Chapter 15

A Comparison of Classification- and Indexing-Based Approaches for Fingerprint Identification

Xuejun Tan, Bir Bhanu, and Rong Wang

15.1 INTRODUCTION

There are two kinds of biometric systems that use fingerprints for the personal identity: verification and identification. In a verification system, the input includes a query fingerprint and a known identity (ID), and the system verifies whether the ID is consistent with the input fingerprint. The output of a verification system is an answer of yes or no. In an identification system, the input only includes a query fingerprint, and the system tries to answer the following question: Are there any fingerprints in the database which resemble the query fingerprint? In this chapter, we are dealing with the identification problem. There are three kinds of approaches to solve the fingerprint identification problem:

1. The *first* approach is to repeat the verification procedure for each fingerprint in the database and select the best match. However, if the size of the database is large, this approach will be time-consuming and it is not practical for real-world applications [1].
2. The *second* approach involves fingerprint classification followed by verification. Traditional classification techniques attempt to classify fingerprints into five classes: right loop (R), left loop (L), whorl (W), arch (A), and tented

Biometrics: Theory, Methods, and Applications. Edited by Boulgouris, Plataniotis, and Micheli-Tzanakou

arch (T). The most widely used approaches for fingerprint classification are based on the number and relations of singular points (SPs) [2]. The problem with this kind of approach is that it is not easy to detect SPs and some fingerprints do not have SPs. Moreover, the uncertainty in the location of SPs is large, which has an undesired effect on the classification results. Based on fingerprint's orientation field [3], other classification approaches use multispace Karhunen–Loeve transform [4] and a combination of different classifiers [5] to improve the performance. The most important problem associated with the classification technique for identification is that the number of principal classes is small and the fingerprints are unevenly distributed (31.7%, 33.8%, 27.9%, 3.7%, and 2.9% for classes R, L, W, A, and T, respectively [6]). The classification approach does not narrow down the search enough in the database for efficient identification.

3. The *third* approach consists of fingerprint indexing followed by verification. Germain et al. [7] integrate indexing and verification in their approach, in which top hypothesis generated by indexing is considered as the final identification result. They use the triplets of minutiae in their identification procedure. The features they use are: the length of each side, the angles that the ridges make with respect to the x-axis of the reference frame, and the ridge count between each pair of vertices. Bhanu and Tan [8] present an indexing approach using novel features of minutiae triplets. They compare the performance of their approach with Germain et al. [7] and demonstrate the improvement in result over Germain et al. [7].

In this chapter, we compare the second and third approaches (see Figure 15.1) that use minutiae features for fingerprint identification. The contributions of this chapter are as follows: (a) It provides the comparison of classification- and indexing-based approaches in a single chapter; *some* of the material is scattered in various recent papers. All the experimental results for comparison are carried out on the entire NIST-4 fingerprint database [6]. (b) It presents a technique based on learned

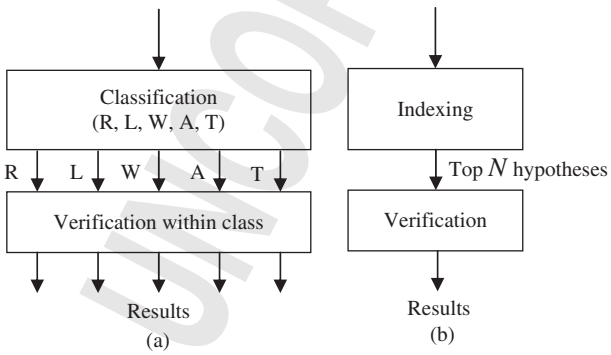

Figure 15.1. Block diagram of two different approaches to solve identification problem: (**a**) Classification followed by verification; (**b**) Indexing followed by verification.

masks for minutiae feature extraction and integrates newly developed classification [9] and indexing [8] techniques with the same verification algorithm that optimizes a criterion function. (c) It presents extensive comparisons between classification- and indexing-based techniques for identification.

15.2 TECHNICAL APPROACH

15.2.1 Minutiae Extraction Using Learned Feature Extraction Masks

There are many approaches in the literature for the minutiae extraction [10]. As compared to the other approaches, we extract minutiae using *learned masks* for all the results reported in this chapter. For each fingerprint, first the background is removed. Local orientation is computed in each local 16×16 block. The fingerprint is adaptively smoothed, binarized, and thinned using the local orientation information. Potential minutiae are found using crossing number [2]. Finally, learned feature extractor masks obtained during offline processing are adaptively applied to purify the potential minutiae.

15.2.1.1 Offline Learning of Feature Extraction Masks

A mask is a 2D filter that is concerned with detecting a minutia. Since a minutia can be an endpoint or a bifurcation, two masks are to be learned, one for each kind of feature. For simplicity, we use endpoints as the example to explain our learning approach. The mask for bifurcations is learned by following a similar process.

Figure 15.2 shows an ideal endpoint mask T that consists of two submasks, T_r (length L_r) and T_g (length L_g), which denote the mask for ridge and gap, respectively. For simplicity, we assume $L_r = L_g$. H and L are the height and the length of the mask T, and $L = L_r + L_g$. The values of each pixel in T_r and T_g are 1 and 0, respectively. Suppose (a) a ridge end E in a binary fingerprint is as ideal as the ideal mask T, (b) the local orientation at the ridge end is θ_l, (c) the correlations between the mask T and the ideal ridge end E with the orientation θ_l and $\theta_l + \pi$ are f_{θ_l} and $f_{\theta_l+\pi}$, respectively, and (d) the difference between f_{θ_l} and $f_{\theta_l+\pi}$ is Δ_{θ_l}, that is,

$$\Delta_{\theta_l} = f_{\theta_l} - f_{\theta_l+\pi} \tag{15.1}$$

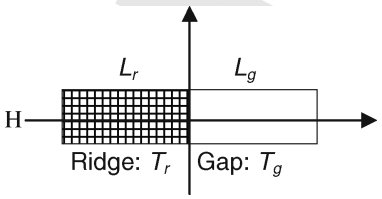

Figure 15.2. Illustration of an ideal endpoint mask T.

where $f_{\theta_l} = \sum_{(h,l)\in(T\cap E_{\theta_l})}\{T(h,l) \times E_{\theta_l}(h,l)\}$ and $E_{\theta_l}(h,l)$ is the ridge with orientation θ_l and the mask $T(h,l)$ is applied along the ridge.

Training Data. Suppose (a) examples of endpoints and bifurcations are obtained from M fingerprint images FI_k, where $k = 1,2,\ldots,M$; (b) in the kth fingerprint image FI_k, there are N_k feature locations $(x_{k,i}, y_{k,i})$, where $i = 1,2,3,\ldots,N_k$; (c) in the local area around $(x_{k,i}, y_{k,i})$, $I_{k,i}(m, n)$ is the gray-scale value at pixel (m, n) of the image FI_k, where $x_{k,i} - d_1 \le m \le x_{k,i} + d_1$, $y_{k,i} - d_2 \le n \le y_{k,i} + d_2$, d_1 and d_2 are constants; and (d) $G = \{(x_{k,i}, y_{k,i})\}$. Then, for each pixel in G, we carry out the following steps: (1) Estimate the local orientation $\theta_{k,i}$ at pixel $(x_{k,i}, y_{k,i})$ in the local area; (2) adaptively smooth $I_{k,i}(m, n)$ in the local area; (3) adaptively binarize $I_{k,i}(m, n)$ in the local area. The details of these steps are the same as those in the run-time minutiae extraction, which are discussed in Section 15.2.1.2.

Optimzation for Feature Extracton Masks Learning. Suppose (a) the mask is $T(h, l)$, where $1 \le h \le H$, $1 \le 1 \le L$, and $H = 2d_1 + 1$ and $L = 2d_2 + 1$; (b) $B_{k,i}(h, l)$ is the binary image of $I_{k,i}(m, n)$; and (c) $B^{\theta k,i}{}_{k,i}(h, l)$ is the rotated binary image of $B_{k,i}(h, l)$, rotation angle is $\theta_{k,i}$, which is the local orientation at pixel $(x_{k,i}, y_{k,i})$. According to Eq. (15.1), the objective of learning algorithm can be defined as [11]

$$\arg\max_{T}\left\{\sum_{k=1}^{M}\sum_{i=1}^{N_k}\sum_{h=1}^{H}\sum_{l=1}^{L}\left[T(h,l) \times Q_{k,i}(h,l)\right]\right\}, \tag{15.2}$$

where $Q_{k,i}(h,l) = B_{k,i}^{\theta_{k,i}}(h,l) - B_{k,i}^{\theta_{k,i}}(h, L-l)$. If we normalize the mask's energy to one—that is, $\sum_{h=1}^{H}\sum_{l=1}^{L} T^2(h,l) = 1$—we can solve the optimization problem with Lagrange's method. Let

$$q(h,l) = \sum_{k=1}^{M}\sum_{i=1}^{N_k} Q_{k,i}(h,l). \tag{15.3}$$

Then, the optimal solution for the mask is

$$T(h,l) = \frac{q(h,l)}{\sqrt{\sum_{h=1}^{H}\sum_{l=1}^{L} q^2(h,l)}}. \tag{15.4}$$

15.2.1.2 Run-Time Feature Extraction

The steps are summarized below:

Remove Background. Since a fingerprint image usually includes some background that does not have any useful information, it is desired to eliminate it. We split an fingerprint into 16×16 blocks and compute the mean μ_s of the gray-scale value of the pixels in each block. If the mean μ_s is greater than δ_s ($\delta_s = 150$), then the block belongs to background.

Compute Local Orientation. The input fingerprint is first smoothed using a 5×5 Gaussian filter of $\mu = 0$ and $\sigma = 1$. Sobel operators are then applied to the smoothed image to estimate the gradient magnitude. After that, the fingerprint is split into $m \times m$ blocks ($m = 16$) with 4 pixels overlap. For each block, the local orientation θ is obtained using a mean square error (*MSE*) criterion [12].

Adaptively Smooth Image. The fingerprint obtained after background removal is adaptively smoothed using guidance from the local orientation. The purpose of this processing is to eliminate most fine details such as islands and pores. We perform uniform smoothing along the local ridge orientation and Gaussian smoothing normal to it. The kernel of the smoothing filter is the normalized product of a 5×1 uniform kernel and a 1×3 Gaussian kernel of $\mu = 0$ and $\sigma = 1$. Possible orientations of the smoothing filters are discretized into 16 values. An appropriate filter is selected according to the local orientation and applied to each pixel.

Adaptively Binarize and Thin Image. The smoothed fingerprint is split into 16×16 blocks with 8 pixels overlap. For each block, we perform histogram equalization and binarize the block by a threshold. Thinned ridges are obtained by thinning the binary image.

Find Potential Minutiae. The initial potential minutiae are selected by crossing number (CN) at each pixel in the thinned image. Generally, initial potential minutiae are very noisy because of binarization, thinning, and error in estimating local orientation. Two simple criteria we use to filter the initial potential minutiae are as follows: (a) In a small local area, if an endpoint and a bifurcation are chosen as the initial potential minutiae, then ignore both of them; (b) in a small local area, if more than one endpoint or one bifurcation are chosen as the initial potential minutiae, then ignore all these minutiae. The result is a relatively good set of potential minutiae.

Adaptively Apply Feature Extraction Mask. At this step, the learned masks are adaptively applied to the potential minutia locations obtained above. Suppose the local orientation at a potential minutia location is θ, then we rotate the learned mask by θ and compute the difference in correlation using Eq. (15.1). In order to compensate for the error in estimating the local orientation, the correlations of the fingerprint are computed with five masks, which are the learned masks rotated with θ, $\theta \pm 5^{\circ}$ and $\theta \pm 10^{\circ}$. The largest of these values is taken as the correlation at this location. (a) Let the number of potential minutiae in an image be N_a; (b) let the correlation of potential minutiae be $V = \{v_i\}$, where $i = 1, 2, 3, \ldots, N_a$; (c) let the mean and the standard deviation of V be μ_v and σ_v, respectively; (d) let k_r be a constant (taken as 1 for all the experiments in this chapter) for adjusting the threshold of rejecting false minutiae. If $v_i > (\mu_v + k_r \times \sigma_v)$, we choose the ith potential minutia as a true minutia, otherwise it is a false minutia.

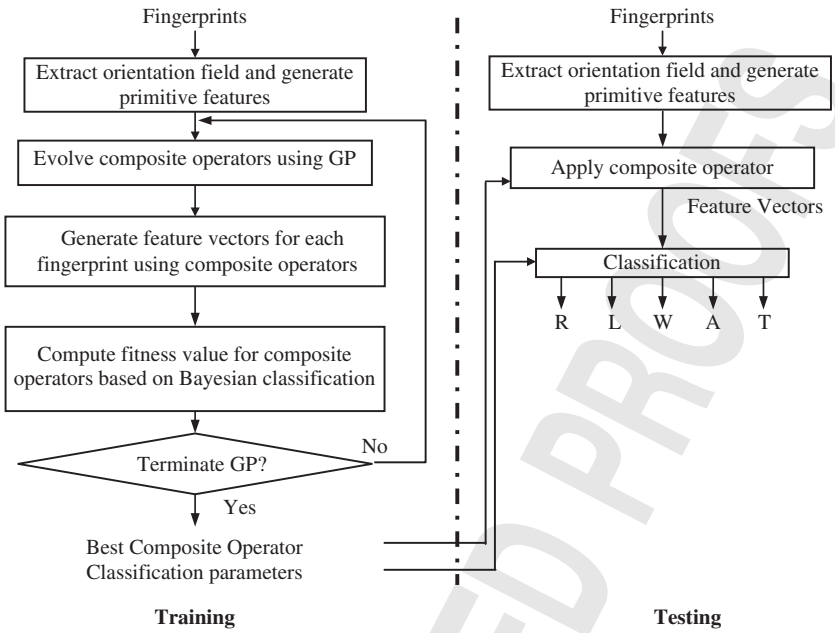

Figure 15.3. Block diagram of the classification approach.

15.2.2 Classification

Figure 15.3 shows the diagram of our classification approach using genetic programming (GP) [9, 3, 14]. During training, GP is used to generate *composite operators*, which can be viewed as a selected combination of primitive operations applied to the primitive features generated from the original orientation field. Features are computed wherever *feature generation operators* (see below) are used. These features are used to form a *feature vector* that represents a particular fingerprint image, and it is used for subsequent fingerprint classification. A Bayesian classifier is used for classification. Fitness value is computed based on the classification result and is used for evolving GP. During testing, composite operators are applied to generate feature vectors.

The individuals in our GP-based learning approach are composite operators represented by binary trees whose internal nodes represent the prespecified primitive operators and leaf nodes represent the primitive feature images. The major design considerations are explained in the following:

The Set of Terminals. For a fingerprint, we can estimate the orientation field [8]. The block size is $m = 32$ in our classification experiments, and $\theta \in [0, 180]$ and is measured in clockwise direction. The set of terminals used in this chapter are called primitive features, which are generated from the orientation field. Primitive features used in our experiments are: (1) original orientation image; (2) mean, standard deviation, min, max, and median images obtained by applying 3×3 and 5×5 filters on

orientation image; (3) edge images obtained by applying sobel filters along horizontal and vertical directions on orientation image; (4) binary image obtained by thresholding the orientation image with a threshold of 90; and (5) images obtained by applying sine and cosine operations on the orientation image. These 16 images are input to the composite operators. GP determines which operations are applied on them and how to combine the results.

The Set of Primitive Operators. A primitive operator takes one or two input images, performs a primitive operation on them, and outputs a resultant image. Suppose (1) A and B are images of the same size and c is a constant; and (2) for operators, which take two images as input, the operations are performed on the pixel-by-pixel basis. Currently, there are two kinds of primitive operators in our approach: *computation operators* and *feature generation operators*, which are described in references 9 and 13. For computation operators, the output is an image, that is generated by applying the corresponding operations on the input image. However, for feature generation operators, the output includes an image and a real number or vector. The output image is the same as the input image and is passed as the input image to the next node in the composite operator. The size of the feature vectors depends on the number and kind of the feature generation operators.

Generation of New Composite Operator. The initial population of the composite operators, represented as binary trees, is randomly generated. The search by GP is done by performing reproduction, crossover, and mutation operations. The reproduction operation used in our approach is the tournament selection. To perform crossover, two composite operators are selected based on their fitness values. One internal node in each of these two parents is randomly selected, and the two subtrees with these two nodes as root are exchanged between the parents. Once a composite operator is selected to perform mutation operation, an internal node of the binary tree representing this operator is randomly selected, and the subtree rooted at this node is replaced by another randomly generated binary tree. The resulting new binary tree replaces the old one in the population. We use *steady-state* GP [9] in our experiments.

The Fitness Measure. During training, at every generation for each composite operator proposed by GP, we estimate the probability distribution function (PDF) of the feature vectors for each class using all the available features. Suppose the feature vectors for each class have a normal distribution, $v_{i,j}$, where $i = 1, 2, 3, 4, 5$ and $j = 1, 2, \ldots, n_i$; n_i is the number of feature vectors in the training for class i, ω_i. Then, for each i, we estimate the mean μ_i and covariance matrix $\sum_i$ by all $v_{i,j}$, and the PDF of ω_i is obtained. A Bayesian classifier is used for classification. The percentage of correct classification (PCC) is taken as the fitness value of the composite operator: *Fitness value* $= \frac{n_c}{n_s} \times 100\%$, where n_c is the number of correctly classified fingerprints in the training set and n_s is the size of the training set.

Parameters and Termination. The key parameters are the population size, the number of generations, the crossover rate, and the mutation rate. The GP stops whenever it finishes the prespecified number of generations.

15.2.3 Indexing

Our approach for fingerprint indexing is based on the use of triplets of minutiae and ridge counts. However, for identification, the indexing and verification in our approach are separated. First, we apply indexing techniques to find the top N hypotheses, and then we apply a verification technique to select a hypothesis with the best match. The hypotheses are generated according to the number of corresponding triangles between two fingerprints. Top N hypotheses, sorted in a descending order of the number of potential corresponding triangles, are the indexing results.

For indexing, we use features based on minutiae triplets [8] in conjunction with the constraints on the transformation to eliminate the false corresponding triangles. Figure 15.4 shows the block diagram of our indexing approach. During the offline processing, the features of each template fingerprint are computed and used to construct the indexing space function $H(\alpha_{\min}, \alpha_{\text{med}}, \phi, \eta, \lambda, \chi, \xi)$ [8].

- Angles $\alpha_{\min}$ and α_{med}. α_{i}'s are the three angles in a triplet, where $i = 1, 2, 3$. $\alpha_{\min} = \min\{\alpha_i\}$, $\alpha_{\max} = \max\{\alpha_i\}$, $\alpha_{\text{med}} = 180^\circ - \alpha_{\min} - \alpha_{\max}$.
- Triangle handedness ϕ. Let $Z_i = x_i + jy_i$ be the complex number corresponding to the location (x_i, y_i) of point P_i, $i = 1, 2, 3$. Define $Z_{21} = Z_2 - Z_1$, $Z_{32} = Z_3 - Z_2$, and $Z_{13} = Z_1 - Z_3$. Let triangle handedness $\phi = \text{sign}(Z_{21} \times Z_{32})$. Points P_1, P_2, and P_3 are noncolinear points, so $\phi = 1$ or -1.

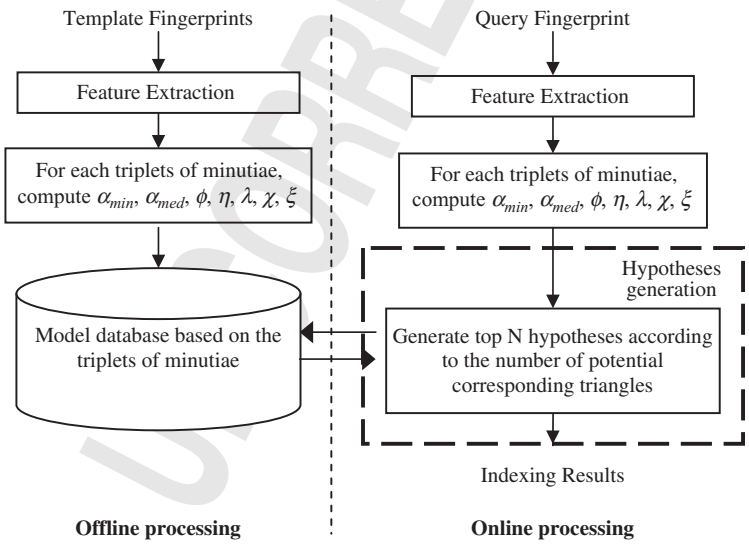

Figure 15.4. Block diagram of the indexing approach.

- Triangle direction η. We search the minutiae in the image from top to bottom and left to right. If a minutiae is the start point of the ridge, then $\nu = 1$; otherwise $\nu = 0$. Let $\eta = 4\nu_1 + 2\nu_2 + \nu_3$, where ν_i is ν value of point P_i, $i = 1, 2, 3$ and $0 \leq \eta \leq 7$.
- Maximum side λ. Let $\lambda = \max\{L_i\}$, where $L_1 = |Z_{21}|$, $L_2 = |Z_{32}|$, and $L_3 = |Z_{13}|$.
- Minutiae density χ. In a local area (32×32 pixels) centered at the minutiae P_i. If there exists χ_i minutiae, then the minutiae density for P_i is χ_i. Minutiae density χ is a 3D vector consisting of all χ_i.
- Ridge counts ξ. Let ξ_1, ξ_2, and ξ_3 be the ridge counts of sides P_1P_2, P_2P_3, and P_3P_1, respectively. Then, ξ is a 3D vector consisting of all ξ_i.

During the online processing, we compute the features for the query fingerprint and use them to search the indexing space $H(\alpha_{\min}, \alpha_{\text{med}}, \phi, \eta, \lambda, \chi, \xi)$. If the feature values of two triangles, which are from two different fingerprints, are within some error tolerance, then they are potential corresponding triangles. The criteria are: $|\alpha'_{\min} - \alpha''_{\min}| \leq T_{\alpha_{\min}}, |\alpha'_{\text{med}} - \alpha''_{\text{med}}| \leq T_{\alpha_{\text{med}}}, \phi' = \phi'', \eta' = \eta'', |\lambda' - \lambda''| \leq T_\lambda$, $|\chi'_i - \chi''_i| \leq T_\chi$, $|\xi'_i - \xi''_i| \leq T_\xi$, $i = 1, 2, 3$, where $(\alpha'_{\min}, \alpha'_{\text{med}}, \phi', \eta', \lambda', \chi'_i, \xi'_i)$ and $(\alpha''_{\min}, \alpha''_{\text{med}}, \phi'', \eta'', \lambda'', \chi''_i, \xi''_i)$ are the local properties of the triangle in different fingerprints; $T_{\alpha_{\min}}$, $T_{\alpha_{\text{med}}}$, T_λ, T_χ, and T_ζ are thresholds to deal with the local distortions.

15.2.4 Verification

Verification follows classification and indexing. It consists of the following two steps: (a) Use local information to estimate transformation between potential corresponding triangles and (b) use global information to eliminate false corresponding triangles and compute matching score. For indexing, verification is simple, since after indexing, for each hypothesis, we know the potential corresponding triangles and we may use this information in the verification directly. However, for classification, we only know the class information. So, we have to find the potential corresponding triangles between the query fingerprint and each template fingerprint that belongs to the same class.

Step 1. Estimate Transformation Between Potential Corresponding Triangles. Suppose the sets of minutiae in the template and the query fingerprints are $\{(t_{n,1}, t_{n,2})\}$ and $\{(q_{m,1}, q_{m,2}\})$ respectively, where $n = 1, 2, 3, \ldots, N, m = 1, 2, 3, \ldots, M$. The number of minutiae in the template and the query fingerprints are N and M, respectively. Let Δ_t and Δ_q be two potential corresponding triangles in the template and the query fingerprints, respectively. The coordinates of the vertices of Δ_t and Δ_q are $(x_{i,1}, x_{i,2})$ and $(y_{i,1}, y_{i,2})$, respectively, and $i = 1, 2, 3$. Suppose $X_i = [\, x_{i,1} \quad x_{i,2} \,]^T$, $Y_i = [\, y_{i,1} \quad y_{i,2} \,]^T$ and that the transformation $Y_i = F(X_i)$ can be expressed as

$$Y_i = s \cdot R \cdot X_i + T, \tag{15.5}$$

where s is the scaling factor, R is the rotation matrix with θ as the angle of rotation in counter clockwise direction between two fingerprints, and $T = [\, t_1 \quad t_2 \,]^T$ is the vector of translation.

There are two possible approaches: Least squares minimization (LSM) over all hypothesized triangles correspondences or over each of the triangle pair. We prefer the second alternative since it may allow better distortion tolerance on different parts of the fingerprint. We estimate the transformation parameters by minimizing error ε^2, which is the sum of the squared distances between the transformed template points and their corresponding query points. That is,

$$\text{error} = \arg\min_{(\hat{S},\hat{R},\hat{T})} \{\varepsilon^2\} \tag{15.6}$$

where $\varepsilon^2 = \sum_{i=1}^{3} ||Y_i - (\hat{s} \cdot \hat{R} \cdot X_i + \hat{T})||^2$ and $\|V\|$ is the L_2 norm of vector V. The solution of Eq. (15.6) is

$$\hat{\theta} = \arctan\left(\frac{B}{A}\right), \qquad \hat{s} = \frac{\sum_{i=1}^{3}\{(X_i - \overline{X})'\hat{R}'(Y_i - \overline{Y})\}}{\sum_{i=1}^{3}\{(X_i - \overline{X})'(Y_i - \overline{Y})\}}, \quad \hat{T} = \overline{Y} - \hat{s}\cdot\hat{R}\cdot\overline{X},$$

where

$$A = \sum_{i=1}^{3}\{(\overline{x}_1 - x_{i,1})(y_{i,1} - \overline{y}_1) + (\overline{x}_2 - x_{i,2})(y_{i,2} - \overline{y}_2)\},$$

$$B = \sum_{i=1}^{3}\{(\overline{x}_1 - x_{i,1})(y_{i,2} - \overline{y}_2) - (\overline{x}_2 - x_{i,2})(y_{i,1} - \overline{y}_1),\}$$

$$\overline{X} = \begin{bmatrix}\overline{x}_1\\ \overline{x}_2\end{bmatrix} = \sum_{i=1}^{3}\overline{X}_i, \quad \overline{Y} = \begin{bmatrix}\overline{y}_1\\ \overline{y}_2\end{bmatrix} = \sum_{i=1}^{3}\overline{Y}_i, \quad \hat{R} = \begin{bmatrix}\cos\hat{\theta} & -\sin\hat{\theta}\\ \sin\hat{\theta} & \cos\hat{\theta}\end{bmatrix}, \quad \hat{T} = \begin{bmatrix}\hat{t}_1\\ \hat{t}_2\end{bmatrix}.$$

If $\hat{s}$, $\hat{\theta}$, $\hat{t}_1$, and $\hat{t}_2$ are within limits, then we take them as the parameters of the transformation between two potential corresponding triangles Δ_t and Δ_q. Otherwise, they are false correspondences.

Step 2. Eliminate False Corresponding Triangles and Compute Match Score. Based on the above transformation $\hat{F}$ $(\hat{s}, \hat{\theta}, \hat{t}_1, \hat{t}_2)$, $\forall j$, $j = 1, 2, 3, \ldots, N$, we compute:

$$d = \arg\min_{k}\left\{\left|\hat{F}\left(\begin{bmatrix}t_{j,1}\\ t_{j,2}\end{bmatrix}\right) - \begin{bmatrix}q_{k,1}\\ q_{k,2}\end{bmatrix}\right|\right\}.$$

If d is less than a threshold T_d, then we define the points $[t_j,_1, t_{j,2}]'$ and $[\, q_{k,1}, q_{k,2}]'$ are corresponding points. If the number of corresponding points based on $\hat{F}(\hat{s}, \hat{\theta}, \hat{t}_1, \hat{t}_2)$ is greater than a threshold T_n, then we define Δ_t and Δ_q as the *corresponding triangles* between the template and the query fingerprints. The final identification score is the number of corresponding triangles between the query and template fingerprints.

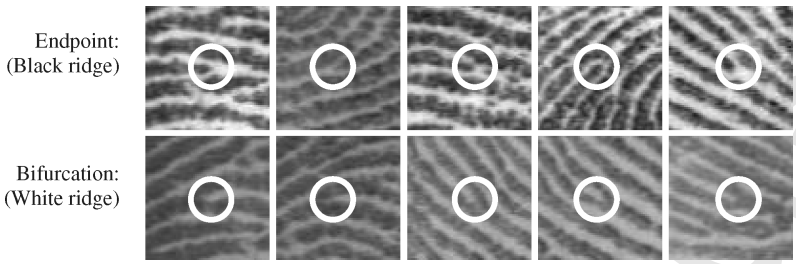

Figure 15.5. Examples of training data: Endpoint (**first row**) and bifurcation (**second row**).

15.3 EXPERIMENTAL RESULTS

NIST Special Database 4 (NIST-4) [6] with 2000 pairs of fingerprints is used in our experiments, where each pair is a different impression of the same finger. The size of the fingerprint images is 480×512 pixels with a resolution of 500 DPI. The fingerprint is coded as an *f* or *s* followed by six numbers, which means the fingerprint image is the first or second impression of certain finger.

15.3.1 Learning Feature Extraction Mask

Training data are manually obtained from 30 fingerprints based on the quality and the location of the minutiae. There are 85 endpoints and 86 bifurcations that are obtained from these 30 images. Figure 15.5 shows five examples of the training data for endpoints and bifurcations (note that each image contains at least one minutia). The masks for endpoint and bifurcation are learned from these binarized examples. Figure 15.6 shows the learned masks that are used to extract minutiae. Note that in order to show the structure of the masks clearly, the masks are normalized such that the minimum and maximum values map to 0 and 1, respectively. Figure 15.7 shows the learned masks superimposed on the examples in Figure 15.5.

Evaluation by Goodness Value. Suppose $M_e = \{e_i,\ i = 1, 2, 3, \ldots, n\}$ is the set of n minutiae extracted by a feature extraction algorithm and $M_g = \{g_j, j = 1, 2, 3, \ldots, m\}$ is the set of m minutiae extracted by an expert in a fingerprint. We define the following terms: (1) *Matched minutiae:* If minutia e_i is located in an uncertainty

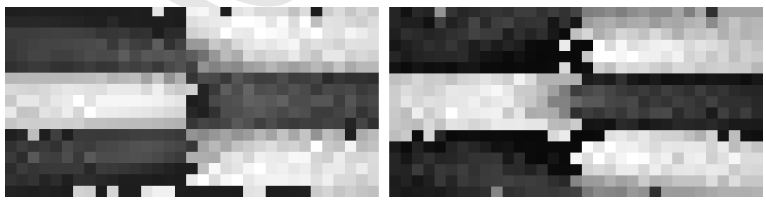

Figure 15.6. Learned feature extraction masks: Endpoint (**left**) and bifurcation (**right**).

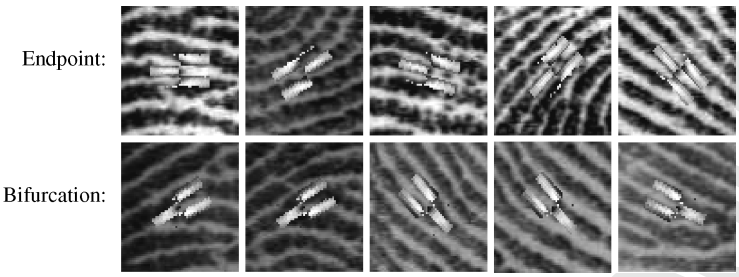

Figure 15.7. Learned feature extraction masks superimposed on examples in Figure 15.5.

region centered around minutia g_j, e_i and g_j are matched minutiae. (2) *Occluded minutia:* If minutia g_j is not in an uncertainty region of any minutia e_i, then g_j is an occluded minutia. (3) *Clutter minutia:* If e_i is not in an uncertainty region of any minutia g_j, then e_i is a clutter minutia.

In our experiments, the size of the uncertainty region is 8×8. goodness value (GV) of extracted feature is defined as:

$$\mathrm{GV} = \frac{n_m}{n_m + n_o + n_c},$$

where n_m, n_o, and n_c are the number of matched, occluded, and clutter minutiae, respectively. We choose 400 pairs of images from the first 1000 pairs of images in NIST-4. These images are chosen visually based on the size of overlapped areas between two images, the number of scars, translation, rotation and scale between images. Figure 15.8 shows goodness value of 15 test fingerprints images (from NIST-4 database). From this figure, we find that the learned masks work better than the fixed masks described in Bhanu et al. [15]. For example, mean of GV on these 15 images is 0.66 for learned masks, and 0.57 for fixed masks, which amounts to an improvement of 15.7%.

Evaluation by Indexing Performance. A query fingerprint, which has a corresponding fingerprint in the database, is said to be correctly indexed if it has enough corresponding triangles in the model database and the correct corresponding fingerprint appears in a short list of hypotheses obtained by the indexing approach. We define correct index power (CIP) as

$$\mathrm{CIP} = \frac{N_{ci}}{N_d} \times 100\%,$$

where N_{ci} is the number of correctly indexed fingerprints and N_d is the number of images in the database. Figure 15.9 shows the comparison of CIP for fixed and learned masks. We observe that the performance of the learned masks is better than that of the fixed masks. CIP for the top 1 hypothesis increases by 2.8%, and by 6.5% and 5.2% when we consider the top 5 and top 10 hypotheses, respectively. Using the fixed masks, the CIP reaches 100% only when we consider the top 26 hypotheses. For learned masks, however, we only need to consider top 10 hypotheses.

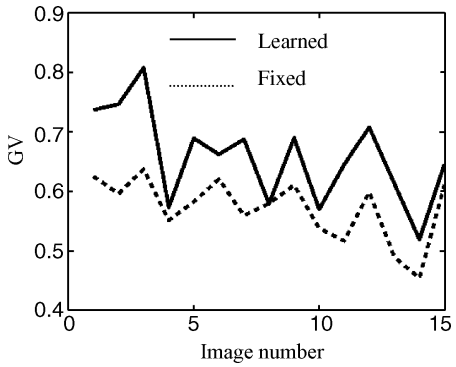

Figure 15.8. Goodness value of test fingerprints.

15.3.2 Classification Results

We use the first 1000 pairs of fingerprints for training. In order to reduce the effect of overfitting, we use only the first 500 pairs to estimate the parameters for each class and use the entire training set to validate the training results. Since we want to compare the results of classification and indexing, we only test the second impression of the second 1000 pairs of fingerprints. The first impressions of the second 1000 pairs of fingerprints are used as templates in verification. The parameters in our experiments are: maximum size of composite operator 150, population size 100, mutation rate 0.05, crossover rate 0.6, and number of generation 100.

We performed the experiments 10 times and took the best result as the learned composite operator. Table 15.1 shows the confusion matrix of our testing results of the second 1000 pairs of fingerprint in NIST-4. The images where tented arch is confused with arch are s1037_06, s1299_07, s1486_03, s1711_02, s1745_09, and s1759_09. The images where arch is confused with tented arch are s1568_08, s1948_05, s1956_10, and s1998_07. Figure 15.10 shows these 10 images where tented arch and arch are confused. Note that because of bad quality, the ground truths of some fingerprints

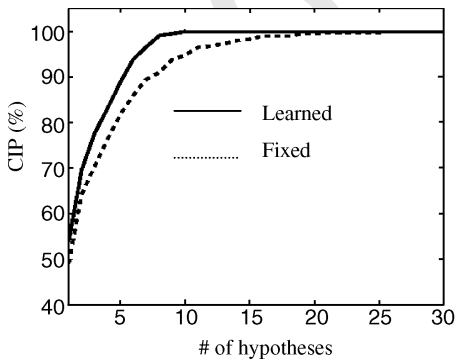

Figure 15.9. Comparison of experimental results.

Table 15.1. Confusion Matrix for Five-Class Classifications

	Assigned Class				
True Class	R	L	W	A	T
R	**180**	5	1	1	14
L	1	**188**	3	3	2
W	6	6	**187**	0	1
A	2	1	0	**208**	4
T	5	10	2	6	**172**

provided by NIST-4 contain two classes, for example, the ground-truth labels of f0008_10 include class T and L. As other researchers did in their experiments, we only use the first ground-truth label to estimate the parameters of the classifier. However, in testing, we use all the ground truth labels and consider a test as correctly classified if the output of the system matches to one of the ground truths. However, if the output of the system does not match any one of them, then we consider it as two incorrect classifications and each of them has an entry in the confusion matrix. Note that some published research work, such as reference 4, only has one entry in the confusion matrix when the input fingerprint has two ground truths and the classification result is incorrect, which inevitably reduces the error rate. Based on the confusion matrix in Table 15.1, the PCC is 92.8% for five-class classification. Considering that we have not rejected any fingerprints from NIST-4, our classification results are excellent [13].

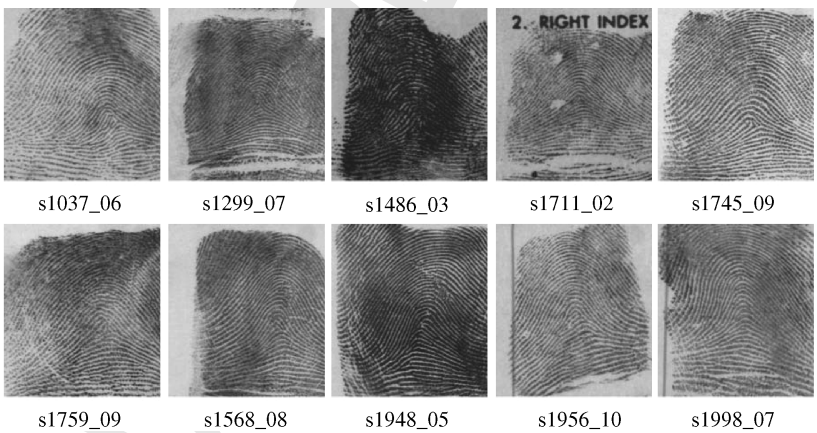

Figure 15.10. NIST-4 database images where tented arch and arch are confused. For the first six images (from left to right and top to bottom), the ground-truth label is tented arch and they are classified as arch. For the last four images, the ground truth is arch and they are classified as tented arch.

15.3.3 Indexing Results

In order to compare the results between indexing and classification, we only do indexing experiments on the second impressions of the second 1000 pairs of fingerprints. The parameters used in our experiments are: $T_{\alpha_{\min}} = 2^\circ$, $T_{\alpha_{\text{med}}} = 2^\circ$, $T_\lambda = 20$, $T_\chi = 2$, $T_\zeta = 2$. Figure 15.11 shows the correct indexing power (CIP). We observe that CIP increases as p, the percentage of the database searched, increases. The CIP are 83.3%, 88.1%, 91.1%, and 92.6%, and p are 5%, 10%, 15%, and 20%, respectively. As p reached about 60%, the relation between CIP and p becomes linear.

15.3.4 Identification Results

For classification, since the number of classes in fingerprint is small, we have to check more hypotheses in verification. For example, the classification result of our approach is one of the best results reported in published papers, however, we can only classify fingerprints into five classes. Since each class is uniformly distributed in NIST-4, after classification, about 200 hypotheses need to be considered in verification. And, this number cannot be tuned. As for indexing, since CIP varies according to the size of the search space, we have different performances of identification by indexing approach, depending on the percentage of the database that is searched. Conceptually, each fingerprint as a query is verified against all the stored fingerprint templates. That is 1,000,000 verifications. Among them, 999,000 verifications are estimating false acceptance rate (FAR) and 1000 verifications are for estimating genuine acceptance rate (GAR). The receiver operating characteristic (ROC) curve is defined as the plot of GAR against FAR. Based on different CIP, we can have different ROCs for identification results for the indexing-based approach and only one ROC for the classification-based approach. The parameters used in the verification step are: threshold to constrain scaling factor $\hat{s}$, $0.85 < \hat{s} < 1.15$; threshold to constrain rotation angle $\hat{\theta}$, $-30^\circ < \hat{\theta} < 30^\circ$; thresholds to constrain translations $\hat{t}_1$ and $\hat{t}_2$, $|\hat{t}_1| < 150$ and

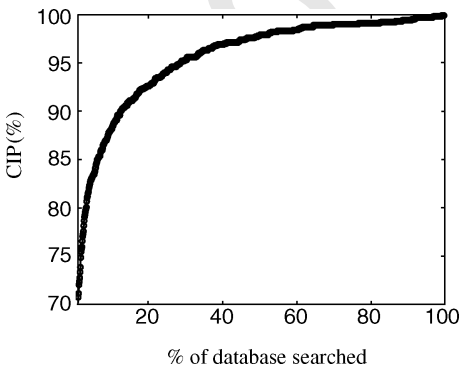

Figure 15.11. Indexing performance.

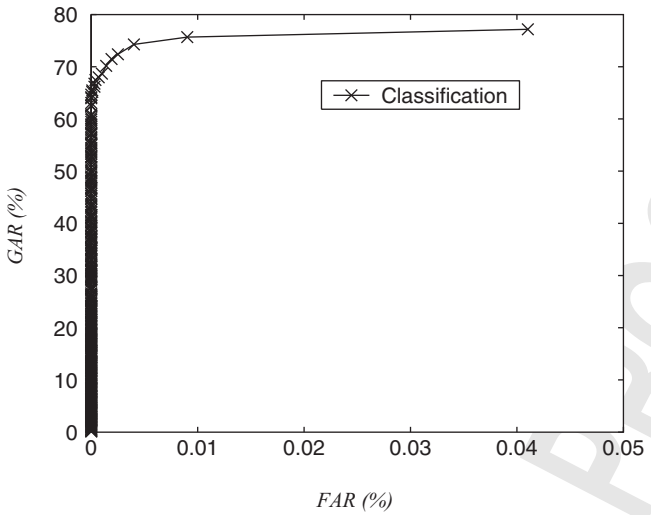

Figure 15.12. Identification results using classification based approach.

$|\hat{t}_2| < 100$; threshold to find the corresponding points, $T_d = 12$; threshold to find the corresponding triangles, $T_n = 8$.

Figures 15.12 and 15.13 show identification results based on classification and indexing, respectively. Note that GAR cannot reach 100.0%. One important reason is that bad-quality images do not provide enough similarity information to be used in verification, and the NIST-4 database is a very difficult database. Using the

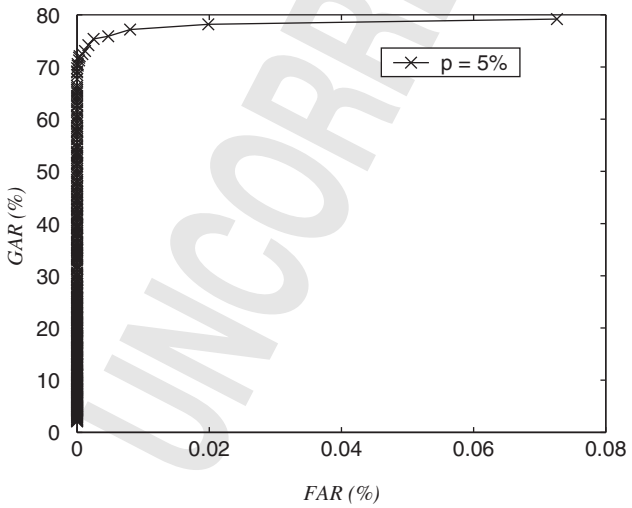

Figure 15.13. Identification results using indexing based approach.

classification-based approach, GAR is 77.2% when FAR is 4.1×10^{-2}%, while using the indexing-based approach with $p = 5$%, GAR is 77.2% and FAR is 8.0×10^{-3}%. It shows that in order to achieve similar GAR in identification, we only need to search 5% of the database by indexing-based approach for identification, while classification-based approach for identification may need to search 20% of the entire search space. FAR for indexing-based approach is much less than that for the classification-based approach. The classes R, L, W, A, and T are uniformly distributed in NIST-4. However, in nature, the frequencies of their occurrence are 31.7%, 33.8%, 27.9%, 3.7%, and 2.9%, respectively. So, using the classification-based approach the search space that needs to be searched will be more than 30.0%, since there are fewer fingerprints that belong to A and T classes in nature than to other classes.

15.4 CONCLUSIONS

In this chapter, we compared the performance of two approaches for identification. One is the traditional approach that first classifies a fingerprint into one of the five classes (R, L, W, A, T) and then performs verification. The alternative approach is based on indexing followed by verification. Using state of the art highly competitive approaches for classification, indexing, and verification, we compared the performance of the two approaches for identification using the NIST-4 fingerprint database. We found that the indexing technique performs better considering the size of search space (5% versus 20%) that needs to be examined. Also, for the same GAR (77.2%) the FAR performance (8.0×10^{-3}% versus 4.1×10^{-2}%) of indexing-based approach is lower. Thus, the indexing based approach provides a potential alternative to the traditional classification-based approach commonly used for fingerprint identification. Also it is possible to use the indexing approach within each of the classes after the classification has been done. This will expedite the identification performance of a classification-based approach.

REFERENCES

1. X. Tan, B. Bhanu, and Y. Lin, Fingerprint identification: classification vs. Indexing, in *IEEE International Conference on Advanced Video and Signal-Based Surveillance*, Miami, FL, July 21–22, 2003, pp. 151–156.
2. K. Karu and A. K. Jain, Fingerprint classification, *Pattern Recognit.* **29**(3):389–404, 1996.
3. R. Cappelli, A. Lumini, D. Maio, and D. Maltoni, Fingerprint classification by directional image partitioning, *IEEE Trans. Pattern Anal. Mach. Intell.* **21**(5):402–421, 1999.
4. R. Cappelli, D. Maio, and D. Maltoni, Fingerprint classification based on multi-space KL, in *Proceedings of the Workshop on Automatic identification Advances Technologies*, 1999, pp. 117–120.
5. Y. Yao, G. L. Marcialis, M. Pontil, P. Frasconi, and F. Roli, Combining flat and structured representations for fingerprint classification with recursive neural networks and support vector machines, *Pattern Recognit.* **36**(2):397–406, 2003.
6. C. I. Watson and C. L. Wilson, NIST special database 4, fingerprint database, U.S. National Institute of Standards and Technology, 1992.
7. R. S. Germain, A. Califano, and S. Colville, Fingerprint matching using transformation parameter clustering, *IEEE Comput. Sci. and Eng.* **4**(4):42–49, 1997.

8. B. Bhanu and X. Tan, Fingerprint indexing based on novel features of minutiae triplets, *IEEE Trans. Pattern Anal. Mach. Intell.* **25**(5):616–622, 2003.
9. X. Tan, B. Bhanu and, Y. Lin, Learning composite operators for fingerprint classification, *Proceeding of the International Conference on Audio- and Video-Based Biometric Person Authentication*, June 2003, pp. 318–326.
10. D. Maltoni, D. Maio, A. K. Jain, and S. Prabhakar, *Handbook of Fingerprint Recognition*, Springer, Berlin, 2003.
11. B. Bhanu and X. Tan, Learned templates for feature extraction in fingerprint images, *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.* **2**:591–596, 2001.
12. A. M. Bazen and S. H. Gerez, Systematic methods for the computation of the directional fields and singular points of fingerprints, *IEEE Trans. Pattern Anal. Mach. Intell.* **24**(7):905–919, 2002.
13. X. Tan, B. Bhanu, and Y. Lin, Fingerprint classification based on learned features, *IEEE Trans. Systems, Man Cybern. Part C* (Special Issue on Biometrics) **35**(3):287–300, 2005.
14. B. Bhanu, Y. Lin, and K. Krawiec, *Evolutionary Synthesis of Pattern Recognition Systems*, Monograph in Computer Science, Springer, Berlin, 2006.
15. B. Bhanu, M. Boshra, and X. Tan, Logical templates for feature extraction in fingerprint images, *Proc. Int. Conf. Pattern Recognit.* **3**:850–854, 2000.
16. S. Prabhakar, A. K. Jain, and S. Pankanti, Learning fingerprint minutiae location and type, *Pattern Recognit.* **36**(8):1847–1857, 2003.