This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication.

Hierarchical Minutiae Matching for Fingerprint and Palmprint Identification

Fanglin Chen, *Member, IEEE*, Xiaolin Huang, *Member, IEEE*, and Jie Zhou, *Senior Member, IEEE*

***Abstract*—Fingerprints and palmprints are the most common authentic biometrics for personal identification, especially for forensic security. Many previous works have been proposed to speed up the searching process in fingerprint and palmprint identification systems, such as those based on classification or indexing, in which the deterioration of identification accuracy is hard to avert. In this paper, a novel hierarchical minutiae matching algorithm for fingerprint and palmprint identification systems is proposed. This method decomposes the matching step into several stages and rejects many false fingerprints or palmprints on different stages, thus it can save much time while preserving a high identification rate. Experimental results show that the proposed algorithm can save almost 50% searching time compared to traditional methods and illustrate its effectiveness.**

***Index Terms*—AFIS, fingerprint verification, minutiae, hierarchical structure.**

I. Introduction

Biometric technologies are playing more and more important role in various security applications, since biometric indicators cannot be easily stolen or shared compared with traditional security methods (such as key and password). Fingerprint and palmprint recognition have one of the highest levels of reliability and extensively used by forensic experts in criminal investigations [1]–[5].

Most automatic fingerprint and palmprint recognition systems are based on a widely-used feature called *minutiae* [5]–[8], which is usually defined as the ridge ending and the ridge bifurcation. Minutiae, including their coordinates and directions, are the most distinctive features to represent the fingerprint and palmprint. The minutiae-based fingerprint and palmprint recognition systems consists of two steps, i.e., minutiae extraction and minutiae matching. In the minutiae matching process, the minutiae feature of a given fingerprint (or palmprint) is compared with the minutiae template, and the matched minutiae will be found out. If the matching score exceeds a predefined threshold, the two fingerprints (or palmprints) can be regarded as belonging to the same finger (or palm) [9]. The minutiae based fingerprint and palmprint matching algorithm usually consists of two steps, i.e., minutiae alignment and matching score computing. Minutiae alignment is essentially a point pattern matching problem. The goal is to find a translation and rotation that aligns two point sets. Thus, the alignment is a three (two of translation: x-coordinate and y-coordinate, and one of rotation) dimensional searching task, and it is time-consuming since the complications. In fingerprint or palmprint verification application, it is tolerated for it conducts one-to-one comparison. But in AFIS/APIS (automatic fingperprint/palmprint identification system) which is one-to-N (conducts N one-to-one comparisons, N is the number of fingerprints or palmprints in the database, and it is usually very huge), it is not acceptable.

There have been some researches to speed up the searching in fingerprint identification problems. These attempts are generally based on two techniques [10], i.e., fingerprint classification and indexing. Fingerprint classification [11]–[14] is a difficult pattern recognition problem due to translations or rotations of fingerprints and noises in fingerprint images. The most significant factor that makes the fingerprint classification problem so difficult is that the small interclass variation and the large intraclass variation in the fingerprint patterns [15]. Indexing techniques [16]–[19] involve pre-selection based on fingerprint classification, or demographic such as sex, age, race, and geographic location. Exploiting the more information, these approaches classify fingerprints into more classes than the classification technique. However, although these approaches can speed up the database searching, the accuracy deterioration of the identification system is hard to avert [20].

Besides the work based on classification and indexing, there are some other state-of-the-art technologies attempt to speed up the minutiae matching. Chan *et al.* [21] proposed a fast fingerprint matching methodology based on localizing the matching regions in captured fingerprint images. Ying *et al.* [22] used the core point to determine the reference point and used a round bounding box for minutiae matching. Using the round bounding box can simplify the calculating processes and speed up the matching. The essential step of detecting a reference point for each fingerprint makes these two methods limited, since the reference point is difficult to detect and the accuracy is not satisfactory yet [23]. Thus, both of the methods are not very robust with respect to errors in the detection of the reference point. Xu *et al.* [24] used the spectral minutiae algorithm as a preselector to speed up the minutiae matching. However, it is not robust to the low quality fingerprints. This limits the application of the spectral minutiae algorithm.

The work reported here, proposes a novel algorithm for fingerprint and palmprint minutiae identification to save time. The following description will mainly analyze fingerprint recognition systems, and it can be generalized to palmprint recognition systems. Drawing lessons from fast cascade face

Fanglin Chen is with the College of Mechatronic Engineering and Automation, National University of Defense Technology, Changsha, Hunan, P.R. China, 410073; Xiaolin Huang and Jie Zhou are with the Department of Automation, Tsinghua University, Beijing, P.R. China, 10084 (e-mail: fanglinchen@nudt.edu.cn; huangxl06@mails.tsinghua.edu.cn; jzhou@tsinghua.edu.cn).

recognition [25]–[27], we proposed a hierarchical strategy and utilized it in the fingerprint matching stage. The hierarchical strategy reject many fingerprints (in the database of the AFIS) which do not belong to the same finger as the input fingerprint quickly, thus it can save much time. Since each hierarchical step can reject many false template fingerprints, it can preserve the identification rate of the system, and have a little improvement. Not as the conventional methods based on classification and indexing, the proposed method does not use more features or information than minutiae, and it can be integrated to the conventional identification systems.

Using some other features other than minutia can improve the accuracy of fingerprint identification [28]. The proposed hierarchical strategy can also be utilized to multi-feature fingerprint identification. Additional experimental results show that the identification ratio of the hierarchical multi-feature method increases a lot with some additional time cost.

The rest of the paper is organized as follows: In Section II, the related conventional minutiae-based fingerprint matching algorithm is introduced. Section III proposes the hierarchical fingerprint matching algorithm. The experimental results and analysis are presented in Section IV. We provide conclusion and discussion in Section V.

II. Related Works

After minutiae extraction, the following triplet structure is used to record each minutia:

$$m = \{x, y, \theta\}, \tag{1}$$

where x is the x-coordinate, y is the y-coordinate, and θ is the local ridge orientation. It should be mentioned that most fingerprint matching algorithms do not differentiate between ridge endings and bifurcations when comparing two fingerprints. One reason for this is that noise or excess pressure can cause a bifurcation to look like a ridge ending, and vice versa [29]. The minutiae sets of the input fingerprint I and the template fingerprint T in the database can be formulated as

$$\begin{aligned} M_I &= \{m_k, k = 1, 2, \cdots, n_I\}, m_k = \{x_k, y_k, \theta_k\}, \\ M_T &= \{m'_k, k = 1, 2, \cdots, n_T\}, m'_k = \{x'_k, y'_k, \theta'_k\}, \end{aligned} \tag{2}$$

where n_I and n_T denote the number of minutiae in I and T, respectively.

A minutia m_i in I and a minutia m'_j in T are regarded as matched when: 1) the differences of their coordinates are less than a given tolerance δx and δy for x-axis and y-axis, respectively, and 2) the angular difference between their directions does not exceed an angular tolerance $\delta\theta$:

$$\begin{aligned} |x_i - x'_j| &< \delta x, \\ |y_i - y'_j| &< \delta y, \\ \delta(i, j) &< \delta\theta, \end{aligned} \tag{3}$$

where $\delta(i, j)$ is the difference between the orientation values of the minutiae, m_i and m'_j, which is formulated as follows:

$$\delta(i, j) = \begin{cases} \delta_0(i, j), & \text{if } \delta_0(i, j) \leqslant \pi, \\ 2\pi - \delta_0(i, j), & \text{otherwise}, \end{cases} \tag{4}$$

and $\delta_0(i, j)$ is defined as:

$$\delta_0(i, j) = |\theta_i - \theta'_j|. \tag{5}$$

Eqn. (4) takes into account that 2π phase jumps may appear (the difference between angles of $3°$ and $358°$ is only $5°$).

A. Alignment

In order to match two minutiae sets M_I and M_T with unknown translation and rotation, the two point sets must be registered with respect to each other [9]. Denote the translation as $(\Delta x, \Delta y)$, and the rotation as θ. A minutia m_i in M_I should first be translated and rotated before matching to a minutia m'_j in M_T:

$$\begin{aligned} \begin{bmatrix} \hat{x}_i \\ \hat{y}_i \end{bmatrix} &= \begin{bmatrix} \cos\theta & -\sin\theta \\ \sin\theta & \cos\theta \end{bmatrix} \begin{bmatrix} x_i \\ y_i \end{bmatrix} + \begin{bmatrix} \Delta x \\ \Delta y \end{bmatrix}, \\ \hat{\theta}_i &= \theta_i + \theta. \end{aligned} \tag{6}$$

To estimate the translation and rotation parameters is certainly a critical and time-consuming step. It is a well studied problem in *point pattern matching* [4], and the robust and widely used method is what Ratha *et al.* [9] proposed using the generalized Hough transform. The Hough transform is normally used for detecting lines in image processing. By converting *point pattern matching* to the problem of detecting peaks in the Hough space of transformation parameters, the Hough transform can be generalized for *point pattern matching* [30]. It discretizes all the transformation parameters into finite sets of values and accumulates voting in the discretized parameter space. For each pair of potentially matching minutiae (one from M_I, and the other from M_T), the translation and rotation necessary to align them is calculated. Voting for this translation and rotation is accumulated in the parameter space. After testing all possible matching minutiae pairs, the parameter space is used to select the most likely translation and rotation parameters (i.e., the one with the maximum accumulated voting). Denote the discrete parameter sets as $\{\Delta x_1^+, \Delta x_2^+, \cdots, \Delta x_a^+\}$, $\{\Delta y_1^+, \Delta y_2^+, \cdots, \Delta y_b^+\}$, $\{\theta_1^+, \theta_2^+, \cdots, \theta_c^+\}$, the process of the generalized Hough transform voting is summarized in Algorithm 1.

B. Matching Score

After the generalized Hough transform voting, the minutiae in the set M_I are translated and rotated according to (6) using the estimated parameters. Then the matched minutiae number $N_{matched}$ can be counted according to (3). The matching score is totally based on the matched number with a normalization by the total number of minutiae. Denote the numbers of the minutiae in Ω (the intersection of the two fingerprints) as N_I and N_T for the fingerprints I and T, respectively. The matching score is usually computed as

$$s_{matching} = \frac{N_{matched}^2}{N_I N_T}. \tag{7}$$

In fingerprint verification systems, if the matching score $s_{matching}$ is higher than a pre-defined threshold, it is determined that the two fingerprints I and T are from the same finger, otherwise not. In fingerprint identification system, if

Input: the minutiae sets, M_I and M_T
Output: the transformation parameters, $(\Delta x, \Delta y, \theta)$
Reset V;
for *each* m_i **do**
    for *each* m'_j **do**
        $k^* = \arg\min_k |\delta(i,j) - \theta_k^+|$;
        // about $\delta(i,j)$, refer to (4)
        $\theta^+ = \theta_{k^*}^+$;
        $\begin{bmatrix}\Delta x\\ \Delta y\end{bmatrix} = \begin{bmatrix}x'_j\\ y'_j\end{bmatrix} - \begin{bmatrix}\cos\theta^+ & -\sin\theta^+\\ \sin\theta^+ & \cos\theta^+\end{bmatrix}\begin{bmatrix}x_i\\ y_i\end{bmatrix}$;
        $k^* = \arg\min_k |\Delta x - \Delta x_k^+|$;
        $\Delta x^+ = \Delta x_{k^*}^+$;
        $k^* = \arg\min_k |\Delta y - \Delta y_k^+|$;
        $\Delta y^+ = \Delta y_{k^*}^+$;
        $V[\Delta x^+, \Delta y^+, \theta^+] = V[\Delta x^+, \Delta y^+, \theta^+] + 1$;
    end
end
$(\Delta x, \Delta y, \theta) = arg \max_{\Delta x^+, \Delta y^+, \theta^+} V[\Delta x^+, \Delta y^+, \theta^+]$;

Algorithm 1: Pseudo-code of the generalized Hough transform voting

the maximum matching score (the maximum one of all the matching scores of the input fingerprint I with each template fingerprint T in the database) is higher than a pre-defined threshold, it is confirmed that the input fingerprint is present in the database, and returns the identifier of the matched fingerprint template T.

C. Matching Time Analysis

The matching time for one-to-one verification consists of two parts: the alignment time t_A and the matching score computing time t_C. Thus the total one-to-one matching time t_M is as

$$t_M = t_A + t_C. \tag{8}$$

For a fingerprint identification system, since it can be regarded as N one-to-one fingerprint verifications, the searching time T_S for an input fingerprint is as

$$T_S = N \cdot t_M. \tag{9}$$

In AFIS, the number N of fingerprints in the database is usually very large (e.g., in the magnitude order of 10^6 or larger).

III. Fingerprint Matching Algorithm based on Hierarchical Strategy

As mentioned in Section II-C, a fingerprint identification system has a long unacceptably response time, which hinders the application of the system. Thus the speeding up of the system becomes more and more important and urgent. Some researchers speeded up the identification process by reducing the number of comparisons that are required to be conducted. These attempts are generally based on two techniques, i.e., fingerprint classification [11]–[14] and indexing [16]–[20]. Fingerprint classification is a difficult pattern recognition problem due to translations or rotations of fingerprints and noises in fingerprint images. The most significant factor that makes the fingerprint classification problem so difficult is that the small interclass variation and the large intraclass variation in the fingerprint patterns [15]. Indexing techniques involve pre-selection based on demographic information such as sex, age, race, and geographic location [16]. However, such information is not always accessible (e.g., criminal identification).

A. Hierarchical Identification Searching Algorithm

The following of this section proposes a hierarchical algorithm for fingerprint minutiae matching to speed up the identification. The proposed algorithm tries to reduce the time t_A of one-to-one verification, then the total searching time can be cut down according to (8) and (9). The alignment time t_A is very high since Algorithm 1 has two loop statements. Denote the average minutiae number of a fingerprint as n, then the time-complexity of the alignment algorithm is about $\circ(n^2)$. Since the spurious minutiae detected in the minutiae extraction step, n is generally 40 or even bigger, which makes the alignment time-consuming.

Drawing lessons from fast cascade face recognition [25]–[27], we proposed a hierarchical strategy to speed up fingerprint identification. Though Adaboost is a well-known and widely used structure in face recognition [31], it needs lots of features to build weak classifiers. Thus, Adaboost is not suitable for minutiae-based fingerprint identification. In this study, we decompose the alignment process to H steps, and each step uses $\frac{n}{H}$ minutiae of the input fingerprint I to perform the alignment with the template fingerprint in the database. The time-complexity of each step is $\circ(\frac{n}{H}n) = \circ(\frac{n^2}{H})$, and is much smaller than $\circ(n^2)$. What is more, the frontal steps can reject many fingerprints that do not belong to the same finger as the input fingerprint I, thus the number of one-to-one verifications of the posterior steps becomes smaller. The rule of rejection is: if the voting score $V[\Delta x, \Delta y, \theta]$ of Algorithm 1 is smaller than a pre-defined threshold, then the template fingerprint T is rejected and judged that it is not from the same finger as the input fingerprint. The whole identification process is summarized as Algorithm 2.

With the advantage of the hierarchical strategy, the proposed identification searching algorithm can reject many fingerprints that do not belong to the same finger as the input fingerprint at the frontal steps, and can save a lot of searching time. Fig. 1 illustrates the flowchart of the method. The following subsection will analyze the time consuming of the proposed searching algorithm.

B. Searching Time Analysis

Before analyzing the time consuming, some assumptions are given as follows:

1) There are N fingerprints in the database: $F_1, F_2, \cdots, F_N$.
2) For the hth step: the average one-to-one time is t_h, the rejection ratio (defined as the rejected fingerprints at this

Input: the minutiae sets of M_I (input fingerprint), and $M_{T_1}, M_{T_2}, \cdots, M_{T_N}$ (template fingerprints)
Output: the identifer ID of the matched fingerprint (if it is nor present, return $N+1$)
divide M_I into H sets $M_I^1, M_I^2, \cdots, M_I^H$ randomly, each contains $\frac{n_I}{H}$ minutiae, where n_I is the number of minutiae in M_I;
define threshold $T_1, T_2, \cdots, T_H, T_{H+1}$, which are learned from the training data;
for $j = 1 : N$ **do**
  for $i = 1 : H+1$ **do**
    // the $H+1$ step computing the matching score
    if $i \leq H$ **then**
      perform Algorithm 1 with input $M_I^1 \bigcup M_I^2 \bigcup \cdots \bigcup M_I^i$ and M_{T_j};
      if $V[\Delta x, \Delta y, \theta] < T_i$ **then**
        break;
      end
    end
    else
      compute matching score using (7);
      if $s_{matching} > T_{H+1}$ **then**
        $ID = j$;
        return;
      end
    end
  end
end
$ID = N+1$;

Algorithm 2: Pseudo-code of the hierarchical identification searching algorithm

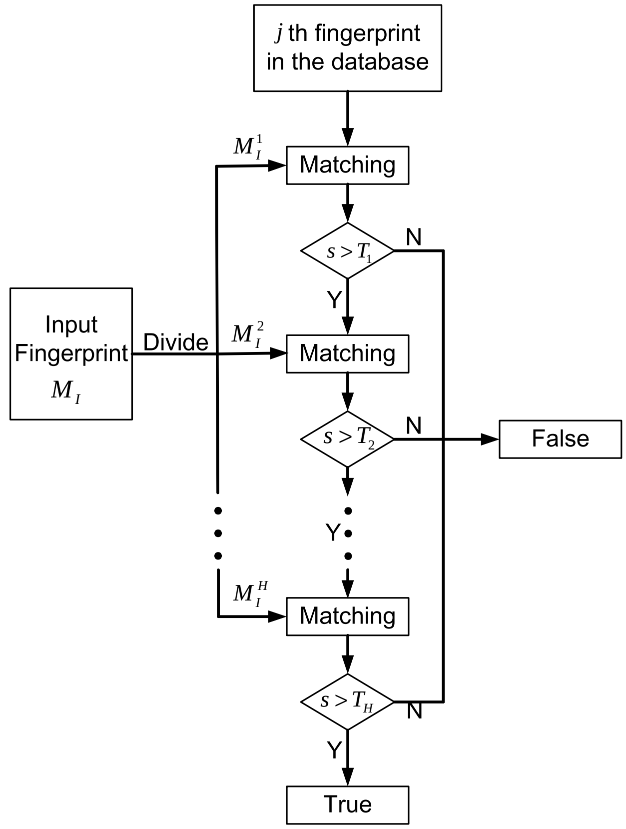

Fig. 1. Flowchart of the hierarchical identification searching algorithm.

step to all the fingerprints of this step) is α_h, the average searching time is T_h.

3) The probability of the input fingerprint belongs to the same finger as the nth fingerprint F_n is p_n. We do not consider the situation that the input fingerprint is not present in the database for simple computing. Suppose p_n is of uniform distribution, then:

$$p_n = \frac{1}{N}, \quad (n = 1, 2, \cdots, N). \tag{10}$$

4) The probability of the input fingerprint belongs to the same finger as the nth fingerprint F_n, and the searching result is the mth fingerprint (the result of Algorithm 2 is $ID = m$) is $q_{nm}(m = 1, 2, \cdots, N+1)$, then:

$$\sum_{m=1}^{N+1} q_{nm} = 1, \quad (n = 1, 2, \cdots, N). \tag{11}$$

Suppose the probability that $m = n$ (namely the searching result is right) is σ:

$$q_{nn} = \sigma, \quad (n = 1, 2, \cdots, N). \tag{12}$$

Suppose when $m \neq n$, q_{nm} is of uniform distribution, then:

$$q_{nm} = \frac{1-\sigma}{N}, (m \neq n, n = 1, 2, \cdots, N). \tag{13}$$

Summing up the above, there is:

$$q_{nm} = \begin{cases} \sigma, & m = n, \\ \dfrac{1-\sigma}{N}, & m \neq n, \end{cases} (n = 1, 2, \cdots, N). \tag{14}$$

For the hth step, when the input fingerprint belongs to the same finger as the nth template fingerprint F_n, and the searching result is the mth template fingerprint F_m, denote the searching time as t_{nm}. Through the frontal $h-1$ steps, the number of fingerprints that are not rejected is $m\prod_{i=0}^{h-1}(1-\alpha_i)$, where $\alpha_0 = 0$. t_{nm} is computed as

$$t_{nm} = \left(m \prod_{i=0}^{h-1} (1-\alpha_i) \right) t_h. \tag{15}$$

The average searching time T_h is as

$$T_h = \sum_{n=1}^{N} p_n \sum_{m=1}^{N+1} q_{nm} t_{nm}. \tag{16}$$

Using (15) to replace t_{nm}, we have

$$\begin{aligned} T_h &= \sum_{n=1}^{N} p_n \sum_{m=1}^{N+1} q_{nm} \left(m \prod_{i=0}^{h-1} (1-\alpha_i) \right) t_h \\ &= \left(\prod_{i=0}^{h-1} (1-\alpha_i) \right) t_h \sum_{n=1}^{N} p_n \sum_{m=1}^{N+1} q_{nm} m. \end{aligned} \tag{17}$$

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication.

According to (10), (14), there is

$$\begin{aligned} T_h &= \left(\prod_{i=0}^{h-1}(1-\alpha_i)\right) t_h \sum_{n=1}^{N} \frac{1}{N}\left(\sum_{m=1,m\neq n}^{N+1} \frac{1-\sigma}{N} m + \sigma n\right) \\ &= \frac{t_h}{2}\left(\prod_{i=0}^{h-1}(1-\alpha_i)\right)\frac{N+1}{N}(N+1-\sigma) \\ &\approx \frac{(N+1)}{2}\left(\prod_{i=0}^{h-1}(1-\alpha_i)\right) t_h. \end{aligned} \tag{18}$$

Suppose there are H steps in the alignments, the total searching time T_S is as

$$T_S = \sum_{h=1}^{H+1} T_h, \tag{19}$$

where the $(H+1)$th step is the matching score computing step. Denote the total alignment time for one-to-one as t_A, the average alignment time of each step for one-to-one is $\frac{t_A}{H}$, that is

$$t_h = \frac{t_A}{H}, \quad (h = 1, 2, \cdots, H). \tag{20}$$

Then the total identification searching time is as

$$\begin{aligned} T_S &= \sum_{h=1}^{H+1} T_h \\ &\approx \sum_{h=1}^{H+1} \frac{(N+1)}{2}\left(\prod_{i=0}^{h-1}(1-\alpha_i)\right) t_h \\ &= \frac{(N+1)}{2} t_A \left(\frac{1}{H}\sum_{h=1}^{H}\left(\prod_{i=0}^{h-1}(1-\alpha_i)\right)\right) \\ &\quad + \frac{(N+1)}{2} t_C \left(\prod_{i=0}^{H}(1-\alpha_i)\right), \end{aligned} \tag{21}$$

where $t_{H+1} = t_C$ is the matching score computing time for one-to-one. When N, t_A, and t_C are given, the total searching time is related to H, and $\alpha_i (i = 1, 2, \cdots, H)$. T_S has descending trend along with the increasing of H. However, when H increases, the verification ability of each step is decreasing, and thus the rejection ratio $\alpha_i (i = 1, 2, \cdots, H)$ may become smaller. This makes T_S have increasing trend. Thus, H should not be too large, and should be determined by experiment.

If $H = 3$, the total searching time is as

$$\begin{aligned} T_S &= \frac{(N+1)}{2} t_A \left(\frac{1}{3}\sum_{h=1}^{3}\left(\prod_{i=0}^{h-1}(1-\alpha_i)\right)\right) \\ &\quad + \frac{(N+1)}{2} t_C \left(\prod_{i=0}^{3}(1-\alpha_i)\right) \\ &= \frac{(N+1)}{2}\frac{t_A}{3}(1 + (1-\alpha_1) + (1-\alpha_1)(1-\alpha_2)) \\ &\quad + \frac{(N+1)}{2} t_C (1-\alpha_1)(1-\alpha_2)(1-\alpha_3). \end{aligned} \tag{22}$$

If the rejection ratio can reach 60%, the proposed identification algorithm can save more than 50% time.

The proposed hierarchical strategy can be also utilized to multi-feature identification, and the above analysis is also tenable.

IV. Experiment

A. Database

Experiments are conducted on two databases. The first database consists of 1422 fingers (with 8 fingerprints per finger), in which 827 fingers come from the THU database [32], while the other 595 fingers are from the elderly database that used in [33].

The second database is a high-resolution palmprint database, which consists of 1500 palms, with 8 palmprints per palm. The image size is 2040×2040 pixels with 500 ppi resolution and 256 grayscales. The database contains palmprints of different types and qualities. About 17% of them are of relatively poor quality due to large amounts of creases, deformity, smudges, blurs, incompleteness and etc.

For each database, we randomly selected 500 fingers or palms to form the training set, and the rest are used for testing. The training set is used to learn the threshold for each step. In each step, the threshold is determined to allow the right template (fingerprint or palmprint, from the same finger or palm as the input one) in the database to pass with the ratio of 99%. Not all the 8 impressions are needed for the testing experiment, since the identification systems uses one impression to identify one finger or palm (the other impression). Thus, we just use two impressions for a finger or a palm.

B. First One Searching

With the hierarchical strategy, for one template finger or palm, if the alignment score $V[\Delta x, \Delta y, \theta]$ in a step is smaller than the threshold, go to the next template finger or palm. Otherwise, if the algorithm reaches the final step, and has a matching score $s_{matching}$ greater than the threshold, the algorithm terminates and returns the identifer of this finger or palm. The results are shown in Table I. For the testing experiment on the fingerprint testing set, if we do not use the hierarchical strategy, namely $H = 1$, the right searching ratio (identification accuracy) is 91.8% with the searching time of 6.61 seconds. When we use the proposed hierarchical strategy ($H = 3$), the right searching ratio is 93.0%, and the searching time is cut down to 3.31 seconds which saves almost 50% time compared to the traditional method. The result illustrates the effectiveness of the proposed algorithm. What is more, the right searching ratio is also increased. Though the matching principles of hierarchical structure use the same minutiae set as non-hierarchical structure, they make use of the minutiae information in a quite different way. The reasons that the right searching ratio increases with applying the hierarchical matching structure are as below: In the hierarchical structure, each step can reject almost 60% of the template fingerprints that do not belong to the same finger as the input fingerprint. The rejection can reduce the affection of wrongly detected minutiae to the final matching result and thus increase the identification accuracy.

This article has been accepted for publication in a future issue of this journal, but has not been fully edited. Content may change prior to final publication.

TABLE I
THE PERFORMANCE (IDENTIFICATION RATE AND COMPUTATIONAL COST) OF THE HIERARCHICAL STRATEGY WITH H LEVELS, H=1 MEANS THE TRADITIONAL NON-HIERARCHICAL METHOD

ratio; time	$H = 1$	$H = 2$	$H = 3$	$H = 4$	$H = 5$	$H = 6$
THU fingerprint database	91.8% ; 6.61s	92.7% ; 4.29s	93.0% ; 3.31s	92.3% ; 3.16s	90.9% ; 3.15s	90.4% ; 3.10s
palmprint database	77.1% ; 7.50s	77.8% ; 4.78s	78.2% ; 3.67s	78.3% ; 3.57s	77.6% ; 3.52s	75.3% ; 3.43s

TABLE II
THE PERFORMANCE (ON NIST-4) OF THE HIERARCHICAL STRATEGY WITH H LEVELS, H=1 MEANS THE TRADITIONAL NON-HIERARCHICAL METHOD

ratio; time	$H = 1$	$H = 2$	$H = 3$	$H = 4$	$H = 5$	$H = 6$
NIST-4 fingerprint database	85.8% ; 11.04s	86.5% ; 6.68s	87.4% ; 6.03s	86.9% ; 5.69s	86.1% ; 5.43s	85.3% ; 5.17s

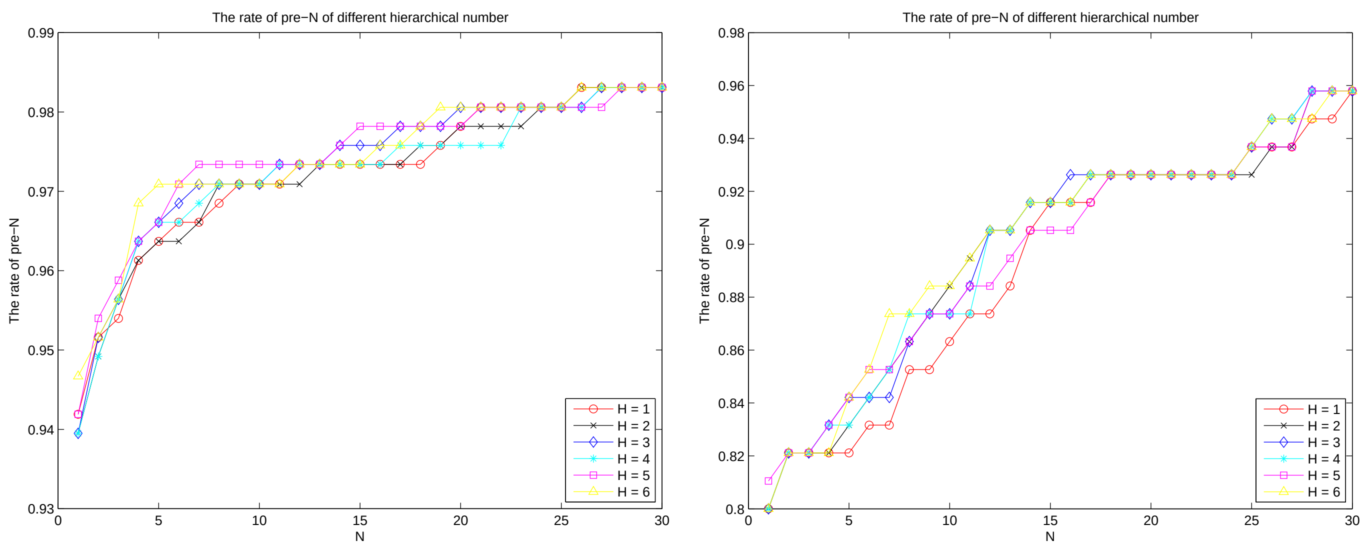

Fig. 2. The pre-N rate with different hierarchical number H on (a) fingerprint database and (b) palmprint database, respectively

As for the palmprint testing database, the results also show that the hierarchical strategy can save much time. When $H = 4$, the system performs the best considering both the ratio and time, comparing with $H = 3$. This is mainly because the number of minutiae in a palm is usually more than that in a finger. In real application, H should be selected according to the average minutiae number or the area of a fingerprint or palmprint.

The THU fingerprint database and palmprint database are both scanned by solid sensors. We also conducted experiments on a non-solid sensor fingerprint database: NIST Special Database 4 (NIST-4) [34]. NIST-4 contains 2000 pairs of rolled fingerprint images (two instances, f and s, for each entity). Table II shows the performance on NIST-4. The results confirm the effectiveness of the proposed hierarchical method, and this illustrates that we can use the hierarchical structure as a strategy to speed up the minutiae-based fingerprint identification.

C. Full Searching

In the above experiment, when the system reaches the final step, and has a matching score $s_{matching}$ greater than a predefined threshold, the algorithm terminates and returns the identifer of this finger or palm. It tries to find one fairish template (fingerprint or palmprint in the database of the system) and then stops to avoid wasting time. However, in some other applications, it needs to search the whole database of the system, and find the best N matching fingerprints or palmprints.

In this experiment, we change the algorithm to find the best N matching templates (fingerprints or palmprints). For one input template finger or palm, if the alignment score $V[\Delta x, \Delta y, \theta]$ in a step is smaller than the threshold, set $s^i_{matching} = 0$ (i denotes the ith template in the database of the system), then go to the next template finger or palm. Otherwise, if the algorithm reaches the final step, record the matching score $s^i_{matching}$. After searching all the templates, the system finds the N templates with the highest $s^i_{matching}$. If the right fingerprint or palmprint (the one comes from the same finger or palm as the input one) is in these N templates, it is said *right-in*. The pre-N rate is defined as the ratio of the *right-in* inputs to the all input fingerprints or palmprints. The pre-N result is shown in Fig. 2. From the result, we can obtain somes conclusion as: (1) if N is increasing, the pre-N rate raises greatly, even when $N < 15$; (2) Totally, the performance of different hierarchical number $H = 3, 4, 5, 6$ is better the that $H = 1, 2$. It illustrates the applicability of the proposed hierarchical algorithm.

D. Multi-feature Fingerprint Identification

Previous studies showed that the performance can be improved by combining multiple features. To show the effectiveness of the proposed hierarchical method, we utilized it to multi-feature fingerprint recognition. We first combined orientation field [35] to the hierarchical method, and the orientation field matching is set as the last step. The comparing results are shown in Table III. Compared with the hierarchical minutiae method, the identification accuracy of the hierarchical fusion (minutiae and orientation) method increases a lot (from 93.0% to 94.9%) with some additional time cost. Besides minutiae and orientation, we further fused FingerCode [36], and density map [37] as the last two steps, for the hierarchical multi-feature fingerprint identification. Table III shows that further accuracy improvement can be gained. The experimental results illustrate the effectiveness of the proposed hierarchical method.

TABLE III
THE RESULTS OF ADDING MORE FEATURES TO THE HIERARCHICAL METHOD

method	ratio	time
the conventional minutiae-based	91.8%	6.61s
the hierarchical minutiae	93.0%	3.31s
the hierarchical minutiae and orientation	94.9%	3.94s
the hierarchical multi-feature	95.7%	5.05s

E. Comparison

We compared the proposed method with some state-of-the-art methods: the localizing method [21] and the spectral minutiae algorithm [24]. In the localizing method, the number of minutiae was set as 10. In the spectral minutiae algorithm, both the CPCA and the LDFT feature reductions are applied. Table IV shows the comparing results. The proposed method is faster than the localizing method. And the accuracy of the proposed method is also higher than the localizing method. This is mainly because the localizing method relies on the accurate detection of reference points, and detection of the reference point is not satisfactory yet; furthermore, the reference point may not even be present in small-sized images [23]. Though the spectral minutiae algorithm is fastest among the three methods, it gains a low identification accuracy since it is sensitive to the minutiae quality as well as the fingerprint quality. Considering both the identification accuracy and the time cost, the proposed hierarchical strategy shows attractiveness for speedup of fingerprint identification.

TABLE IV
THE COMPARING RESULTS OF DIFFERENT METHODS

method	ratio	time
the localizing method [21]	89.6%	3.59s
the spectral minutiae algorithm [24]	84.5%	2.19s
the proposed hierarchical minutiae	93.0%	3.31s

V. CONCLUSION

In this paper, a novel algorithm for fingerprint and palmprint minutiae identification is proposed to save time. A hierarchical strategy is proposed and utilized in the matching stage. The hierarchical strategy can reject many fingerprints (in the database of the AFIS) which do not belong to the same finger as the input fingerprint quickly, thus it can save much time. Experimental results show that the proposed algorithm can save almost 50% searching time compared to the traditional method and illustrate its effectiveness. Not as the conventional methods based on classification and indexing, the proposed method does not use more features or information than minutiae, and it can be integrated to the conventional identification systems in future.

ACKNOWLEDGEMENT

This work was supported by the Natural Science Foundation of China under Grant 61203263, by the National 973 Program of China under Grant 2011CB707802. The work of Jie Zhou was supported by the National Natural Science Foundation of China under Grants 61225008, 61020106004, 61005023, and 61021063, and by the Ministry of Education of China under Grant 20120002110033.

REFERENCES

[1] D. Zhang, *Automated biometrics: Technologies and systems*. Kluwer Academic Publishers, New York, 2000.
[2] A. K. Jain, R. M. Bolle, and S. Pankanti, *Biometrics: Personal Identification in Networked Society*. Springer, October 2005.
[3] A. K. Jain, P. Flynn, and A. A. Ross, *Handbook of biometrics*. Springer-Verlag New York Inc, 2007.
[4] D. Maltoni, D. Maio, A. K. Jain, and S. Prabhakar, *Handbook of Fingerprint Recognition (Second Edition)*. Springer, 2009.
[5] A. W. Kong, D. D. Zhang, and M. S. Kamel, "A survey of palmprint recognition," *Pattern Recognition*, vol. 42, no. 7, pp. 1408–1418, 2009.
[6] A. K. Hrechak and J. A. McHugh, "Automated fingerprint recognition using structural matching," *Pattern Recognition*, vol. 23, no. 8, pp. 893–904, 1990.
[7] A. K. Jain, L. Hong, and R. M. Bolle, "On-line fingerprint verification," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 19, no. 4, pp. 302–314, 1997.
[8] A. K. Jain and J. Feng, "Latent palmprint matching," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 31, no. 6, pp. 1032–1047, 2009.
[9] N. K. Ratha, K. Karu, S. Chen, and A. K. Jain, "A real-time matching system for large fingerprint databases," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 18, no. 8, pp. 799–813, 1996.
[10] X. Tan, B. Bhanu, and Y. Lin, "Fingerprint identification: classification vs. indexing," in *Proceedings. IEEE Conference on Advanced Video and Signal Based Surveillance, 2003.*, 2003, pp. 151–156.
[11] A. K. Jain, S. Prabhakar, and L. Hong, "A multichannel approach to fingerprint classification," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 21, no. 4, pp. 348–359, 1999.
[12] C. Klimanee and D. T. Nguyen, "Classification of fingerprints using singular points and their principal axes," in *ICIP*, 2004, pp. 849–852.
[13] S. Shah and P. S. Sastry, "Fingerprint classification using a feedback-based line detector," *IEEE Transactions on Systems, Man, and Cybernetics, Part B*, vol. 34, no. 1, pp. 85–94, 2004.
[14] J.-H. Hong, J.-K. Min, U.-K. Cho, and S.-B. Cho, "Fingerprint classification using one-vs-all support vector machines dynamically ordered with naive bayes classifiers," *Pattern Recognition*, vol. 41, no. 2, pp. 662–671, 2008.
[15] N. Yager and A. Amin, "Fingerprint classification: a review." *Pattern Anal. Appl.*, vol. 7, no. 1, pp. 77–93, 2004.
[16] R. Germain, A. Califano, and S. Colville, "Fingerprint Matching Using Transformation Parameter Clustering," *IEEE Computational Science & Engineering*, vol. 4, no. 4, pp. 42–49, 1997.

[17] B. Bhanu and X. Tan, "Fingerprint indexing based on novel features of minutiae triplets," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 25, no. 5, pp. 616–622, 2003.
[18] S. Chikkerur, S. Pankanti, A. Jea, N. K. Ratha, and R. M. Bolle, "Fingerprint representation using localized texture features," in *ICPR (4)*, 2006, pp. 521–524.
[19] R. Cappelli, M. Ferrara, and D. Maltoni, "Fingerprint indexing based on minutia cylinder-code," *IEEE Transactions on Pattern Analysis and Machine Intelligence.*, vol. 33, no. 5, pp. 1051–1057, 2011.
[20] M. Liu, X. Jiang, and A. C. Kot, "Efficient fingerprint search based on database clustering," *Pattern Recognition*, vol. 40, no. 6, pp. 1793–1803, 2007.
[21] K. C. Chan, Y. Moon, and P. Cheng, "Fast fingerprint verification using subregions of fingerprint images," *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 14, no. 1, pp. 95–101, 2004.
[22] J. Ying, Y. Yuan, R. Zhang, and Q. Song, "Fingerprint minutiae matching algorithm for real time system," *Pattern Recognition*, vol. 39, no. 1, pp. 143–146, 2006.
[23] A. Ross, A. K. Jain, and J. Reisman, "A hybrid fingerprint matcher," *Pattern Recognition*, vol. 36, no. 7, pp. 1661–1673, 2003.
[24] H. Xu, R. N. Veldhuis, T. A. Kevenaar, and T. A. Akkermans, "A fast minutiae-based fingerprint recognition system," *IEEE Systems Journal*, vol. 3, no. 4, pp. 418–427, 2009.
[25] R. Schapire, Y. Freund, P. Bartlett, and W. Lee, "Boosting the Margin: A New Explanation for the Effectiveness of Voting Methods," *The Annals of Statistics*, vol. 26, no. 5, pp. 1651–1686, 1998.
[26] P. A. Viola and M. J. Jones, "Robust real-time face detection," *International Journal of Computer Vision*, vol. 57, no. 2, pp. 137–154, 2004.
[27] J. Wu, S. C. Brubaker, M. D. Mullin, and J. M. Rehg, "Fast asymmetric learning for cascade face detection," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 30, no. 3, pp. 369–382, 2008.
[28] X. Jiang, M. Liu, and A. C. Kot, "Fingerprint retrieval for identification," *Information Forensics and Security, IEEE Transactions on*, vol. 1, no. 4, pp. 532–542, 2006.
[29] N. Yager and A. Amin, "Fingerprint verification based on minutiae features: a review," *Pattern Analysis & Applications*, vol. 7, no. 1, pp. 94–113, 2004.
[30] D. H. Ballard, "Generalizing the hough transform to detect arbitrary shapes," *Pattern Recognition*, vol. 13, no. 2, pp. 111–122, 1981.
[31] J. Ren, X. Jiang, and J. Yuan, "A complete and fully automated face verification system on mobile devices," *Pattern Recognition*, vol. 46, no. 1, pp. 45–56, 2013.
[32] J. Gu, J. Zhou, and C. Yang, "Fingerprint recognition by combining global structure and local cues," *IEEE Transactions on Image Processing*, vol. 15, no. 7, pp. 1952–1964, 2006.
[33] J. Zhou, F. Chen, N. Wu, and C. Wu, "Crease detection from fingerprint images and its applications in elderly people," *Pattern Recognition*, vol. 42, no. 5, pp. 896–906, 2009.
[34] C. Watson and C. Wilson, "NIST Special Database 4," *Fingerprint Database, National Institute of Standards and Technology*, vol. 17, 1992.
[35] X. Jiang, "On orientation and anisotropy estimation for online fingerprint authentication," *IEEE Transactions on Signal Processing*, vol. 53, no. 10, pp. 4038–4049, 2005.
[36] A. K. Jain, S. Prabhakar, L. Hong, and S. Pankanti, "Filterbank-based fingerprint matching," *IEEE Transactions on Image Processing*, vol. 9, no. 5, pp. 846–859, 2000.
[37] D. Wan and J. Zhou, "Fingerprint recognition using model-based density map," *IEEE Transactions on Image Processing*, vol. 15, no. 6, pp. 1690–1696, 2006.

Fanglin Chen received the B.S. degree of Engineering in Automatic Control Science and Technology, and B.S. degree (dual degree) of Science in Applied Mathematics, from Xian Jiaotong University, Xian, China, in 2006. Then he received his M.S. and Ph.D. degrees from Department of Automation, Tsinghua University, Beijing, China, in 2008 and 2011, respectively. He has been on the faculty in the Department of Automatic Control at National University of Defense Technology, Changsha, China. His research interests include pattern recognition, classifier fusion, machine learning, computer vision, biometrics, neuroimage, and BCI. He received the Academic Newcomer Award (Ministry of Education of China) in 2010, the First Prize of Electronic Information Science and Technology Award (Chinese Institute of Electronics) in 2011, the Second Prize of Natural Sciences (Ministry of Education of China) in 2012, the Second Group of Robot Vision Challenge (ImageCLEF) in 2013. He is a member of the IEEE.

Xiaolin Huang received the B.S. degree in control science and engineering, and the B.S. degree in applied mathematics from Xi'an Jiaotong University, Xi'an, China in 2006. In 2012, he received the Ph.D. degree in control science and engineering from Tsinghua University, Beijing, China. Since then, he has been working as a postdoctoral researcher in ESAT-SCD-SISTA, KU Leuven, Leuven, Belgium. His current research areas include optimization, classification, and identification for nonlinear systems via piecewise linear analysis.

Jie Zhou received B.S. degree and M.S. degree both from Department of Mathematics, Nankai University, Tianjin, China, in 1990 and 1992, respectively. He received Ph.D. degree from Institute of Pattern Recognition and Artificial Intelligence, Huazhong University of Science and Technology (HUST), Wuhan, China, in 1995. From then to 1997, he served as a postdoctoral fellow in Department of Automation, Tsinghua University, Beijing, China. From 2003, he has been a full professor in Department of Automation, Tsinghua University. His research area includes computer vision, pattern recognition and image processing. In recent years, he has authored more than 100 papers in peer-reviewed journals and conferences. Among them, more than 30 papers have been published in top journals and conferences such as PAMI, T-IP and CVPR. He is an associate editor for International Journal of Robotics and Automation, Acta Automatica and two other journals. Dr. Zhou is a senior member of IEEE and a recipient of the National Outstanding Youth Foundation of China.