To Appear in BTAS 2012

Fingerprint Image Synthesis based on Statistical Feature Models

Qijun Zhao
Sichuan University
qjzhao@msu.edu

Anil K. Jain
Michigan State University
jain@cse.msu.edu

Nicholas G. Paulter Jr., Melissa Taylor
National Inst. of Standards and Technology
{paulter, melissa.taylor}@nist.gov

Abstract

Fingerprint image synthesis has received considerable attention because of its potential use in generating large databases to evaluate the performance of fingerprint recognition systems. Existing fingerprint synthesis algorithms (e.g., SFinGe) focus on rendering realistic fingerprint images, but the features (e.g., minutiae) in these fingerprints are formed in an uncontrollable manner. However, generating synthetic fingerprint images with specified features is more useful in developing, evaluating and optimizing fingerprint recognition systems by providing ground truth features in the synthesized images. In this paper, we propose a method to synthesize fingerprint images that retain prespecified features (i.e., singular points, orientation field, and minutiae). To obtain realistic fingerprints, these features are sampled from appropriate statistical models which are trained by using real fingerprints in public domain databases. We validate the proposed method by comparing the synthesized images with those generated by SFinGe and by investigating the match score distributions on synthesized and real fingerprint databases. Furthermore, the synthesized fingerprint images and their minutiae are used to evaluate the matching capabilities of two commercial off-the-shelf (COTS) fingerprint matchers.

1. Introduction

Fingerprints have been widely used in many forensics and civilian applications, such as criminal and victim identification, access and attendance control, and border control [16]. The past decade has witnessed a vast growth in the number of fingerprints housed in automated fingerprint recognition systems; e.g., FBI's IAFIS [4] holds more than 100 million fingerprints, and DHS's US-VISIT [6] stores over 600 million fingerprints. Such large scale fingerprint databases impose enormous challenges to the development and evaluation of fingerprint recognition systems.

Fingerprint recognition systems are usually evaluated based on their recognition accuracy (or error rates) on test data [18]. Three main steps are generally involved in the evaluation: (i) collecting a set of fingerprint images, (ii) extracting and matching the features in the fingerprint images to generate genuine and imposter match scores, and (iii) analyzing the obtained match scores to compute error rates (e.g., false match rates and false non-match rates) of the fingerprint recognition system. Such score-based evaluation methods were employed in many public domain evaluations of biometric technologies, e.g., Fingerprint Vendor Technology Evaluation (FpVTE) [2,3] organized by the National Institute of Standards and Technology (NIST).

Score-based evaluation has two major limitations. First, from the viewpoint of end-users of fingerprint recognition systems, while such a technology evaluation can give a rough estimate of the accuracy of the systems, its scope is limited by the size and representativeness of the test databases. Large and representative databases are necessary for reliable and scalable performance evaluation. However, it is both expensive and time consuming to manually collect fingerprint images (along with ground truth features in them) from a large population. As a result, none of the available fingerprint databases in public domain are comparable in size to the operational databases in large-scale applications. For example, the database used in FpVTE 2003 [2] contains fingerprints from about twenty-five thousand subjects, and the database in FpVTE 2012 [3] contains fingerprints from 10 million subjects. Compared with the ever-expanding operational databases (e.g., India's UID system [5] has already collected ten prints for about 150 millions residents), these databases are relatively small. In order to construct large fingerprint databases at a very low cost, some researchers [9] have proposed to generate synthetic fingerprint images.

Second, from the viewpoint of designers of fingerprint recognition systems, system accuracy alone is not enough to identify the technical limitations of the systems or ways that can improve the system performance. For example, the low system accuracy might be due to deficient sensors which generate low quality fingerprint images, inadequate feature extractors which result in spurious and missing minutiae, and fragile feature matchers which cannot efficiently distinguish similar/dissimilar features in

the presence of distortion and noise. It is necessary to identify the underlying sources of error in a fingerprint recognition system in order to improve the system performance. This can be done by utilizing the ground truth features in fingerprint images, e.g., the features in individual fingerprint images and the feature correspondences between pairs of fingerprints. However, such feature information is not available in traditional fingerprint databases because of the prohibitive cost of manually marking ground truth features for the large number (e.g., tens of thousands) of fingerprints in the databases. Features (e.g., minutiae) in the fingerprint images synthesized by existing synthesis algorithms are formed in an uncontrollable manner and separate algorithms have to be applied to detect the features [7].

In this paper, we make an attempt to synthesize fingerprint images which retain prespecified features (i.e., singular points, orientation field, and minutiae). See Fig. 1. Statistical feature models are established for each of the five major types of fingerprints (i.e., arch, tented arch, left loop, right loop, and whorl). Given a fingerprint type and the fingerprint image size, singular points, ridge orientation field, and minutiae are sequentially sampled from their respective statistical models (see Fig. 2). A master fingerprint containing the specified features is then generated by using a fingerprint reconstruction algorithm [12]. Multiple impressions are generated by applying nonlinear plastic distortion and global rigid transformation to the master fingerprint. Finally, image rendering is performed on the impressions, which simulates finger dryness and noise. Note that features in different fingerprint impressions are traced during the synthesis.

The rest of this paper is organized as follows. Section 2 reviews related work on fingerprint image synthesis and reconstruction. Section 3 introduces the proposed algorithm for fingerprint image synthesis. Section 4 then validates the synthesized fingerprints, and reports the evaluation results with two commercial off-the-shelf (COTS) fingerprint matchers (denoted as COTS1 and COTS2). Section 5 finally concludes the paper with discussion on future work.

2. Related Work

Techniques to generate fingerprint images can be divided into two categories: (i) fingerprint synthesis and (ii) fingerprint reconstruction. Fingerprint synthesis aims to generate a large number of realistic-looking synthesized fingerprint images to evaluate fingerprint recognition algorithms, whereas fingerprint reconstruction algorithms reconstruct fingerprint images from a given set of features (typically, minutiae) to evaluate the security of fingerprint templates.

SFinGe [9] is state-of-the-art fingerprint image synthesis algorithm. Taking fingerprint type, image size, region of interest, and singular points as input, SFinGe first

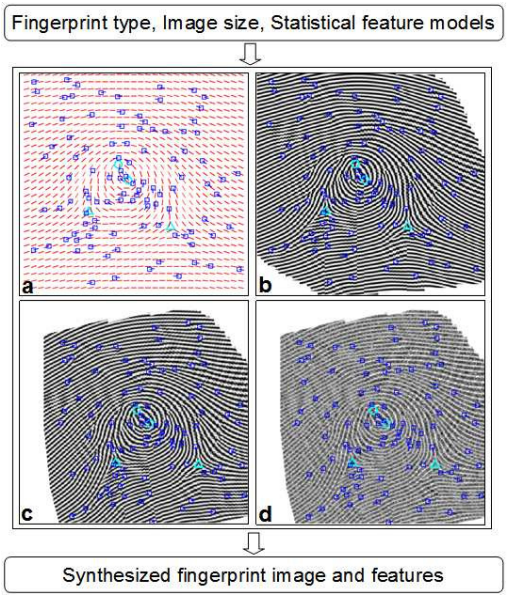


Figure 1. The proposed fingerprint image synthesis method includes four main modules: (a) sampling features (singular points, orientation field, and minutiae) from appropriate statistical feature models; (b) generating a master fingerprint; (c) generating multiple fingerprint impressions from the master fingerprint via distortion (one such impression is shown here); and (d) rendering fingerprint images by simulating finger dryness and adding noise.

generates an orientation field based on a modified Zero-Pole model according to the fingerprint type and singular points, and then iteratively applies Gabor filters to a seed image (e.g., a white image with a few randomly placed black points) to generate a master fingerprint based on the orientation field and a fixed value of ridge frequency. SFinGe finally generates multiple impressions of the master fingerprint by distorting and rendering the master fingerprint with randomly-chosen distortion and noise parameters. Databases generated by SFinGe have been used in a series of fingerprint verification competitions, called FVC [16]. It has been observed that the performance of different fingerprint matchers on the synthetic fingerprint databases generated by SFinGe follows the same trend as on databases of real fingerprints collected from human subjects.

One limitation of SFinGe is that it cannot control the number and location of minutiae since the minutiae in the synthesized fingerprint images are formed randomly during the iterative application of Gabor filters. To obtain the minutiae in the synthesized fingerprint images, either manual markup or automatic extraction is needed [16]. Moreover, it has not been verified whether the minutiae generated by Gabor filtering in SFinGe follow the distribution of minutiae in real fingerprints (see Fig. 4).

Another body of related work [8, 12, 19] deals with algorithms for reconstructing fingerprints from stored minutiae templates. Their objective is to study the template inversion problem with the purpose of showing that minutiae templates are invertible and need to be secured [14]. These algorithms first reconstruct an orientation field based on the given minutiae, then reconstruct the fingerprint based on the orientation field and the minutiae (similar to SFinGe, a fixed ridge frequency value is assumed for the reconstructed fingerprints), and finally render the reconstructed fingerprint image to make it appear more realistic. Among various fingerprint reconstruction algorithms, the algorithm proposed by Feng and Jain [12] is the state-of-the-art, which results in a relatively small number of spurious and missing minutiae compared with its counterparts. In this paper, we will employ Feng and Jain's algorithm to generate synthetic fingerprints from given orientation field and minutiae that are not extracted from real fingerprints, but sampled from statistical models according to the specified fingerprint type.

3. Proposed Method

The proposed fingerprint image synthesis algorithm consists of four main modules: sampling fingerprint features from statistical models, generating a master fingerprint, generating multiple impressions from the master fingerprint, and rendering fingerprint images (see Fig. 1).

3.1. Sampling Features

In this paper, we consider synthesizing typical 500 ppi fingerprint images with level-1 and level-2 features, i.e., singular points (cores and deltas), orientation field, and minutiae (ridge endings and ridge bifurcations). Unlike existing fingerprint synthesis and reconstruction algorithms, the proposed method samples features from their statistical distribution models. Different types of fingerprint features are essentially dependent on each other. For example, orientation field is partially determined by singular points [20], minutiae density tends to be higher in regions around singular points than in regions far from singular points, and minutia directions are determined by their types and the ridge orientations at their locations. Therefore, given a fingerprint type to be synthesized, we sequentially sample its features from statistical models, i.e., first singular points, followed by orientation field, and finally, minutiae (see Fig. 2).

3.1.1 Sampling Singular Points

Cappelli and Maltoni [10] observed that the spatial locations of singular points in fingerprints can be approximated by a mixture of Gaussians. They aligned fingerprint images so that the centroid of singular points is in the image center (which is taken as the origin), and constructed a feature vector to represent the singular points by concatenating their coordinates. Further, for each of the four major types of fingerprints (i.e., tented arch, left loop, right loop, and whorl) that have singular points, they assumed that the feature vectors of singular points follow a finite mixture of Gaussian distributions. In this paper, we employ Cappelli and Maltoni's statistical models of singular points for sampling. To sample the singular points for a given fingerprint type, a feature vector of singular points is randomly generated via realizing the corresponding Gaussian mixture model. The coordinates of the singular points can be then retrieved from the feature vector, and aligned to the image coordinate system (whose origin is at the top left corner of the image) according to the given fingerprint image size.

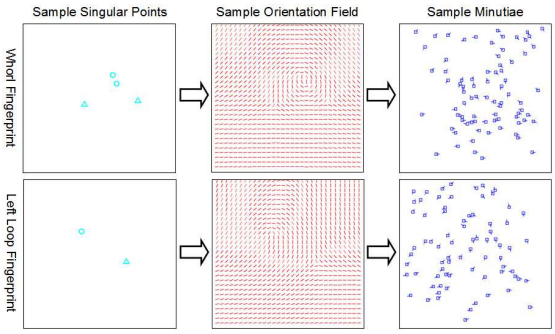


Figure 2. Singular points, orientation field, and minutiae are sampled sequentially from their respective distribution models according to the specified fingerprint type.

3.1.2 Sampling Orientation Field

The orientation field of a fingerprint represents the dominant local ridge orientations. Let Ω denote the region of interest in the fingerprint image. The fingerprint orientation field can be then viewed as a function $\theta(x, y)$ of the location $(x, y) \in \Omega$, where $\theta(x, y) \in [0, \pi)$ represents the dominant local ridge orientation at (x, y). In order to establish the statistical distribution model of orientation field, we need to first represent the orientation field by a fixed-length feature vector. We achieve this by decomposing the orientation field into singular and residual components and approximating the two components by Zero-Pole model [20] and cosine peripheral model [22], respectively.

The Zero-Pole model [20] for the singular component of orientation field is defined as

$$\theta^{SP}(z) = \frac{1}{2} \arg \left(\frac{\prod_{t=1}^{N_c} (z - z_{c_t})}{\prod_{j=1}^{N_d} (z - z_{d_j})} \right), \qquad (1)$$

where $z = x + i \cdot y$ is a point in the fingerprint represented in

the complex plane, z_{c_t} and z_{d_j} correspond, respectively, to the t^{th} core and the j^{th} delta, and N_c and N_d are, respectively, the numbers of cores and deltas in the fingerprint. The Zero-Pole model describes the influence of singular points on fingerprint orientation field.

The residual component of orientation field is defined as the difference between the orientation field θ and its singular component θ^{SP}, i.e.,

$$\theta^R = \theta - \theta^{SP}. \tag{2}$$

It can be approximated by using the following cosine peripheral model [1] [22]:

$$\theta(x, y|k_1, k_2) = \arctan\left\{\max\left(0, k_1 - \frac{k_2 \cdot y}{W}\right) \cdot \cos\left(\frac{\pi x}{H}\right)\right\}, \tag{3}$$

where the parameters k_1 and k_2 control the curvature changes and the lifting level of bottom lines in the residual orientation field, respectively, and H and W are the height and width of the fingerprint image. The parameters, k_1 and k_2, are estimated by solving the following optimization problem:

$$\arg\min_{k_1,k_2} \sum_{(x,y)\in\Omega} \sin^2(\theta(x, y|k_1, k_2) - \theta^R(x, y)). \tag{4}$$

In this way, the residual orientation field in a fingerprint image is represented by a two-tuple $\nu = \{k_1, k_2\}$.

The distribution of ν for each of the five major fingerprint types is modeled as a mixture of Gaussians, i.e.,

$$f_\nu(\nu|\Theta) = \sum_{l=1}^{L} \tau_l \cdot G_l(\nu|\mu_l, \Sigma_l), \tag{5}$$

where Θ is the set of parameters which includes: L, the number of mixture components, τ_l, the probability of the l^{th} component, and the mean μ_l and covariance Σ_l of the Gaussian density function $G_l(\nu|\mu_l, \Sigma_l)$. The parameters in the mixture models are estimated by applying the Expectation-Maximization method [13] to a set of training data. We trained the mixture models by using the "F" fingerprints in the NIST SD4 database [1] which have an exclusively assigned fingerprint type, are of good quality with NFIQ [21] value from 1 to 3, have all singular points visible, and are free from rotation (i.e., vertically aligned).

To sample the orientation field for a given fingerprint type, the residual orientation field is generated by realizing the distribution model of $\{k_1, k_2\}$ associated with the given fingerprint type and substituting them into the cosine peripheral model in Eq. (3). This residual orientation field is then combined with the singular orientation field generated by the sampled singular points according to Eqs. (1) and (2) to generate the sampled orientation field.

[1]Here, we assume that the fingerprint is centered in the fingerprint image and vertically aligned.

3.1.3 Sampling Minutiae

Several distribution models [11, 17, 24] have been proposed for the minutiae in fingerprints, among which Chen and Jain's model [11] is state-of-the-art. In [11], Chen and Jain showed that minutiae in different types of fingerprints are characterized by different distributions and proposed models of minutiae distributions for the five major fingerprint types. The spatial locations and directions of minutiae are modeled as mixture of Gaussians and mixture of von-Mises distributions, respectively. In this paper, we employ the Gaussian mixture based spatial distribution models of minutiae proposed by Chen and Jain [11].

To sample minutiae, the number of minutiae in the fingerprint is first randomly chosen according to the empirical minutia density (i.e., the number of minutiae in a unit area) and the fingerprint image size. The minutia locations are then obtained by realizing the spatial distribution model in [11], and each minutia is randomly assigned either as a ridge ending or as a ridge bifurcation with equal probability. The minutia directions are finally determined according to their types (ending or bifurcation) and the ridge orientations at their locations: the direction of ridge bifurcation is the same as the local ridge orientation and the direction of ridge ending is computed by adding 180 degrees to the local ridge orientation.

3.2. Generating Master Fingerprints

We employ the AM-FM based method in [12] to reconstruct a master fingerprint from a set of sampled features, because it generates a relatively small number of spurious/missing minutiae in the synthetic fingerprints. In the AM-FM method, a fingerprint is represented as

$$I(x, y) = 255 \times \frac{1 + \cos(\Psi(x, y))}{2}, \tag{6}$$

where $\Psi(x, y)$ is the composite phase and can be uniquely decomposed into continuous phase, $\Psi_c(x, y)$, and spiral phase, $\Psi_s(x, y)$, i.e.,

$$\Psi(x, y) = \Psi_c(x, y) + \Psi_s(x, y). \tag{7}$$

The spiral phase is determined by the N_m minutiae in the fingerprint:

$$\Psi_s(x, y) = \sum_{n=1}^{N_m} p_n \arctan\left(\frac{y - y_n}{x - x_n}\right), \tag{8}$$

where (x_n, y_n) denotes the coordinates of the n^{th} minutia and $p_n \in \{-1, 1\}$ denotes its type (i.e., ridge ending or bifurcation).

To obtain the continuous phase, the gradient of the composite phase, $G(x, y)$, is first computed from the orientation field. $G(x, y)$ is called the instantaneous frequency,

whose direction is normal to the local ridge orientation and whose magnitude is proportional to the local ridge frequency, i.e.,

$$G(x, y) = 2\pi f \exp(i(\theta_u(x, y) + \pi/2)), \quad (9)$$

where $\theta_u(x, y)$ is the unwrapped orientation field and f is the ridge frequency (in the experiments, we set $f = 0.12$).

Since the orientation field and singular points are given together with the minutiae, we directly unwrap the orientation field [12] and compute the gradient of the composite phase by Eq. (9). Once we have the gradient of the composite phase and take the derivative of the sprial phase in Eq. (8), the gradient of continuous phase is obtained by the difference between the gradients of the composite and the spiral phases. Then, the continuous phase can be reconstructed by

$$\Psi_c(x, y) = G_{cx}(x, y)x + G_{cy}(x, y)y + P(x, y), \quad (10)$$

where G_{cx} and G_{cy} are the $x-$ and $y-$ derivatives of continuous phase, and $P(x, y)$ is the phase offset. The phase offset is estimated by using the block-wise planar model based method [12]. With the reconstructed continuous phase and the spiral phase, the master fingerprint can be obtained according to Eqs. (6) and (7). Note that this master fingerprint is free from distortion and noise, and the features in the master fingerprint are known.

3.3. Generating Impressions From a Master Fingerprint

Multiple impressions from the same master fingerprint (i.e., genuine pairs of fingerprint images) are generated by distorting the master fingerprint (see Figs. 3(a) and 3(b)). Nonlinear plastic distortion [15] is applied followed by global rigid transformation (i.e., rotation and translation). The nonlinear plastic distortion model simulates the skin distortion (e.g., compression and stretching) observed in contact-based plain fingerprints [2]. The model divides a fingerprint into three regions: close-contact, transitional, and external regions. The close-contact (central) region is free from skin distortion, and it is defined as an ellipse by its center (c_x, c_y) and variances (s_x, s_y). The external region is characterized by a slight rigid transformation caused by dragging skin or moving finger. The rigid transformation is defined by a rotation β_N and translation (dx_N, dy_N). The transitional region is the region connecting the central and external regions. It is subject to a nonlinear distortion controlled by the skin plasticity parameter k — a higher k indicates a larger transitional region and thus a smoother transition between the central and external regions.

[2]Note that in this paper we focus on synthesizing plain fingerprint images collected by using contact-based sensors. To synthesize other types of fingerprint images, such as rolled or contact-less fingerprint images, different distortion models are required.

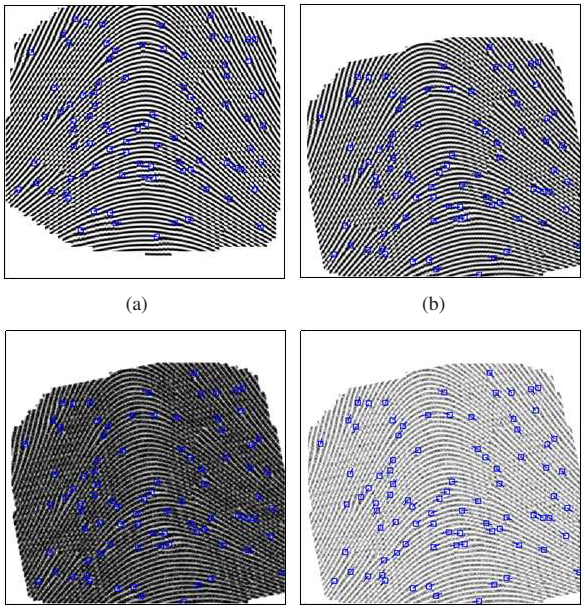


Figure 3. Multiple impressions of a master fingerprint are generated by applying nonlinear plastic distortion and global rigid transformation to the master fingerprint and rendering the fingerprint images. (a) A master fingerprint, (b) an impression obtained by distorting (a), and (c) and (d) are fingerprint images by rendering (b) as wet and dry finger, respectively.

The global rigid transformation is applied by first rotating the fingerprint with respect to the image center by an angle of β_R and then translating the fingerprint by dx_R and dy_R along the $x-$ and $y-$ axes, respectively. Let us denote the nonlinear plastic distortion and the global rigid transformation as T_N and T_R, respectively. The correspondences between the pixels $\tilde{v}$ in the distorted image and the pixels v in the original fingerprint can be represented by the following mapping

$$v \mapsto \tilde{v} = T_R T_N(v). \quad (11)$$

Some pixels in the distorted image might not have corresponding pixels in the original fingerprint. Values of these pixels are estimated by interpolation of their neighboring pixels. The range of the parameter values involved in the distortion is empirically specified.

The locations of singular points and minutiae in the distorted images are computed by applying the mapping in Eq. (11) to their original locations. To determine the local ridge orientation and minutia directions after distortion, we define a unit vector at each local block or at the location of each minutia, which points along the local ridge orientation or the minutia direction. The initial and terminal points of the unit vector are then distorted according to Eq.

(11). Finally, the local ridge orientation and the minutia direction after distortion are computed as the direction of the distorted unit vector. The correspondences between the minutiae in different impressions are traced while distorting the master fingerprint. The minutiae are removed when they are outside of the fingerprint image.

3.4. Rendering Fingerprint Images

The fingerprint impressions of a master fingerprint obtained in Section 3.3 do not contain any noise and thus do not look very realistic. Therefore, we further render the impressions by simulating finger dryness and adding noise. Similar to [9], we define five levels of finger dryness $\{-2, -1, 0, 1, 2\}$. Negative dryness level means that the fingers are wet and the fingerprint images are dilated to increase the ridge width. On the other hand, positive dryness level indicates dry fingers and the fingerprint images are eroded to make the ridges thinner. After dryness simulation, each of the ridge pixels is subject to a random perturbation according to a specified probability (i.e., noise level). Finally, a 3×3 average filter is applied to smooth the ridge pixels, resulting in the final fingerprint images (see Figs. 3(c) and 3(d)).

4. Experiments

4.1. Validation of Synthesized Fingerprints

Figure 4 shows example fingerprint images synthesized by SFinGe and the proposed method. The extracted minutiae in these fingerprints show that a large portion of the fingerprints synthesized by SFinGe might contain parallel ridges without any minutiae (e.g., see the bottom right area in Fig. 4(a)). This is quite rare in real fingerprints. Figure 5 shows the histograms of the number of minutiae (extracted by COTS1) in foreground blocks in the fingerprints in NIST SD4, 1,000 master fingerprints synthesized by SFinGe, and 1,000 master fingerprints synthesized by the proposed method (the probability of different fingerprint types follows their natural distribution, i.e., 3.7% for arch, 2.8% for tented arch, 33.8% for left loop, 31.7% for right loop, and 27.9% for whorl [23]). These results demonstrate that, by sampling minutiae from the statistical models that are learned from real fingerprints, the proposed method can generate fingerprints with more naturally distributed minutiae (see Fig. 4(b)).

We evaluated the degree of preservation of prespecified features on the 1,000 master fingerprints generated by the proposed method by comparing the minutiae extracted by the COTS matchers with the prespecified minutiae. The average numbers of true, missing, and spurious minutiae [3] are 41, 23, and 25 for COTS1, and 44, 20, and 36 for COTS2. Many true minutiae are retained in the synthesized fingerprints. However, there are also many missing and spurious minutiae. The relatively large number of missing and spurious minutiae might be caused by two reasons: (i) the statistical feature models and the feature sampling process are not accurate enough; (ii) the fingerprint reconstruction algorithm [12] we employ is initially designed for minutia template inversion, whose goal is to match the reconstructed fingerprints to the source fingerprints (hence, a large number of true minutiae are usually sufficient, even though there are missing and spurious minutiae).

(a) (b)

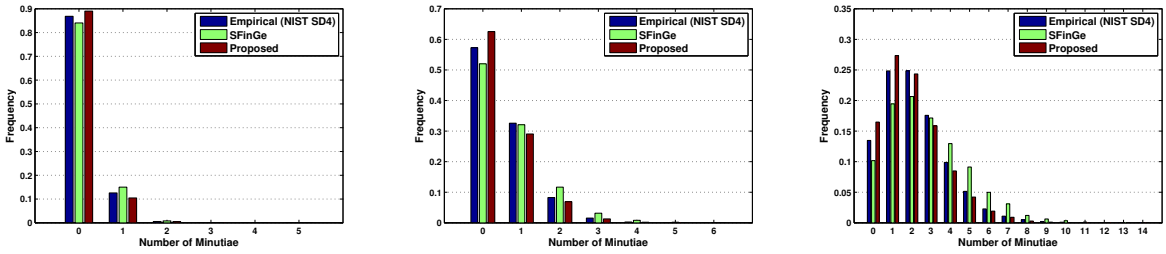
Figure 4. Example fingerprint images synthesized by (a) SFinGe and (b) the proposed method with minutiae extracted by COTS1. A large portion of SFinGe generated fingerprints might contain parallel ridges without any minutiae (see bottom right portion of (a)), which is quite rare in real fingerprints. Minutiae in fingerprints generated by the proposed method appear to follow a more natural distribution.

We compared the match score distributions obtained from synthesized fingerprint databases and those from real fingerprint databases. Five impressions were generated from each of the 1,000 master fingerprints to construct a synthesized fingerprint database. The NIST SD4 database (2,000 pairs of fingerprints) [1] was used to generate the match score distributions on real fingerprints. In the matching experiments, the first image of each finger was used as a reference and the remaining images were used as query. Figure 6 shows the obtained genuine and imposter score distributions of COTS2. Similar trends can be observed in the distributions for the synthesized fingerprint database and for real fingerprint databases.

4.2. Evaluation of COTS Fingerprint Matchers

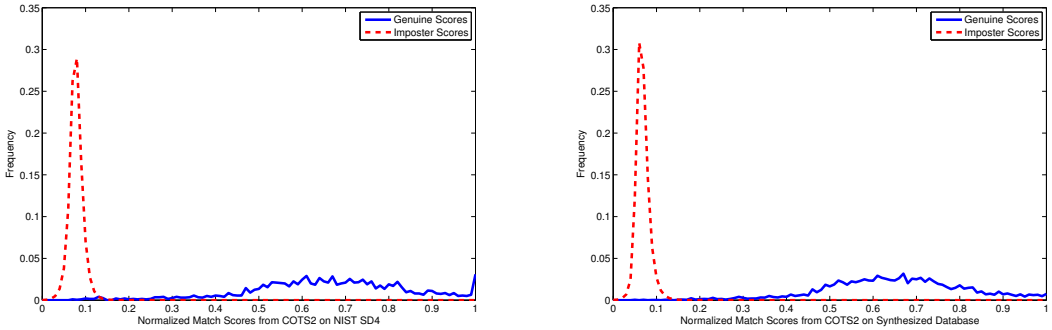
The evaluation of COTS fingerprint matchers using the synthesized fingerprint database was conducted in two

[3] A prespecified minutia is correctly detected if there exists an automatically extracted minutia whose distance to it is less than 10 pixels, otherwise it is called missing; an automatically extracted minutia is a spurious minutia if its distance to any of the prespecified minutiae is larger than 10 pixels, otherwise it is a true minutia



(a) (b) (c)

Figure 5. Histograms of the number of minutiae in (a) 16×16, (b) 32×32, and (c) 64×64 foreground blocks.



(a) (b)

Figure 6. Match score distributions of COTS2 on (a) the NIST SD4 and (b) the synthesized fingerprint databases. Scores are normalized to [0, 1] by min-max normalization.

scenarios: (i) using the prespecified minutiae as ground truth to generate standard templates for the fingerprint matchers, and (ii) using the fingerprint matchers to extract proprietary templates from the synthetic fingerprint images. The first scenario evaluates the minutia matching performance under the assumption of zero feature extraction error (ground truth features are used). The second scenario evaluates the overall fingerprint matching accuracy when proprietary templates are used.

The receiver operating characteristic (ROC) curves of COTS1 and COTS2 matchers in both scenarios are shown in Fig. 7, from which the following inferences can be made about the two matchers. These observations can serve as potential guidance for improving the matchers:

- COTS1 has a higher matching accuracy than COTS2 on the standard minutiae templates generated from the ground truth minutiae. This indicates that COTS1 has a better matching module for minutiae only input (i.e., only the location and direction of minutiae are available in fingerprint matching).
- COTS2 has a higher accuracy than COTS1 when their proprietary templates are used. This is potentially due to the following two reasons: (i) COTS2 has higher feature extraction accuracy, and (ii) COTS2 is more effective in utilizing various discriminative features (not merely minutiae) in fingerprints.
- COTS1 performs better using the ground truth minutiae than using its proprietary template, which again demonstrates its potentially low feature extraction accuracy and relatively low efficiency in exploiting non-minutiae features.

5. Conclusion and Future Work

A new method has been proposed for fingerprint image synthesis based on statistical feature models. Compared with the well-known method SFinGe, the proposed method provides more control on the features in synthesized fingerprints via sampling the features from statistical models and generating fingerprint images containing the sampled features. This additional ability enables us to conduct efficient evaluation and optimization of fingerprint recognition systems as demonstrated by the experiments with two COTS fingerprint matchers.

As this approach, to our knowledge, is the first attempt in synthesizing fingerprints based on statistical feature

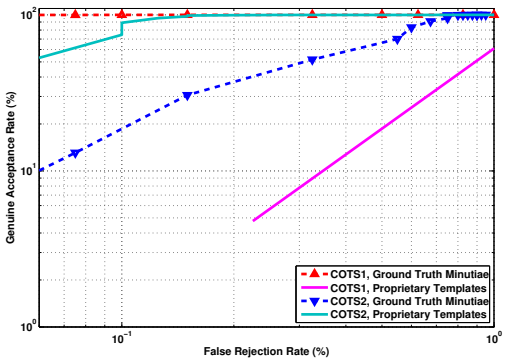


Figure 7. The receiver operating characteristic (ROC) curves of the two COTS matchers used here on the synthesized fingerprint image database (5,000 images of 1,000 fingers, 5 images per finger).

models, the proposed method is still limited in retaining the prespecified features in the synthesized fingerprint images. We are currently enhancing the proposed method along two directions. (i) Improving the statistical feature models and the sampling process. For example, correlation between different individual minutiae in a fingerprint is not adequately considered in the existing spatial distribution models of minutiae, and the dependency of minutia locations on the orientation field is not exploited in the current method. (ii) Enhancing the fingerprint image reconstruction algorithm to further reduce the number of missing and spurious minutiae.

Acknowledgments

This work is supported by the NIST 2011 Measurement Science and Engineering Research Grants Programs.

References

[1] NIST Special Database 4, NIST 8-Bit Gray Scale Images of Fingerprint Image Groups (FIGS). http://www.nist.gov/srd/nistsd4.htm.

[2] Fingerprint Vendor Technology Evaluation (FpVTE) 2003. http://www.nist.gov/itl/iad/ig/fpvte03.cfm.

[3] Fingerprint Vendor Technology Evaluation (FpVTE) 2012. http://www.nist.gov/itl/iad/ig/fpvte2012.cfm.

[4] IAFIS. http://www.fbi.gov/about-us/cjis/fingerprints_biometrics/iafis/iafis.

[5] Unique Identification Authority of India. http://uidai.gov.in.

[6] US-VISIT. http://www.dhs.gov/files/programs/usv.shtm.

[7] R. Cappelli. Use of synthetic data for evaluating the quality of minutia extraction algorithms. In *Proc. 2nd NIST Biometric Quality Workshop*, 2007.

[8] R. Cappelli, A. Lumini, D. Maio, and D. Maltoni. Fingerprint image reconstruction from standard templates. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 29(9):1489–1503, 2007.

[9] R. Cappelli, D. Maio, and D. Maltoni. Synthetic fingerprint database generation. In *Proc. 16th International Conference on Pattern Recognition (ICPR'02)*, pages 1–4, 2002.

[10] R. Cappelli and D. Maltoni. On the spatial distribution of fingerprint singularities. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 31(4):742–448, 2009.

[11] Y. Chen and A. K. Jain. Beyond minutiae: A fingerprint individuality model with pattern, ridge and pore features. In *Proc. 2nd International Conference on Biometrics*, pages 523–533, June 2009.

[12] J. Feng and A. K. Jain. Fingerprint reconstruction: From minutiae to phase. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 33(2):209–223, 2011.

[13] M. Figueiredo and A. K. Jain. Unsupervised learning of finite mixture models. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 24(3):381–396, 2002.

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

[15] D. Maltoni and R. Cappelli. Advances in fingerprint modeling. *Image and Vision Computing*, 27:258–268, 2009.

[16] D. Maltoni, D. Maio, A. K. Jain, and S. Prabhakar. *Handbook of Fingerprint Recognition (Second Edition)*. Springer-Verlag, 2009.

[17] S. Pankanti, S. Prabhakar, and A. K. Jain. On the individuality of fingerprints. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 24(8):1010–1025, 2002.

[18] P. J. Phillips, A. Martin, C. L. Wilson, and M. Przybocki. An introduction to evaluating biometric systems. *Computer*, 33(2):56–63, 2000.

[19] A. Ross, J. Shah, and A. K. Jain. From template to image: Reconstructing fingerprints from minutiae points. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 29(4):544–560, 2007.

[20] B. G. Sherlock and D. M. Monro. A model for interpreting fingerprint topology. *Pattern Recognition*, 26(7):1047–1055, 1993.

[21] E. Tabassi, C. Wilson, and C. Watson. Fingerprint Image Quality. NISTIR 7151, August 2004. http://fingerprint.nist.gov/NFIS/ir_7151.pdf.

[22] Y. Wang and J. Hu. Global ridge orientation modeling for partial fingerprint identification. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 33(1):72–87, 2011.

[23] C. L. Wilson, G. T. Candela, and C. I. Watson. Neural network fingerprint classification. *Journal of Artificial Neural Networks*, 1(2):203–228, 1994.

[24] Y. Zhu, S. C. Dass, and A. K. Jain. Statistical models for assessing the individuality of fingerprints. *IEEE Transactions on Information Forensics and Security*, 2(3):391–401, 2007.