Short Papers

Online Fingerprint Template Improvement

Xudong Jiang and Wee Ser, *Senior Member*, *IEEE*

Abstract—This work proposes a technique that improves fingerprint templates by merging and averaging minutiae of multiple fingerprints. The weighted averaging scheme enables the template to change gradually with time in line with changes of the skin and imaging conditions. The recursive nature of the algorithm greatly reduces the storage and computation requirements of this technique. As a result, the proposed template improvement procedure can be performed online during the fingerprint verification process. Extensive experimental studies demonstrate the feasibility of the proposed algorithm.

Index Terms—Fingerprint verification, minutia set, template improvement, multiple fingerprints.

1 Introduction

An automatic fingerprint verification system matches fingerprint inputs with prestored fingerprint templates, each of which consists of a set of features extracted from a fingerprint image. Since the most reliable feature for fingerprint matching is the minutia, most current automatic fingerprint verification systems are based on minutia matching. A fingerprint template of such systems is thus a minutia set. The two most prominent kinds of minutiae are ridge ending and ridge bifurcation, which can be extracted using techniques such as those proposed in [1], [2], [3], [11]. Unfortunately, noise, inadequate contrast, and other image acquisition artifacts often make reliable minutia extraction very difficult. The resulting undesirable results include spurious minutiae being produced, valid minutiae being lost, and the minutia type (ending or bifurcation) being wrongly labeled. The employment of various image enhancement techniques [4], [5] merely alleviate these problems to a limited extent since they operate only on a single fingerprint image. Maio and Maltoni [6] implemented five different minutia extraction techniques [7], [8], [9], [10], and compared their performances. The best technique in their experiment produced 8.52 percent spurious minutiae, lost 4.51 percent genuine minutiae, and caused the type labeling error for 13.03 percent minutiae, resulting in a total error of 26.07 percent. For the other approaches, the total errors were 33.83 percent, 119.80 percent, 207.52 percent, and 216.79 percent, respectively. From this experiment, we can see that perfect minutia extraction from a single fingerprint image is a very difficult task.

Whereas the improvement of minutia extraction from a single fingerprint image is limited, multiple fingerprint images captured at different times can be used to achieve more significant improvements since the imaging conditions that cause the minutia extraction error change with time due to the changes in skin condition, climate, and on-site environment. However, improving the image quality based on multiple fingerprints is unfeasible due to the high memory and computation consumption required by processing multiple images that are not rotation and translation invariant. Instead, it is more feasible to improve the template minutia set by using multiple minutia sets of fingerprints captured at different times. If this template improving process requires only a small memory space and short computation time, it can be performed online in a fingerprint verification system, which receives fingerprint inputs of users during the day-to-day normal operation. Such online template improving functions work by merging the input data into the template database during the actual application of the fingerprint verification system.

This paper proposes an online fingerprint template improvement algorithm with which spurious minutiae can be removed, dropped minutiae be recovered, and wrongly labeled minutia type be corrected. The proposed algorithm works online during the day-to-day operation of the fingerprint verification system. As a result, users will find the system more and more reliable.

• *The authors are with the Centre for Signal Processing, Nanyang Technological University, 50 Nanyang Avenue, Singapore 639798. E-mail: {exdjiang, ewser}@ntu.edu.sg.*

Manuscript received 7 Mar. 2001; revised 11 July 2001; accepted 17 Oct. 2001. Recommended for acceptance by M. Pietikainen.
For information on obtaining reprints of this article, please send e-mail to: tpami@computer.org, and reference IEEECS Log Number 113757.

2 Minutia Set Estimation from Multiple Minutia Sets

To extract the minutiae, the image outputted from a fingerprint sensor has to be segmented into the background (invalid fingerprint region) and the valid fingerprint region that usually covers only a part of a finger. Thus, different fingerprint images captured from the same finger usually have different (valid) fingerprint regions. A fingerprint region can be represented by a point set, which contains *x*- and *y*-coordinates of all pixels within this region. Suppose that we have M fingerprint images captured from the same finger and obtained M minutia sets $\mathbf{F}^m = \{F_k^m\}$ and fingerprint regions $\mathbf{S}^m$ by applying a minutia extraction algorithm [11], where

$$F_k^m = (\,x_k^m, y_k^m, \varphi_k^m, t_k^m\,) \tag{1}$$

is a parameter vector describing the location (x_k^m, y_k^m), the direction φ_k^m and the type t_k^m of minutia k in fingerprint m. Although the position and direction of a finger on the sensor is usually different for the various acquisitions, pose transformation can be performed using a minutia matching program [12] in order for the minutia sets and fingerprint regions of different images to be aligned. By matching the minutia sets, we can determine whether two minutiae from two different minutia sets are matched. Without losing generality, we assume that all minutia sets $\mathbf{F}^m$ and fingerprint regions $\mathbf{S}^m$ have been aligned, i.e., they are invariant to the rotation and translation of the finger, and minutiae in different sets have the same index k if and only if they are matched.

For a particular physical minutia, we obtain M' sample measurements of its parameter vector from M' different fingerprints ($M' \leq M$). Our task is to estimate an optimal parameter vector based on these M' measurements, i.e., learning from samples of experimental data. This problem could be approached in the context of minimizing a suitable cost function. If the cost function is chosen to be the negative logarithm of the likelihood function derived from the sample data, this becomes equivalent to maximum likelihood (ML) learning. By considering a generalization of the Gaussian distribution of the data with a constant variance, the ML approach leads to a cost function of the form

$$E = \sum_m |F_k^m - F_k^P|^R,$$

known as the Minkowski-*R* error [13], where F_k^P is the optimal representative minutia parameter vector to be estimated. When the distribution of the data is assumed to be standard Gaussian, i.e., $R = 2$, the cost function reduces to the sum-of-squares error. If a Laplacian distribution is assumed, i.e., $R = 1$, the cost function becomes the city block metric [13]. One problem of the standard sum-of-squares error is that the solution can be dominated by outliers if the distribution of data has heavy tails [14]. The use of the city block metric ($R = 1$) reduces the sensitivity to outliers but minimizing it leads to a median operation [13] on the acquired data. This is more intensive to compute compared with the simple mean operation that minimizes the sum-of-squares error. In our context of fingerprint template improvement, there are really no significant data outliers since minutiae with large measurement errors cannot be matched with other corresponding minutiae with small measurement errors. This provides the motivation in using the sum-of-squares error as the cost function.

Although the biological characteristics of fingerprints ensure minutia features to be permanent and unchanging for a given finger [1], acquisition of minutiae information is affected by the skin and imaging conditions at the time of measurement and the exact manner the finger was making contact with the sensor. As a result, the measured minutia parameter inevitably changes with time and the measurements F_k^m can thus be seen as a temporal sequence of data. As such, the machine-learning task could be viewed as a problem of regression estimation, i.e., function or model learning. There are a number of well-developed approaches in the literature for these problems, for example, LPC [15], Kalman filtering [16], Hidden Markov model [17], MCMC methods [18] and EM-C algorithm [19]. However, changes to minutia parameters may occur abruptly with these changes being maintained for quite a long time due to the skin nature and human's habits. F_k^m is thus typically an abrupt rather than a smooth function of m, which makes it difficult to apply the above-mentioned function learning approaches. Furthermore, fingerprint samples for a particular user are not collected in even time intervals; duration between two subsequent presentations of a finger to the system may vary between several minutes to several months. This again makes the above-mentioned approaches unsuitable.

Having considered the above factors and the computational efficiency required for an online application, we employ the weighted least-squares with predetermined weights as the learning rule. The weights are chosen based on the nature of minutia set series, the objective of the integration of the multiple minutia sets, and the computation efficiency. For instance, a higher weight should be assigned to the registered template than the query fingerprint received in the verification process since the original template obtained during the registration phase is generally more reliable than the input minutia sets obtained during the day-to-day verification process. More recent fingerprint inputs should also be assigned with higher weights than earlier ones since the integration of the multiple minutia sets is aimed at increasing the reliability of future matching process. The weights will be chosen in the next section based on these desired factors and the computational resources required.

The estimation errors for all minutiae k of all minutia sets m are expressed as

$$e_k^m = F_k^m - F_k^P, \quad \text{for } \forall (k,m) F_k^m \in \mathbf{F}^m. \tag{2}$$

The estimated minutia F_k^P is obtained by minimizing the weighted sum of the squared errors

$$\sum_{m, F_k^m \in \mathbf{F}^m} w_k^m (e_k^m)^2 = \sum_{m, F_k^m \in \mathbf{F}^m} w_k^m \left(F_k^m - F_k^P\right)^2 \Rightarrow \text{Minimum}, \quad \text{for } \forall k, F_k^m \in \mathbf{F}^m, \tag{3}$$

where w_k^m are predetermined weights. Based on this criterion, it is straightforward to obtain

$$F_k^P = \frac{1}{\sum\limits_{m, F_k^m \in \mathbf{F}^m} w_k^m} \cdot \sum_{m, F_k^m \in \mathbf{F}^m} w_k^m F_k^m, \quad \text{for } \forall k, F_k^m \in \mathbf{F}^m. \tag{4}$$

The above estimated minutia parameter F_k^P generally has better accuracy than F_k^m since it is a weighted arithmetic average over all matched minutiae. As a result, wrongly labeled minutiae type can be statistically corrected during the averaging process.

If all estimated minutiae by (4) are collected in the estimated template, a template synthesis will be performed. If we match an input fingerprint with this template, we will face the problem of matching a partial input fingerprint with a much larger full fingerprint. False nonmatch rate may decrease but false match rate may increase simultaneously, or some matching criteria have to be changed by compromise. Further, template synthesis can be simply offline performed by enrolling several fingerprints for each finger and there is no point in online synthesizing the templates. The problem of the template synthesis was addressed in [20]. This work is not aimed at solving the problem that the template represents only a partial fingerprint, but aimed at improving the quality of the template online, i.e., reducing the minutia extraction error. Thus, our estimated template is restricted to an estimated fingerprint region $\mathbf{S}^P$, which can be chosen to be one of the M original fingerprint regions $\mathbf{S}^m$ or be determined by the synthesized fingerprint region in case the template synthesis is performed beforehand in the registration phase.

Our estimated minutia set $\mathbf{F}^P = \left\{F_k^P | \left(x_k^P, y_k^P\right) \in \mathbf{S}^P\right\}$ contains all minutiae in the region $\mathbf{S}^P$ extracted from the M fingerprints. As a result, genuine minutiae that are not extracted from some fingerprints can be recovered in the estimated minutia set $\mathbf{F}^P$ if they are successfully extracted from some other fingerprints. However, any spurious minutia extracted from any fingerprint is also transferred to the estimated minutia set if it is located within $\mathbf{S}^P$. Therefore, a technique has to be developed to identify the spurious minutiae of the estimated template $\mathbf{F}^P$.

If an estimated minutia k is successfully extracted from fingerprint m, a certainty level $c_k^m = 1$ is defined. If this minutia fails to be extracted from fingerprint m but its location is within this fingerprint region, a certainty level $c_k^m = 0$ is defined. However, if the region of fingerprint m does not cover this minutia, no information about the reliability of minutia k is provided by fingerprint m. Thus, the reliability of each estimated minutia is described by M certainty levels defined by

$$c_k^m = \begin{cases} 1, & \text{if } F_k^m \in \mathbf{F}^m \\ 0, & \text{if } F_k^m \notin \mathbf{F}^m \wedge (x_k^P, y_k^P) \in \mathbf{S}^m, \\ \text{unknown}, & \text{if } \left(x_k^P, y_k^P\right) \notin \mathbf{S}^m \end{cases} \quad \text{for } \forall\, k, F_k^P \in \mathbf{F}^P, m = 1, 2, \ldots M, \tag{5}$$

Similar to the calculation of the estimated minutia parameter in (4), a certainty level c_k^P of the estimated minutia k can be estimated by the weighted average of c_k^m over all fingerprints whose regions cover the minutia k

$$c_k^P = \frac{1}{\sum\limits_{m, (x_k^P, y_k^P) \in \mathbf{S}^m} w_k^m} \cdot \sum_{m, (x_k^P, y_k^P) \in \mathbf{S}^m} w_k^m c_k^m, \quad \text{for } \forall k, F_k^P \in \mathbf{F}^P. \tag{6}$$

For authenticating a future input fingerprint, only the minutiae whose certainty levels are equal to or higher than a threshold Cv, $0 < Cv < 1$, will be used in the matching. This means that minutiae with certainty levels lower than Cv are regarded as spurious minutiae, which must still remain in the template for the further template improvement in the future or can be removed from the template to reduce the template size if further template improvement is not conducted.

If we choose equal weights, the estimation of a minutia based on (4) and (6) is simplified to

$$F_k^P = \frac{1}{M_k} \cdot \sum_{m, F_k^m \in \mathbf{F}^m} F_k^m, \quad \text{for } \forall k, F_k^m \in \mathbf{F}^m, \tag{7}$$

$$c_k^P = \frac{1}{N_k} \cdot \sum_{m, (x_k^P, y_k^P) \in \mathbf{S}^m} c_k^m, \quad \text{for } \forall k, F_k^P \in \mathbf{F}^P, \tag{8}$$

where M_k is the number of fingerprints from which minutia k is extracted and N_k is the number of fingerprints that cover the position of minutia k. In this case, we see that it is the minutia occurrence frequency that is used to recover dropped minutiae and remove spurious minutiae.

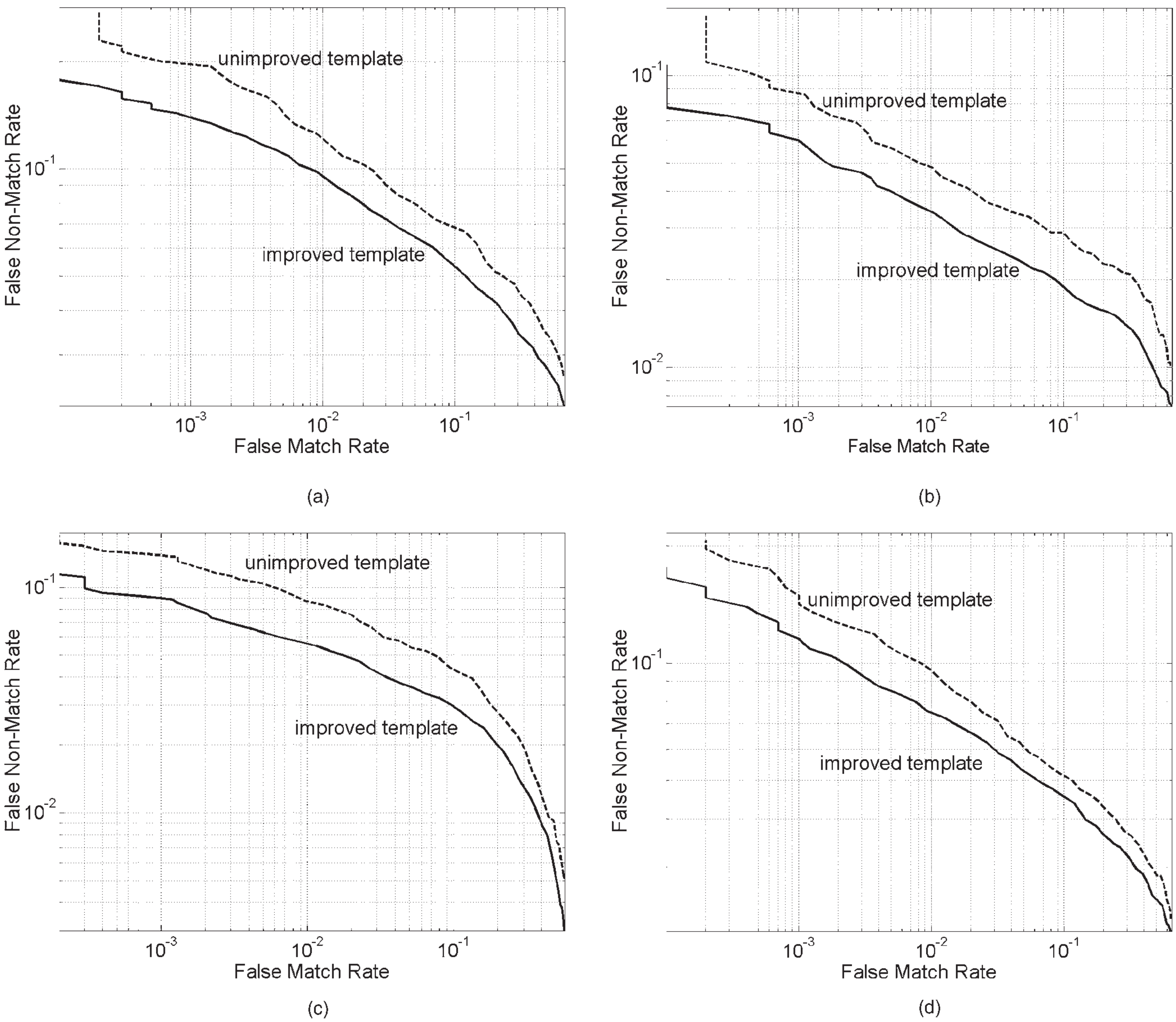

Fig. 1. ROC curves on (a) DB1_a, (b) DB2_a, (c) DB3_a, and (d) DB4_a.

3 ONLINE FINGERPRINT TEMPLATE IMPROVEMENT

It is not user-friendly to capture a number of fingerprints of the same finger at long intervals in the registration phase. However, during the verification operation of a fingerprint verification system, input fingerprints are successively received and compared with the templates. If an input fingerprint is successfully matched with a template, these two fingerprints are verified to have originated from the same finger. Therefore, input fingerprints can be used to improve the matched template online during the day-to-day operation of the fingerprint verification system. However, to use (4) and (6) to improve a template, all matched input minutia sets and fingerprint regions need to be stored and an arithmetic average over all matched minutia sets need to be calculated for every template update. This requires a large storage space and significant computation time. However, storage space and verification time are often serious constraints, especially in stand-alone application. To reduce the storage space and processing time requirements, we need a simplified fingerprint region representation and a recursive algorithm.

It is not difficult to simplify the fingerprint region representation. A polygon represented by a few points can be used to approximate a fingerprint region [21] and to determine whether a minutia point of another fingerprint is located within this region. It is easy to prove that a point with location (x, y) is within a polygon represented by L points (x_j, y_j), $j = 1, 2, \ldots L$, if and only if:

$$A_j x + B_j y + C_j < 0, \quad \text{for all } j, j = 1, 2, \ldots, L, \tag{9}$$

where $A_j = y_j - y_{j+1}$, $B_j = x_{j+1} - x_j$, and $C_j = -A_j x_j - B_j y_j$ with $(x_{L+1}, y_{L+1}) = (x_1, y_1)$.

A recursive algorithm that implements the weighted averaging in (4) and (6) can be derived by choosing the weights properly. As mentioned earlier, the finger skin and imaging condition changes with time and the template improvement is aimed at increasing the reliability of future matching processes. Thus, the more recent fingerprint inputs should be assigned higher weights than earlier ones. In addition, a fingerprint image is usually captured with much more caution during the registration process compared with those acquired during the day-to-day verification. Therefore, a higher weight should be assigned to the original registered template. We thus choose a power series α^n to weight the fingerprint sequence and another constant λ to distinguish the weights of input fingerprints from that of the original template fingerprint.

Let $F_k^P(N)$ denote the improved template minutia by using an original template minutia $F_k^P(0)$ and N input minutiae $F_k^I(n)$ of N input fingerprints, $n = 1, 2 \ldots \mathrm{N}$, where fingerprint n is received

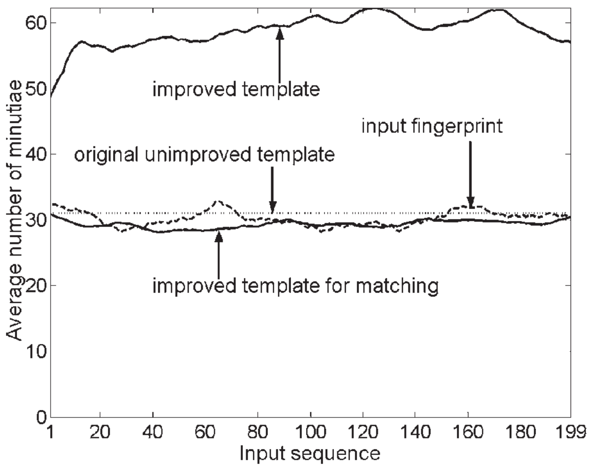

Fig. 2. Average numbers of minutiae.

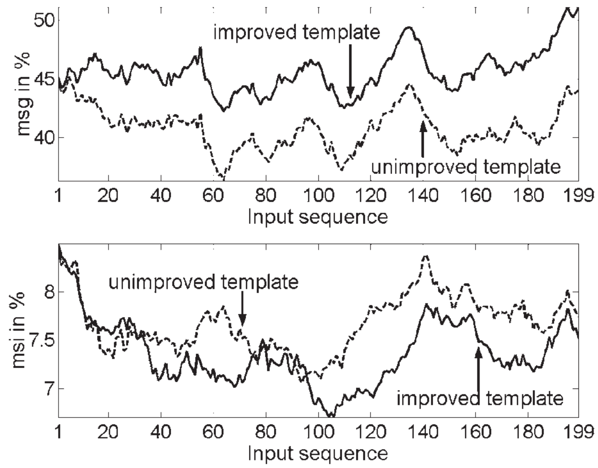

Fig. 3. Average matching scores.

earlier than fingerprint $n+1$. Without losing generality, (4) can be rewritten as

$$F_k^P(N) = \alpha^N F_k^P(0) + \lambda \sum_{n=0}^{N-1} \alpha^n F_k^I(N-n), \tag{10}$$

with the condition of

$$\alpha^N + \lambda \sum_{n=0}^{N-1} a^n = 1. \tag{11}$$

The power series coefficients $\alpha^n (\alpha < 1)$ weight the more recent fingerprint entries more heavily than earlier ones, while the constant $\lambda (\lambda < 1)$ scales the weights of input fingerprints with respect to the original template. By choosing $\lambda = 1 - \alpha$, it is not difficult to prove that condition (11) holds independent of the values of α and N. Thus, we can use the same value of α in (10) for different number of entries N, i.e., we can have

$$F_k^P(N+1) = \alpha^{N+1} F_k^P(0) + \lambda \sum_{n=0}^{N} \alpha^n F_k^I(N+1-n). \tag{12}$$

From (10) and (12) with $\lambda = 1 - \alpha$, it is straightforward to obtain

$$F_k^P(N+1) = \alpha F_k^P(N) + (1-\alpha) F_k^I(N+1). \tag{13}$$

Equation (13) is the recursive template minutia parameter update formula where $F_k^P(N)$ is the old template minutia parameter and $F_k^P(N+1)$ the new template minutia parameter after the fingerprint verification system receives a new entry $F_k^I(N+1)$. If an input minutia $F_k^I(N+1)$ is located within the template fingerprint region, but there is no old template minutia $F_k^P(N)$ matched with it, this input minutia will be merged into the template, i.e., $F_k^P(N+1) = F_k^I(N+1)$.

In a similar way, a recursive certainty level update formula can be easily derived from (6) as

$$c_k^P(N+1) = \alpha c_k^P(N) + (1-\alpha) c_k^I(N+1), \tag{14}$$

where $c_k^P(N)$ is the old certainty level of template minutia k and $c_k^P(N+1)$ the new certainty level of template minutia k after the system receives a new entry with certainty leve $c_k^I(N+1)$.

It is worth noting that falsely matched input fingerprints will have an adverse effect on the template improvement process. To reduce the probability of this happening, an input fingerprint is used to update the matched template only if their matching score ms is higher than a threshold Mu that is set to be larger than the verification threshold Mv of a fingerprint verification system. Furthermore, we could limit the shortest time interval Ti between two successive updates of a template to prevent any intentional repeated abuse of the template update process. This way, we can diminish the negative effects of the online fingerprint template improvement procedure.

The proposed online fingerprint template improvement algorithm is summarized as follows:

1. Users enroll for the fingerprint verification system. For each enrolled fingerprint, a fingerprint region $\mathbf{S}^p$ is segmented out and represented by L points and a minutia set $\{F_k^P\}$ is extracted. A certainty level $c_k^P = 1$ is initialized for each extracted minutia. Template update time tu is initialized to be the current time tc.
2. The fingerprint verification system waits until an input fingerprint is received.
3. Segment out the fingerprint region and represent it with L points. Extract input minutia set $\{F_k^I\}$. Match it with each template $\{F_k^P | c_k^P \geq Cv\}$. If the maximal matching score $ms \leq Mv$, reject this input and go to Step 2, otherwise output the corresponding finger ID.
4. Read the current time tc. If $ms \leq Mu \, (Mu > Mv)$ or $(tc - tu) < Ti$, go to Step 2, otherwise $tu = tc$.
5. The matched template $\{F_k^P\}$ is updated as follows:

   a. For all matched template minutiae, F_k^P and c_k^P are updated by

   $$\alpha F_k^P + (1-\alpha) F_k^I \Rightarrow F_k^P$$

   and $\alpha c_k^P + 1 - \alpha \Rightarrow c_k^P$.

   b. Find all unmatched template minutiae located within the input fingerprint region by using (9). The certainty levels of these minutiae are updated by $\alpha c_k^P \Rightarrow c_k^P$.

   c. The certainty levels of other template minutiae (i.e., those located outside the input fingerprint region) are unchanged, i.e., $c_k^P \Rightarrow c_k^P$.

   d. Find all unmatched input minutiae located within the template fingerprint region by using (9). Merge these minutiae into the template, i.e., $F_k^I \Rightarrow F_k^P$ with $c_k^P = 1 - \alpha$.

   e. Remove all template minutiae whose certainty levels are lower than a threshold $Cu (0 < Cu < 1 - \alpha < Cv < 1)$ to limit the enlargement of the template size. Store the updated template and go to Step 2.

The above proposed online fingerprint template improvement algorithm updates a fingerprint template using a recursive algorithm that implements a weighted averaging over all matched fingerprints. It increases the precision of minutia parameter, corrects

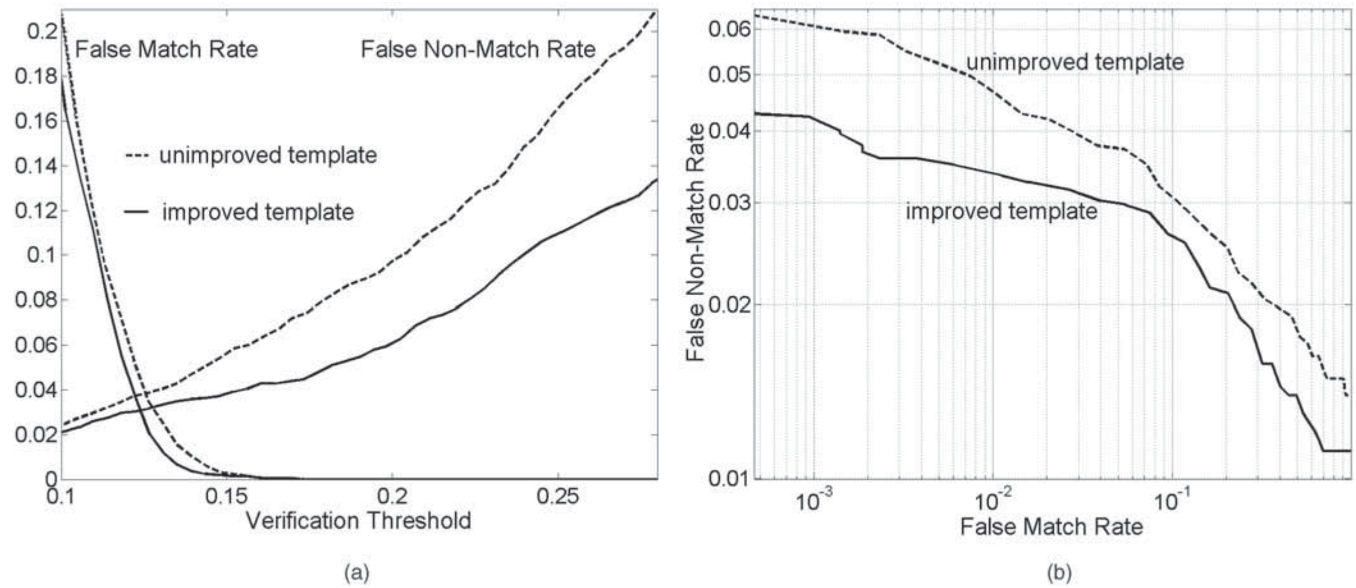

Fig. 4. (a) False match rate and false nonmatch rate versus threshold and (b) ROC curves on DB_T.

wrongly labeled minutia type, recovers dropped minutiae (from $c_k^P < Cv$ to $c_k^P \geq Cv$) and removes spurious minutiae (from $c_k^P \geq Cv$ to $c_k^P < Cv$). Furthermore, the algorithm accords greater importance to more recently acquired fingerprints so that it weakens the effect of older skin and imaging conditions while strengthening recent ones. The recursive algorithm works only on the current input fingerprint and the stored template. This greatly reduces the processing time and storage space requirements and, therefore, enables our proposed approach to be online employed.

4 Experimental Studies

For a meaningful performance evaluation of our online fingerprint template improvement algorithm, the test database should contain not only a large number of finger IDs, but also a large number of sample fingerprints per finger. Unfortunately, it is very difficult to find or build up such a database. Thus, we decided to conduct experiments with two different kinds of databases. The first experiment used the FVC2000 [22] databases that contain a large number of finger IDs and eight sample fingerprints per finger. The second experiment used a database collected by us that contains only 12 finger IDs but 200 sample fingerprints per finger. The algorithm parameters chosen for both experiments were: $L = 8$, $\alpha = 0.8$, $C_u = 0.15$, $C_v = 0.5$, $Mu = Mv = 0.25$, and $Ti = 0$.

There are four databases, DB1_a, DB2_a, DB3_a, and DB4_a, used in FVC2000. Each database contains 100 finger IDs and eight fingerprints per finger (800 fingerprints in all). Let $\mathbf{F}_j^i$ denote the minutia set extracted from the ith sample fingerprint of the jth finger, $i = 1, 2, \dots 8$, $j = 1, 2, \dots 100$. To test the verification performance with the unimproved template, each template $\mathbf{T}_j^i = \mathbf{F}_j^i$ was matched against the inputs $\mathbf{F}_j^k$ $(i < k \leq 8)$ to obtain the genuine matching scores and each template $\mathbf{T}_j^1 = \mathbf{F}_j^1$ was matched against the inputs $\mathbf{F}_k^1$ $(j < k \leq 100)$ to obtain the impostor matching scores. Let $\mathbf{T}_j^i(k)$ denote the improved template by using the original template $\mathbf{F}_j^i$ and six inputs $\mathbf{F}_j^m$ $(1 \leq m \leq 8, m \neq i, m \neq k, i < k \leq 8)$. To test the verification performance with the improved template, each improved template $\mathbf{T}_j^i(k)$ was matched against the input $\mathbf{F}_j^k$ to obtain the genuine matching score and each template $\mathbf{T}_j^1(2)$ was matched against the inputs $\mathbf{F}_k^1$ $(j < k \leq 100)$ to obtain the impostor matching scores. In both tests, a total of $((8 \times 7)/2) \times 100 = 2,800$ genuine matches and $(100 \times 99)/2 = 4,950$ impostor matches were performed on each database. These numbers of matches are the same as that used in the FVC2000 [22]. Fig. 1 illustrates the ROC curves on the four FVC2000 databases. These ROC curves on the four different databases consistently show that our template improvement algorithm causes a significant improvement in the verification accuracy.

In the second experiment, a Veridicom CMOS sensor of size 300×300 pixels was used to capture fingerprints. Twelve untrained users were enrolled by capturing one template fingerprint per user. Each of these 12 users was asked to represent the enrolled finger to the system to produce input fingerprints several times (no more than five times) every working day until 199 input fingerprints per user were received. Each input fingerprint was matched online with the 12 templates and used to update (online improve) the template that had the maximal matching score if this matching score was higher than Mu. In this way, 2,400 fingerprints were collected in database DB_T and the numbers of genuine matches and imposter matches are $199 \times 12 = 2,388$ and $199 \times 11 \times 12 = 26,268$, respectively.

To show the template improvement progress clearly against the fingerprint input sequence, we averaged the results over the 12 users. Fig. 2 plots the average numbers of minutiae of the input fingerprints, the original templates, and the improved templates as well as the average numbers of improved template minutiae that were valid for matching. From Fig. 2, the improved template size increased with the first few fingerprint inputs and then stabilized at around twice the original unimproved template size. However, Fig. 2 also shows that the improved template had averaged fewer valid minutiae for matching than the original template. It means that, in most cases, the template improvement process removed more spurious minutiae compared with recovering dropped minutiae. This is because our minutia extraction algorithm, like most other minutia extraction approaches [6], usually produces more spurious minutiae than dropped minutiae. This experiment also tells us that the improved verification performance is not due to the increased number of minutiae in the improved template but the improved template quality.

Fig. 3 illustrates the average genuine matching scores msg and the average imposter matching scores msi against the fingerprint input sequence. Fig. 3 clearly shows that the improved template produced not only higher genuine matching scores but also lower imposter matching scores than the unimproved template. Obviously, the higher genuine matching score is due to the template improvement process. The lower imposter matching score also does not surprise us as the template improvement process reduced the number of spurious minutiae and, thus, lowered the impostor matching score. Both the higher genuine matching and the lower imposter matching scores improved the fingerprint verification accuracy, as shown in Fig. 4.

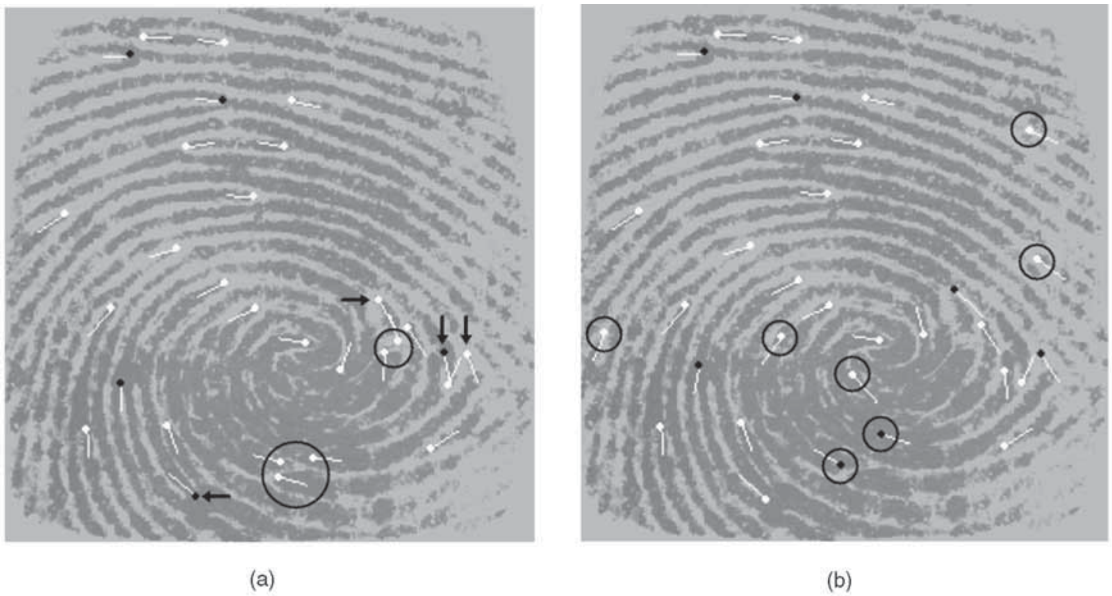

Fig. 5. (a) Original template minutiae and (b) improved template minutiae.

The average time taken by the fingerprint verification process (one minutia set extraction and one matching) was 0.147 seconds and the average time taken by the template improvement process was only 0.0003 seconds (for a Pentium III–733 MHz PC). Our template improvement algorithm only decelerates the fingerprint verification system by a negligible 0.2 percent.

Fig. 5a shows an original template minutia set while Fig. 5b shows the improved template minutia set (valid for matching, i.e., $\{F_k^P | c_k^P \geq 0.5\}$), generated using the template minutia set in Fig. 5a and 26 other input minutia sets. In these two figures, white dots represent endings, dark dots represent bifurcations, while white short lines represent the minutia directions. There are 30 minutiae in Fig. 5a and 32 minutiae in Fig. 5b. After the template improvement, five spurious minutiae were removed (see circles in Fig. 5a), seven dropped minutiae were recovered (see circles in Fig. 5b) and four minutiae had their type relabeled (see the arrows in Fig. 5a).

5 Conclusions

In this work, an online fingerprint template improvement algorithm is proposed. The proposed algorithm improves the reliability of a fingerprint template by using weighted averaging over all matched fingerprints that a fingerprint verification system receives. It reduces minutia extraction errors, such as spurious minutiae, dropped minutiae, and wrongly labeled minutia type, which are difficult to avoid using only a single fingerprint image. Furthermore, the template is gradually changed to reflect changes in the finger skin and imaging conditions by weighting recent fingerprints more heavily. A recursive algorithm minimizes the storage space and computation requirements of the template improvement process. As a result, the proposed fingerprint template improvement process can be performed online during the day-to-day operation of a fingerprint verification system. Extensive experimental studies demonstrate that the proposed online template improvement technique significantly increase the verification accuracy at a negligible cost in time. One problem of this technique is the enlargement of the template size. However, the improved template size is well limited to within twice the size of the original template.

References

[1] A.K. Jain, L. Hong, and R. Bolle, "On-Line Fingerprint Verification," *IEEE Trans. Pattern Analysis and Machine Intelligence,* vol. 19, no. 4, pp. 302-314, Apr. 1997.

[2] A.K. Jain, L. Hong, S. Pankanti, and R. Bolle, "An Identity-Authentication System Using Fingerprints," *Proc. IEEE,* vol. 85, no. 9, pp. 1365-1388, Sept. 1997.

[3] X.D. Jiang, W.Y. Yau, and W. Ser, "Minutiae Extraction by Adaptive Tracing the Gray Level Ridge of the Fingerprint Image," *Proc. IEEE Sixth Int'l Conf. Image Processing,* vol. 2, pp. 852-856, 1999.

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

[5] X.D. Jiang, W.Y. Yau, and W. Ser, "Fingerprint Image Processing for Automatic Verification," *Proc. IEEE Second Int'l Conf. Information, Comm. & Signal Processing,* 1999.

[6] D. Maio and D. Maltoni, "Direct Gray-Scale Minutiae Detection in Fingerprints," *IEEE Trans. Pattern Analysis and Machine Intelligence,* vol. 19, no. 1, pp. 27-39, 1997.

[7] R.M. Stock and C.W. Swonger, "Development and Evaluation of a Reader of Fingerprint Minutiae," Technical Report CAL No. XM-2478-X-1: 13-17, Cornell Aeronautical Laboratory, 1969.

[8] B. Moayer and K. Fu, "A Tree System Approach for Fingerprint Pattern Recognition," *IEEE Trans. Pattern Analysis and Machine Intelligence,* vol. 8, no. 3, pp. 376-388, 1986.

[9] M.R. Verma, A.K. Majumdar, and B. Chatterjee, "Edge Detection in Fingerprints," *Pattern Recognition,* vol. 20, no. 5, pp. 513-523, 1987.

[10] L. O'Gorman and J.V. Nickerson, "An Approach to Fingerprint Filter Design," *Pattern Recognition,* vol. 22, no. 1, pp. 29-38, 1989.

[11] X.D. Jiang, W.Y. Yau, and W. Ser, "Detecting the Fingerprint Minutiae by Adaptive Tracing the Gray Level Ridge," *Pattern Recognition,* vol. 34, no. 5, pp. 999-1013, May 2001.

[12] X.D. Jiang and W.Y. Yau, "Fingerprint Minutiae Matching Based on the Local And Global Structures," *Proc. 15th Int'l Conf. Pattern Recognition,* vol. 2, pp. 1042-1045, 2000.

[13] Ch.M. Bishop, *Neural Networks for Pattern Recognition.* Oxford Univ. Press, 1995.

[14] P.J. Huber, *Robust Statistics.* New York: John Wiley, 1981.

[15] L. Rabiner and J.B.-H. Juang, *Fundamentals of Speech Recognition.* Prentice-Hall, 1993.

[16] Y. Bar-Shalom and T. Fortmann, *Tracking and Data Association.* Academic Press, 1988.

[17] G.E. Kopec and P.A. Chou, "Document Image Decoding Using Markov Source Models," *IEEE Trans. Pattern Analysis and Machine Intelligence,* vol. 16, no. 6, pp. 602-617, June 1994.

[18] P. Magni, R. Bellazzi, and G.D. Nicolao, "Bayesian Function Learning Using MCMC Methods," *IEEE Trans. Pattern Analysis and Machine Intelligence,* vol. 20, no. 12, pp. 1319-1331, Dec. 1998.

[19] B. North, A. Blake, M. Isard, and J. Rittscher, "Learning and Classification of Complex Dynamics," *IEEE Trans. Pattern Analysis and Machine Intelligence,* vol. 22, no. 9, pp. 1016-1034, Sept. 1997.

[20] K.A. Toh, W.Y. Yau, X.D. Jiang, T.P. Chen, J. Lu, and E. Lim, "Minutiae Data Synthesis for Fingerprint Identification Application," *Proc. IEEE Int'l Conf. Image Processing,* 2001.

[21] R.C. Gonzalez and R.E. Woods, *Digital Image Processing.* Addison Wesley, 1993.

[22] D. Maio, D. Maltoni, R. Cappelli, J.L. Wayman, and A.K. Jain, "FVC2000: Fingerprint Verification Competition," *Proc. 15th Int'l Conf. Pattern Recognition,* 2000.