Approximate Fingerprint Matching Using Kd-tree

Partha Bhowmick
Computer Science & Technology Dept.
Bengal Engineering College (DU)
Shibpur, Howrah 711103, India
partha@becs.ac.in

Bhargab B. Bhattacharya
Advanced Computing & Microelectronics Unit
Indian Statistical Institute
203 B. T. Road, Kolkata 700108, India
bhargab@isical.ac.in

Abstract

Fast and robust fingerprint matching is a challenging task today in fingerprint-based biometric systems. A fingerprint matching algorithm compares two given fingerprints and returns either a degree of similarity or a binary decision. Minutiae-based fingerprint matching is the most well-known and widely used method [11]. This paper reveals a new technique of fingerprint matching, using an efficient data structure, combining the minutiae representation with the individual usefulness of each minutia, to make the matching more powerful. Experimental results exhibit the strength of this method.

1. Introduction

Given an existing database representation of a fingerprint image and a similar input representation extracted from a query image, the matching stage in a fingerprint verification system determines the similarity of the two fingerprint representations and decides whether they are from the same finger. The most elegant and efficient representation of fingerprints, which is also adopted in the conventional automatic fingerprint identification systems (AFIS), is based on a common hypothesis called minutiae owing to their unique ability to capture the invariant and discriminatory information present in a fingerprint image. In this work, we have considered two most prominent kinds of minutiae, namely, ridge termination and ridge bifurcation, which are used by the Federal Bureau of Investigation [14] and also adopted in many AFIS. There exist several techniques [3, 7, 10] to extract the minutiae from a gray-scale fingerprint image.

Being a non-ideal mapping of some part of a three-dimensional finger to a two-dimensional plane, the acquired fingerprint image, and its corresponding representation there of, inevitably suffers from unpredictable complications, some of which are: (i) spurious minutiae may creep in and genuine minutiae may drop during minutiae detection phase; (ii) global translation, rotation, and scaling of the minutiae pattern due to unknown alignment of the finger during image acquisition; (iii) local non-linear deformations due to uneven pressure at the fingertip or/and digitization error in the acquisition mechanism; (iv) reduction in actual region of interest due to partial overlap between the two impressions; (v) cut marks and skin imperfections developed in the time span elapsed between acquisitions of the two impressions. Considering all these factors, a necessary foundation for achieving a good matching performance is to construct a reliable, realistic and robust model of fingerprint matching that takes into account all sort of deviations and anomalies, which are very likely to occur between two sets of minutiae extracted from different impressions of the same finger. A number of approaches have been proposed, and most of them are by minutiae matching based on some variety of point pattern matching [5], and by structural matching [4, 13]. However, even these methods fail to perform well [6] because they did not make use of the rich information content present in a fingerprint pattern. Presently, most fingerprint matching algorithms follow a dual strategy that combines the minutiae matching with some method that captures the ridge structure properties in order to improve the overall matching performance [6, 8, 9].

2. Proposed method

In this paper, we present a new fingerprint matching technique that exploits both the local topological structures of a valid minutia and the global geometric structure of the minutiae set as a whole. Both the local and global structures have been used adaptively in our algorithm, making the matching procedure more meaningful, efficient, and robust. Though our method also follow a dual strategy by combining both the local and global perspectives, it is notably different from the existing methods [6, 8, 9] as explained below.

We describe each minutia point \( p_i \), detected from a fingerprint image, by a 6-element feature vector given by:

\[
p_i = (x_i, y_i, t_i, \phi_i, \lambda_i, s_i)
\]

where, \( x_i, y_i \) are the rectangular coordinates of \( p_i \) w.r.t. image frame; \( t_i \) is the type of minutia (ridge termination
or bifurcation); \( \phi_i \in [0, 360) \) is the local ridge direction at \( p_i \), measured in the counterclockwise direction w.r.t. +ve x-axis; \( \lambda_i \) is the local inter-ridge distance at \( p_i \); \( s_i \) is the score associated with \( p_i \). \( s_i \) signifies the degree of authenticity of \( p_i \) in its local neighborhood in the respective fingerprint image. An elaborate procedure to estimate \( s_i \) for each \( p_i \) has been discussed in [2]. Depending on the ridge and valley characteristics and the prevailing noise level in the local topological neighborhood of \( p_i \), its score \( s_i \) is estimated based on \( t_i \), \( \phi_i \), and \( \lambda_i \), and the value is normalized within 1-100 scale so that a higher value of \( s_i \) implies the genuineness of \( p_i \) in its way to participate in any matching algorithm. Our consideration of local structural characteristics of a minutia differs from the existing ones [6, 8, 9].

In [6], grayscale variance within a tessellated cell quantifies the underlying ridge structure and is used as a feature; in [9], only the associated ridge on which the minutia lies is considered; in [8], the inter-minutia distance and ridge count are considered as local structural measure to facilitate the preliminary matching. On the contrary, in this work, for all detected minutiae, we have considered their scores, which embrace all the relevant and useful topological properties and discard the detrimental ones, thereby enriching the minutia-based matching by final evaluation of a matching score. An example of detected set of minutiae with positive scores is shown in Fig.1, where, the minutiae with positive scores [2] are encompassed by circles, the radius of the circle centred at a minutia being proportional to its score, and label ‘i’ of a minutia corresponds to the leaf node ‘Pi’ in \( K_k \) in Fig.2.

Let there are \( N \) images in the database, and \( \mathcal{F}_k, 1 \leq k \leq N \), represents the \( k \)th database image having \( n_k \) minutiae represented by the feature set \( S_k = \{ p_i \}_{i=1}^{n_k} \). In the preprocessing (offline) phase, a primary data structure \( \mathcal{T}_k \) is defined over each \( S_k \). \( \mathcal{T}_k \) is an AVL tree that permits 1-dimensional range searching [1] in fastest possible time. The Euclidean distance \( d(p_i, p_j) \) between two distinct minutiae \( p_i \) and \( p_j \) is stored in a node \( \nu_{ij} \) of \( \mathcal{T}_k \), which has, therefore, \( \binom{n_k}{2} \) nodes in total. Apart from the usual attributes present in an AVL tree, each node \( \nu_{ij} \) contains two additional links pointing to the corresponding feature vectors of \( p_i \) and \( p_j \) present in \( S_k \).

When a query image \( \mathcal{F}_q \) comes in the online stage, it is first processed to extract the minutiae feature set \( S_q = \{ q_i \}_{i=1}^{n_q} \), where, each minutia \( q_i \) is represented conforming to Eqn. 1. The extracted set of minutiae are enumerated in \( S_q \) in nonincreasing order of their score values, which is necessary to systematically generate all \( \binom{n_q}{2} \) inter-minutia Euclidean distances in an ordered list \( \mathcal{L}_q = \{ d(q_i, q_j) \}_{1 \leq i < j \leq n_q} \). Each entry \( d(q_i, q_j) \) in \( \mathcal{L}_q \) additionally has two links pointing to the corresponding minutiae \( q_i \) and \( q_j \) occurring in \( S_q \). The ordering of the inter-minutia distances in \( \mathcal{L}_q \) ensures the maximum likelihood of finding a matching distance(s) in \( T_k \) in the soonest possible time, without considering a core point. A core point may not be always present, or may be present in a noisy region, in a given fingerprint impression.

It is very likely that, if \( \mathcal{F}_k \) and \( \mathcal{F}_q \) are from the same finger, then in their overlapped region, a minutia with high score in one is present in the other, though in the later it may or may not have a high score. Hence, it will be wise to search for those minutiae which are having high scores in \( S_q \) into the database set \( S_k \). Based on this novel idea, the search space in the parameter domain is reduced.

### 2.1. Range query of a minutia in a Kd-tree

A Kd-tree stores a set of points in K-dimensional space, and a 2-dimensional Kd-tree can be used for partial match queries in \( O(\sqrt{n} + m) \) time, where, \( n \) is the number of points stored, and \( m \) is the number of (partial) matches reported in the query [1]. The elegance of Kd-tree, say 2-dimensional, meant for a range query, is that, the query on the Kd-tree may be of polygonal shapes like triangle, rectangle, hexagon, octagon, etc., and as the number of edges of the query polygon increases, the shape gradually approximates a circle in the discrete domain since the number of axes of symmetry of the polygon increases, thereby making the searching an ideal one, without compromising with
the linear size of the data structure; however, with the increase in number of edges of the query polygon, the dimensionality of the Kd-tree also increases, thereby slowing down the search operations. Considering all these factors, we have resorted to the 2-dimensional Kd-tree for orthogonal range searching in our minutiae matching algorithm.

In this work, we consider the coordinates of the minutiae for dimensional splitting while storing them in a Kd-tree. For each database image $F_k$, $1 \leq k \leq N$, we maintain a secondary data structure $K_k$, which is a 2-dimensional Kd-tree containing $n_k$ leaf nodes, where, each leaf node of $K_k$ represents a distinct feature vector of $S_k$. The structure of $K_k$ for the image in Fig.1 is shown in Fig.2, where, a non-leaf node splits the preceding region by an appropriate axis in our experiments and also agreed to by others [6, 8], depending on the current condition. For each such possibility, starting from the beginning of $L_k$, till $\frac{1}{\lambda}(s_{\ell'} + s_{j'}) \geq \overline{\pi}_k$, evaluate the average $\lambda$ across the respective pair of minutiae: $\lambda_{\ell,j'} = \frac{1}{\lambda}(s_{\ell'} + s_{j'})$, and the corresponding tolerance: $\epsilon_{\ell,j'} = \frac{1}{\lambda}(\lambda_{\ell,j'} - s_{\ell'})$, and search (range query) for every possible match $d(p_i, p_j)$ with $\lambda_{ij} = \frac{1}{\lambda}(\lambda_{\ell,j'} - s_{\ell'})$ in $T_k$, such that:

$$d(p_i, p_j) - \epsilon_{\ell,j'} \leq \frac{d(p_i, p_j)}{\lambda_{\ell,j'}} \leq d(p_i, p_j) + \epsilon_{\ell,j'}$$

Let $L_{kij} = \{d(p_i, p_j), d(p_i, p_{j'}), \ldots\}$ be the ordered sequence of the matching distances found in $T_k$, where, $(s_{i1} + s_{j1}) \geq (s_{i2} + s_{j2}) \geq \ldots$. For each such possible match $d(p_i, p_j) \in L_{kij}$, starting from the beginning of $L_{kij}$, if $[(\phi_i - \phi_j) - (\phi_j - \phi_j')] \leq 2\pi_0^0$ (angular tolerance in our experiments and also agreed to by others [6, 8]), then transform the entire query set $S_l$, such that $x_{\ell'} = x_i$ and $y_{\ell'} = y_j$, i.e., the query vector $q_{\ell'}q_{j'}$ exactly fits the matching database vector $p_ip_j$. Eqn. 3 provides the necessary translation $\Delta x$ along x-axis and $\Delta y$ along y-axis, rotation $\Delta \phi$, and scaling $\Delta \zeta$, depending on the current match.

$$\begin{bmatrix} \Delta x \\ \Delta y \\ \Delta \phi \\ \Delta \zeta \end{bmatrix} = \begin{bmatrix} \frac{1}{2}(x_i + x_j) - \frac{1}{2}(x_{\ell'} + x_{j'}) \\ \frac{1}{2}(y_i + y_j) - \frac{1}{2}(y_{\ell'} + y_{j'}) \\ \tan^{-1}\left(\frac{y_i - y_j}{x_i - x_j}\right) - \tan^{-1}\left(\frac{y_{\ell'} - y_{j'}}{x_{\ell'} - x_{j'}}\right) \\ \frac{d(p_i, p_j)}{d(q_{\ell'}, q_{j'})} \end{bmatrix}$$

Step 3. After all points in $S_l$ are given the subject transformation to obtain a new ordered feature set $\hat{S}_l = \{q_{m}, q_{2}, \ldots, q_{n_l} \mid s_1 \geq s_2 \geq \ldots \geq s_{n_l}\}$, for each $q_k = (x_{\ell'}, y_{\ell'}, \ell', j', \lambda_k, \delta_k) \in \hat{S}_l$, apply a 2-dimensional range searching in $K_k$ with an orthogonal square query box with the diagonal dimension $\lambda_k$ and centered about the coordinates $(x_{\ell'}, y_{\ell'})$ of $q_k$. We keep a counter $m(i \mapsto \ell', j \mapsto j')^{(k)}$, initialized to 0, which gets incremented by unity if a match $p_{\ell'}, 1 \leq \ell' \leq n_k$, is found.

Step 4. For each matching pair of minutiae $(p_\ell, p_j \in S_k, q_{\ell', q_{j'}} \in \hat{S}_l)$ obtained in step 3, a matching score function $\xi(s_u, \delta_u)$ is defined by Eqn. 4, which contributes into the final overall matching score $\mu^{(k)}$ between the two corresponding feature sets $S_k$ and $\hat{S}_l$ as shown in Eqn. 5.

$$\xi(s_u, \delta_u) = \frac{1}{2}(s_u + \delta_u) - \omega|s_u - \delta_u|$$

$$\mu^{(k)} = \frac{1}{m(i \mapsto \ell', j \mapsto j')^{(k)}} \sum_{u=1}^{m(i \mapsto \ell', j \mapsto j')^{(k)}} \xi(s_u, \delta_u)$$

Step 5. Reverse the roles of $q_{\ell'}$ and $q_{j'}$ in step 2 such that $x_{\ell'} = x_j$ and $y_{\ell'} = y_j$, i.e., the query vector $q_{\ell'}q_{j'}$ exactly fits the matching database vector $p_ip_j$, and repeat the process. If the feature set matching score, thus obtained, is greater than the previous one, then update the final matching score given in Eqn. 5 accordingly.

Step 6. If max $\{m(i \mapsto \ell', j \mapsto j')^{(k)}\}$ is at least 6, and $\mu^{(k)}$ is at least as high as $\min(\overline{\pi}_k, \overline{\pi}_l)$, then $F_l$ is reported as a matching image of $F_k$ with a matching score of $\mu^{(k)}$.

There are several intrinsic adaptive properties of this algorithm. Eqn. 2 takes care of non-linear deformations in the region lying in between and around the minutiae pairs. The condition $\frac{1}{\lambda}(s_{\ell'} + s_{j'}) \geq \overline{\pi}_k$ ensures that, for registration of $F_l$ with $F_k$, a minutia pair having a high degree of authenticity in $F_l$ is considered, thereby encouraging a real match and discouraging a faked one. In step 3, since there can not lie two minutiae $p_{\ell'}$ and $p_{j'}$ within a distance of $\min(\lambda_{\ell'}, \lambda_{j'})$ (if so, they have been already eliminated as spur or bridge endpoints in the minutiae detection stage), there can not be more than one minutia in the database set that lies within the range query box. As a result, if there is (at most) one minutia $p_{\ell'} \in S_k$ lying within the query box, then $p_{\ell'}$ is considered as a valid match with the corresponding query $q_{\ell'}$, provided they are of identical types, and their directions differ by at most $2\pi_0^0$. In Eqn. 4, the parameter $\omega$ represents the weightage attached to the difference of scores of two matching minutiae in their way of participating in the estimate of $\mu^{(k)}$. It can be shown that the value of $\omega$ should lie between 0 and $\frac{10}{100}$ so that $\xi(s_u, \delta_u)$ is never negative. In our experiments, we have considered $\omega = \frac{1}{2}$. 

0-7695-2128-2/04 $20.00 (C) 2004 IEEE
3. Experimental results

We used fingerprint images from (i) NIST Special Database 4 [12] (size 480 × 512, 500 dpi resolution each), and (ii) Database dB1a of FVC2000 [11] (300 × 300, 500 dpi each). Set (i) contains 100 images, captured from 50 different fingers, 2 images per finger, and set (ii) contains 800 images from 100 fingers, 8 images per finger.

The proposed method is implemented in C on a Sun_Ultra 5_10, Sparc, 233 MHz, the OS being the SunOS Release 5.7 Generic. The average execution time for both the sets are given in Table 1. The times shown in braces are required when minutiae scores are not considered, which are substantially higher than those, shown outside braces, when scores are included. Fig. 3 shows the different Receiver Operating Characteristic (ROC) curves by plotting the Authentic Acceptance Rate (AAR) vs. the False Acceptance Rate (FAR), for the two databases. The firm lines (type 1) represent the ROC when the individual scores of both the database image and the query image are taken into consideration, whereas, the dotted lines (type 2) the ROC when the conventional method is followed without considering the local topological properties (scores) of the minutiae. Type 1 curves are obtained by choosing different threshold values on number of matching minutiae pairs (default value is 6 in our experiments) and on matching score (default $\min(\sigma_k, \sigma_l)$), and type 2 curves by different threshold values only on number of matching minutiae pairs.

4. Conclusion and future works

This work elucidates a novel fingerprint matching technique that employs an efficient data structure and exploits the local topological properties of each minutia to strengthen a fingerprint matching process. Improvements on execution time, AAR and FAR may be still possible by further experiments on shape of the query box by higher dimensional Kd-tree, optimization on area covered under the query box, and refining the minutia score finding method. Furthermore, research provision also lies in automatic thresholding of parameters, which play a vital role in the matching process.

<table>
<thead>
<tr>
<th></th>
<th>NIST 4</th>
<th>FVC dB1</th>
</tr>
</thead>
<tbody>
<tr>
<td>% of invalid images (&lt; 6 minutiae)</td>
<td>6.00</td>
<td>4.36</td>
</tr>
<tr>
<td>Avg. time (secs.) for acceptance</td>
<td>0.10 (0.79)</td>
<td>0.06 (0.51)</td>
</tr>
<tr>
<td>Avg. time (secs.) for rejection</td>
<td>0.07 (1.02)</td>
<td>0.04 (0.64)</td>
</tr>
</tbody>
</table>

Table 1. Results for 2 sets of images

References