MITSUBISHI ELECTRIC RESEARCH LABORATORIES
http://www.merl.com

Alignment and Bit Extraction for Secure Fingerprint Biometrics

A. Nagar, S. Rane, A. Vetro

TR2010-003 February 2010

Abstract

Security of biometric templates stored in a system is important because a stolen template can compromise system security as well as user privacy. Therefore, a number of secure biometrics schemes have been proposed that facilitate matching of feature templates without the need for a stored biometric sample. However, most of these schemes suffer from poor matching performance owing to the difficulty of designing biometric feature that remain robust over repeated biometric measurements. This paper describes a scheme to extract binary features from fingerprints using minutia points and fingerprint ridges. The features are amenable to direct matching based on binary Hamming distance, but are especially suitable for use in secure biometric cryptosystems that use standard error correcting codes. Given all binary features, a method for retaining only the most discriminable features is presented which improves the Genuine Accept Rate (GAR) from 82% to 90% at a False Accept Rate (FAR) of 0.1% on a well-known public database. Additionally, incorporating singular points such as a core or delta feature is shown to improve the matching tradeoff.

Proceedings of the SPIE Workshop on Electronic Imaging, Media Forensics

Alignment and Bit Extraction for Secure Fingerprint Biometrics

A. Nagar*, S. Rane† and A. Vetro†

*Michigan State University, East Lansing, MI, USA
†Mitsubishi Electric Research Laboratories, Cambridge, MA, USA

ABSTRACT

Security of biometric templates stored in a system is important because a stolen template can compromise system security as well as user privacy. Therefore, a number of secure biometrics schemes have been proposed that facilitate matching of feature templates without the need for a stored biometric sample. However, most of these schemes suffer from poor matching performance owing to the difficulty of designing biometric features that remain robust over repeated biometric measurements. This paper describes a scheme to extract binary features from fingerprints using minutia points and fingerprint ridges. The features are amenable to direct matching based on binary Hamming distance, but are especially suitable for use in secure biometric cryptosystems that use standard error correcting codes. Given all binary features, a method for retaining only the most discriminable features is presented which improves the Genuine Accept Rate (GAR) from 82% to 90% at a False Accept Rate (FAR) of 0.1% on a well-known public database. Additionally, incorporating singular points such as a core or delta feature is shown to improve the matching tradeoff.

Keywords: Secure biometrics, fingerprints, secure alignment

1. INTRODUCTION

Biometric recognition is a preferred mode of personal authentication today, mainly due to the inalienable and distinctive nature of biometric traits such as fingerprints, faces, and irises. As a result, biometric sensors are being deployed on a large scale in a wide range of applications from door locks to border security. In a typical biometric recognition system, a user is first enrolled by capturing his biometric trait and storing a template extracted from it. During authentication, the user - or an imposter - provides to the system a probe biometric which is matched with the corresponding template in the database to confirm the user's identity. Such a system is susceptible to a number of attacks, among which template theft is the most crucial and unique to biometric systems (See Jain *et al.*[1] for a list of attacks on biometric systems). It has been shown that biometric templates can be effectively used to create spoof biometrics[2] that are easily accepted by the current systems. This further emphasizes the need for the security of stored biometric templates. In addition to creating spoof biometrics, unsecured templates can also be used to link records corresponding to the same individual in different databases. The feasibility of such linkage entails a significant threat to personal privacy as hackers can stage such linkage attacks and accumulate significant information about the users. Also, biometric templates tend to reveal private information about a user such as race, gender and certain health conditions.[3] In order to mitigate such concerns, a number of techniques are being proposed to limit the the amount of information that can be easily extracted from a stored template. Clearly, this makes the very task of matching biometric templates more difficult, and the design of a secure biometric system involves tradeoffs between the matching performance and template security.

Template protection techniques can be broadly categorized into feature transformation techniques and biometric cryptosystems. In feature transformation techniques, the features extracted from a biometric trait are transformed using a transformation function indexed by a user-specific password. This allows cancelability in the system; a user can revoke a template that has been compromised and generate a new template which cannot be easily guessed using the compromised template. The security of such techniques depends heavily on the transformation function and it is difficult to provide provable guarantees on the security of cancelable transforms.

Work performed during A. Nagar's internship at MERL. Send correspondence to S. Rane: rane@merl.com

In a biometric cryptosystem, on the other hand, a secret key is associated with the biometric, and a secure template is derived such that the template does not reveal much information about the enrolled biometric or the key. Authentication is performed by presenting a probe biometric which is used to recover the enrolled biometric as well as the key used. Error correcting codes (ECC), which have been deeply studied in the context of reliable communication, are normally used to achieve this functionality. In this case, provable security guarantees can be provided based on the coding rate of the ECC. However, systems based on this approach have traditionally suffered from poor matching performance. In the case of fingerprint biometrics, this is attributed to the fact the ECC, often an algebraic or a graph-based code is designed for a simple channel model while the dependency between multiple fingerprint measurements tends to be very complicated. Misalignment of fingerprints, dirt on the finger or sensor, difference in pressure applied to the sensor, injury to the finger are some of the main reasons why the "channel" between an enrolment fingerprint and a probe fingerprint is very complicated. This realization drives our approach in this paper: We are concerned with the extraction of features that are well-matched to existing ECCs. This entails the transforming fingerprint images into a (usually binary) feature vector such that, after the transformation, the channel between the enrolment and probe biometrics is reduced to a simple dependency, such as a binary symmetric channel.[4]

Fingerprints are the most popular biometrics and thus are of particular interest. A fingerprint essentially consists of a ridge flow pattern and is most commonly represented as a set of points called minutiae, which are the coordinates of the endings and bifurcations of the ridge lines. Although minutiae are a compact and highly salient representation of a fingerprint and a number of high performance algorithms are available to match two fingerprints based on minutiae, such a representation is not very desirable when the templates are required to be protected especially using biometric cryptosystems, for the reasons mentioned above. While ECC for unordered sets of minutia points have been designed,[5] the cryptosystems have limitations: The decoding process of the underlying ECC is either very complex[6] or has lower error correcting capacity.[5] It is also difficult to combine such a representation with other biometric cryptosystems. In principle, a fingerprint fuzzy vault can be combined with other vector-based biometrics[7] such as iris, but the security of the combined system is not optimal. Due to such limitations of set-of-points based representation of fingerprints, it is desirable to represent fingerprints as a binary vector. Binary features are desirable since a number of error correcting codes are available for binary vectors. Moreover, binary features are efficient in terms of matching and storage requirements. A number of techniques are thus being designed in order to convert minutiae into a vector-based representation.[4,8–10]

In Farooq *et al.*,[8] a histogram of minutiae triplets is binarized and used to represent a fingerprint. In Sutcu *et al.*,[4] minutiae features extracted from local regions are used to obtain a binary representation. In Gyaourova and Ross,[9] match scores with respect to a particular matcher, with a set of fixed prototype fingerprints are used as elements of a vector. In this paper, we extend the technique proposed in Sutcu *et al.*[4] by including a richer set of minutia-based features as well as additional features related to ridge orientation and ridge wavelength in order to improve the matching performance of the system.

2. RELATED WORK

Securing fingerprint templates using a ECC-based technique was first proposed in Draper *et al.*[11] In this biometric cryptosystem, the fingerprint is tessellated into a fine regular grid and each grid element is represented as a 0 or a 1 depending on whether the element contains a minutia or not. A syndrome of this binary vector, based on a Low Density Parity Check (LDPC) code, is computed and is stored as the template during enrollment. During authentication, the query fingerprint is also similarly binarized and then the stored syndrome is used to recover the enrolled binary pattern. Figure 1 shows the schematic diagram of a syndrome-based secure fingerprint recognition system.

This technique, however, has high false reject rate due to its lower tolerance to noise in minutiae positions. Further, the matching performance was sensitive to the number of minutia in a fingerprint. To improve the performance within the syndrome coding framework, Sutcu *et al.*[4] proposed a new feature extraction technique based on random selection of cuboidal regions in the (x, y, θ) space associated with minutiae. First, a number of cuboidal regions are randomly generated, and then filtered by eliminating cuboids having significant overlap with other cuboids. Each cuboid is then associated with a value corresponding to the number of minutiae present in that cuboid. Thus, each fingerprint is represented as a vector whose elements correspond to the different

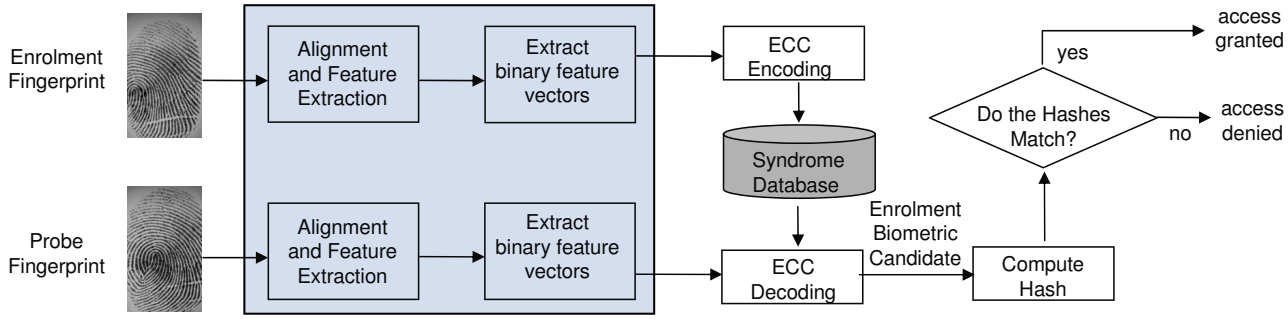


Figure 1. Schematic diagram for a syndrome-based secure fingerprint recognition system.

cuboids. This feature vector is then binarized and some correlated bits are removed to obtain the final set of binary features. In this paper, we proceed to significantly augment the cuboid-based feature extraction algorithm to include a much richer set of fingerprint features and present a method to select only the most discriminable features for matching. In addition to improving the inherent matching performance associated with the feature vectors, this has an additional advantage when applied within a syndrome-based framework: The feature vector is now much longer, and thus admits a longer code which can perform closer to channel capacity than the shorter codes used in earlier works. A direct consequence of this is that for a given matching tradeoff, the syndrome - which is the secure biometric - contains a smaller fraction of the bits contained in the original feature vector. The attacker's task, which is to guess the remaining bits, is thus made more difficult.

3. LOCAL AGGREGATE REGION SELECTION

As proposed by Sutcu *et al.*,[4] a number of local regions are identified in a fingerprint image in order to locally extract various fingerprint features. For minutiae-based features, 3D rectilinear regions or cuboids are obtained with sides along x, y, and θ directions. For ridge-based features 2D regions are obtained with sides along x and y directions. Figure 2 shows examples of 3D and 2D regions. A 2D rectangular region, say R_{2D}, can be represented by a set of four values $[x_{min} \quad x_{max} \quad y_{min} \quad y_{max}]$ such that

$$(x, y) \in R_{2D} \quad \text{if} \quad (x_{min} \leq x \leq x_{max}) \quad \& \quad (y_{min} \leq y \leq y_{max}). \tag{1}$$

Similarly, a 3D region, say R_{3D}, is represented by a set of six values $[x_{min} \quad x_{max} \quad y_{min} \quad y_{max} \quad \theta_{min} \quad \theta_{max}]$ such that

$$(x, y, \theta) \in R_{3D} \quad \text{if} \quad (x_{min} \leq x \leq x_{max}) \quad \& \quad (y_{min} \leq y \leq y_{max}) \quad \& \quad (\theta_{min} \leq \theta \leq \theta_{max}), \tag{2}$$

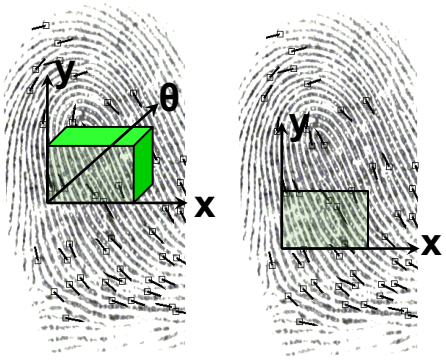
where $0 < x_{min}, x_{max} \leq w$, $0 < y_{min}, y_{max} \leq h$, and $0 < \theta_{min}, \theta_{max} \leq 360^o$, and w, h are the randomly chosen lengths of the cuboids in the x and y directions respectively.

We refer to both the 3D as well as 2D local regions for feature extraction as local aggregate regions. Since it is not desirable to have a small local aggregate region in a corner of the fingerprint image, we require the regions to be centered at the most likely minutia configurations. For this, we center all the fingerprints in a database as follows: Suppose that the i^{th} fingerprint $I_i(x, y)$ has n_i minutiae given by $\{m^1, m^2, ..., m^{n_i}\}$ and each minutiae is represented as $m^j = (x_i^j, y_i^j, \theta_i^j)$. The centered fingerprint $I_{ic}(x, y)$ is given by

$$I_{ic}(x, y) = I_i(x + \delta_{ix}, y + \delta_{iy}) \tag{3}$$

where $\delta_{ix} = (\max_j x_i^j + \min_j x_i^j)/2 - x_c$ and $\delta_{iy} = (\max_j y_i^j + \min_j y_i^j)/2 - y_c$, and (x_c, y_c) is the center of the fingerprint image.

We then make the center of each randomly generated local aggregate region to be the configuration of a randomly selected minutia among the centered fingerprints. Further it is required that the regions obtained do not have volume smaller than 5% of the volume corresponding to the maximum range in a fingerprint both for 2D as well as 3D regions. To ensure this, we retain only those cuboids whose randomly chosen dimensions result in a volume that is large enough.



(a) (b)

Figure 2. Example of a randomly chosen local aggregate regions in (a) 3D for minutia-based features and (b) 2D for ridge-based features.

4. FINGERPRINT ALIGNMENT

In order to extract similar features from two different impressions from the same finger, they should be appropriately aligned before feature extraction. Two fingerprints can be best aligned using the minutiae correspondences. However, minutiae are not explicitly stored in the system database in order to ensure template security. In this paper, we use the secure alignment technique proposed by Uludag and Jain[12] which does not require storage of minutiae. In this technique, high curvature points are extracted from a fingerprint and are matched using a trimmed iterative closest point (ICP) matching algorithm. Obtaining helper data for alignment in the absence of the original fingerprint involves four steps:[6] (1) Orientation field estimation, (2) extraction of flow curves, (3) determination of maximum curvature points, (4) clustering of high curvature points. The orientation field is the orientation of ridge lines at each point in a fingerprint and is estimated using Markov random fields.[13] Flow curves are a set of curved lines which follow the orientation field at every point. Flow curves are obtained by tracing lines along the orientation values.[14] Local maxima along the ridge flow curves are identified and the values that are larger than certain threshold are included in the set of high curvature points. Further, since more than one cluster of high curvature points can be present in a fingerprint due to multiple singularities, such clusters are identified using single link clustering. Figure 3 shows the various stages of high curvature point extraction. In case the fingerprints being aligned have more than one set of high curvature points, each set of high curvature points in one fingerprint are matched to each set of high curvature points in the other fingerprint. A trimmed iterative closest point (TrICP) algorithm[15] is used to align two sets of high curvature points. This algorithm provides some correspondences among the two sets of points which are used to obtain the point set transformation function.

The technique for high curvature point estimation has a number of limitations. In many fingerprints, due to non-linear deformation in fingerprints, the high curvature points are not reliably obtained in certain images and this leads to a slight misalignment. Also, in many images, it is not possible to extract high curvature points. In order to further improve the alignment between fingerprints, we investigate the use of singular points such as fingerprint core or delta in addition to the high curvature points. In order to achieve this, we implement the following steps:

1. Obtain correspondence between high curvature points and the transformation function using the TrICP algorithm
2. Align the singular points using the transformation function
3. Given a singular point in the enrolled fingerprint, check if there is any core point of same type in the aligned query fingerprint at distance less than certain threshold, say th_{SP}. If yes, consider the two singular points as having a correspondence.

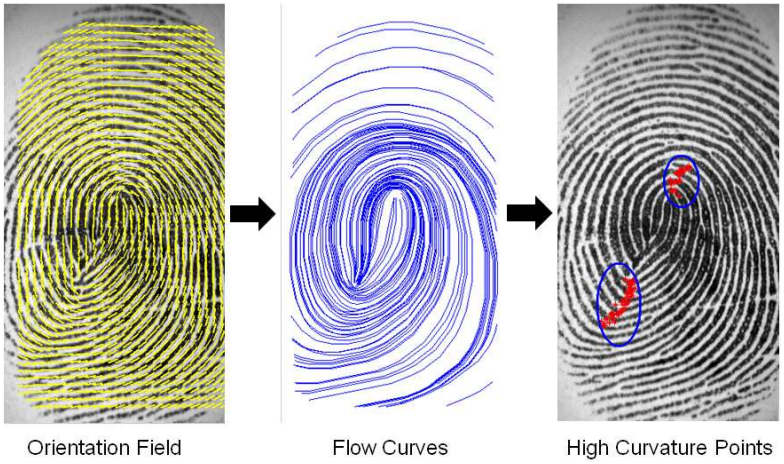
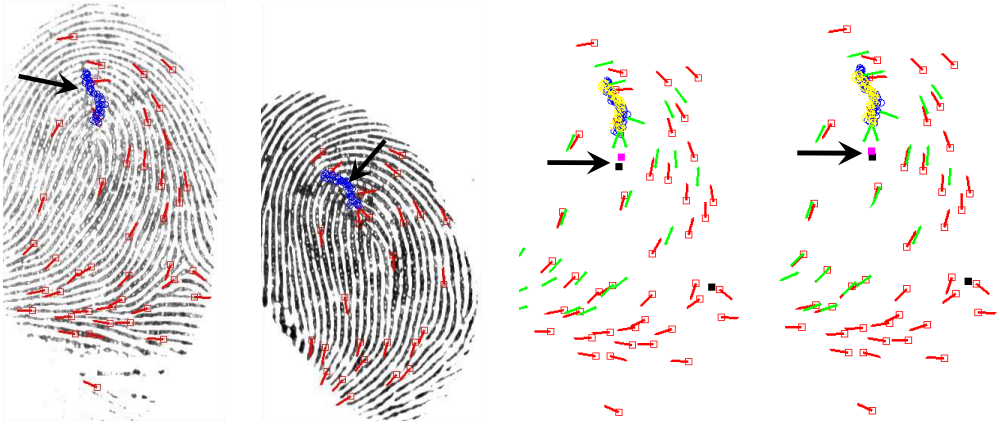


Figure 3. Extraction of high curvature points from a fingerprint. These high curvature points are stored on the access control device to facilitate secure alignment in the absence of the enrolled biometric sample.



(a) (b) (c) (d)

Figure 4. Aligned minutiae and high curvature points from two impressions of the same finger. (a) First impression with minutiae and high curvature points overlaid. Arrow points to the high curvature points (b) Second impression with minutiae and high curvature points overlaid. Arrow points to the high curvature points. (c) Aligned minutiae and high curvature points when singular points are not considered. Arrow points to the singular points. Note that the singular points (squares) from the two impressions are not aligned, and (d) Aligned minutiae and high curvature points when singular points are considered. Arrow points to the singular points. Note that the singular points (squares) from the two impressions are nearly aligned.

4. If there are n_{SP} corresponding high curvature points, augment this set by including the corresponding singular points. However each corresponding singular point is included n_{SP} times in order to give more importance to the singular points in the further processing.
5. Obtain the transformation function using the augmented set of correspondences.

Figure 4 shows two different impressions of the same finger that are aligned with and without considering singular points. In case singular points are used for alignment, the minutiae are more closely aligned as compared to the case when no singular points are used.

5. FINGERPRINT FEATURE EXTRACTION

In this paper, we extract features related to the three main sources of information in a fingerprint, i.e., minutiae, ridge orientation map and ridge frequency. Note that these three features have sufficient information in order to reconstruct a fingerprint, as shown by Cappelli *et al.*[16] and Feng and Jain.[2]

(a) (b)

Figure 5. Two fingerprint impressions taken from the same finger. The extracted minutiae points are superimposed to show the effect of minutia insertion, deletion and movement.

5.1 Minutiae Features

Minutiae are considered as the most discriminative information in a fingerprint. For instance, in the Fingerprint Verification Competition (FVC) 2004,[17] the second best fingerprint matching algorithm in the "light" category used only minutiae and singular points as the fingerprint features. Figure 5 shows minutiae extracted from two different impressions obtained from the same finger. The two sets of minutiae are not exactly the same as there are missing or spurious minutiae. Further, there is noise in the fingerprint image and non-linear deformation, i.e., one set of minutiae is not obtained by simple translation or rotation of the minutiae in the other impression.

To extract minutia-based features, a set of minutiae aggregate regions in the form of cuboids oriented along x, y, and θ directions are obtained as described in section 3. Then, corresponding to each randomly chosen cuboid, three features are extracted: (1) *Aggregate wall distance:* Summation of the closest distance of each minutia from the cuboid boundary, (2) *Minutiae Average:* Average coordinate of all the minutiae present in each cuboid in a given measurement, and (3) *Minutiae Deviation:* Standard deviation of minutiae coordinates present in each cuboid in a given measurement.

Let $\{m_1(=(x_1,y_1,\theta_1)), m_2(=(x_2,y_2,\theta_2)), ..., m_t(=(x_t,y_t,\theta_t))\}$ be the set of minutiae in a fingerprint that lie inside an aggregate region represented by $(x_{min}, x_{max}, y_{min}, y_{max}, \theta_{min}, \theta_{max})$ such that

$$(x_{min} \leq x_i \leq x_{max}) \quad , \quad y_{min} \leq y_i \leq y_{max} \quad \& \quad \theta_{min} \leq \theta_i \leq \theta_{max} \quad \forall i \in 1,2,...,t. \tag{4}$$

The aggregate wall distance (δ) for this configuration is computed as:

$$\delta = \sum_{i=1}^{t} \min(\delta_x, \delta_y, \delta_\theta, \tau_\delta) \tag{5}$$

where τ_δ is a threshold used for wall distance, and δ_x, δ_y, and δ_θ are calculated as $\min(|x_i - x_{min}|, |x_i - x_{max}|)$, $\min(|y_i - y_{min}|, |y_i - y_{max}|)$, and $\min(|\theta_i - \theta_{min}|, |\theta_i - \theta_{max}|)$, respectively.

as defined above, aggregate wall distance is a multiple of the number of minutiae in a cuboid if all the minutiae are at a distance of at least τ_δ from the sides of the cuboid. However if a minutiae is close to the boundary of the cuboid, its contribution to the feature value is reduced. This is useful because a minutia close to cuboid boundary has a greater chance of migrating outside the cuboid when another impression of the same fingerprint is taken.

Both minutiae average and minutia deviation features consist of three components each corresponding to the x, y, and θ coordinates, these are μ_x, μ_y, μ_θ, σ_x, σ_y, σ_θ. Standard formulae are used for computing the average

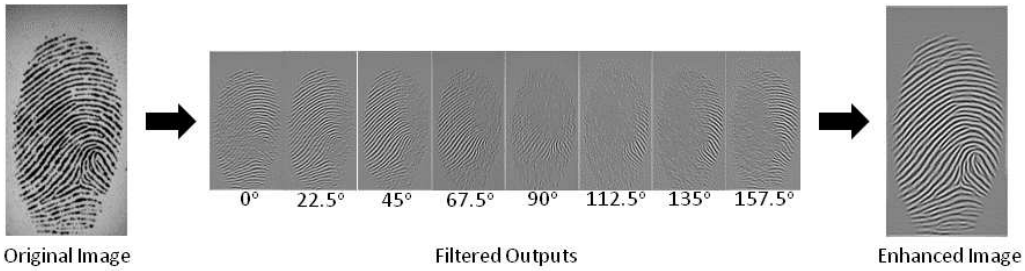


Figure 6. Enhancement procedure using Gabor filtering.

and the standard deviation for the x and y coordinates whereas for θ coordinate, the mean μ_θ and the standard deviation σ_θ are computed as follows:

$$\mu_s = \frac{1}{t}\sum_{i=1}^{t}\sin\theta_i,\ \ \mu_c = \frac{1}{t}\sum_{i=1}^{t}\cos\theta_i,\ \ \mu_\theta = \tan^{-1}\left(\frac{\mu_s}{\mu_c}\right)$$

$$\sigma_\theta = \sqrt{\frac{1}{t-1}\sum_{i=1}^{t}\left[\min(|\theta_i - \mu_\theta|, 360 - |\theta_i - \mu_\theta|)\right]^2}$$

where θ_i is the angle corresponding to the i^{th} minutia. If there is no minutia in a particular cuboid, the average features assume the value corresponding to the center of the cuboid whereas the deviation features are set to zero. The deviation features are also set to zero when there is only a single minutia in the cuboid.

5.2 Ridge Orientation Features

Ridge orientation is another discriminative source of information in a fingerprint. Information captured by ridge orientation is sufficiently independent of that captured by minutiae. Note that in many cases even if the minutiae extraction is unreliable due to noisy fingerprints, ridge orientation can still be estimated. In order to extract ridge based features, we use the method similar to the one proposed in Ross *et al.*[18]

First, the fingerprints are enhanced using Gabor filters according to the fingerprint enhancement technique as proposed by Lin Hong[19] in order to remove unwanted noise as follows. Let the fingerprint image be represented as $I(x, y)$. This image is convolved with an even symmetric Gabor filter represented by

$$G(x, y, f, \theta) = \mathcal{G}(x', y', \delta_x, \delta_y)\cos(2\pi f x') \tag{6}$$

where $\mathcal{G}(x', y', \delta_x, \delta_y) = e^{-.5(x'^2/\delta_x^2 + y'^2/\delta_y^2)}$, $x' = x\sin(\theta) + y\cos(\theta), y' = x\cos\theta - y\sin\theta$, f is the frequency of the sinusiodal plane wave along the direction θ from the x-axis, and δ_x and δ_y are the space constants of the Gaussian envelope along x and y axes, respectively. Eight values of $\theta \in \{0^o, 22.5^o, 45^o, 67.5^o, 90^o, 112.5^o, 135^o, 157.5^o\}$ are used to obtain eight different filters which are convolved with $I(x, y)$ to obtain the corresponding filtered outputs given by $E_\theta(x, y) = G(x, y, f, \theta) * I(x, y)$. The enhanced fingerprint image is given by

$$I'(x, y) = a(x, y)E_{\theta_1}(x, y) + (1 - a(x, y))E_{\theta_2}(x, y) \tag{7}$$

where $\theta_1(x, y), \theta_2(x, y) \in \{0^o, 22.5^o, 45^o, 67.5^o, 90^o, 112.5^o, 135^o, 157.5^o\}$ are the two angles closest to the ridge orientation $o(x, y) \in \{0, 180\}$ at (x, y) and $a(x, y) = min(|(o(x, y) - \theta_1(x, y))|, 180 - |(o(x, y) - \theta_1(x, y))|)/22.5$. Fingerprint orientation is obtained using the algorithm proposed in Dass, 2004.[13] Figure 6 shows the fingerprints obtained after enhancement.

In order to obtain the orientation-based features, the enhanced image I' is filtered using four different Gabor filters described by eq (6) corresponding to four different values of θ i.e. 0^o, 45^o, 90^o, and 135^o. The four filtered images are represented as F_θ, $\theta \in \{0^o, 45^o, 90^o, 135^o\}$. Given a local aggregate region, four different values are obtained corresponding to the standard deviations of the values inside the aggregate region in F_θ. Figure 7 shows the four filtered images obtained from an input fingerprint.

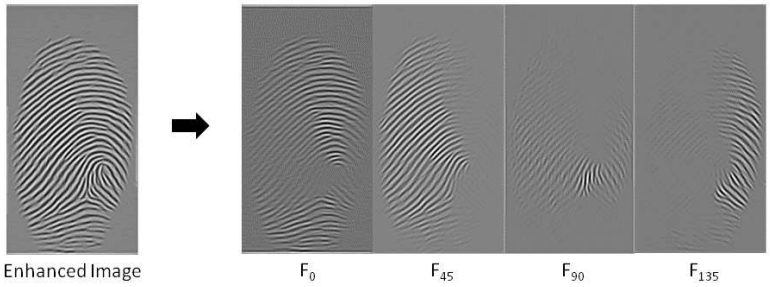


Figure 7. Four different filtered images are obtained from the enhanced image. One bit is extracted from each of these filtered images.

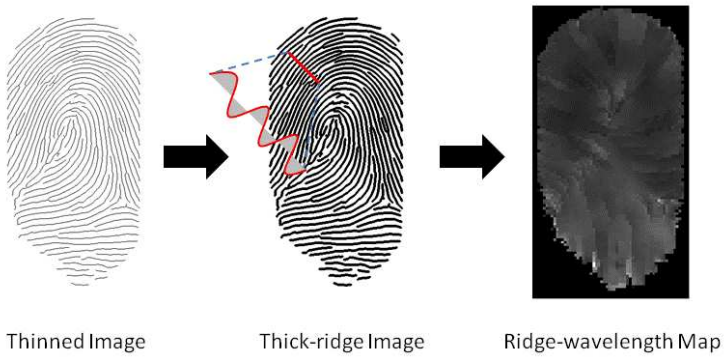


Figure 8. Extraction of ridge-wavelength features.

5.3 Ridge Wavelength Features

Feng[20] showed that the ridge wavelength is a salient fingerprint feature with the third best matching performance among a set of 17 different features selected. In order to obtain the ridge wavelength, we obtain a thinned image from the fingerprint.[21] This thinned image is then thickened using morphological operation to obtain 5-pixel wide ridges. Ridge orientation is obtained using the algorithm of Dass[22] and at each point in the fingerprint, a scan line is obtained in direction perpendicular to the ridge orientation. We use Bresenham scan line algorithm to obtain the pixels corresponding to the scan line. Then distances between consecutive ridge pixels are measured to obtain inter ridge distances. The mean value of the inter-ridge distances is obtained as the ridge wavelength value corresponding to the point in consideration. A ridge wavelength map of the whole fingerprint is thus obtained. Given a local aggregate region, the feature value is obtained as the average of the ridge wavelength values corresponding to the pixels lying in the local region. Figure 8 shows a schematic diagram for extraction of ridge wavelength features. In order to account for certain noisy regions, we consider all the values that are less than 5 or greater than 25 as missing values.

In order to achieve efficiency without significant loss in information content, we tessellate the whole fingerprint image into a rectangular grid whose each element is of size 5×5 and compute a single wavelength value for each grid element.

Let $W(x, y)$ represent the value of the wavelength features at the $(x, y)^{th}$ pixel in the image. The value of the wavelength feature corresponding to region R_{2D} represented as $[x_{min} \quad x_{max} \quad y_{min} \quad y_{max}]$ is given as $W_{R_{2D}} = \text{avg}\{W(x, y) | (x, y) \in R_{2D}, 5 \leq W(x, y) \leq 25\}$. The average computation discards the missing values as explained above and computes the mean corresponding to the available values.

6. BIT EXTRACTION

After the minutiae, ridge orientation and ridge frequency features have been extracted from the fingerprints, it is required to binarize those features. We use the median values computed over the training set as thresholds in order to binarize the features. First, a set of n fingerprints from the database are centered as described by (3)

and then for each 3D local aggregate region, the seven minutiae features are extracted and for each of the 2D local aggregate regions the four ridge orientation and ridge wavelength features are extracted. For each of these features, n different values are obtained corresponding to the n different fingerprints. The threshold used to binarize these features is then computed as the median of these n values. If some of the features contain missing values, the median of the remaining values is considered as the threshold. Note that as a result of median based thresholding, each bit will be 0 or 1 with roughly equal probability. This leads to the maximum ambiguity in each bit.

Once the binary features have been obtained from the fingerprints, it is important to select the most relevant features that lead to maximum matching performance. Feature selection is an important problem of general interest and a number of approaches have been used to obtain the most relevant bits among a set of bits that are used to represent an object. The two main criteria of feature selection are dependence among the features (or redundancy) and discriminability (or relevance) of each feature.[23] If the features have high correlation then some of the features can be discarded in order to accommodate additional unrelated features, thereby improving the matching performance. As a first step in eliminating inferior features, those cuboids that have large overlap with other cuboids are discarded, because these would naturally generate highly correlated features.

The following procedure is used for this[4] :

1. Compute the relative overlap, i.e., the ratio of volume of intersection to the volume of union, for all pairs of cuboids.
2. Select the pair having highest relative overlap.
3. Discard the cuboid whose maximum relative overlap with the other cuboids is greater.
4. Repeat steps 2 and 3 until desired number of nearly non-overlapping cuboids or the desired reduction in overlap is obtained.

Further, some of the correlated bits are also discarded using the above procedure where the relative partial overlap criterion is replaced with bit correlation.

In order to compute the discriminability of each bit, let

- b_e^{ij} be the value of the i^{th} bit corresponding to the j^{th} enrolled user
- b_{qG}^{ijk} be the value of the i^{th} bit corresponding to the k^{th} genuine query for to the j^{th} enrolled user
- b_{qI}^{ijk} be the value of the i^{th} bit corresponding to the k^{th} impostor query for to the j^{th} enrolled user

Then, the discriminability of the i^{th} bit corresponding to the j^{th} user is given by

$$d_i^j = I_i^j - G_i^j \tag{8}$$

where

$$G_i^j = 1/N_G \sum_k d_{Gi}^{jk}, \quad d_{Gi}^{jk} = b_e^{ij} \oplus b_{qG}^{ijk} \tag{9}$$

$$I_i = 1/N_I \sum_k d_{Ii}^{jk}, \quad d_{Ii}^{jk} = b_e^{ij} \oplus b_{qI}^{ijk} \tag{10}$$

and, N_G and N_I are the number of genuine and impostor queries. In case d_{Gi}^k (or d_{Ii}^{jk}) cannot be computed due to presence of missing values in either b_e^{ij} or b_{qg}^{ijk} (or b_{qi}^{ijk}), its value is assumed to be 0.5. Based on these discriminability values, a subset of bits are finally extracted separately for each user.

7. EXPERIMENTS

We performed all experiments on the Fingerprint Verification Competition 2002 database-2a. This publicly available database contains fingerprints from 100 different fingers each having 8 different impressions captured at a resolution of 569 dpi using an optical sensor. First we obtain a set of 750 3D rectangular regions and the same number of 2D regions using the technique described in Section 3. Relative pairwise overlap is then used to discard 250 regions from each set leading to 500 2D and 500 3D regions as described in Section 6.

We binarize the features based on the median values. The median values are computed for each feature using a set of 100 enrolled fingerprints. For this, the local rectangular regions are evaluated on the first impression of each of the 100 fingers in the database. Seven minutiae based features (three average minutia, three deviation minutia, and wall distance) are evaluated on the 500 3D regions and four orientation based features and the ridge wavelength are evaluated on the 500 2D regions. This leads to $6000(= 12 \times 500)$ different features per fingerprint. For each feature, the median value is used as the threshold to binarize the features, i.e., the bit corresponding to a feature is zero if the feature value is less than the corresponding median value and one otherwise.

After binarization, we choose 1500, 700, and 300 bits each from the minutiae, texture and ridge frequency features such that they have the least correlation. Section 8 contains a discussion of the entropy of the binary features before and after discarding correlated bits.

Once the correlated bits have been discarded, discriminability of bits is computed for each user as described in Section 6. We select 900 most discriminable bits from the pool of $2500(= 1500 + 700 + 300)$ bits separately for each user.

In our experiments, the first impression of each finger is used as enrolled fingerprint and second impression is used as a genuine query during testing. The remaining six impressions are used during training as genuine queries. Impostor queries during training are obtained using the second impressions of 20 different users leading to a total of $2000(= 20 \times 100)$ impostor matches during training. During testing, the first impression of each of the 99 other fingerprints is used as impostor queries leading to a total of 9900 impostor matches during testing. In order to compute the distance between two bit vectors we use modified normalized Hamming distance in order to account for the missing values in the data. Let $b_e = \{b_e^1, b_e^2, ..., b_e^n\}$ be the enrolled binary vector and $b_q = \{b_q^1, b_q^2, ..., b_q^n\}$ be the binary vector obtained from the query biometric. First we remove the bits from both these vectors which have a missing value in the enrolled vector to obtain $b'_e = \{b_e^{a_1}, b_e^{a_2}, ..., b_e^{a'_n}\}$ and $b'_q = \{b_q^{a_1}, b_q^{a_2}, ..., b_q^{a'_n}\}$ where $a_1, a_2, ..., a'_n$ are the positions where b_e does not have missing values. Let there be r bits that have missing value in b'_q and besides these there are s positions in b'_q which differ from b'_e. The modified normalized Hamming distance between b_e and b_q is given by

$$MNHD(b_e, b_q) = (2s + r)/n'. \tag{11}$$

Figure 9 shows the Receiver Operating Characteristic (ROC) curves corresponding to the 900 most discriminable bits based on the modified normalized hamming distance. For the sake of comparison, ROC curves corresponding to all the bits extracted from minutiae, ridge orientation and ridge wavelength are also plotted in the same plot. Also, we consider the case when an impostor does not know which bits have been selected for the given user based on discriminability. This curve has been named as 'Unknown Regions" in the Figure 9. Also a curve corresponding to all the 6000 bits extracted from a fingerprint is plotted as 'Allbits' in Figure 9.

The above results are computed using a new alignment technique described in Section 4 where the singular points such as the core and the delta are used to improve the alignment obtained based on the high curvature points. Figure 10 shows the ROC curves corresponding to the 900 most discriminable bits obtained from features in case the alignment is performed using just the high curvature points as well as the case when singular points are incorporated in alignment. As can be observed from the figure, improved alignment leads to an increase in GAR from 82% to 89% at an FAR of 0.01%.

8. SECURITY ANALYSIS

We now discuss the security of the system in two different attack scenarios: template inversion and template linkage. Template inversion refers to an attack in which the adversary tries to obtain the original template which

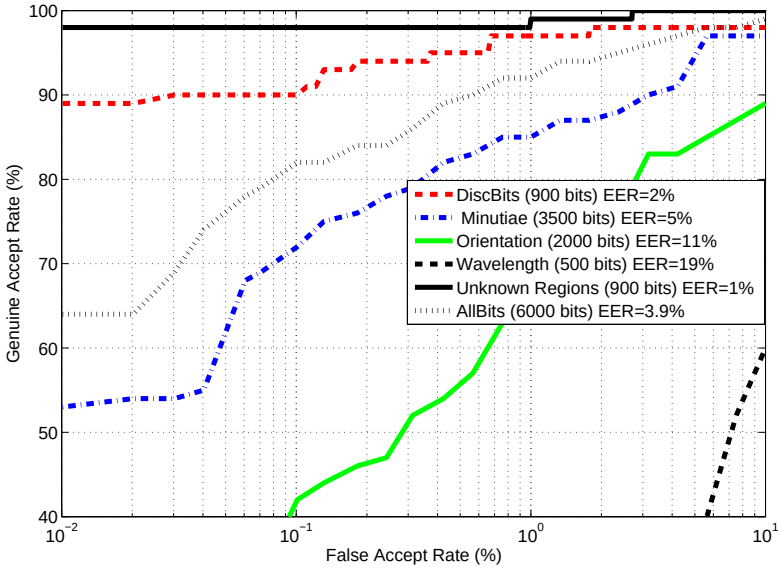


Figure 9. ROC curve corresponding to the various combinations of fingerprint features. "Minutiae" refers to the ROC curve for minutiae-based features alone. "Orientation" refers to the ROC curve for ridge orientation alone. "Wavelength" refers to the ROC curve for ridge wavelength alone. "AllBits" refers to all bits taken together. "DiscBits" refers to the discriminable bits with the knowledge of these discriminable features available to the attacker. "Unknown Regions" refers to the case in which the attacker does not know which features have been chosen as discriminable features.

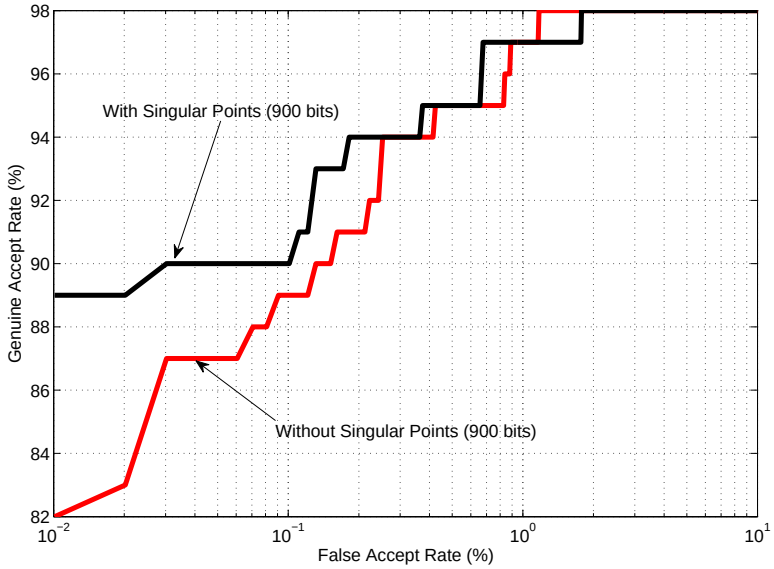


Figure 10. ROC curve with and without consideration of singular points (core and delta) for improved alignment.

constitute the set of minutiae, the ridge orientation and ridge wavelength features from the transformed version which is the binary vector in the present case. Template linkage refers to an adversary attack where the adversary tries to identify templates corresponding to the same individual that have been differently transformed. The two properties that measure the effectiveness of a template protection technique against these attacks are commonly referred to as non-invertibility and cancelability respectively.

Non-invertibility of the system relates to the complexity of obtaining the original features given the transformed features. Considering that a minutia is required to be estimated with a tolerance of p pixels in x and y dimensions and q pixels in the θ dimension, we can tessellate the whole x, y, θ-space into a set of $\mathcal{N}$ rectangular elements of size $p \times p \times q$ each. Each element, $R_i, i = 1, ..., \mathcal{N}$, is considered as a variable that can take value 1, if a minutiae is present in that configuration and zero otherwise. If the size of image is 500×300 and $p, q = 5$ the number of different configurations possible are $\sim 420,000$. Since each local aggregate region corresponds to a set of these rectangles, the features viz. wall distance, minutia average and minutiae deviation can be written in terms of R_i. The availability of binarized feature values lead to inequalities in $R_i's$ which can be solved using an interior point detection algorithm. The other features can also be similarly estimated by considering an appropriate tessellation. However, solving these problems is, in general, computationally very expensive and there is no guarantee that a solution exists or if an obtained solution is unique. The proposed technique is thus highly non-invertible.

Cancelability of a system refers to the fraction of times a transformed template is accepted by a biometric recognition system in which a differently transformed version of the same biometric feature is enrolled. In the current system cancelability can be achieved in two main ways: (1) Random transformation or (2) Cuboid replacement. In a random transformation technique the extracted features can be transformed using a random matrix of real numbers or a permutation matrix.[24] In this case however the transformation/permutation matrix is required to be kept secret. The other technique is to generate a completely new set of local aggregate regions for each new enrollment. Note, however, that some of the newly generated regions will have significant overlap with the earlier regions leading to some bits from one template being correlated with possibly some bits from the other template. The scheme is completely secure if the configuration of the cuboids are not publicly available, for example, when the user-specific cuboids are held by a user on a smart card.

Further the attacker can try to guess the binary features extracted using the proposed technique which can be used to stage an inversion attack or a linkage attack. This is one of the reasons for eliminating the correlated bits and cuboids as described in Section 6. Since the bits are not totally independent, the entropy of the bits is usually less than n. In order to evaluate the entropy in a binary feature vector, we compute the degrees of freedom of the vector considering each bit as a Bernoulli random variable, similar to a method of Daugman.[25]

First we compute 4950 pairwise normalized hamming distances by considering all pairs of the enrolled feature vectors. The mean of the pairwise distances is an estimate of the probability that the bernoulli random variable corresponding to each element of the feature vector is 1. The degrees of freedom in this case is given by $p(1-p)/\sigma^2$ where p is the mean NHD and σ is the standard deviation of the NHD. Figure 11 shows the histogram of NHD corresponding to all the features and that corresponding to the ones selected based on correlation.

NHD corresponding to all the features has a mean of 0.504 and a variance of 0.0072 whereas those corresponding to features selected based on correlation have mean of 0.504 and variance is 0.0042. Thus, eliminating the correlated values leads to a greater separation among the feature vectors. We also computed the degrees of freedom before and after selection of bits based on correlation for individual feature modalities. The degrees of freedom corresponding to all the features increased from 34.7 to 59.6. Note that degree of freedom is a measure of randomness in the binary features and thus the security of the technique. The degrees of freedom for minutiae features increased from 54 to 157, that of ridge orientation features increased from 12.3 to 15.4 and that of ridge wavelength features increased from 2.9 to 3.2. Clearly, minutiae features have significantly greater variability than the ridge orientation based or the wave based features. Note that degree of freedom only depends on the randomness of the signal whereas matching performance is affected by the intra class variations as well.

9. CONCLUSIONS

In this paper, we augment the binary fingerprint feature extraction technique proposed in Sutcu *et al.*[4] by incorporating minutia average, minutia deviations, ridge orientations and ridge wavelengths in order to improve

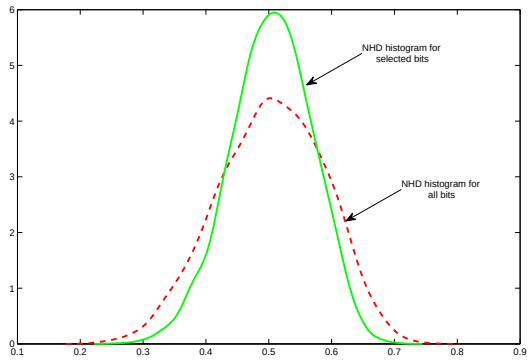


Figure 11. Normalized Hamming distance among the feature vectors before and after correlated bits have been removed.

the matching performance. If these feature vectors are used in a syndrome-based secure biometrics framework, then the security is improved because the longer feature vectors now allow the use of stronger codes. The proposed approach produced 90% GAR at 0.1% FAR. We also propose a technique to incorporate singular points from a fingerprint such as a core or a delta in order to improve the matching performance. As shown in Figure 10, GAR at 0.1% FAR improved from 89% to 90% with the use of singular points. The improvement is even larger at 0.01% FAR, for which the GAR increased from 82% to 89%. Further, the randomness present in the binary features is analyzed by computing the degree of freedom of the binary feature vector considering each bit of the feature vector as a Bernoulli variable. This paper focussed solely on the properties of the binary feature vectors and the matching tradeoffs that are obtained with a rich feature set. Evaluating the performance of a secure biometric system in which these feature vectors are combined with a syndrome code will be considered in our future work. A subset of the syndrome coding results are scheduled to appear shortly in Nagar *et al.*[10] .

REFERENCES

[1] Jain, A., Nandakumar, K., and Nagar, A., "Biometric template security," *EURASIP Journal on Advances in Signal Processing* **2008**, 1–17 (2008).

[2] Feng, J. and Jain, A., "FM model based fingerprint reconstruction from minutiae template," in [*International conference on Biometrics (ICB)*], (2009).

[3] Mordini, E. and Massari, S., "Body, biometrics and identity," *Bioethics* **22**(9), 488–498 (2008).

[4] Sutcu, Y., Rane, S., Yedidia, J., Draper, S., and Vetro, A., "Feature extraction for a slepian-wolf biometric system using ldpc codes," in [*Proceedings of the IEEE International Symposium on Information Theory*], (July 2008).

[5] Juels, A. and Sudan, M., "A Fuzzy Vault Scheme," in [*Proceedings of IEEE International Symposium on Information Theory*], 408 (2002).

[6] Nandakumar, K., Jain, A. K., and Pankanti, S., "Fingerprint-based Fuzzy Vault: Implementation and Performance," *IEEE Transactions on Information Forensics and Security* **2**, 744–757 (December 2007).

[7] Nandakumar, K. and Jain, A., "Multibiometric Template Security Using Fuzzy Vault," in [*International Conference on Biometrics: Theory, Applications and Systems*], 1–6 (2008).

[8] Farooq, F., Bolle, R., Jea, T., and Ratha, N., "Anonymous and revocable fingerprint recognition," in [*Proc. Computer Vision and Pattern Recognition*], (June 2007).

[9] Gyaourova, A. and Ross, A., "A Novel Coding Scheme for Indexing Fingerprint Patterns," in [*Proceedings of the 2008 Joint IAPR International Workshop on Structural, Syntactic, and Statistical Pattern Recognition*], (2008).

[10] Nagar, A., Rane, S., and Vetro, A., "Privacy and Security of Features Extracted from Minutiae Aggregates," in [*International Conference on Acoustics, Speech and Signal Processing*], (2010). To appear.

[11] Draper, S. C., Khisti, A., Martinian, E., Vetro, A., and Yedidia, J. S., "Using Distributed Source Coding to Secure Fingerprint Biometrics," in [*Proceedings of IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP)*], **2**, 129–132 (April 2007).

[12] Uludag, U. and Jain, A. K., "Securing Fingerprint Template: Fuzzy Vault With Helper Data," in [*Proceedings of CVPR Workshop on Privacy Research In Vision*], 163 (June 2006).

[13] Dass, S. C., "Markov Random Field Models for Directional Field and Singularity Extraction in Fingerprint Images," *IEEE Transactions on Image Processing* **13**, 1358–1367 (October 2004).

[14] Dass, S. C. and Jain, A. K., "Fingerprint Classification Using Orientation Field Flow Curves," in [*Proceedings of Indian Conference on Computer Vision, Graphics and Image Processing*], 650–655 (December 2004).

[15] Chetverikov, D., Svirko, D., Stepanov, D., and Krsek, P., "The Trimmed Iterative Closest Point Algorithm," in [*Proceedings of International Conference on Pattern Recognition*], 545–548 (August 2002).

[16] Cappelli, R., Lumini, A., Maio, D., and Maltoni, D., "Fingerprint Image Reconstruction From Standard Templates," *IEEE Transactions on Pattern Analysis and Machine Intelligence* **29**(9), 1489–1503 (2007).

[17] Cappelli, R., Maio, D., Maltoni, D., Wayman, J., and Jain, A., "Performance evaluation of fingerprint verification systems," *IEEE Transactions on Pattern Analysis and Machine Intelligence* **28**(1), 3–18 (2006).

[18] Ross, A., Jain, A., and Reisman, J., "A Hybrid Fingerprint Matcher," *Pattern Recognition* **36**, 1661–1673 (July 2003).

[19] Hong, L., "Automatic personal identification using fingerprints," tech. rep. (1998).

[20] Feng, J., "Combining minutiae descriptors for fingerprint matching," *Pattern Recognition* **41**(1), 342–352 (2008).

[21] Jain, A. K., Hong, L., and Bolle, R., "On-line Fingerprint Verification," *IEEE Transactions on Pattern Analysis and Machine Intelligence* **19**, 302–314 (April 1997).

[22] Dass, S., "Markov Models for Directional Field and Singularity Extraction in Fingerprint Images," *IEEE Transactions on Image Processing* , 1358–1367 (2004).

[23] Peng, H., Long, F., and Ding, C., "Feature selection based on mutual information: criteria of max-dependency, max-relevance, and min-redundancy," *IEEE transactions on pattern analysis and machine intelligence* **27**(8), 1226–1238 (2005).

[24] Lumini, A. and Nanni, L., "An improved BioHashing for human authentication," *Pattern Recognition* **40**(3), 1057–1065 (2007).

[25] Daugman, J., "The importance of being random: statistical principles of iris recognition," *Pattern Recognition* **36**(2), 279–291 (2003).