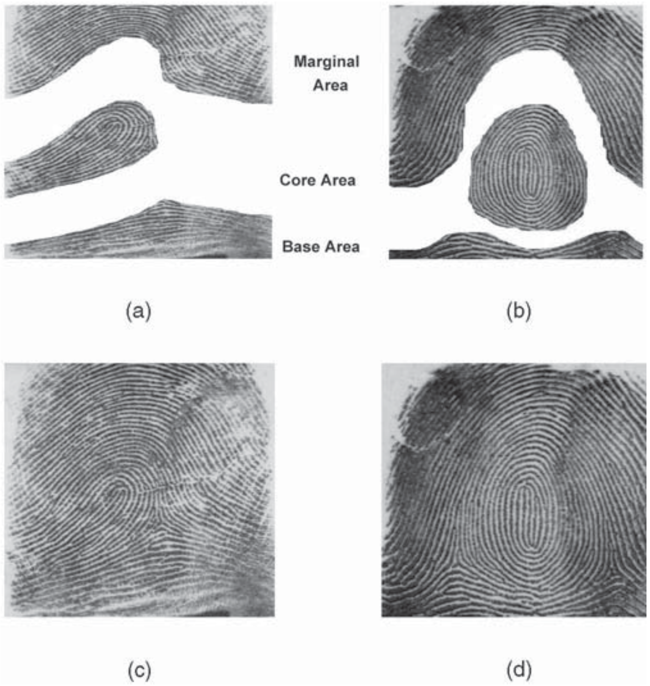

Fig. 8. Finding a region containing "salient" minutiae: (a) and (b) show two fingerprints of class $\mathbf{L}$ and $\mathbf{W}$, respectively, divided into several regions. Note that the region in the vicinity of the core attains the most discriminating information for type classification. (c) and (d) are the original fingerprints corresponding to (a) and (b), respectively.

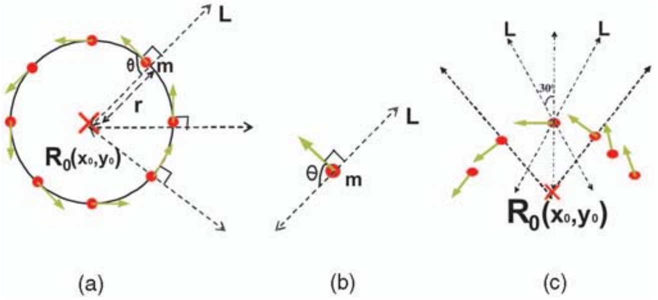

Fig. 9. Estimating the registration point (R_0): (a) Circular plot of minutiae. (b) Line L is perpendicular to the orientation of minutiae m. (c) Line L made nearly perpendicular $(\pm 30^{\circ})$ to the orientation θ.

between image pairs). Fig. 4b clearly indicates that the estimated orientations of the genuine fingerprint pairs are correlated and, thus, confirms the efficacy of the orientation estimation algorithm that is based on minutiae points alone.

To determine the accuracy of the orientation estimation technique, we compute the Mean Square Error (MSE) between the estimated $\hat{\boldsymbol{\theta}}$ and true orientation $\boldsymbol{\theta}$ maps. Fig. 5 shows the histogram of MSE computed for the NIST 4 database. It suggests that the estimated ridge orientations are reasonably consistent with the underlying true ridge flow. About 72 percent of the orientation pairs are observed to have MSE values less than than 0.25. This further illustrates the ability of the proposed algorithm to predict the underlying ridge flow.

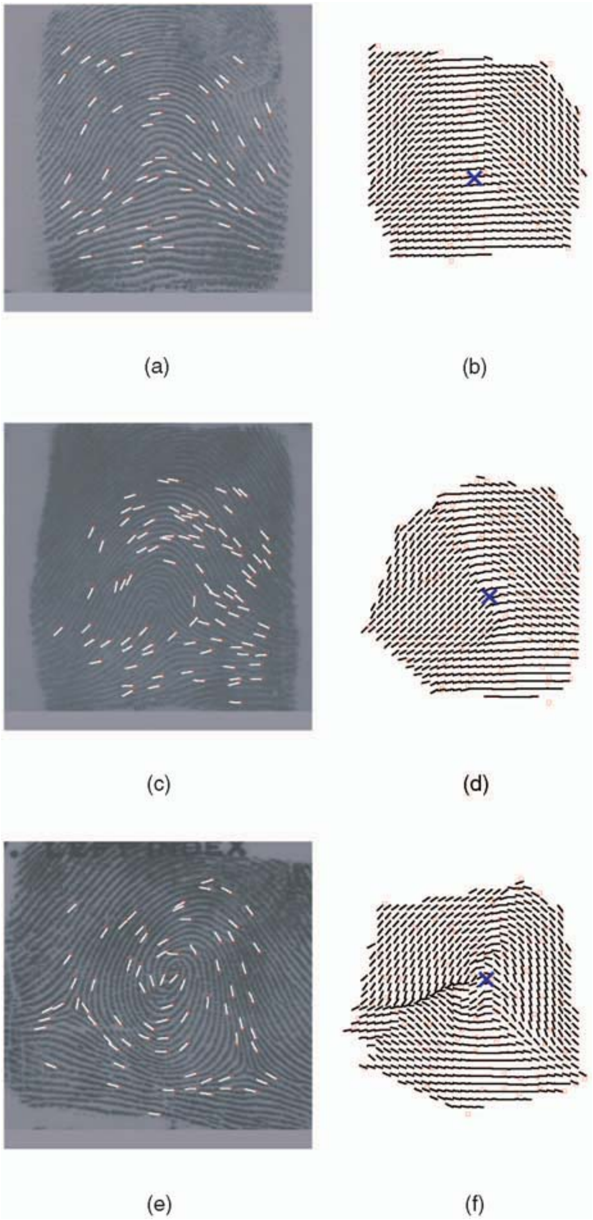

Fig. 10. Minutiae plot overlaid on original fingerprint belonging to classes $\mathbf{A}$, $\mathbf{L}$, and $\mathbf{W}$ are shown in (a), (c), and (e), respectively. The corresponding R_0 points marked in blue ("X") detected using the estimated orientation maps and minutiae points are shown in (b), (d), and (f).

3 Predicting Fingerprint Class from Minutiae Template

The ridge pattern in a fingerprint allows for its systematic classification. According to the Galton-Henry classification scheme [13], fingerprints can be classified into four main classes, viz., Arch $(\mathbf{A})$, Right loop $(\mathbf{R})$, Left loop $(\mathbf{L})$, and Whorl $(\mathbf{W})$. Tented Arch are included in the Arch $(\mathbf{A})$ class. Loops $(\mathbf{L}$ and $\mathbf{R})$ constitute about 65 percent of the total fingerprint patterns; whorls make up about 30 percent and arches and tented arches together account for the other 5 percent. A visual glance at the minutiae plots of the four classes (Fig. 1) suggests the possibility of predicting the fingerprint class from the minutiae points.

Fingerprint classification refers to the problem of assigning a fingerprint to a predefined class in a consistent and reliable way. Usually, fingerprint matching is performed using local features such as local ridge and minutiae details, whereas fingerprint classification is accomplished using global features, such as global ridge structure and singularities. Most fingerprint classification schemes in the literature use the fingerprint image along with one or more features like ridge line flow, singularities, orientation image, and Gabor filter responses [23]. Table 1 summarizes the various features that have been used in the literature for fingerprint classification.

We describe a novel algorithm for fingerprint classification which uses only the minutiae information. Galton [11] showed that there is a strong correlation between the class of a fingerprint and the occurrence of a minutiae at a specific location in the image. This is evident in Fig. 6. Here, the fingerprint area has been divided into fixed blocks (for each class) and the probability of minutiae occurring in each block is estimated based on a sample of 30 fingerprints per class. The minutiae density is observed to vary across the fingerprint image for the four fingerprint classes. Similar observations have been reported in the forensic literature also (see [28], for example), further substantiating our hypothesis that the seemingly random distribution of minutiae in a fingerprint can reveal important information about the *class* of the fingerprint.

We assume that only the position and orientation (x, y, θ) of a minutia point is available in the template and that information about the singularity points is not known.[3] Fig. 7 summarizes the main steps in the proposed classification algorithm.

3.1 Detecting Registration Point

A visual analysis of the fingerprint ridge patterns of various classes reveals that they have almost the same ridge structure in the base and marginal areas, as shown in Fig. 8. However, it is the irregularities in the vicinity of the core region that are significant for classification (such as the circular ridge pattern in the case of whorls or the curving back of ridges in loops). In order to select these "salient" minutiae, we attempt to detect a registration point (R_0) using the Hough transform [3].

The ridges around the core point have a high curvature and form a nearly circular pattern. Accordingly, the orientations of the minutiae in such regions define a nearly circular pattern. Consider a circle $(x - x_0)^2 + (y - y_0)^2 = r^2$ defined by a group of minutiae, as shown in Fig. 9a. Here, x_0 and y_0 are the coordinates of the center of the circle and r is its radius. Our goal here is to detect the center of the circle. It can be observed that the minutiae depicted in Fig. 9a have orientations almost tangential to the circumference of the circle. Using this property, for each minutiae m we traverse a line L that is perpendicular to its orientation θ. This line is viewed as a set of discrete points, as shown in Fig. 9b. These points correspond to the probable center points of the circle. The radius r of each of these circles is the distance between the minutiae and the corresponding center points since the minutia m lies on their circumferences. An accumulator in 3D Hough space corresponding to the center (x, y) and radius r is used to detect the center. For a circular minutiae pattern, there will be a well-pronounced peak in the Hough parameter space corresponding to the center (x_0, y_0) (point marked with red "x" in Fig. 9a). This is the registration point characterized by significant minutiae activity around it. Since the ridge structure in fingerprints is not exactly circular, we use lines that are nearly perpendicular $(\pm 30^\circ)$ to the orientation θ, as shown in Fig. 9c. The results of the Hough transform using the estimated orientation maps and the minutiae points are shown in Fig. 10. The purpose of this exercise is *not* to detect the core of the fingerprint; rather, it is to detect a registration point that can be used to extract the "salient" minutiae. Only the minutiae located in a 300×300 pixels region about R_0 are used for classification (Fig. 11).

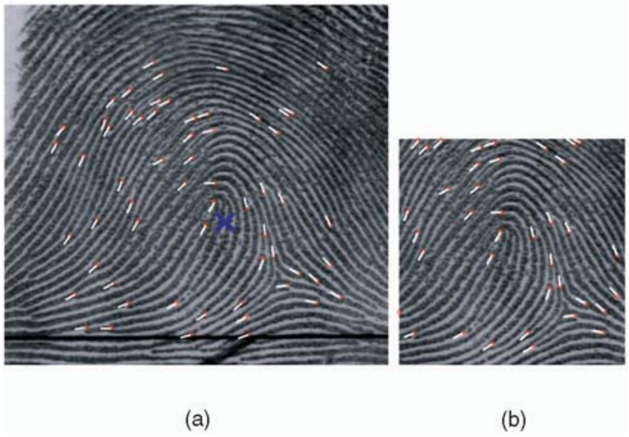

Fig. 11. Extracting salient minutiae. (a) The orientation map and the detected registration point. (b) Salient minutiae in a 300×300 region about R_0.

3.2 Generating Feature Vectors

A feature vector is next extracted from the set of salient minutiae identified in the previous stage. The features in this vector capture various properties of the minutiae such as the relationship between minutiae location and orientation, the clustering property of minutiae, the relationship between minutiae pairs, etc. The features we have designed in this regard are invariant to the rotation and translation of fingerprint images. The 11-dimensional feature vector $F = \{F_1, F_2, \cdots, F_{11}\}$ is constructed as follows.

3.2.1 *Features Based on Minutiae Orientations (F_1, F_2)*

The orientations represented by the minutiae vary across the four classes. For instance, a whorl fingerprint has at least one ridge which traverses a 360° closed path in the central region of the fingerprint. Thus, the orientations of these minutiae range from 0-360 degrees. On the other hand, the minutiae orientations of arches have only two dominant directions. In order to understand the distribution of minutiae orientations for each class, we examine the

[3] Note that the basic shape of the fingerprint is decided by the location and orientation of singularities. So, if the minutiae template were to store the position and orientation of singularities, then determining the class of the fingerprint is a rather simple task.

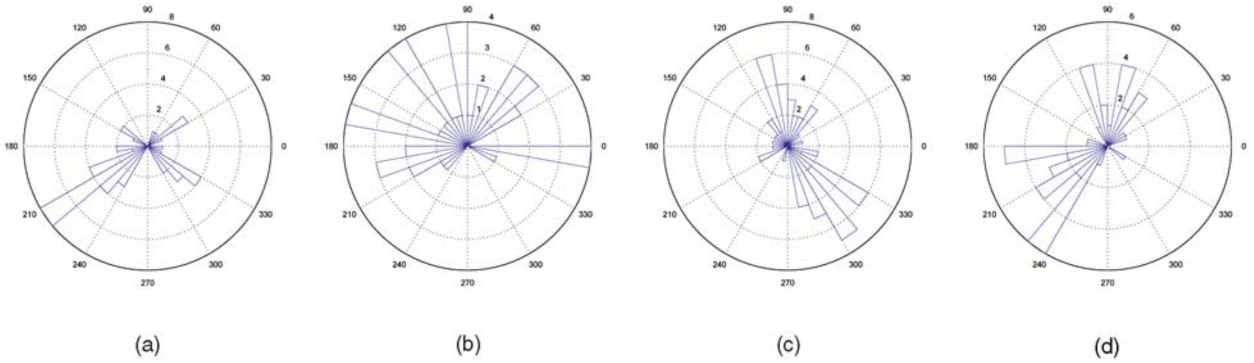

Fig. 12. Rose plots of (a) $\mathbf{A}$, (b) $\mathbf{W}$, (c) $\mathbf{L}$, and (d) $\mathbf{R}$.

rose plots[4] of minutiae orientations (Fig. 12). The range of dominant minutiae orientations is captured by counting the number of empty bins in the rose plot (feature F_1). This feature effectively discriminates between whorl and arch types, though it may not discriminate between the other classes. Feature F_2 is used to denote the variance in minutiae orientations present in the template.

3.2.2 Features Based on Minutiae Pairs (F_3 to F_6)

Minutiae pairs are fundamental units for representing variations in fingerprints [32]. The properties of neighboring minutiae change across classes. For instance, the neighboring minutiae in the central region of $\mathbf{W}$ have large orientation differences, whereas minutiae neighbors in $\mathbf{A}$ have similar orientations. The correlation between spatial location and orientation of minutiae pairs can be examined by estimating the joint distribution of R and Φ, where R is the distance between two minutiae and Φ is the difference in their orientation (as shown in (4) to (7)). Let $P(R,\Phi)$ denote the probability of observing a minutiae pair which are separated by a distance R and with difference in orientation Φ. Then, F_3 is the number of minutiae pairs that are spatially compact and have almost similar orientations; F_4 is the number of pairs that are spatially compact but have a large orientation difference; F_5 is the number of minutiae pairs that are spatially separated and have similar orientations; F_6 is the number of minutiae pairs that are spatially separated but have large differences in orientation. Fig. 13 illustrates these four types of minutiae pairs.

$$F_3 = \int_{0\le R\le R_1}\int_{0\le\Phi\le\Phi_1} P(R,\Phi)\, dR\, d\Phi, \tag{2}$$

$$F_4 = \int_{0\le R\le R_1}\int_{\Phi_1\le\Phi\le\Phi_2} P(R,\Phi)\, dR\, d\Phi, \tag{3}$$

$$F_5 = \int_{R\ge R_2}\int_{0\le\Phi\le\Phi_1} P(R,\Phi)\, dR\, d\Phi, \tag{4}$$

$$F_6 = \int_{R\ge R_2}\int_{\Phi_1\le\Phi\le\Phi_2} P(R,\Phi)\, dR\, d\Phi. \tag{5}$$

In our experiments with the NIST-4 database, we have set $R_1 = 60$ pixels, $R_2 = 180$ pixels, $\Phi_1 = 30^\circ$, and $\Phi_2 = 180^\circ$. It is observed that these features are significantly different for the four fingerprint classes. For arches, F_3 takes relatively larger values compared to the other three features, whereas, for whorls, F_4 is typically the largest.

3.2.3 Features Based on Minutiae Clusters (F_7, F_8)

Minutiae tend to cluster in certain regions of the fingerprint. For example, minutiae activity increases in the core and delta regions of a fingerprint [21], [32], [8]. Variations in ridge flow seem to contribute to a high incidence of minutiae points. In order to capture these variations across fingerprint classes, we compute the minutiae density in circular regions representing a radii of 50 pixels. Feature F_7 is defined to be the maximum minutiae density corresponding to a particular fingerprint template. The value of F_7 is relatively high for whorls and small for arches. Feature F_8 is the maximum variance in minutiae orientations observed across all the circular regions considered.

3.2.4 Features Based on Global Ridge Information (F_9, F_{10}, F_{11})

Visually (Fig. 1), it is apparent that features F_1 to F_8 can possibly distinguish classes $\mathbf{A}$ and $\mathbf{W}$, but are not sufficient for reliably resolving ambiguity between classes ($\mathbf{L}$, $\mathbf{R}$), ($\mathbf{A}$, $\mathbf{L}$, $\mathbf{R}$), and ($\mathbf{L}$, $\mathbf{R}$, $\mathbf{W}$). It is necessary to include information about global ridge pattern in conjunction with the local minutiae properties to address this issue. To capture the global ridge structure of the fingerprint, we define geometric kernels which model the shape of the fingerprint around the core region for the $\mathbf{W}$, $\mathbf{L}$, and $\mathbf{R}$ classes. In a left loop, the ridges in the core region form a

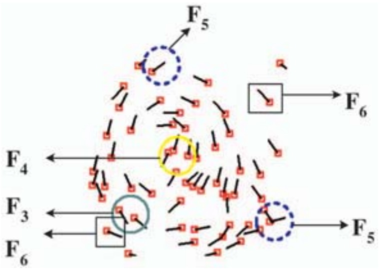

Fig. 13. A minutiae plot illustrating the four types of minutiae pairs.

[4] The rose plot is a a polar plot showing the histogram of angles.

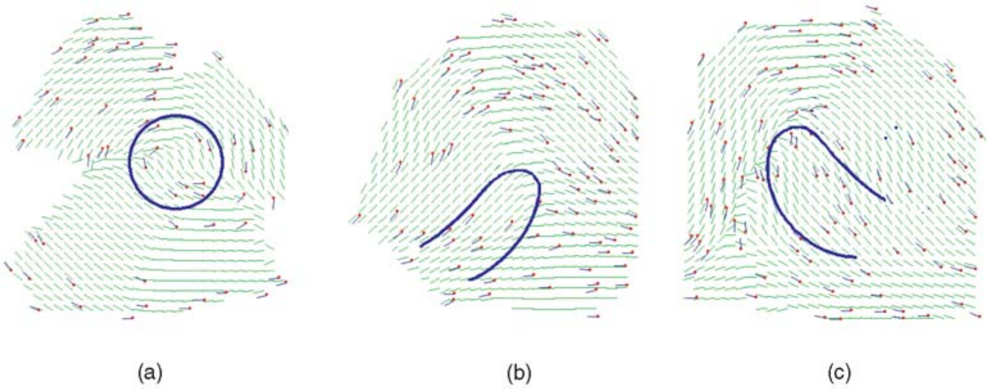

Fig. 14. Kernels for (a) $\mathbf{W}$, (b) $\mathbf{L}$, and (c) $\mathbf{R}$.

loop by recurving to the left side of the fingerprint. This is captured by a kernel constructed using two semi-ellipses corresponding to the concave and convex portions of the loop. The kernel for $\mathbf{R}$ is merely a mirror image of the $\mathbf{L}$ kernel. The circular ridge structure of $\mathbf{W}$ is captured using a simple circular kernel. Since the marginal area of every fingerprint has arch-like characteristics (Fig. 8), we do not define a separate kernel for the $\mathbf{A}$ class as it would align well with fingerprints of all the other classes. See Fig. 14.

In order to determine a goodness-of-fit for these kernels, we use a model-based scheme by modifying the hierarchical kernel fitting approach proposed by Jain and Minut [18]. In this approach, fingerprint classification is achieved by finding the kernel that best fits the flow field (i.e, orientation field) of a given fingerprint template. Consider V to be a smooth vector field defined over some region in the plane R^2 and let β be its argument. Let γ be a circular kernel curve in R^2, as shown in Fig. 15.

Let $\dot{\gamma}$ be the tangent to γ and α be its argument. Consider a point $\gamma_t = (x(t), y(t))$ present on the kernel γ. An energy functional capturing the difference between the direction of $\dot{\gamma}$ and that of the vector field V at point γ is defined by

$$E(\gamma) = \frac{\int_\gamma sin^2(\alpha - \beta(\gamma))\, d\gamma}{\int_\gamma d\gamma}. \tag{6}$$

To determine how well each kernel fits the estimated orientation map of the fingerprint, energy values (6) for various discrete points on the kernel are computed. The average of these values is used to determine the goodness of the fit. A lower energy average means a better fitting kernel. For each estimated orientation map (from minutiae information), we find three values for the energy functionals corresponding to the $\mathbf{L}$, $\mathbf{R}$, and $\mathbf{W}$ kernels. These functionals form features (F_9, F_{10}, F_{11}), respectively. For an orientation map corresponding to $\mathbf{W}$, it is expected that the minimum of the three features will be F_{11}, whereas, for $\mathbf{L}$ and $\mathbf{R}$, the minimum will correspond to F_9 and F_{10}, respectively. We observed that, due to the inherent similarity in ridge structure of loops and arches, both the $\mathbf{L}$ and $\mathbf{R}$ kernels fit well for fingerprints belonging to class $\mathbf{A}$. Thus, F_9 and F_{10} are similar-valued for arches. This property is useful for resolving the ambiguity between loops and arches.

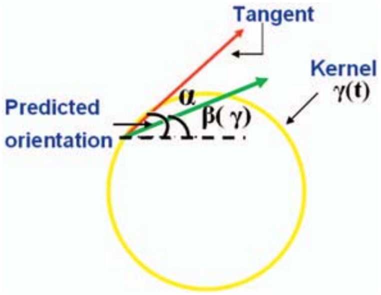

Fig. 15. Fitting kernels to flow fields. Red vector: Unit tangent vector to the kernel. Green vector: Direction of flow field at point γ_t [18].

The class-specific kernels are defined with respect to the registration point, R_0, obtained using the Hough transform, i.e., for the $\mathbf{W}$ kernel, R_0 is the center of the circle, whereas, for the $\mathbf{L}$ and $\mathbf{R}$ kernels, R_0 is the focus of the elliptical kernel. The rotation and translation of fingerprints are taken into account by subjecting these kernels to certain predefined transformations prior to applying them. For $\mathbf{W}$, the radius of the kernel is varied between 100 to 160 pixels. For $\mathbf{L}$ and $\mathbf{R}$, the semi-major axis is varied between 120 to 180, while the semi-minor axis is varied between 60 to 100 pixels. The angle that the ellipse subtends with the horizontal is varied between -10 to 10 degrees. Further, the kernels are moved in a 20×20 window around R_0. The features (F_9, F_{10}, F_{11}) correspond to the *minimum* of the energy functional obtained across all transformations for *each* kernel.

3.3 Classification of Fingerprints

A K-nearest neighbor classifier employing the Manhattan distance was used to classify the minutiae sets in the NIST-4 fingerprint database based on the 11 features described above.[5] A good quality fingerprint typically consists of between 40-100 minutiae [16], but this is not true for some fingerprints in the NIST-4 database. Since a sufficient number of minutiae triplets is necessary to compute the orientation map, we rejected fingerprints with less than 25 minutiae. Also,

[5] Since $\mathrm{K} = 5$ resulted in the best performance, we report the classification performance for the 5-NN classifier only.

TABLE 2
Confusion Matrix Indicating Classification Performance Using Minutiae Information Alone

True	Assigned Class			
Class	A	L	R	W
A	467	45	32	6
L	61	464	7	18
R	69	26	448	7
W	4	61	68	417

in order to ensure an equal distribution of templates across the four classes, we considered only 2,800 minutiae templates extracted from the NIST-4 database (i.e., 700 templates per class).

From each of the four classes, we used 150 fingerprints (randomly chosen) for training and 550 fingerprints for testing the classification performance. We invoked an exhaustive feature selection technique to select the optimal features resulting in the best classification performance. We observed that, by reducing the dimensionality of the feature vector from 11 to 8, the classifier performance could be improved. The 8 selected features were F_1, F_2, F_3, F_4, F_7, F_9, F_{10}, F_{11}. A 4-class classification rate of 84 percent was obtained; the resultant confusion matrix is shown in Table 2.

3.4 Analyzing Classifier Performance

Though our classification result is inferior to the state-of-the-art classification techniques [23], it suggests an inherent relationship between the class of the fingerprint and the underlying minutiae distribution. Most of the misclassifications represent cases where ridges contributing to important pattern characteristics (e.g., recurving ridges) do not have minutiae. This can be seen in Fig. 16, where the minutiae plot does not capture the recurving of ridges in the core region of a right loop resulting in a classification error. Furthermore, the definition of a fingerprint class can be ambiguous in some instances. Even human fingerprint experts cannot uniquely classify certain fingerprints. (About 17 percent of the 4,000 images in the NIST-4 database have been assigned two different ground truths.) The NIST-4 data set also has quite a few low quality and partial fingerprints that do not include the delta region. Fingerprint images, in some cases, may have large intraclass variability (i.e., prints of the same class may have very different characteristics) and small intraclass variability (i.e., prints of different classes look similar), as shown in Fig. 17.

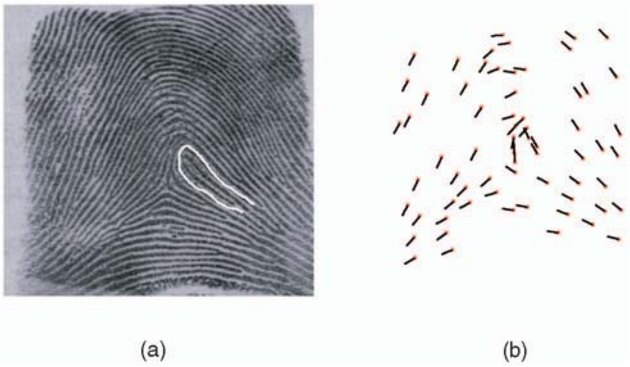

Fig. 16. The minutiae plot of a fingerprint belonging to the $\mathbf{R}$ class. This was misclassified as $\mathbf{A}$ in our experiments. The minutiae plot does not appear to capture the shape of the recurving ridge (marked in (a)).

4 Fingerprint Reconstruction

By the term "reconstruction" of fingerprints, we refer to the regeneration of fingerprints from the stored minutiae points. While various approaches have been proposed for generating synthetic fingerprints [7], [1], to the best of our knowledge, no technique for reconstructing fingerprints with minutiae points at desired locations has been proposed in the literature.

Given the estimated orientation field, we generate the ridge structure associated with it. In our earlier work [27], we attempted to use a Gabor filter-based technique to generate fingerprints from the estimated orientation field. However, such a scheme does not permit control over the location, number, and type of minutiae in the generated image. Here, we propose a novel algorithm for recovering the ridge structure from the minutiae template, which enables us to place minutiae at desired locations.

Streamlines and Line Integral Convolution (LIC) have been widely used for imaging arbitrary two and three-dimensional vector fields [4], [22]. Given a vector field, streamlines are the curves that are tangential to the vector field at every point. These are also called integral curves as they are generated by integrating the vector field. LIC is basically a texture synthesis technique that is used to

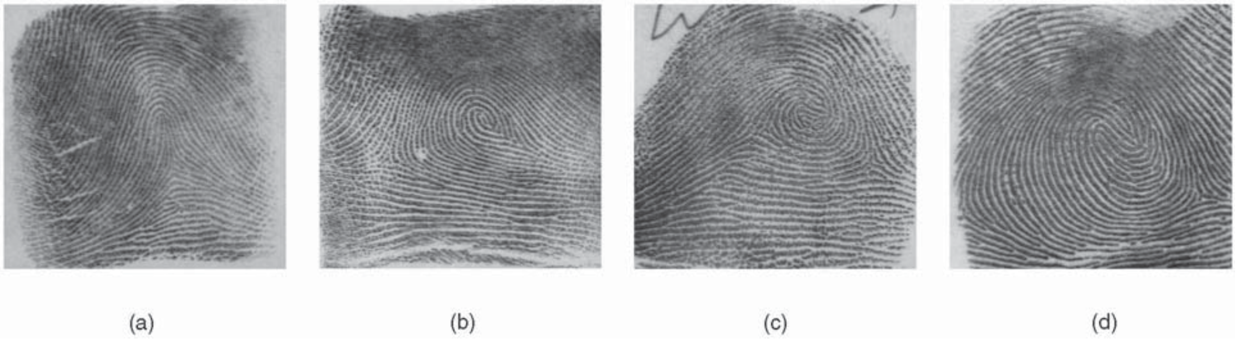

Fig. 17. Small interclass variability between classes (a) $\mathbf{L}$ and $\mathbf{A}$. (b) $\mathbf{R}$ and $\mathbf{W}$. (c) and (d) Large intraclass variability between two prints of class $\mathbf{W}$.

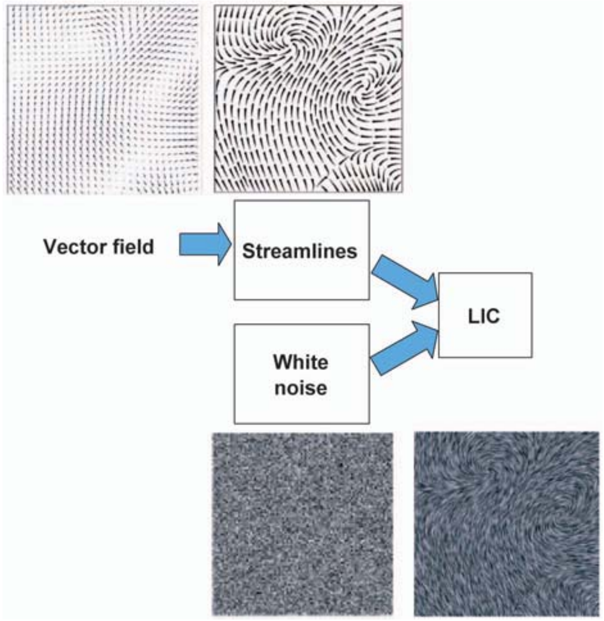

Fig. 18. Visualizing vector field using LIC.

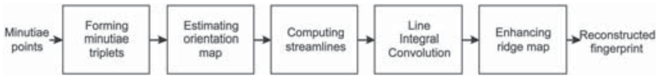
Minutiae points
Forming minutiae triplets
Estimating orientation map
Computing streamlines
Line Integral Convolution
Enhancing ridge map
Reconstructed fingerprint

Fig. 19. Proposed fingerprint reconstruction algorithm.

visualize 2D data. Cabral and Leedom [4] proposed LIC for imaging vector fields and to produce novel special effects such as motion blurring in images. Fig. 18 shows the key steps for visualizing vector fields using streamlines and LIC. For a given vector field, the streamlines are first generated. Next, by using streamlines in conjunction with a spatially uncorrelated white noise image, LIC computes the intensity values for the coordinates of each streamline. It applies a one-dimensional filter to blur the noise image along the streamlines, resulting in a texture image which aids in the clear visualization the vector field. A fingerprint is an oriented texture pattern and, therefore, the use of streamlines and LIC for generating the ridge structure is quite appropriate; streamlines are used to generate the thinned ridges, whereas LIC is used to impart texture-like appearance to the ensuing ridges. The main stages in regenerating fingerprints are shown in Fig. 19.

1. **Estimating orientation map $\hat{\boldsymbol{\theta}}$**. Given a minutiae template, we first obtain the estimated orientation map $\hat{\boldsymbol{\theta}}$ using minutiae triplets described in Section 2.
2. **Constructing streamlines using $\hat{\boldsymbol{\theta}}$**. A streamline is a path in the orientation map $\hat{\boldsymbol{\theta}}$ whose tangent vectors coincide with $\hat{\boldsymbol{\theta}}$ (Fig. 20a).

   Let P be a point in the orientation map $\hat{\boldsymbol{\theta}}$ and S be a streamline passing through P. Let the curvature at point P be due to a circle of radius $r(t)$. Then, the tangent dr at point P is given by,

$$\frac{dr}{dt} = \hat{\boldsymbol{\theta}}[r(t)], \tag{7}$$

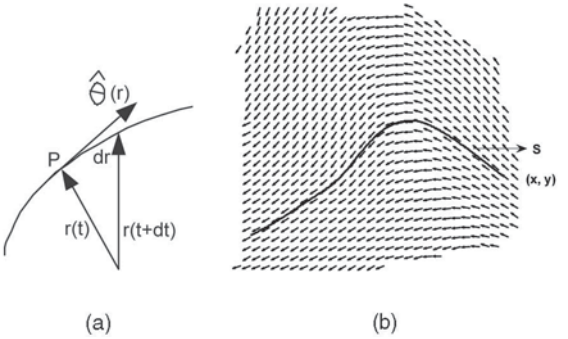

Fig. 20. Constructing streamlines. (a) A streamline S tangent to the orientation field $\hat{\boldsymbol{\theta}}$. (b) Streamline originating from a seed point (x, y) in the orientation field $\hat{\boldsymbol{\theta}}$.

where dt represents the next position. A numerical integration technique or an interpolation technique can be used to solve this ordinary differential equation (ODE). Integrating (7) yields,

$$r(t) = r(0) + \int_0^t \hat{\boldsymbol{\theta}}(r(t))dt. \tag{8}$$

With the help of (8), a streamline can be initiated from a *seed point* and evolved in small steps of size dt. If we solve the above ordinary differential equation for $\hat{\boldsymbol{\theta}}(x, y)$ with the seed point as (x, y), we get a streamline as shown in Fig. 20b. In order to generate streamlines for constructing ridge lines, the following issues have to be addressed: 1) selection of seed points, 2) criteria for streamline termination, 3) distance between adjacent streamlines, and 4) generation of minutiae in predetermined positions.

a. **Seed point selection**. The seed points are the prespecified points in an orientation map $\hat{\boldsymbol{\theta}}$ from which the streamlines originate. Here, the seed point can be considered to be a minutia (ridge ending). Thus, in order to generate minutiae at desired locations, we use the minutiae locations as seed points. However, using only minutiae as seed points results in the sparse distribution of streamlines over the image plane (Fig. 21b). Thus, we also use the boundary points of $\hat{\boldsymbol{\theta}}$ as seed points. This enables us to capture the global shape of the parent fingerprint as shown in Fig. 21c.

   In high minutiae-density areas, the seed points will occur in close proximity, resulting in a clutter of streamlines in local regions. To solve this problem, we place all the seed points, S_i, $i = 1 \cdots k + n$ (where k and n are the number of border points and minutiae, respectively), on a lattice with cell size $D_s \times D_s$ such that two seed points are at least $D_s = 10$ pixels apart from each other, as shown in Fig. 22a. Due to this mapping, a single grid point may correspond to multiple minutiae points. This prevents the excessive proliferation of streamlines in a local region.

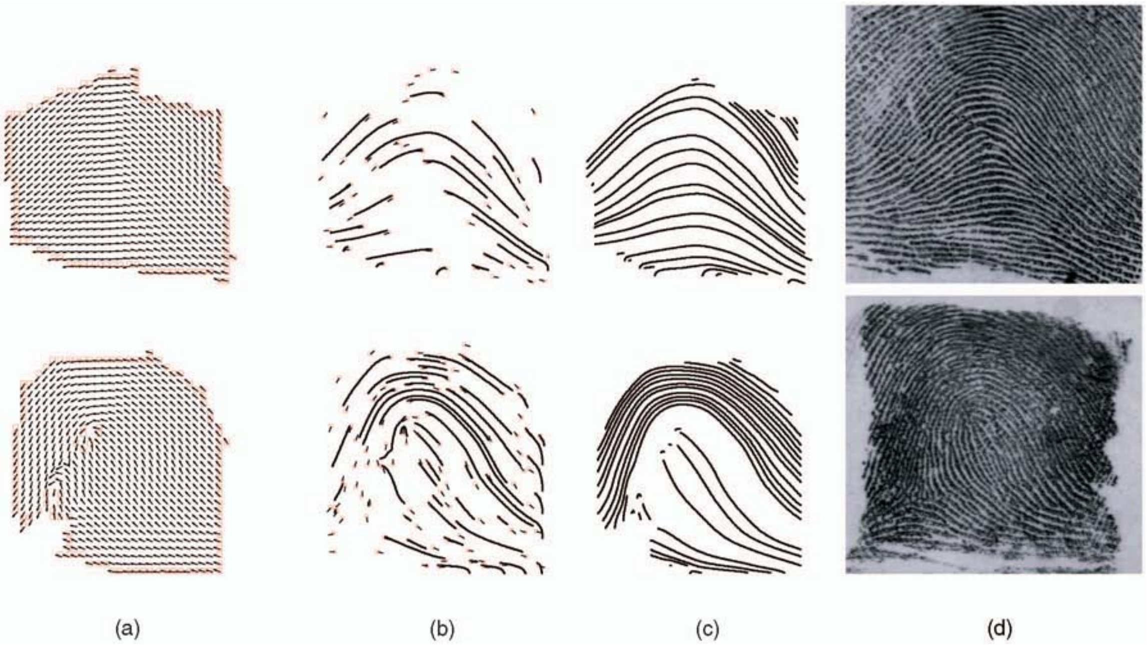

Fig. 21. Streamline generation. (a) Estimated orientation map $\hat{\boldsymbol{\theta}}$ with border points marked in red. (b) Streamlines generated using minutiae points as seed points. (c) Streamlines generated using border points as seed points. (d) Original fingerprint from $\mathbf{A}$ and $\mathbf{R}$ classes.

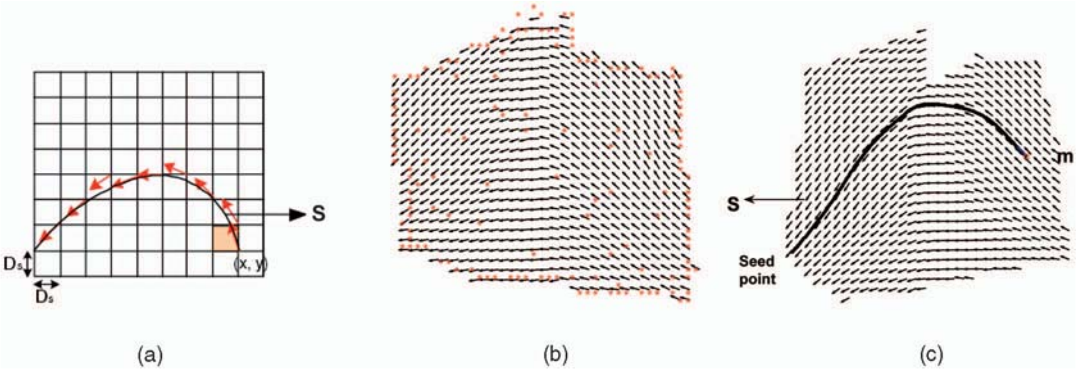

Fig. 22. Streamline construction: (a) Lattice of seed points ($D_{seed} = 10$). (b) Seed points "S" overlaid on the orientation field, $\hat{\boldsymbol{\theta}}$, of an arch. (c) Streamline terminated near a minutia "m."

b. **Streamline construction**. While most particle tracking algorithms (e.g., [29]) use various numerical integration techniques to solve (8), we use a linear interpolation scheme for constructing streamlines.[6] Once a streamline is initiated from a seed point, the orientations of its four immediate neighbors are used to linearly interpolate the next location of the streamline. The streamline is terminated if

   i. it encounters a boundary point in the grid,
   ii. it arrives in the vicinity of a minutia point (when a streamline is within a predetermined distance from a minutia point, i.e., five pixels, as shown in Fig. 22c), and
   iii. if the number of coordinates computed is greater than 10,000.

A minutia (ridge ending) is generated if either a streamline initiates from that point or another streamline appears in its proximity. In the current implementation, we generate only ridge endings at the desired minutiae locations since it is assumed that the "type" of the minutia is not stored in the template.

It has been observed in the literature that a constant density of seed points does not necessarily ensure an even distribution of the streamlines [4]. The generated streamlines may not be evenly distributed, resulting in a frenzy of streamline activity in certain local regions. While several elegant techniques exist for creating evenly spaced streamlines (see [19], for example), we employ a simple technique to control interridge distances (D_r): Ridges are generated with uniform interridge distances over the entire fingerprint image since we assume that the minutiae template does not

[6] The "stream3c" function in Matlab (version 7.0) was used for this purpose.

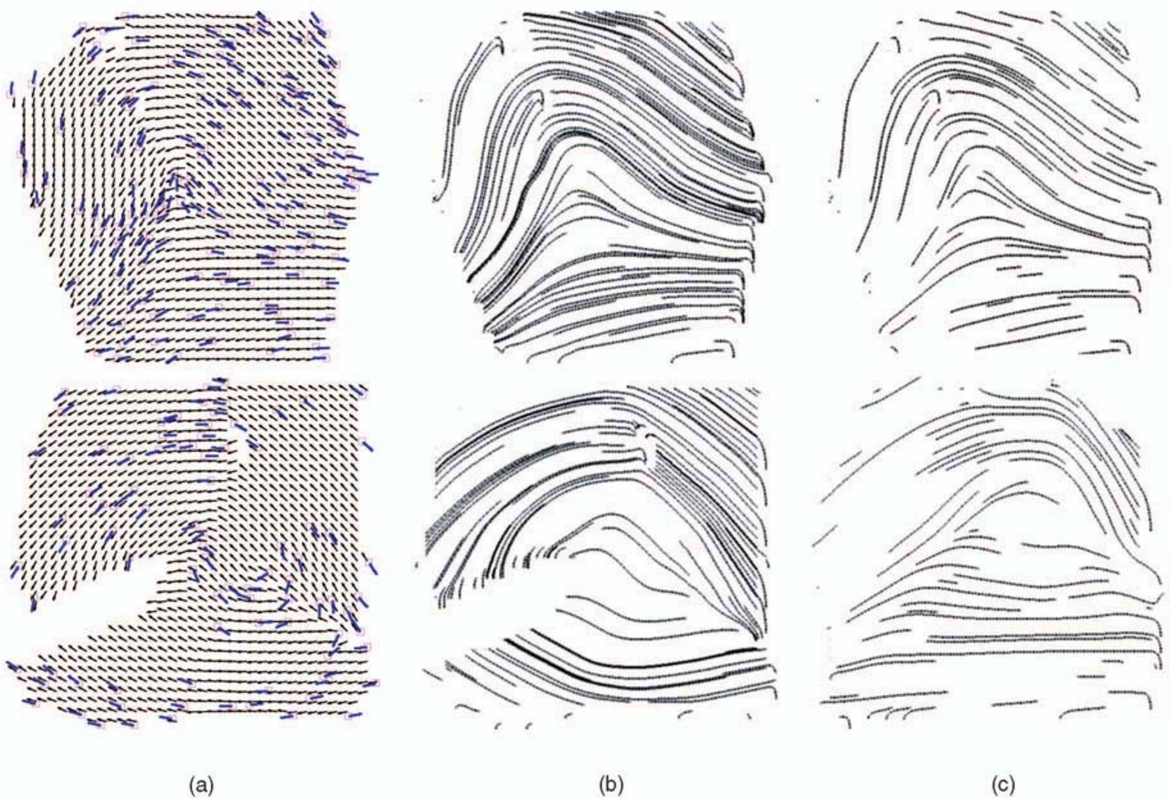

Fig. 23. Streamline generation. (a) Estimated orientation map $\hat{\boldsymbol{\theta}}$. (b) Streamlines generated without controlling D_r. (c) Streamlines generated by setting $D_r = 5$.

store any information about the interridge distance of the parent fingerprint. During streamline construction, if a streamline is in the proximity of another streamline $(D_r = 5)$, then it is discarded. This avoids the cluttering of streamlines in local regions (Fig. 23).

Fig. 24 shows the results of streamline generation for different fingerprint classes.

3. **Generating ridge structure using LIC**. The application of streamlines results in the generation of binarized thin ridge lines for the estimated orientation map $\hat{\boldsymbol{\theta}}$. In order to impart texture-like appearance to the ridges, we use Linear Integral Convolution (LIC). Given a streamline S, the LIC technique involves calculating the intensity of all pixels constituting the streamline. It locally blurs an uncorrelated input texture image, such as white noise, along the path of the streamlines to impart a dense visualization of the flow field. Consider a pixel at location $x_0 = S(p_0)$ on the streamline. Its intensity is computed using one-dimensional filtering as

$$I(x_0) = \int_{p_0-L}^{p_0+L} k(p - p_0)T(S(p))dp, \qquad (9)$$

where T is the texture (white noise) image. The kernel, k, is a one-dimensional low-pass filter with $L = 25$ pixels. The convolution results in the generation of ridge-like patterns whose orientations correspond to the estimated vector/orientation field (Fig. 25). The filter length L determines how much the texture is smeared in the direction of the vector field. Due to the application of LIC, the binary image generated using streamlines is converted to a gray scale image (Fig. 26b).

4. **Enhancing the ridge map**. Although LIC imparts texture-like appearance to the thin ridges (streamlines), the ridges are still one-pixel thick. In order to increase the ridge width, we first use a lowpass filter to smooth the texture image generated using LIC and then perform histogram equalization of the ridge structure for contrast enhancement, as shown in Fig. 26c. The reconstructed fingerprints using minutiae alone are actually partial fingerprints. This is because a minutiae template does not capture all the global properties of the parent fingerprint. We demonstrate that it is indeed possible to reconstruct at least those parts of the fingerprint which are useful for authentication.

Some of the fingerprints in the NIST 4 database are noisy, as shown in Fig. 27a. The minutiae extracted from a poor quality fingerprint image might be erroneous (in terms of their location and orientation). This will affect the orientation estimation procedure and, consequently, the streamlines will not conform to the original fingerprint (Figs. 27b and 27c).

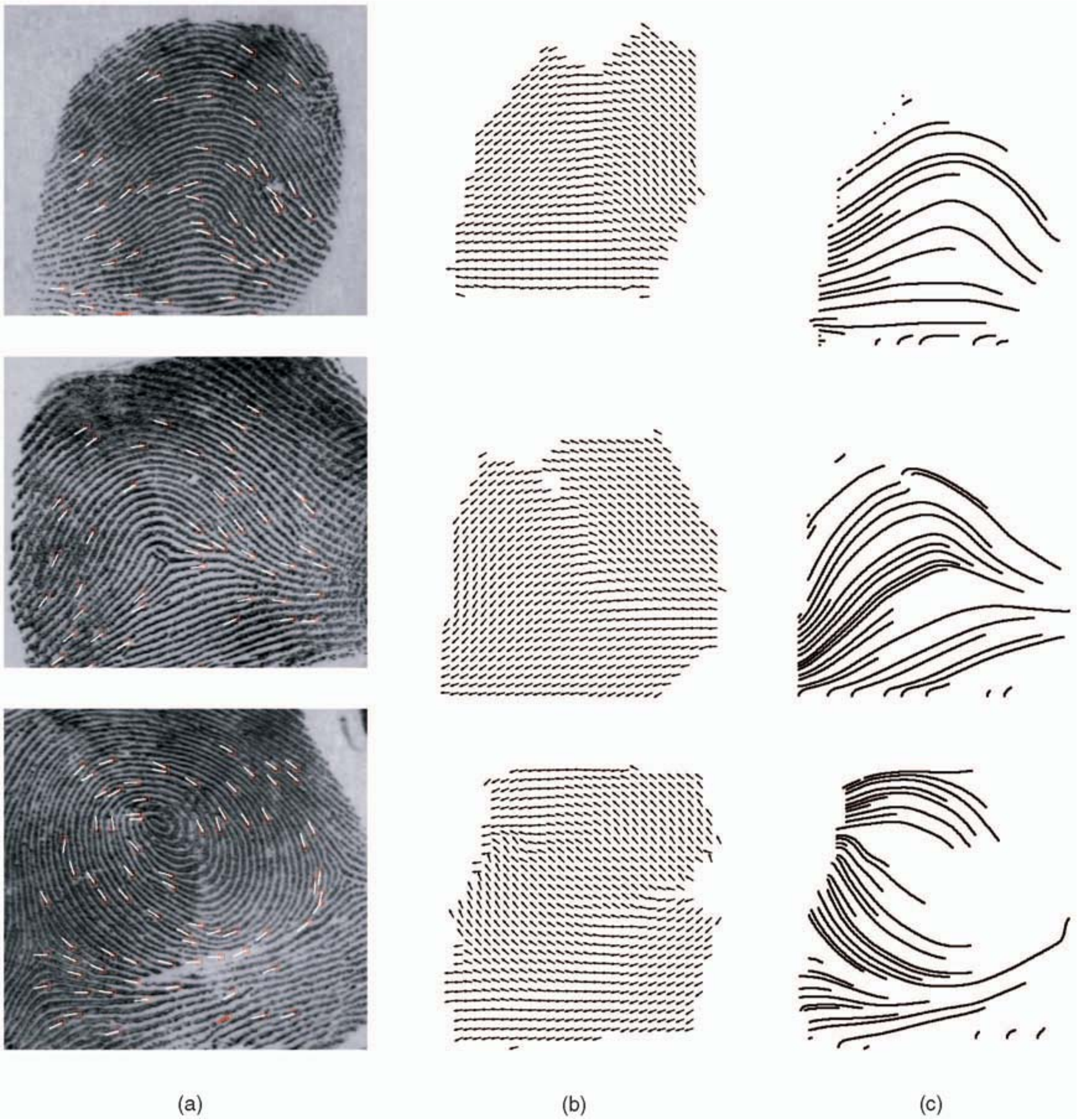

Fig. 24. Streamline generation for three fingerprint classes ($\mathbf{A}$, $\mathbf{L}$, and $\mathbf{W}$): (a) Minutiae plot. (b) Estimated orientation map $\hat{\boldsymbol{\theta}}$. (c) Streamlines.

4.1 Validation of Reconstructed Fingerprints

Fig. 28 shows an overlay of the original and reconstructed fingerprints for two classes of fingerprints: $\mathbf{A}$ and $\mathbf{L}$. Visually, it is evident that the reconstructed fingerprints are consistent with the underlying ridge structures. We conducted experiments to quantitatively determine the similarity between the reconstructed ridge structure and the parent fingerprint.

To test our reconstruction algorithm, we used the NIST-4f database. NIST-4f contains 2,000 fingerprints (500 fingerprints per class) of size (512×512) and one fingerprint per individual. Using the VeriFinger 4.1 SDK developed by Neurotechnologija,[7] we computed the similarity scores generated when matching the reconstructed fingerprint against the respective parent fingerprint. The VeriFinger application first enhances the input fingerprint before matching it against other fingerprints as, shown in Fig. 29.

We considered two matching scenarios. In the first scenario, each reconstructed fingerprint was matched against every parent fingerprint in the database. In the

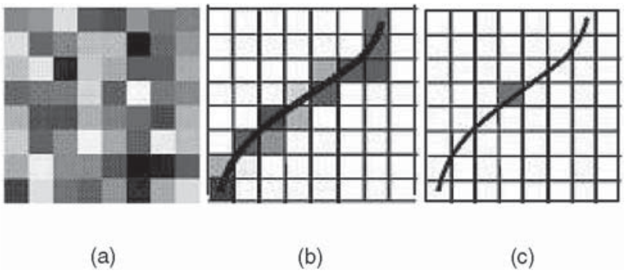

Fig. 25. Texture rendering using LIC. (a) White noise image, T. (b) Convolving T with a streamline. (c) Output of the convolution process.

[7] http://www.neurotechnologija.com.

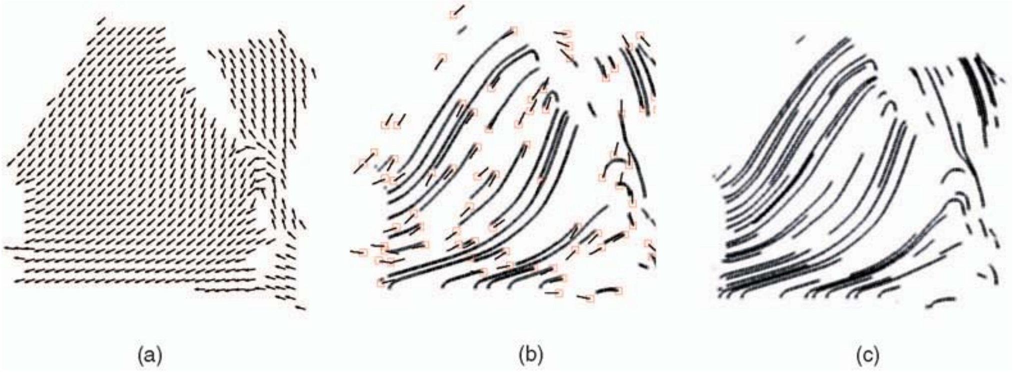

Fig. 26. (a) Estimated orientation map $\hat{\boldsymbol{\theta}}$. (b) Result of applying LIC. (c) Enhanced fingerprint using a 5×5 lowpass filter.

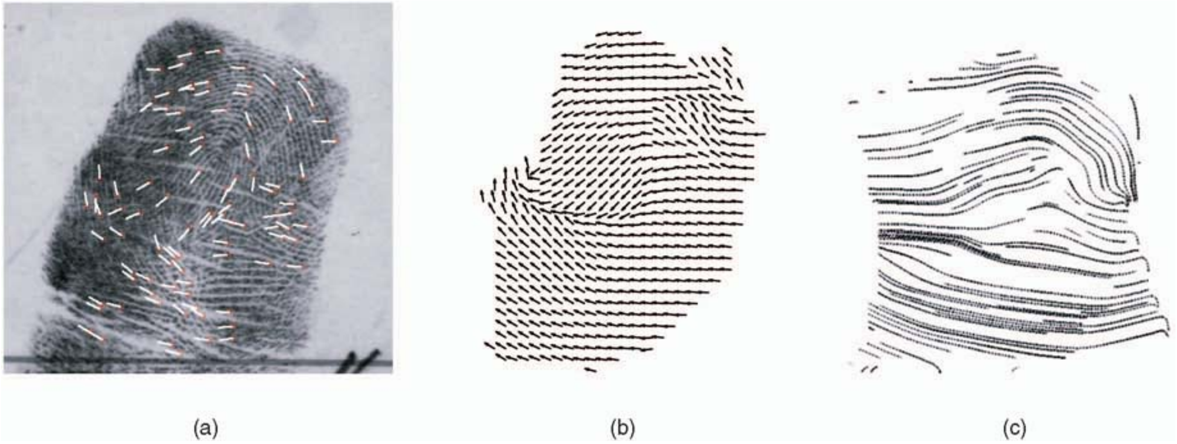

Fig. 27. Reconstruction of a poor quality fingerprint. (a) Original left loop and its minutiae plot. (b) Estimated orientation map $\hat{\boldsymbol{\theta}}$. (c) Reconstructed fingerprint.

second scenario, it is assumed that the class of the parent fingerprint is known, so the reconstructed fingerprint is matched with only parent fingerprints corresponding to this class. For each reconstructed fingerprint, the top matches are recorded and the CMC (Cumulative Match Characteristics) curve is generated to summarize the identification performance. The CMC graph plots the identification rate as a function of the number of top matches (ranks) [25].

Fig. 30 shows the CMC curves corresponding to these experiments. About 23 percent of the regenerated fingerprints are observed to match successfully (at the rank one level) with the parent fingerprint. When the class information is used to narrow the search, then $\sim$ 30 percent of type $\mathbf{L}$ fingerprints and $\sim$ 35 percent of type $\mathbf{A}$ fingerprints correctly match with their respective parent fingerprints. It is observed that fingerprints of type $\mathbf{W}$ result in the lowest identification rate. A fingerprint of class $\mathbf{W}$ is characterized by high ridge activity near the core and delta region. The orientation estimation algorithm cannot accurately capture the ridge orientations near the core or delta region from the minutiae information alone. This results in the incorrect generation of streamlines in such regions. Hence, the associated minutiae points may not match well with those of the original fingerprint, resulting in a lower identification rate. Also, during streamline construction, we do not permit a streamline to be generated too close to an existing streamline. As the interridge distance information is not available in the minutiae template, we use a constant interridge distance over the entire image plane. Thus, very few streamlines are generated in regions with high minutiae density resulting in missing minutiae in the reconstructed fingerprint.

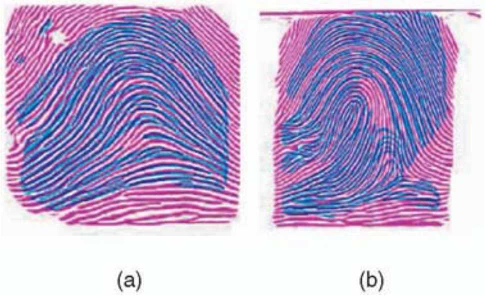

Fig. 28. Overlay of original (magenta) and reconstructed (blue) fingerprints for two minutiae templates.

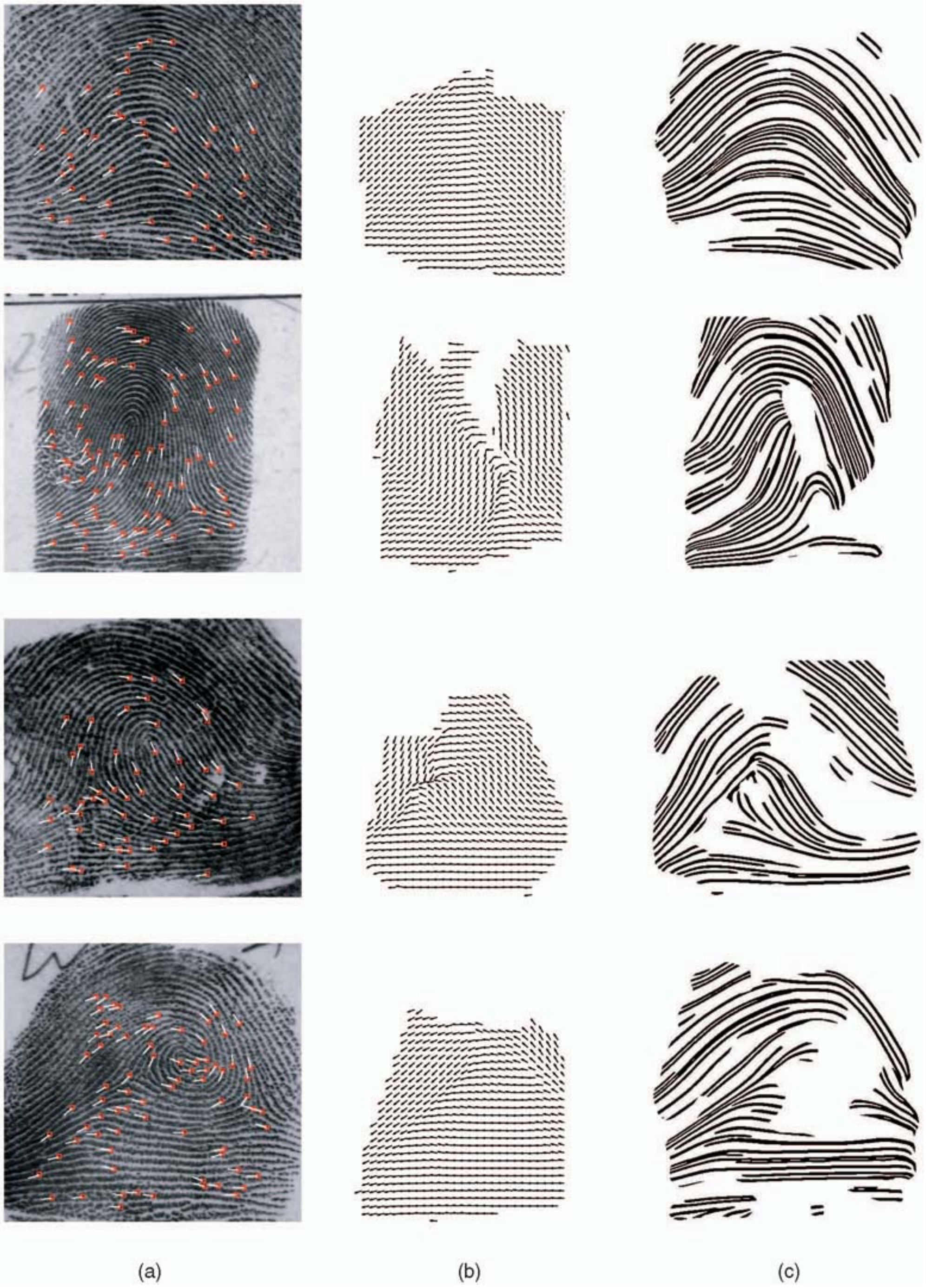

Fig. 29. Reconstructing the ridge structure. (a) Original fingerprint and its minutiae plot. (b) Estimated orientation map $\hat{\boldsymbol{\theta}}$. (c) Enhanced ridge structure after application of the Verifinger software.

5 Summary and Future Work

It has been traditionally assumed that a minutiae template does not reveal substantial information about the parent fingerprint. In this paper, it was demonstrated that three levels of information about the parent fingerprint can be elicited from a given minutiae template: the orientation field, the fingerprint class, and the friction ridge structure. The orientation field is estimated using minutia triplet information. The class of the parent fingerprint is inferred from specific attributes of the minutiae distribution in conjunction with the estimated orientation field. The ridge structure of the parent fingerprint is generated by applying streamlines and LIC on the estimated orientation field. The ridge generation technique controls the location and the number of minutiae in the generated ridge map and, hence, may be used as an alternative scheme for generating synthetic fingerprints with minutiae placed at predetermined locations. The reconstructed ridge structure is observed to be visually similar to that of the parent fingerprint. Furthermore, the reconstructed image may be

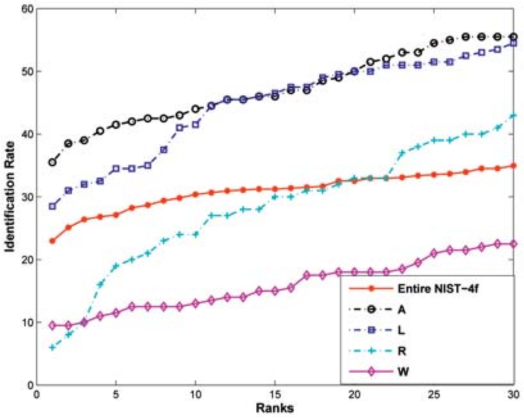

Fig. 30. CMC curves generated by matching the reconstructed fingerprint against all fingerprints of the NIST-4f database. In the plots labeled **A**, **L**, **R**, and **W**, the fingerprint class is assumed to be known, so the reconstructed fingerprint is matched with fingerprints pertaining to the associated class only.

used to generate synthetic prints that can be used to compromise the security of a biometric system. If other information, such as the location of singular points, the class of the fingerprint, the type of minutiae, and interridge attributes, are available in the template, then, perhaps, the original fingerprint can be reconstructed in its *entirety*.

The inferred fingerprint class could be used during the streamline construction process to control the density and occurrence of minutiae across the fingerprint. For example, each fingerprint class could have a different seeding strategy. Verma et al. [33] discuss a seed placement strategy based on flow features in the data set. In their technique, the seeds are placed in the vicinity of critical points in the flow field to capture important flow patterns. Such an approach may be used here to enhance the accuracy of the generated friction ridge structure. Also, more sophisticated numerical integration methods, like the fifth order Runge-Kutta technique, may be used to improve the quality of the streamlines.

The fingerprint reconstruction process can be moderated by employing an iterative hill climbing approach (see Fig. 31). In such a scheme, the reconstructed artifact can be matched against the minutiae template stored in the database via the matching system. Based on the match score released by the system, the fingerprint reconstruction process can be modified so as to improve the match score. We are also currently investigating ways to enhance the visual appearance of the reconstructed fingerprints.

ACKNOWLEDGMENTS

This work was supported in part by an Information Technology Research (ITR) grant (#CNS-0325640) from the US National Science Foundation (NSF). The authors would like to acknowledge useful discussions with Professor Terry Boult, University of Colorado, Colorado Springs. This work was done when J. Shah was at West Virginia University.

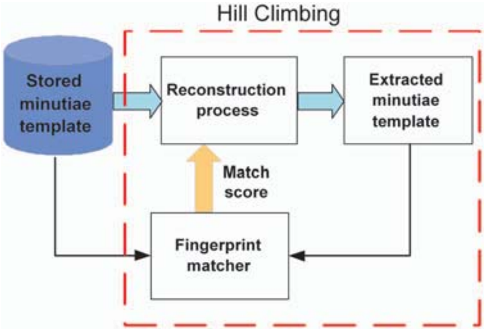

Fig. 31. Reconstructing fingerprints using a hill-climbing approach.

REFERENCES

[1] Optel, Poland, www.optel.com.pl, 2007.

[2] A. Adler, "Can Images be Regenerated from Biometric Templates?" *Proc. Biometrics Consortium Conf.*, Sept. 2003.

[3] D.H. Ballard, "Generalizing the Hough Transform to Detect Arbitrary Shapes," *Pattern Recognition,* vol. 13, no. 3, pp. 111-112, 1997.

[4] B. Cabral and L. Leedom, "Imaging Vector Fields Using Line Integral Convolution," *Proc. 20th Ann. Conf. Computer Graphics and Interactive Techniques,* pp. 263-270, 1993.

[5] G. Candela, P. Grother, C. Watson, R. Wilkinson, and C. Wilson, "PCASYS—A Pattern-Level Classification Automation System for Fingerprints," Technical Report NIST TR 5647, Aug. 1995.

[6] R. Cappelli, A. Lumini, D. Maio, and D. Maltoni, "Fingerprint Classification by Directional Image Partitioning," *IEEE Trans. Pattern Analysis and Machine Intelligence,* vol. 21, pp. 402-421, 1999.

[7] R. Cappelli, D. Maio, and D. Maltoni, "Synthetic Fingerprint Image Generation," *Proc. 15th Int'l Conf. Advances in Pattern Recognition,* vol. 3, no. 5, pp. 475-478, 2000.

[8] C. Champod and P. Margot, "Computer Assisted Analysis of Minutiae Occurrences on Fingerprints," *Proc. Int'l Symp. Fingerprint Detection and Identification,* p. 305, 1996.

[9] B. Cho, J. Kim, I. Bae, I. Bae, and K. Yoo, "Core-Based Fingerprint Image Classification," *Proc. 15th Int'l Conf. Pattern Recognition,* vol. 2, pp. 863-866, 2000.

[10] M. Chong, T. Ngee, L. Jun, and R. Gay, "Geometric Framework for Fingerprint Image Classification," *Pattern Recognition,* vol. 30, no. 9, pp. 1475-1488, 1997.

[11] F. Galton, *Finger Prints.* McMillan, 1892.

[12] I.B. Group, "Generating Images from Templates," technical report, I.B.G. white paper, 2002.

[13] E. Henry, *Classification and Uses of Fingerprints.* Routledge, 1900.

[14] C. Hill, "Risk of Masquerade Arising from the Storage of Biometrics," master's thesis, Australian Nat'l Univ., 2001.

[15] L. Hong and A. Jain, "Classification of Fingerprint Images," *Proc. 11th Scandinavian Conf. Image Analysis*, June 1999.

[16] L. Hong, Y. Wan, and A.K. Jain, "Fingerprint Image Enhancement: Algorithm and Performance Evaluation," *IEEE Trans. Pattern Analysis and Machine Intelligence,* vol. 20, no. 8, pp. 777-789, Aug. 1998.

[17] A. Jain, S. Prabhakar, and L. Hong, "A Multichannel Approach to Fingerprint Classification," *IEEE Trans. Pattern Analysis and Machine Intelligence,* vol. 21, pp. 348-359, 1999.

[18] A.K. Jain and S. Minut, "Hierarchical Kernel Fitting for Fingerprint Classification and Alignment," *Proc. 16th Int'l Conf. Pattern Recognition,* vol. 2, pp. 469-473, Aug. 2002.

[19] B. Jobard and R. Wilfrid, "Creating Evenly-Spaced Streamlines of Arbitrary Density," *Proc. Eurographics Workshop Visualization in Scientific Computing,* 1997.

[20] K. Karu and A. Jain, "Fingerprint Classification," *Pattern Recognition,* vol. 29, pp. 389-404, 1996.

[21] C. Kingston, "Probabilistic Analysis of Partial Fingerprint Patterns," PhD thesis, Univ. of California, Berkeley, 1964.

[22] D. Laidlaw, R. Kirby, C. Jackson, J. Davidson, T. Miller, M. Silva, W. Warren, and M. Tarr, "Comparing 2D Vector Field Visualization Methods: A User Study," *IEEE Trans. Visualization and Computer Graphics,* vol. 11, no. 1, pp. 59-70, Jan./Feb. 2005.

[23] D. Maltoni, D. Maio, A.K. Jain, and S. Prabhakar, *Handbook of Fingerprint Recognition,* first ed. Springer-Verlag, 2003.

[24] A. Moenssens, *Fingerprint Techniques.* Chilton Book Company, 1971.

[25] H. Moon and P. Phillips, "Computational and Performance Aspects of PCA-Based Face-Recognition Algorithms," *Perception,* vol. 30, pp. 303-321, 2001.

[26] N. Ratha, S. Chen, and A. Jain, "Adaptive Flow Orientation-Based Feature Extraction in Fingerprint Images," *Pattern Recognition,* vol. 28, no. 11, pp. 1657-1672, 1995.

[27] A. Ross, J. Shah, and A. Jain, "Towards Reconstructing Fingerprints from Minutiae Points," *Proc. SPIE Conf. Biometric Technology for Human Identification II,* pp. 68-80, 2005.

[28] T. Roxburgh, "On the Evidential Value of Fingerprints," *Sankhya: Indian J. Statistics,* pp. 189-214, 1993.

[29] A. Sadarjoen, T. Walsum, A. Hin, and F. Post, "Particle Tracing Algorithms for 3D Curvilinear Grids," *Proc. Fifth Eurographics Workshop Visualization and Scientific Computing,* 1994.

[30] A. Senior, "A Combination Fingerprint Classifier," *IEEE Trans. Pattern Analysis and Machine Intelligence,* vol. 23, no. 10, pp. 1165-1174, Oct. 2001.

[31] B. Sherlock and D. Monro, "A Model for Interpreting Fingerprint Topology," *Pattern Recognition,* vol. 26, no. 7, pp. 1047-1055, 1993.

[32] D.A. Stoney, "Quantitative Assessment of Fingerprint Individuality," PhD thesis, Univ. of California, Davis, 1985.

[33] V. Verma, D. Kao, and A. Pang, "A Flow-Guided Streamline Seeding Strategy," *Proc. Conf. Visualization,* pp. 163-170, 2000.

[34] C. Wilson, G. Candela, and C. Watson, "Neural Network Fingerprint Classification," *J. Artificial Neural Networks,* vol. 1, no. 2, pp. 203-228, 1994.

[35] Y. Yao, P. Frasconi, and M. Pontil, "Fingerprint Classification with Combination of Support Vector Machines," *Proc. Third Int'l Conf. Audio- and Video-Based Biometric Person Authentication,* pp. 253-258, 2001.

Arun Ross received the BE (Hons.) degree in computer science from the Birla Institute of Technology and Science, Pilani, India, in 1996 and the MS and PhD degrees in computer science and engineering from Michigan State University, East Lansing, in 1999 and 2003, respectively. Between 1996 and 1997, he was with the Design and Development Group of Tata Elxsi (India) Ltd., Bangalore, India. He also spent three summers (2000-2002) with the Imaging and Visualization Group at Siemens Corporate Research, Inc., Princeton, New Jersey, working on fingerprint recognition algorithms. He is currently an assistant professor in the Lane Department of Computer Science and Electrical Engineering, West Virginia University, Morgantown. His research interests include pattern recognition, classifier combination, machine learning, computer vision, and biometrics. He is actively involved in the development of pattern recognition and biometrics curricula at West Virginia University. He has authored several journal and conference articles in biometrics and has presented his research work at various international meetings. He was an invited speaker at the Frontiers of Science Symposium organized by the US National Academy of Sciences in November 2006. He also served as one of the US experts in multibiometrics at the ISO/IEC JTC1 SC37 meeting for biometric standards in 2004. Dr. Ross is the coauthor of the book *Handbook of Multibiometrics* (Springer, 2006). He is a member of the IEEE and the IEEE Computer Society.

Jidnya Shah received the BE degree in electronics engineering from the Vishwakarma Institute of Technology, Pune, India, in 2002. After graduation, she worked with Axis Technologies, India, as a software engineer (2002-2003). She received the MS degree in electrical engineering from West Virginia University in 2005. She worked on fingerprint enhancement in her internship at Siemens Corporate Research, New Jersey, between September and December 2005. She is now working as a research scientist at L-1 Identity Solutions Corporate Research Center developing face recognition algorithms. Her research interests include image processing, computer vision, and biometrics.

Anil K. Jain received the BTech degree from the Indian Institute of Technology, Kanpur, and the MS and PhD degrees from The Ohio State University. He is a University Distinguished Professor in the Department of Computer Science and Engineering at Michigan State University. He received a distinguished alumni award from The Ohio State University. His research interests include statistical pattern recognition, data clustering, and biometric authentication. He received the 1996 *IEEE Transactions on Neural Networks* Outstanding Paper Award and best paper awards from the Pattern Recognition Society in 1987 and 1991. He was the editor-in-chief of the *IEEE Transactions on Pattern Analysis and Machine Intelligence*. He is a fellow of the IEEE, ACM, IAPR, SPIE, and AAAS. He has received a Fulbright Research Award, a Guggenheim fellowship, the Alexander von Humboldt Research Award, and the 2003 IEEE Computer Society Technical Achievement Award. He is the holder of six patents in the area of fingerprint matching, and is the author of a number of books, including the *Handbook of Multibiometrics* (Springer, 2006), *Biometric Systems, Technology, Design and Performance Evaluation* (Springer, 2005), the *Handbook of Face Recognition* (Springer, 2005), the *Handbook of Fingerprint Recognition* (Springer, 2003) (he received the PSP award from the Association of American Publishers), *BIOMETRICS: Personal Identification in Networked Society* (Kluwer, 1999), and *Algorithms for Clustering Data* (Prentice-Hall, 1988). ISI has designated him as a highly cited researcher. He is an associate editor of the *IEEE Transactions on Information Forensics and Security* and *ACM Transactions on Knowledge Discovery in Data*. He is a member of the National Academy panels on whither biometrics and improvised explosive device.