Proceedings of the 5th WSEAS International Conference on Signal Processing, Istanbul, Turkey, May 27-29, 2006 (pp234-237)

An Alignment Based Fingerprint Matching Algorithm

Liu Wei[1], Yan Puliu[1], Xia Delin[1], Zhou Cong[2],
College of Electronic Information[1], Department of Civil Engineering[2]
Wuhan University[1], Hubei University of Technology[2]
Wuhan, 430000, Hubei
China

Abstract: - In this paper an enhanced alignment based matching algorithm for fingerprint verification is discussed which modified Jain et al.'s algorithm. A new method is introduced to evaluate the correlation of two ridges. A matching-origin-pair searching algorithm based on such method is also presented accordingly. Experiments done on a set of fingerprint images show it is effective, fast and has high accuracy in fingerprint verification.
Key Words: - matching; alignment; bounding box; corresponding pair

1. Introduction

Among all the biometric indicators, fingerprint have one of the highest levels of reliability [1,3] and have been extensively used. In automated fingerprint identification, the performance is often dictated by the matching technique being employed.

A.K.Jain proposed an alignment-based matching algorithm in [1], in which the ridges associated with the minutiae were used to align the input minutiae with the template minutiae and an adaptive elastic matching algorithm was proposed to match the aligned minutiae. However, this approach results in a large template size. Tian Jie[4] improved Jain's algorithm by introduce ridge information into the process of matching and use a changeable sized box in the matching process, but the algorithm can't get the accurate corresponding pairs and need too much time for alignment.

The algorithm proposed in this paper is originated from the algorithm of Jain. But the algorithm employed new minutiae alignment algorithm and new ridge comparable evaluation method to accelerate the matching speed without increasing FNMR or FMR.

Section 2 introduced the matching method in detail. Section 3 gives the experimental results and the performance. We draw conclusion in section 4.

2. Minutia matching

For every minutia candidate, the following parameters are recorded.

① x and y coordinate of the point.
② orientation which is defined as the local ridge orientation of the associated ridge.
③ the type of the point: ridge ending or bifurcation.
④ the direct link grade with the other minutia in the neighborhood.

Define 1: if two minutia are linked in the framework and at least there's no other minutiae in one path, they are direct linked. The direct link grade is the path numbers of the direct link.

2.1 Alignment of minutiae set

Let $I=\left((x_1^I,y_1^I,\theta_1^I,t_1^I,l(r)_1^I),\ldots,(x_M^I,y_M^I,\theta_M^I,t_M^I,l(r)_M^I)\right)$

denote the set of I minutiae in the input image and $T=\left((x_1^T,y_1^T,\theta_1^T,t_1^T,l(r)_1^T),\ldots,(x_N^T,y_N^T,\theta_N^T,t_N^T,l(r)_N^T)\right)$ denote the set of T in the template image. $l(r)_1$ represents the direct link grade in the minutiae's neighborhood.

$$l(r)_1 = l_{End}(r)_i + l_{Bif}(r)_i \quad 1\le i\le M \text{ or } 1\le i\le N \qquad (1)$$

In formula 1, $l_{End}(r)_1$ is the sum of all endings link grade with point i in neighborhood r. And $l_{Bif}(r)_1$ is the sum grade of all bifurcation link grade with point i in neighborhood r.

We can get the ridge information by sampled it equidistance. The distance is the average distance of the ridges $\bar{\lambda}$. But many ridges look like beeline by the influence of $\bar{\lambda}$ and the numbers of sampling points. Algorithms in [1] and [4] can't resolve the question well, and the fake corresponding pairs are easy to be taken. So we employed 3 new parameters to justify the comparability of the ridges. For the ridge of input minutiae $(x_m^I, y_m^I, \theta_m^I, t_m^I, l(r)_m^I)$ and the ridge of template minutiae $(x_n^T, y_n^T, \theta_n^T, t_n^T, l(r)_n^T)$:

① $\sigma_1(m,n)$ is the change of the distance between the minutiae and the sampling points.

$$\sigma_1(m,n) = \sqrt{\sum_{i=1}^{N_{mn}} a_i \left(\frac{d_i^I}{d_i^T} - \tau\right)^2 \Bigg/ \sum_{i=1}^{N_{mn}} a_i} \qquad (2)$$

a_i is the weight increasing with i and $a_i > 0$. d_i^I and d_i^T are the distance of the minutiae and the sampling point in input image and template. See figure 1. τ is an amplification parameter, always $\tau = 1$. $N_{mn} = \min(N_m^I, N_n^T)$.

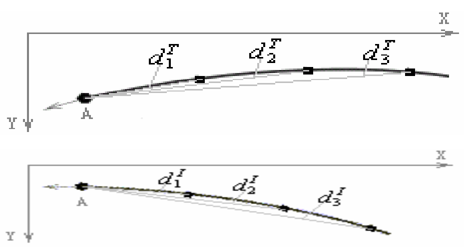

Figure 1: ridge in template and the input image

② ϕ_i^I、ϕ_i^T are the angles between x-axis and the line of minutiae and sampling point. $\sigma_2(m,n)$ is the change of the angles.

$$\sigma_2(m,n) = \sqrt{\frac{\sum_{i=1}^{N_{mn}} a_i[(\phi_i^I - \theta_m^I) - (\phi_i^T - \theta_n^T)]^2}{\sum_{i=1}^{N_{mn}} a_i}} \qquad (3)$$

③ $v(m,n)$ is the mark of similarity.

$$v(m,n) = \frac{C(k_1 \cdot \sigma_1(m,n) + k_2 \cdot \frac{\sigma_2(m,n)}{\pi})}{N_{mn}} \qquad (4)$$

In formula 4, C is a positive number. k_1, k_2 are the weights of (1) and (2), $k_1, k_2 \ge 1$. The ridges are more comparable while the mark is less.

Here we give the alignment algorithm of the corresponding pairs with the parameters above.

① An empty candidate pairs set is built, and the max length is L.

② Chose $I[m] = (x_m^I, y_m^I, \theta_m^I, t_m^I, l(r)_m^I)$ from input set and $T[n] = (x_n^T, y_n^T, \theta_n^T, t_n^T, l(r)_n^T)$ from template set.

③ If $t_m^I = t_n^T$ and $l(r)_m^I = l(r)_n^T$, go to ④, else back to ②.

④ If $\sigma_1(m,n) < T_1$ and $\sigma_2(m,n) < T_2$, go to ⑤, else back to ②. T_1 and T_2 are two thresholds.

⑤ Add the two minutia into the candidate set and compute the $v(m,n)$. If the set is full, delete the pair with max $v(m,n)$.

2.2 Minutia matching

Proceedings of the 5th WSEAS International Conference on Signal Processing, Istanbul, Turkey, May 27-29, 2006 (pp234-237)

Giving $(x_o^I, y_o^I, \theta_o^I, t_o^I, l(r)_o^I)$, $(x_o^T, y_o^T, \theta_o^T, t_o^T, l(r)_o^T)$ is a corresponding pair in input set and template set as the reference minutiae. Convert each minutiae in the corresponding pair set to the polar coordinate system with the respect to $(x_o^I, y_o^I, \theta_o^I, t_o^I, l(r)_o^I)$ and $(x_o^T, y_o^T, \theta_o^T, t_o^T, l(r)_o^T)$. All the minutia can be represented as $(r_m^I, e_m^I, \theta_m^I - \theta_o^I)$ and $(r_n^T, e_n^T, \theta_n^T - \theta_o^T)$ in the polar coordinate system where $r_m^I (r_n^T)$ represents the radial distance, $e_m^I (e_n^T)$ represents the radial angle, $\theta_m^I - \theta_o^I (\theta_n^T - \theta_o^T)$ represents the orientation of the minutiae with respect to the reference minutiae.

Represent the template and the input minutiae in the polar coordinate system as symbolic strings by concatenating each minutia in the increasing order of radial angles:

$$I' = \left((r_1^I, e_1^I - \Delta\theta, \theta_1^I - \theta_o^I), \ldots (r_M^I, e_M^I - \Delta\theta, \theta_M^I - \theta_o^I)\right)$$
$$T' = \left((r_1^T, e_1^T, \theta_1^T - \theta_o^T), \ldots (r_N^T, e_N^T, \theta_N^T - \theta_o^T)\right) \quad (5)$$

A bounding box [4] is also used in the matching process. A bounding box is a box placed around the template minutia, see fig. 2. Two sides of the bounding box have constant radial angle, while the other two have constant radius.

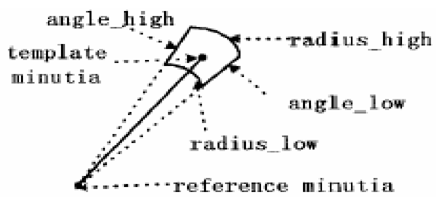

Figure 2 Bounding box

The size of the bounding box is constant or changeable, We use the changeable sized bounding box to compensate calibration error. The size is regulated after every new pair is found. The regulation is based on the suppose: the calibration error is adjacent of different pairs in a local area.

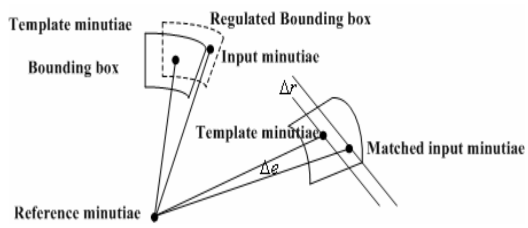

Figure 3 Changeable sized bounding box

In fig. 3, $\Delta e, \Delta r$ is the calibration error of the former pair ($I'[m]$ and $T'[n]$), $\Delta e = (e_m^I - \Delta\theta) - e_n^T$, $\Delta r = r_m^I - r_n^T$. The currently parameter of the bounding box: r_{Low}、r_{High}、θ_{Low}、θ_{High} represent the underside and upside of the radial distance, the underside and upside of the radial angle. The parameters will be regulated as:

$$\begin{cases} r_{Low} = r_{Low} + k \cdot \Delta r \\ r_{High} = r_{High} + k \cdot \Delta r \\ \theta_{Low} = \theta_{Low} + k \cdot \Delta e \\ \theta_{High} = \theta_{High} + k \cdot \Delta e \end{cases}$$

(6)

The following is the matching algorithm.

```
P=0; //represents the number of corresponding pairs
while（1≤n≤N ） {
    while（1≤m≤M ）{
        if（T'[n]和I'[m]match the condition）{
            P++;
            Compute the calibration error;
              Regulate the next bounding box;
              break;
        }// end if
    }// end while
}// end while
```

The match condition is:

$$\begin{cases} r_{Low} \le r_m^I \le r_{High} \\ \theta_{Low} \le e_m^I - \Delta\theta \le \theta_{High} \\ |(\theta_m^I - \theta_o^I) - (\theta_n^T - \theta_o^T)| < \varepsilon \end{cases} \quad (7)$$

ε is the threshold. After matching at most L times, we chose the largest match number as the number of corresponding pairs. If it is larger than the threshold, we justify the input image and the template originated from the same finger.

3 Samples and Performance

The fingerprint image databases we used come from FVC2002 DB and NIST. Two images are considered to match well if the matching score is higher than a certain threshold. Figure 4 and 5 are the ROC curves of FVC2002 DB1 by the proposed algorithm and [4]. Figure 6 is the FMR（t）and FNMR（t）curves of the DB4. (Match Score>12)

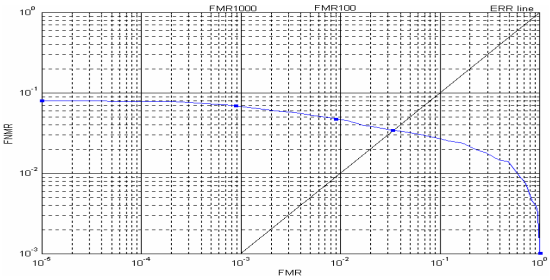

Figure 4 our ROC curve

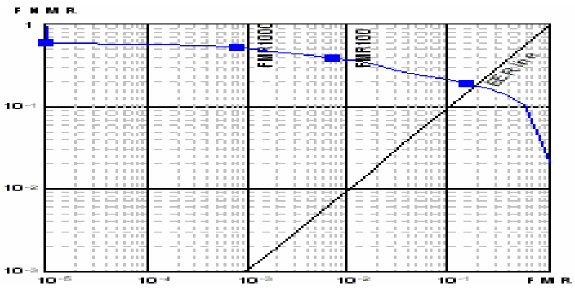

Figure 5 the ROC curve of algorithm in [8]

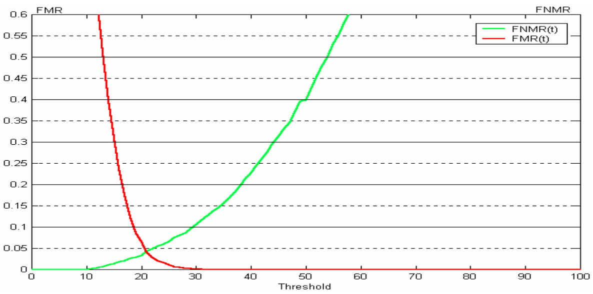

Figure 6 FMR（t）and FNMR（t）curves

Table 1 is the integrate comparison between two algorithms in DB3. (REJ_{MATCH}=0 not list)

Table 1 comparison between two algorithms

%	ERR	FMR 100	FMR 1000	Zero FMR	Match Ti
Ours	6.07	12.25	18.62	24.00	0.22s
[4]	14.96	39.07	53.71	69.43	0.47s

We test our algorithm in all FVC2002, some figures and tables are not list here for the limit of the paper length. And all the results show that our algorithm has a high performance than algorithm in [1] and [4].

4. Conclusion and Discussion

The paper introduced a fingerprint matching algorithm. Based on [1] and [4], it used more efficient corresponding pairs searching algorithm, proposed new parameters for ridgelines' comparability and improved minutiae matching algorithm. The experiments show good performance. But there still remain some questions. The algorithm used more ridgelines' information, and the space consuming is larger. This needs further research.

Reference:

[1]Anil J, LingHong, RuudB. On Line Fingerprint Verification. IEEE Transaction on Pattern Analysis and Machine Intelligence, Vol. 19, No.4, 1997, pp: 302-313.

[2]Anil J, Salil P, LinHong, etal. Filterbank Based Fingerprint Matching. IEEE Transaction on Pattern Analysis and Machine Intelligence, Vol.9, No.5, 2000, pp: 846-859.

[3] LingHong. Automatic Personal Identification Using Fingerprints. USA: Michigan State University, 1998.

[4] Luo Xi ping, Tian Jie, Wu Yan. A Minutia Matching Algorithm in Fingerprint Verification. ICPR'00, 2000.12: 4833-4836

[5] Gorman L. An Overview of Fingerprint Verification Technologies. Information Security Technical Report, Vol.3, No.1, 1998, pp:21-32.

[6] Stockman G, Kopstein S, Benett S. Matching Images to Models for Registration and Object Detection via Clustering. IEEE Transaction on Pattern Analysis and Machine Intelligence, Vol.4, No.3, 1982, pp: 229-241.