

Available online at <http://www.mecs-press.net/ijwmt>

Fast Matching Algorithm Based on Fingerprint Classification Information

Na Li^{a,b}, Wenbi Rao^a, Tiecheng Xu^a

^a*Department of Computer Science and technology, Wuhan University of Technology, Wuhan, China*

^b*Department of Electrical and Information Engineering, Xuchang University, Xuchang, China*

Abstract

This paper focuses on fingerprint minutia matching algorithm. A special minutia neighbor structure is proposed during the matching process in this algorithm. It can locate fingerprints using the singular from classification information. In addition, minutia structure can be used to save the time of matching minutia in a simple but effective way. Then, the matching of minutia is based on the changeable sized boundary box. At the same time, possible reference position is computed to make sure the algorithm more robust to nonlinear deformation from fingerprint images. Experimental results on Fingerprint verification competition FVC2004 databases show that this algorithm can speed up the matching of fingerprint database with a preferable performance.

Index Terms: Fingerprint identification; classification; matching; singular point

© 2011 Published by MECS Publisher. Selection and/or peer review under responsibility of the Research Association of Modern Education and Computer Science

1. Introduction

From the operation mode, the fingerprint matching algorithm can be divided into: verification and identification. For a large scale fingerprint database, the traditional verification algorithm can't satisfy the need of on_line matching in execution efficiency, and therefore the identification algorithm with a good matching rate and speed could be used. In order to quickly retrieve fingerprints in a database, all of them are divided into fixed categories through the classification algorithm, and therefore the speed of system operation is improved. Classification method usually only focuses on the research of database classification, but ignores fingerprint matching speed which determines the system running time. Because of the isolation between classification and matching, many algorithms with good performance in the small database don't fully display to their speed advantage. Considering the fingerprint classification and matching entirely, a singular neighbor structures is used, which modifies the structure presented by Shi Peng [1]. In addition, a fast matching algorithm based on the changeable sized boundary box is proposed. The improvement includes the following aspects: First, the structure can locate fingerprints by the singular from classification message as a reference point. Second, the minutia

matching scores based on the changeable sized boundary box is computed, which makes this algorithm more robust to nonlinear deformation. At last, Using the minutia matching scores computes possible adjust position for reference in the second match, which not only improves the calculation speed, but also increases the algorithm robustness.

2. The singular in fingerprint classification

Common classification method is Henry system which bases on the fingerprint natural shape [3]. It divides fingerprint into 5 categories, or extends to 6 ~ 8 categories, with the important characteristic: the fingerprint singular, including the Core and the Delta, orientation field, ridge etc. The singular calculation based on Poincare index has been widely used for its simplicity, since kawagoe proposed it [4].

The singular detection algorithms based on Poincare index could be able to detect the position and types of singular effectively, including the Core and the Delta. If the value of poicare index is $1/2$, then the minutia is considered as the Core; If the value of poicare index is $-1/2$, then the minutia is considered as the Delta. To enhance the accuracy of the singular extraction, multiresolution, and sliding window technology, low pass filtering smooth method can be used. When there is more than one Core in fingerprint image, the above Core could be chosen as a reference. When the fingerprint is the type in which the Core does not exist, such as the arch type, we can use the feature extraction algorithm proposed by Liu [5], which extract the biggest curvature changeable position as a reference point.

3. The singular neighbor structure

The singular neighbor structure is like Fig.1.

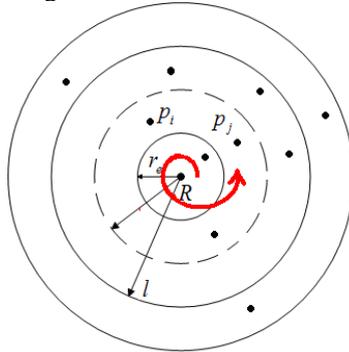


Figure 1. The singular neighbor structure.

the reference point R is used for the centre, predefined value r_e is used for radius in m concentric rings, and the unit of r_e is the multiples of pixel. The singular neighborhood can be divided into m areas through concentric rings, the circle l is called the class l , minutiae in the singular neighborhood are divided into m classes. The optimal number of concentric is found through the experiment due to fingerprint image properties in different database. According to above rules, the singular neighbor structure not only can facilitate fingerprint continuous classification, but also can achieve fast matching [1]. The minutia in class l is defined

as: $p_i = (r, e, \theta, l, n)$, thereinto, $l = \frac{r}{r_e}$. where (r, e, θ) represent the corresponding radius, radial angle, and

normalized minutia orientation with respect to the reference minutia R , respectively. Attribute l is the class of p_i . Attribute n , the number of ridges between minutia and the reference minutia, is invariant to transition,

rotation and nonlinear distortion. Comparing the directions of the reference point in template fingerprint and input fingerprint, each input minutia direction e could be added rotation angle $A\theta_{RR}$ (as in (1)) for rotary alignment. At last, M input minutiae and N template minutiae could be sorted respectively in spiral increasing sequence, according to class $l = 1, \dots, n$, and polar angle from small to big, such as Fig.1, and connected into alignment, as follows:

$$P = ((r_i^P, e_i^P, \theta_i^P, l_i^P, n_i^P)^T), i = 1, \dots, M$$

$$Q = ((r_j^Q, e_j^Q, \theta_j^Q, l_j^Q, n_j^Q)^T), j = 1, \dots, N.$$

So, the fingerprint identification algorithm is the matching between minutiae.

$$A_{ij} = \min\left(\left|\bullet_j^Q - \bullet_i^P\right|, 2\pi - \left|\bullet_j^Q - \bullet_i^P\right|\right) \quad (1)$$

4. Fast fingerprint matching algorithm

According to minutiae structure characteristics, we set one conditions $\left|l_L^Q - l_K^P\right| \leq 1$ to reduce impossible similar minutiae (Because the position of them is far apart from each other), and set another conditions $n_L^Q = n_K^P$ to eliminate obviously dissimilar minutiae to match (Because attribute n is invariant to transition, rotation and nonlinear distortion).

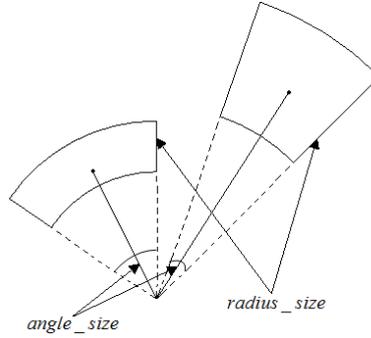


Figure 2. The changeable sized boundary box.

After achieving above conditions, minutiae could be matched using the changeable sized boundary box [2]. The advantage of doing so is that it can decrease the number of matching minutiae, degrade the calculation of matching algorithm, and therefore make matching speed significantly improved. Fast matching algorithm based on fingerprint classification information is described as follows:

- 1) $S = 0$.
- 2) While $1 \leq k \leq N$ do
 - While $1 \leq L \leq M$ and $n_L^Q = n_K^P$ and $\left|l_L^Q - l_K^P\right| \leq 1$ do
 - If *template_point[L]* and *input_point[K]* satisfy condition1 as in (2) then Compute $S[K][L]$ as in (3);
 - $S = S + S[K][L]$;
- end if

Increase L ;

end while

Increase k ;

end while

3) $S > Thre_1$, output :match , goto step 5);

4) if $S > Thre_2$, compute Δr as in (4), compute Δe as in (5), along with S , all stored in Stack[T] ;

5) end.

Above all, condition1 and $S[K][L]$ are defined as follow:

$$condition1 = \begin{cases} true & \text{if } \begin{cases} \text{both in boundary box} \\ A\theta_{KL} < \varepsilon, \text{ as in(1)} \end{cases} \\ false & \text{else} \end{cases} \quad (2)$$

thereinto, ε is a threshold.

$$S[K][L] = 1 - \frac{1}{2} \left(\frac{|\Delta r_{KL}|}{radius_size} + \frac{|\Delta e_{KL}|}{angle_size} \right) \quad (3)$$

thereinto, Δr_{KL} as in (6), Δe_{KL} as in(7), $radius_size$ and $angle_size$ presents radius, radial angle of the boundary box, as in Fig. [2].

$$\Delta r = \sum_{s_{KL} \neq 0} s_{KL} \Delta r_{KL} \quad (4)$$

$$\Delta e = \sum_{s_{KL} \neq 0} s_{KL} \Delta e_{KL} \quad (5)$$

thereinto, $s_{KL} = \frac{S[K][L]}{S}$, Δr_{KL} as in (6), Δe_{KL} as in (7).

$$\Delta r_{KL} = r_L^O - r_K^P \quad (6)$$

$$\Delta e_{KL} = \begin{cases} 360 - a & \text{if } (a = e_L^O - e_K^P) > angle_size, \\ 360 + a & \text{else if } a < -angle_size, \\ a & \text{else} \end{cases} \quad (7)$$

$S[K][L]$ presents similar degrees between minutiae, while S presents the similarity between the input fingerprint and the template fingerprint. If S is bigger than the threshold $Thre_1$, then the input fingerprint and the template fingerprint are considered as the same fingerprint. Otherwise, the second threshold $Thre_2$ is set to tolerate the error from fingerprint images and imprecise pretreatment algorithms. The template fingerprint which has $S_{in} (Thre_2, Thre_1)$ is stored in Stack[T], the stored number is no more than T due to S from big to small, the stored content includes the matching score, transition Δr and rotation Δe . The aim of doing so is convenient for possible reference point calculation.

When the input fingerprint doesn't match all template fingerprints in database, we can do the second match like above algorithms using the message stored in Stack[T]. According to S from big to small, the most possible reference can be computed by transition Δr and rotation Δe . But minutia attributes of input fingerprint must be extracted again, and step 3) must be modified as follows: if $S > Thre_1$, match; else mismatch.

5. Experimental Results

The matching algorithm is based on classification information, so, the quality of both input fingerprints and template fingerprints, classification method, and singular extraction algorithm will all affect the fingerprint matching speed. This paper uses the algorithm in [1] to pretreat fingerprint image, and uses algorithm in [6] to classify fingerprints. The experiment is done on fingerprint database DB1 from Fingerprint verification competition FVC2004 [7]. Average Enroll Time: 0.28s, and Average Match Time: 0.21s. Experimental results in Fig.3 and Fig.4 present that the recognition accuracy is degraded in some degree in order to keep the fast speed of recognition, but the overall recognition performance of the algorithm still remains in a good level.

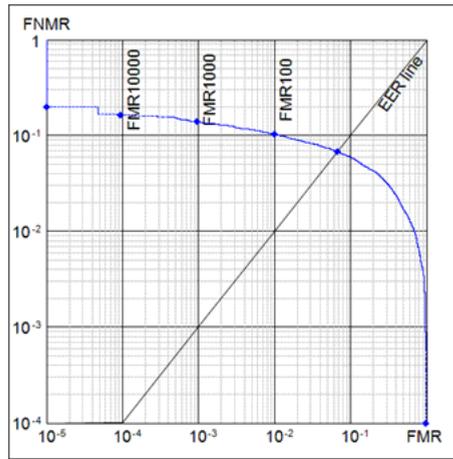


Figure 3. The ROC curve of the algorithm.

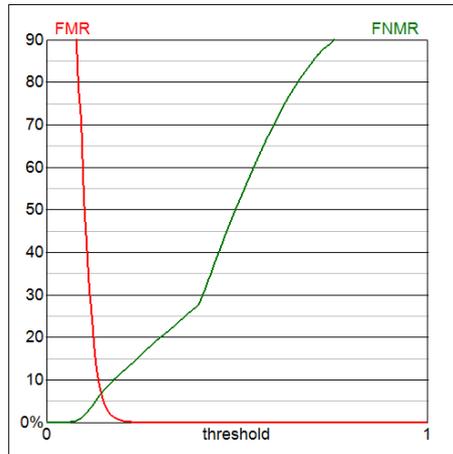


Figure 4. FMR(t) and FNMR(t).

EER, FMR and FNMR etc play an important role in the user's perception of the system performance[7]. In addition, fingerprint images in database DB1 from FVC2004 come from humans, so, the results also prove that the algorithm is fit for on_line matching.

6. Conclusions

This paper proposes the matching algorithms based on fingerprint classification information, in which classification and fast fingerprint matching of two acceleration methods are unified. The singular for reference can increase the efficiency of locating minutiae. In addition, minutiae structure could be used to reduce the calculation of matching minutiae, and save time of matching algorithm in the matching process. Experimental results show that the algorithm improves the system operation speed, meanwhile, the identification accuracy is only reduced minimally. However, the scale of the fingerprint database in this paper is so small (only 10×8 fingerprint) that cannot represent the demand of on_line matching in the fingerprint database of large capacity. Therefore, we need to continue testing performance on larger fingerprint database, and improving, optimizing algorithms, especially focus on the minutia matching strategy.

References

- [1] Shi Peng, Tian Jie, Su Qi, Yang Xin. Fast fingerprint identification based on neighborhood structure around singular point. *Journal of Software*, 2008, 19(12): 3134-3146. (in Chinese)
- [2] Luo Xiping, Tian J. Image enhancement and minutia matching algorithms in automated fingerprint identification system. *Journal of Software*, 2002, 13(5): 946-956. (in Chinese)
- [3] Maltoni D, Maio D, Jain AK, Prabhakar S. *Handbook of Fingerprint Recognition*. New York: Springer-Verlag, 2003. 173-202.
- [4] Kawagoe M, Tojo A. Fingerprint pattern classification. *Pattern Recognition*, 1984, 17(3): 295-303.
- [5] Liu MH, Jiang XD, Kot AC. Reference point detection for fingerprint recognition. In: Zhang D, Jian AK, eds. *Biometric Authentication*. Berlin: Springer-Verlag, 2004. 272-279.
- [6] Yang Limin, Yang Jie, Li Gang. Pattern Matching Based on Fingerprint Classification. *ACTA ELECTRONICA SINICA*, 2003, 31(7): 1030-1034. (in Chinese)
- [7] FVC2004: fingerprint verification competition 2004. <http://bias.csr.unibo.it/fvc2004>