A NEW FINGERPRINT TRANSLATION FINDING ALGORITHM

HUI-MIN MA, RU-KE HUANG

Dept. of Electronic Engineering, Tsinghua University, Beijing 100084, China
E-MAIL: mhmpub@tsinghua.edu.cn, ruke@163.net

Abstract :

Judging the translation and rotation between the input fingerprint and the template fingerprint is the prerequisite for fingerprint matching, and the process is rather time consuming. In this paper, we propose a new algorithm of finding the translation between two point sets and its application in fingerprint minutiae matching. This new algorithm is based on the frequency statistic character of the translation vector matrix of two point sets, and has fast processing speed and independence of the locating accuracy of reference points. This method is not only applicable in judging two identical point sets, but also valid in judging two point sets with partial overlapped regions. By using this technique, we achieve a fast speed in finding the translation between two matched minutiae sets, much faster than the traditional method.

Keywords:

fingerprint matching; minutiae point sets; translation judging

1. Introduction

Judging the translation between two fingerprints from the same finger is the key problem in fingerprint matching, and it is the main obstacle for speeding up the matching process [1]. Although reference point is introduced to solve this problem, it is not adequate because reference points cannot always represent precisely the same place in different fingerprints generated from the same finger.

First, we assume the rotation between the input fingerprint and the template in the database is ranged from $-\theta$ to θ, and the searching step is $d\theta$. Second, for each possible rotation angle, we rotate the input fingerprint minutiae set in that angle and store it in a buffer. Third, we calculate the displacement between the template minutiae set and each rotated input minutiae set by our fast algorithm. Our algorithm will return a maximum frequency of the displacement vector matrix that indicates the number of matched minutiae, and the estimation of the displacement between the template and the rotated minutiae set. Only when these two fingerprints are matched and the rotation angle is correct will the returned frequency be the max. By using this information, we can judge the rotation and the displacement simultaneously.

Some one may point out that reference point can be used to locate the fingerprint and thus solve the shift between matched fingerprints. It is all right when the reference point is precise, but it is not always the case.

The reference point used in our fingerprint verification system is the focus of the ridge of the fingerprint calculated automatically by the computer. Though it is based on the statistic feature of the ridge and is not sensitive to the quality of the images, it does have some shift in different images. Moreover, when used in fingerprint pairs with insufficient common regions, the shift of reference points is significant.

We gathered some fingerprints by the CMOS sensor of Veridicom Company. It has only a 1.5cm×1.5cm sensor area, thus we can obtain only partial of a fingerprint. We call them "small fingerprints". The different ridges in these "small fingerprints" affect the position of the reference points largely. By testing these fingerprint samples, we found that reference points may shift in a range from 0 to 30 pixels(The scale of the fingerprint image is 300×300pixels).

Even by using an optical fingerprint scanner, which has larger sensor area and can obtain more complete fingerprint, reference points may still shift due to distortion of the ridges and rolling of the finger [2]. Therefore, we still have to adjust the translation of two minutiae sets before matching them.

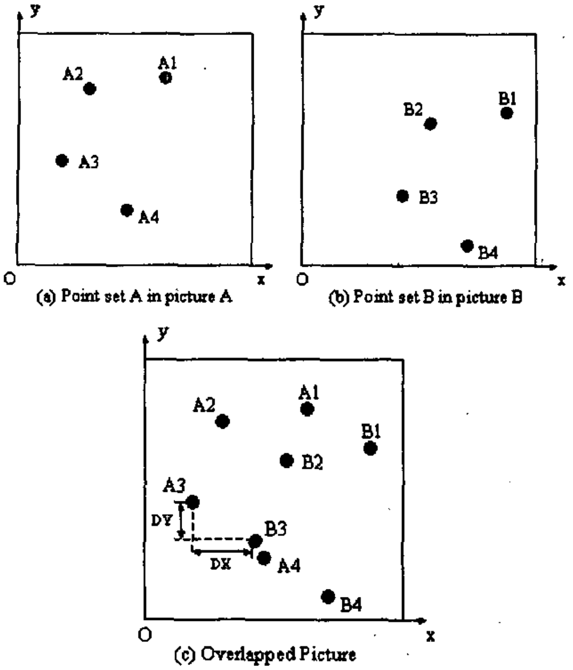
In this paper, we present a new method of judging the translation of two point sets. By comparing with the [3][4], our method statistic the translation between two point sets independent of the reference points, but the generalized Hough Transform constructed it's point sets dependent on the reference points, and the locating accuracy of reference point has no effect on our algorithm. Experimental results are given to support our reasoning.

2. Judging the translation of two identical point sets

The so-called identical point sets means a group of

0-7803-8403-2/04/$20.00 ©2004 IEEE

point sets with the same number of points, the same pattern and the same structure [5]. Fig.1(a) and Fig.1(b) show two identical point sets A and **B** in picture A and picture B respectively. Fig.1(c) is the overlapped picture of picture A and B.



(a) Point set A in picture A

(b) Point set B in picture B

(c) Overlapped Picture

Figure 1. Alignment of Point Set A and B

From Fig.1(c), we can see clearly that point set A and point set B has a translation of DX in the x-coordinate direction and DY in the y-coordinate direction.

However, under the applicable condition of minutiae matching, we won't know point set A and point set B are matched before the matching process, nor do we know the translation DX and DY. The traditional algorithm of matching is given as follows:

```
for dx=-MAX_DX to MAX_DX
    for dy=-MAX_DY to MAX_DY
        Count the matching score of point set A and
        B adjusted the translation by dx and dy.

        Record the max score and corresponding dx
        and dy.
    Endfor dy
Endfor dx
Output the max score and its corresponding dx and dy.
```

The MAX_DX and MAX_DY are the max searching translation in x-coordinate and y-coordinate directions.

Totally different from the traditional algorithm, I invented a new method which is described below:

1) Define the Translation Vector Matrix **D**: **D** =

$$\begin{bmatrix} \vec{D}_{11} & \vec{D}_{12} & \dots & \vec{D}_{1M} \\ \vec{D}_{21} & \vec{D}_{22} & \dots & \vec{D}_{2M} \\ \dots & \dots & \dots & \dots \\ \vec{D}_{N1} & \vec{D}_{N2} & \dots & \vec{D}_{NM} \end{bmatrix} \quad (1)$$

N is the number of points in point set A and M is the number of points in point set B. So **D** is an N×M vector matrix. Its element $\vec{D}_{ij}$ is the two-dimensional translation vector of point A_i and B_j. It is given as follows:

$$\mathbf{D}[i\ ,\ j] = \vec{D}_{ij} = \vec{B}_j - \vec{A}_i \quad (2)$$

$$\vec{A}_i = x_{Ai}\vec{I}_x + y_{Ai}\vec{I}_y \quad (3)$$

$$\vec{B}_j = x_{Bj}\vec{I}_x + y_{Bj}\vec{I}_y \quad (4)$$

$$\vec{D}_{ij} = \vec{B}_j - \vec{A}_i = (x_{Bj} - x_{Ai})\vec{I}_x + (y_{Bj} - y_{Ai})\vec{I}_y$$
$$= dx_{ij}\vec{I}_x + dy_{ij}\vec{I}_y \quad (5)$$
$$i = 1,2,\dots,N \qquad j = 1,2,\dots,M\ .$$

$\vec{A}_i$ and $\vec{B}_j$ are the vectors of points $A_i(x_{Ai},y_{Ai})$ and B_j (x_{Bj},y_{Bj}) . $\vec{I}_x$ and $\vec{I}_y$ are the unit vectors in x and y directions.

2) Define **FreqMap** as the statistic frequency map of matrix **D.** The meaning of its element **FreqMap**[dx,dy] is the frequency of the vector $\vec{D} = dx\vec{I}_x + dy\vec{I}_y$ occurring in **D.** For example, if $\vec{D}_{11} = \vec{D}_{21} = \vec{D}$, and only these two vectors in matrix **D** are equal to $\vec{D}$, then **FreqMap**[dx,dy]=2. Actually, **FreqMap** is the two dimensional histogram of the vector matrix **D.** The scale of the **FreqMap** is determined by the minimum value of dx and dy, which may be negative and the maximum value of dx and dy. For example, if MIN_DX and MIN_DY denote the minimum value of dx and dy, and MAX_DX, MAX_DY denote their maximum value respectively, then the scale of **FreqMap** is (MAX_DX-MIN_DX+1)(MAX_DX-MIN_DX+1).

Absolutely, if the two point sets are totally in a frame with dimension W×H just as point set A and B in Fig.1(c), then a (2W+1)×(2H+1) matrix must be large enough for the FreqMap.

3) **FreqMap** can be calculated in the following way:

(I)Set all the elements in FreqMap zero.
(II)Calculate the translation vector matrix **D** from point set A and B.
(III)Scan the matrix **D**. For each $\vec{D}_{ij}$, increase its corresponding FreqMap[dx_{ij},dy_{ij}] by one.
(IV)After that, we will get a resulting FreqMap.

If point sets A and B are identical and each of them has N points, the FreqMap has two attributes:
<1> The peak value of FreqMap[dx,dy] is N.
<2> If the translation between point set A and B is (DX,DY), then FreqMap[DX,DY]=N.

In order to prove these propositions, I will give the expression of FreqMap in a different form:

$$FreqMap[dx,dy]=\sum_{i=1}^{N}\sum_{j=1}^{M}f(\vec{A}_i,\vec{B}_j) \qquad (6)$$

$$f(\vec{A}_i,\vec{B}_j)=\begin{cases}1........\vec{B}_j-\vec{A}_i=\vec{D}\\ 0........\vec{B}_j-\vec{A}_i\neq\vec{D}\end{cases} \qquad (7)$$

$$\vec{D}=dx\bullet\vec{I}_x+dy\bullet\vec{I}_y \qquad (8)$$

Here, for each absolute $\vec{D}$, we can calculate its occurring frequency in matrix **D** by formula (6).

First we will prove the attribute <1>:

Because no two points in the same point set will have the same vector (otherwise they will be regarded as one point), we can see that for each $\vec{A}_i$ (i=1,2,...,N), there is at most one $\vec{B}_j$ (j=1,2,...,N) to satisfy $\vec{B}_j-\vec{A}_i=\vec{D}$. So from formula (7) we can get that

$$\max\left(\sum_{j=1}^{N}f(\vec{A}_i,\vec{B}_j)\right)=1 \qquad (9)$$

So

$$\max\left(\sum_{i=1}^{N}\sum_{j=1}^{N}f(\vec{A}_i,\vec{B}_j)\right)=N \qquad (10)$$

So, from formula (6), we can get:

$$\max(FreqMap[dx,dy])=N$$

Second, we will prove the attribute <2>:

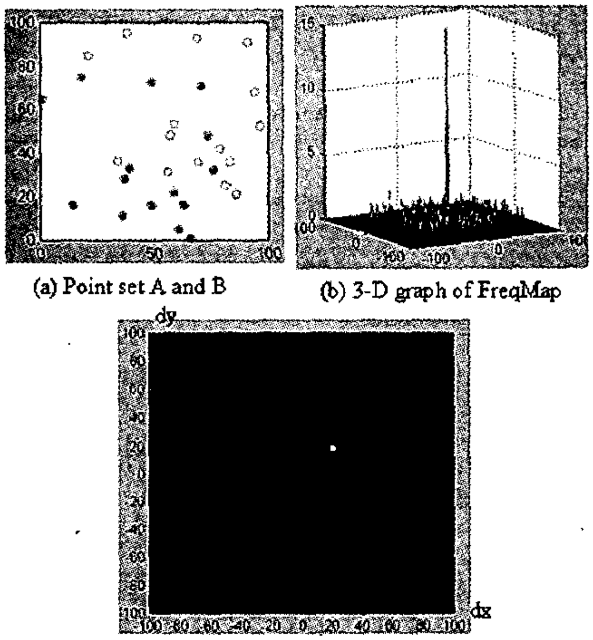
Here we denote $\vec{D}1=DX\bullet\vec{I}_x+DY\bullet\vec{I}_y$. From the meaning of point set translation, for each point A_i, there is a corresponding point B_i which satisfies the equation that $\vec{B}_i-\vec{A}_i=\vec{D}1$, and because there are N points in each set, we will get FreqMap[DX,DY]=N definitely.

So from the reasoning above, we will get the conclusion that if point set A and B are identical and have a translation (DX,DY) between them, then the FreqMap will achieve the peak value N at FreqMap[DX,DY].

On the other hand, if point set A and B both have N points, and FreqMap[DX,DY]=N, then they are identical and have a translation of (DX,DY). The proof is given below:

Because FreqMap[DX,DY]=N, and from the conclusion given by formula (9), we learn that for each A_i, there must be one and only one B_j which satisfies $\vec{B}_j-\vec{A}_i=\vec{D}1$. So all A_i can find corresponding points in point set B with a translation vector $\vec{D}1$, which means point set A and point set B are identical and have a translation of (DX,DY) between them.

From the discussion above, we have proved that our method is reliable for judging the translation of two identical point sets. Following I will give an example of using this method.



(a) Point set A and B

(b) 3-D graph of FreqMap

(c) Gray scale map of FreqMap

Figure 2. Visual display of FreqMap

Fig.2 visually shows the FreqMap of two 15-point point sets with a translation of (20,20). The pattern of the point set is created randomly. The bright white spot in Fig.2(c) is correspondent to the sharp peak in the Fig.2(b). We can see that the coordinate of the spot is (20,20), which is the translation of these two point sets.

3. Judging the translation of two partial overlapped point sets

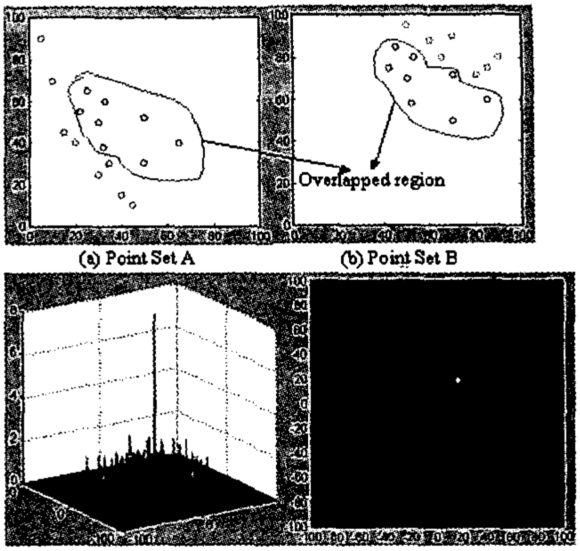
Considering the actual condition of minutiae matching, we should apply our method to the non-totally identical point set matching. Because under most situations, two matched minutiae sets may have an overlapped area and also some different regions, how our method would work determines its practical value. Luckily, it works well in this situation. The reasons are given as follows:

First, FreqMap is the statistic feature of the translation of two point sets. Its element FreqMap[dx,dy] records the number of point pairs with translation (dx,dy). The FreqMap[DX,DY] which has the peak value indicates that the number of point pairs with translation (DX,DY) is the max, thus it gives the estimation that these two point sets have a translation of (DX,DY). It is an optimal estimation based on the rule of making the number of matching point pairs max.

Second, because the self-correlation of fingerprint minutiae is little, the rule of optimal estimation (making the number of matching point pairs max) is valid for most of the time, especially when the number of minutiae in the overlapped region is larger than that of minutiae in the non-overlapped region.

Third, if our algorithm cannot correctly estimate the translation of minutiae sets, nor can the traditional algorithm. Our algorithm may fail only when the optimal estimation rule becomes invalid due to strong self-correlation of the minutiae set or insufficient points in the overlapped region. In the first situation, the pattern of the minutiae has periodicity. In the second, the common area of this two point sets is too small to support their identity. Because the traditional algorithm is also based on the optimal estimation rule to make as many points as possible to match, it will also give the wrong answer. However, the first situation is so rare that it may never occur in a practical fingerprint minutiae set. The failure in the second situation is practically acceptable because we cannot affirm that they are the same if the two fingerprints have only a very small common region. So our method can be applied to practical fingerprint matching. Fig.3 demonstrates the result of applying our method in two point sets with overlapped and different regions.

In Fig.3(a) and (b), we see point set A and B have an overlapped region with 8 points. There are some other points in each of them and point set B has a translation of (20,20) from point set A. This pattern of point sets is quite common in fingerprint matching.



(a) Point Set A (b) Point Set B

(c) 3-D view of FreqMap (d) Gray scale view of FreqMap

Figure 3. Result of applying in partial overlapped point sets

The result of calculating the FreqMap is shown in Fig.3(c) and (d). As we had expected, there is a peak value of 8 at FreqMap[20,20], correctly indicating the translation of these two point sets.

4. Fast algorithm of implementing our method

Based on our innovative method, we contrived a fast algorithm to judge the translation of minutiae sets. Let's denote the template minutiae as MinuTEM[i] (i=1,...,N), and the input minutiae as MinuIN[j](j=1,...,M).

First, MinuTEM[i] and MinuIN[j] are sorted in an order shown in Fig.1(a) and (b). The minutia with a smaller label is placed on the top of those with larger label, or it is on the left of them when they have the same y-coordinate. Actually, this sorting of the template minutiae is done when it was added into the fingerprint database, so we only need to sort the input minutiae. Because for one input minutiae set, we have to compare it with huge numbers of template minutiae sets, the time spent in sorting is trivial compared to thousands times of matching.

Furthermore, because the shift of the reference point ranges form 0 to 30 pixels, the peak value of FreqMap must occur at FreqMap[dx,dy] with $-30 \leqslant dx \leqslant 30$ and $-30 \leqslant dy \leqslant 30$. So we needn't calculate the FreqMap[dx,dy] when dx or dy is not in this range. Practically, because of the plastic distortion and noise, the estimation of translation need not and cannot be as precise as 1-pixel resolution. Plastic matching is used to resist the distortion. So we make our resolution of estimation 5 pixels. It is done by dividing the actual (dx,dy) by a factor of 5 and using the integer part of the result as the (dx,dy) used in FreqMap[dx,dy]. For example, FreqMap[0,0] is correspondent to the number of point pairs with translation of $0 \leqslant dx \leqslant 4$ and $0 \leqslant dy \leqslant 4$. FreqMap[6,6] is correspondent to the number of point pairs with translation of $30 \leqslant dx \leqslant 34$ and $30 \leqslant dy \leqslant 34$. Using this technique, we can shrink the size of FreqMap from $(2*30+1)^2=3721$ to $(2*6+1)^2=169$, but still can judge the translation in the range of $-30 \leqslant dx \leqslant 30$ and $-30 \leqslant dy \leqslant 30$.

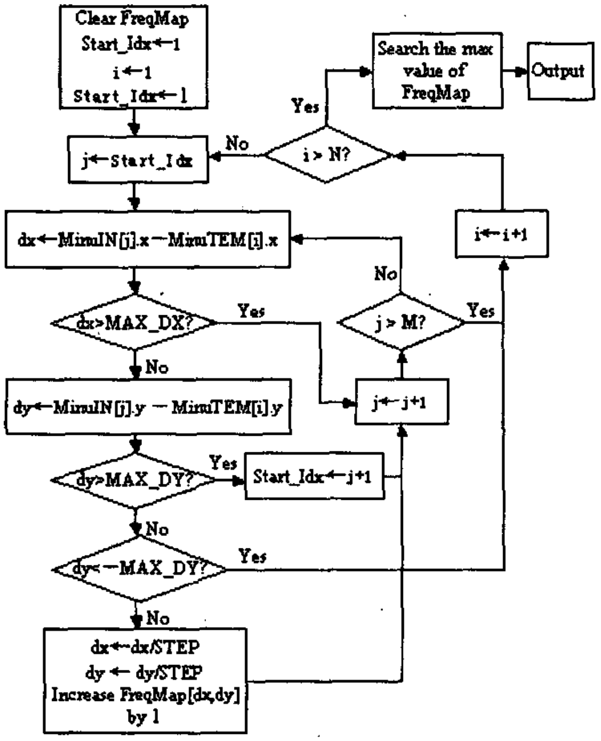


Figure 4. Flow chart of algorithm

Fig.4 shows the flow chart of the algorithm. Here the MAX_DX and MAX_DY are both equal to 30 and the STEP is equal to 5, which is the resolution of the estimation. The variant Start_Idx is used to dynamically adjust the boundary of effective band shown in Fig.5:

As you see in Fig.5, any MinuIN[j] located outside the band must have a dy larger than MAX_DY or smaller than -MAX_DY. So we don't need to count them. Because MinuINs are sorted, Start_Idx marks the index of MinuIN that is nearest to the topside of the effective band.

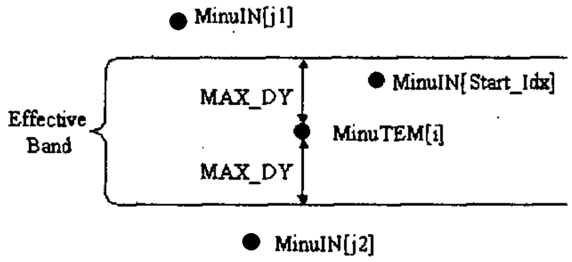


Figure 5. Meaning of Effective Band

Then for the MinuIN[j]s with j>Start_Idx, they are located in the band or beneath the bottom side of the band which is detected in the algorithm and breaks the loop of j. Using this technique, the counting of FreqMap is done in the effective band only for each MinuTEM[i]. Let's find the number of loops in calculating the FreqMap. Denote E(MAX_DY) as the average number of MinuINs in the effective band for all the MinuTEMs, then the average loop times is:

$$\overline{N}_{LOOP} = E(MAX\_DY) \bullet N \qquad (11)$$

A traditional algorithm with resolution of 5 pixels and the ability of estimating a translation up to (30,30) needs to loop $[2*(30/5)+1]^2=169$ times to test a specific group of dx and dy. In each loop, it still has to loop M*N times or E(MAX_DY)*N times (when applying the "effective band" technique) to count the matching score. It is clear that our algorithm is much faster than the traditional one.

5. Experimental results

We conduct an experiment to test the speed of our algorithm and compare it with the traditional one. The experiment is done on two groups of fingerprint minutiae sets. One is the "Small Fingerprint" with average 35 minutiae in each set; the other one is the "Big Fingerprint" with average 68 minutiae in each set. Each group has 500 fingerprint minutiae sets. Our experiment is the same for both groups. First, we randomly select one minutiae set from one group, then artificially add some shift (DX,DY) to it, then find the matched set in its group and their translation. Both our algorithm and the traditional one give the right answer, but their speeds are quite different. Test is done on a PC Pentium III 733MHz. Table 1 shows the result:

	Group 1 Avg. 35/Set	Group 2 Avg. 68/Set
Fast Algorithm	0.0259 ms/Set	0.0814 ms/Set
Traditional	9.993 ms/Set	41.91 ms/Set
Gain	385.8 Times	514.9 Times

From the gain of speed we can see that our algorithm is hundreds of times faster than the traditional algorithm. For a template minutiae set with 35 minutiae, it takes only *0.0259ms to judge its translation with the input minutiae set.* This expense is rather small. By using this algorithm, our fingerprint matching system achieves an average speed of matching 4000 fingerprints in a second.

6. Conclusions

This paper introduces a new method of judging translation between two point sets and proposes a fast algorithm used in fingerprint minutiae matching based on this method. We give theoretical proof of the validity of the new method, and the experimental result of the algorithm. Our technique is useful not only in fingerprint matching, but also in other image processing applications such as finding the overlapped region of two images or joining two images which are partial overlapped.

Acknowledgements

We wish to thank professor Su Guang-da for providing us with the research environment and the fingerprints database.

References

[1] Xiping Luo,Jie Tian and Yan Wu, A Minutia Matching Algorithm in Fingerprint Verification, 15th ICPR, Barcelona, Vol.4, pp.833-836, 2000.

[2] R.Cappelli, D.Maio and D.Maltoni, Modelling Plastic Distortion in Fingerprint Images, In proceedings Second International Conference on Advances in Pattern Recognition(ICAPR2001), Rio de Janeiro, pp.369-376, March 2001.

[3] D. H. Ballard. Generalized Hough transform to detect arbitrary patterns, IEEE Trans. Pattern Anal. and Machine Intell., 3(2), pp.111-122, 1981.

[4] S. H. Chang, F. H. Cheng, W. H. Hsu, G. Z. Wu. Fast algorithm for point pattern matching: Invariant to translations, rotations, and scale changes, Pattern Recognition, 30(2), pp.321-339, 1997.

[5] Marius Tico and Pauli Kuosmanen, New approach of automated fingerprint matching, Proceedings of SPIE Vol.4303-19 SPIE•0277-786X/01, 2001.