An Algorithm of Fingerprint Feature Extraction Based on Macroscopic Curvature

Zhi—*Xiong Liu Gui-Ming He*

School of Electrical Engineering , Wuhan University,Wuhan 430072,Hubei,China

Lzxgdf@Hotmail.com

Abstract

In the Automatic Fingerprint Identification System (AFIS) , extracting the feature of fingerprint is very important .The local curvature of ridges of fingerprint is irregular , so people have the barrier to effectively extract the fingerprint curve features to describe fingerprint. This article proposes a novel algorithm; it embraces information of few nearby fingerprint ridges to extract a new characteristic which can describe the curvature feature of fingerprint. Experimental results show the algorithm is feasible , and the characteristics extracted by it can clearly show the inner macroscopic curve properties of fingerprint. The result also shows that this kind of characteristic is robust to noise and pollution.

Key words: fingerprint recognition; minutiae ; feature extraction ;curvature

In the Automatic Fingerprint Identification System (AFIS) verification is performed by comparing the similarity of fingerprint features between two different fingerprint images. So the characteristics extraction is a critical step. Fingerprint image is made of ridges and valleys. Except the structural features above the curvature of ridges is also important to describe fingerprint. But , because of the local bifurcations , discontinues , ending and distortion of fingerprint ridges , it is difficult to use the single ridges curve to describe the structure of fingerprint. So we face the barrier of efficiently applying the huge curve information. So how to extract a characteristic which can describe the global curve feature of fingerprint and omit the local disorders of fingerprint is important.

By observing , we find that though the single finger curve is irregular , the change of the fingerprint image strips , which are made up of several neighboring fingerprint bridges , is regular. The curvature of this strip is not sensitive to the change of single fingerprint ridge and can reflect the common curvature feature of all the ridges in this fingerprint strip. These different strips together reflect the curve feature of the whole fingerprint . The objective of our algorithm is to pick this strip out and extract the feature of it.

The proposed algorithm includes the following steps : selecting the interest region , ridge tracking , curve fitting and feature extraction. The proposed algorithm operates on the thinned binary fingerprint images. The width of the ridge in the thinned image is a single pixel , while some short spurs and sharp angles of it have been eliminated.

1. Selection of the Interest Region

The region below the center of fingerprint image is an area where a lot of different directional fingerprint ridges encounter each other. It is difficult to compare different kinds of fingerprint images in this area because different kinds of fingerprint image of this area have different growing trends[2-4] .But , the region above the center (core) has a nearly consistent structure which is feasible to be analyzed by algorithm and compared. So we select this area as our interest region. We select a rectangle area ABCD shown in Fig. 1 as our interest area. In the Fig. 1 , the $|AB|$ and $|CD|$ can be decided by users ($|\cdot|$ is the length) ; the point O is the core of fingerprint and it is also the middle point on the line AD.

2. Ridge Tracking

Ridge tracking is critical in the proposed algorithm. The objective of this step is to select a strip which includes several neighboring fingerprint ridges. Then we will operate on this strip . The traditional region tracking method based on 4 or 8 nearest neighbors does a search in all 4 or 8 orientations with a step length of 1. Being used in ridge research , it has several disadvantages : every point will be picked out repeatedly ;the area grows slowly because there is no prior search direction ; a enormous memory is required for saving the searched points ; it cannot overpass the discontinue ridges , *etc*. So it is improper for being used in ridge tracking. The proposed algorithm will overcome these demerits. Let G be a thinned binary fingerprint image and $t(x, y)$ be the gray value of the point (x, y) in the image. Fig.2 shows the eight orientations of local ridge.

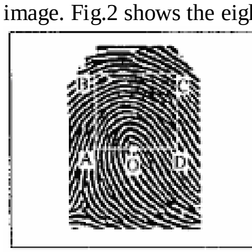

Fig.1 Interest region

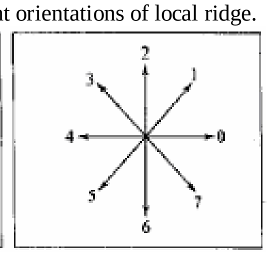

Fig.2 Eight orientations Of local ridge

Firstly , we make the following definitions :

Definition 1 positive tracking :as shown in Fig. 3 (a) .It means searching the points which satisfy the condition $t(x, y) = 1$ in the area $G(i, j)$, where $G(i, j) = \{ t(i, j-2), t(i+1, j-2), t(i+2, j-2), t(i+2, j-1), t(i+2, j), t(i+2, j+1), t(i+2, j+2), t(i+1, j+2), t(i, j+2) \}$.

Definition 2 reverse tracking :as shown in Fig. 3 (b) .It means searching the points which satisfy the condition $t(x, y) = 1$ in the area $G(i, j)$, where $G(i, j) = \{ t(i, j+2), t(i-1, j+2), t(i-2, j+2), t(i-2, j+1), t(i-2, j), t(i-2, j-1), t(i-2, j-2), t(i-1, j-2), t(i, j-2) \}$.

Definition 3 tracking along the direction k: where k ($k = 0, 1, \cdots, 7$) is the direction of fingerprint. Tracking along the direction k means searching the points which satisfy the condition $t(x, y) = 1$ in the area $Gk(i, j)$, Where $G0(i, j) = \{ t(i+5, j-2), t(i+5, j-1), t(i+5, j), t(i+5, j+1), t(i+5, j+2) \}$, $G1(i, j) = \{ t(i+5, j+2), t(i+4, j+3), t(i+3, j+4), t(i+2, j+5) \}$, and the other six regions are symmetrical with region $G0(i, j)$ and $G1(i, j)$ respectively. As shown in Fig. 3 (c) , (d) .Then , we begin our algorithm:

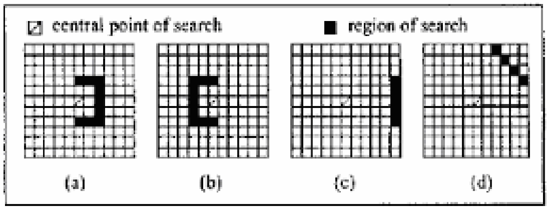

Fig.3 Search definitions

(a) Positive search;(b) Inverse search;(c)search along orientation 0; (d) Search along orientation 1

Initiate condition : Let M be the times of the search and it is initialized with 1. $S0$ which has an initial value O is the point on the ridge along the path OAB ; we use it as the point where to begin the search. Let P be a queue to preserve our found points which are ordered from the big to the small. The initial value of P is null.

Step 1 :if $M > 3$,

Terminate this algorithm

else

Beginning from $S0$, select the first point in the nearby area of the path S0AB which satisfies the formula $t(x, y) = 1$ (this ensues the point is in the ridge) and mark it with S .Then look for a point in the queue P whose distance to S is not longer than two pixels. If cannot find this point , let $M = M + 1$, $S0 = S$, $t0 = S0$ and go to Step 2 ; else , let $M = M + 1$, $S0 = S$ and return to Step 1.

The aim of Step 1 is to control executing times of the algorithm , here we let M be 3 , this demonstrates we will select three ridges to form a fingerprint ridge strip. Step 1 also ensures we can select out a ridge which has never be tracked before , and begin a new track.

Step 2 : With $t0$ as the searching center point begin a positive search. Mark the found points with $D1, D2, \cdots, DN$, where N is the account of points found. In the case of $N > 1$, we can conjoin these points to 1 or 2 points using the rule of conjoining nearby points (which will be introduced in Section 3) . So according to the value of N , there are three cases :

Case 1 : if $N = 1$,

$t0 = D1$, return to Step 2.

Case 1 deals with the instance that only one point has been found on the ridge , under this condition the algorithm will use the just found point as the central point and continue the positive search.

Case 2 : If $N = 2$, respectively use $D1$ and $D2$ as the central point and do reverse search once. Note the found point queue as $D'1, D'2, \cdots, D'n$. The relationship of these points in the search nearby area has two cases , one is that they make up of a continual area (decided by 8 nearby domain) , and the other is that they make up of two areas which are not adjacent. The former shows that the search meets with a forward bifurcation point ; in this instance respectively use $D1$ and $D2$ as the central point (let $t0 = D1$ or $t0 = D2$) and return to Step 2 to begin a new positive search. . The latter shows the search meets with a backward bifurcation point ; in this instance we random select a point as $D'i$ in the area which don't include $t0$. Use $D'i$ as central point ($t0 = D'i$)and return to Step 2 , at the same time , change the positive search to reverse search in Step 2 ,and then continue. Meanwhile , if $D'i$ s found in the searching queue of $D1$ ($D2$) use $D2$ ($D1$) as the other searching process ($t0 = D2$ ($D1$)) and return to Step 2.

Case 2 shows the instance of finding a bifurcation point. The algorithm will decide the kind of bifurcation by an inverse search. If the points found by the inverse search compose an area, the bifurcation is a forward bifurcation. Otherwise we can say it is a backward bifurcation. For the forward bifurcation, the positive search process will divided into two positive search processes which will continue search along one of the two branches respectively. For the backward bifurcation , the original search will also divided into two processes :one continues the positive search along the forward branch and the other does the inverse search along the backward branch. As shown in Fig. 4.

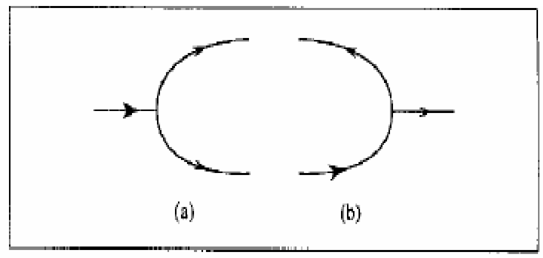

Fig.4 The original search(dark arrow) divides into two branches

(a)Meeting with a forward; (b)Meeting with a backward bifurcation

Case 3 : If $N = 0$, there is no point found. In this instance , by using directional information of the central point ,the algorithm will increase the step of search and do the search again. Let k ($k = 0, 1, 2, \cdots, 7$) be the local direction of $t0$ .In the case of the positive search , if direction k is included in the directions of the positive search the algorithm will search along the direction k ; else , the search will be done along the negative direction of k.In the case of the inverse search :If k is included in the directions of the negative search the algorithm will search along the direction k also ; else , search along the negative direction of k.If there is no point found with increased search step ,this search process will terminate (in this instance the centralpoint of search is a termination) . Else , if the found point(points) is obtained in a positive search , return to Step 2 ; if the found point (points) is obtained in a negative search , return to Step 2 while change the positive research of Step 2 to a reverse one .

Case 3 shows the case of searching meeting with a discontinuous bridge. Here , the discontinuity may be a termination or a false termination cased by a short discontinuous bridge. The false termination may interrupt the search process. To avoid this , the algorithm increase the search step ;and to avoid searching points which don't belong to this ridge , the local direction information of central point is used to limit the size of searching nearby domain. By done this ,the algorithm can escape the discontinuity caused by false termination and its efficiency is improved. For each search process , the search will be terminated under the following conditions :Condition 1 : meeting a point which is a termination point (as discussed in Case 3) .Condition 2 : meeting with the boundaries of the rectangle ABCD.

3. The Rule of Conjoining Nearby Points

Rule 1 : Fig. 5 (a) . If there are two separate points , *D*1 and *D*2 , are found in the positive (or inverse) search whose center is the point P , these two points are just the points we want to obtain. Rule 2 : Fig. 5 (b) . If there are two adjacent points ,*D*1 and *D*2 , are found in the positive (or inverse) search of point *P* , delete one of the two points and then note the left one as *D*1 . Rule 3 : Fig. 5 (c) (d) . If there are more than two points found (*D*1 , *D*2 , ···, *Dm*) , no matter they adjoin or not , choose the farthest two points as *D*1 and *D*2 , and desert others.

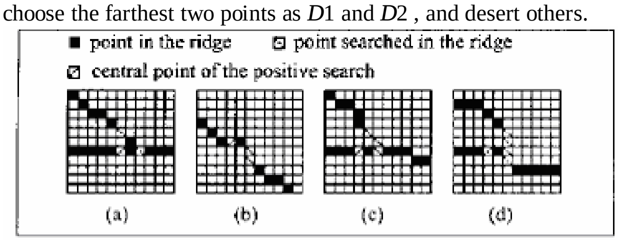

Fig.5 More than one points searched in the region
(a)Two points not adjacent;(b)Two adjacent points;
(c),(d) More than two points

4. Curve Fitting and Feature Extraction

The objective of this step is to find a curve fitting for the points found by the above steps , then to extract the feature of this curve.

4.1. Curve Fitting

For the points in the queue *P* , we can do curve fitting by using the method of least squares , i. e. , look for a function $\varphi(x)$, in *H* , which satisfies the following condition :

$$\sum_{i=1}^{n}(y_i - \phi(x_i))^2 = \min_{\Psi \in H}\sum_{i=1}^{n}(y_i - \Psi(x_i))^2$$

Where (x_i , y_i) , is a point in *P* ; and *H* , is a function class of lower order polynomials. The fitting curve has a continually changed curvature.

4.2. Max Curvature Point

Supposed that the curve calculated by above can be expressed as : $z = \varphi(x, y)$, the direction angle of this curve can be expressed as :

$$\theta = \arctan(\phi'(z)) + \frac{\pi}{2}$$

where *z* , is the point in the curve. And the curvature of z_i(z_i is a point of the curve) can be expressed as the change of direction angles of the neighboring points : $u_i = \theta_{i+1} - \theta_i$. So the max curvature of this curve is : $u_i = \max(u_i)$ .And the point z_k which is relative to the position of u_k , is the feature point extracted by executing the proposed algorithm once. This feature point can be described as : $M(x, y, \theta)$. Where *x* and *y* , are the coordinates of *M* ; and θ, is the direction of it.

5. Experiment Result s and Discussion

We obtained fingerprint images of 300 ×300 pixels by a capacitive sensor. The algorithm is executed on the image after removing the system error, filtering, enhancement , binarization and thinning.

Fig. 6 (a) is a group of ridges picked out by executing ridge tracking algorithm once. After curve fitting for this group of ridges , Fig. 6 (b) shows the extracted characteristic and its orientation (marked on a strip which is made up of these ridges) .

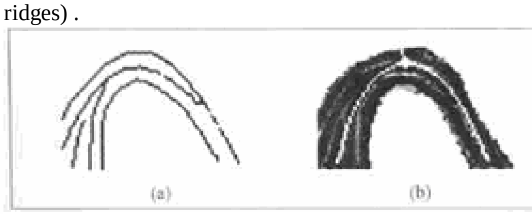

Fig.6 Results executed the proposed algorithm once
(a)A group of ridges picked out by executing ridge tracking algorithm once;
(b)The extracted characteristic and its orientation

Fig. 7 (a)-(d) show the features extracted by the proposed algorithm on different fingerprint images. Every feature shows the max curvature position and orientation of a group of ridges. Fig. 7 (e) , (f) are the tracks of feature points in Fig. 7 (d) , (c) . They clearly describe the max curvature position and its tendency of change , and are consistent with people's subjective perceive to the change of the curve of fingerprint ridge strips. These indicate this kind of characteristic extracted by the proposed algorithm can embody the macroscopic curvature feature of fingerprint , consequently it embody the inner property of fingerprint images.

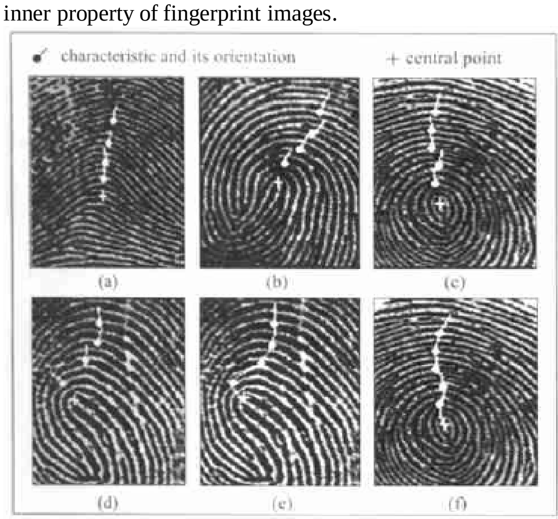

Fig.7 Features extracted by the proposed algorithm
(a)-(d) Extracted characteristic;(e),(f) The track of characteristic in (d),(c) respectively

A fingerprint image is under the interference of all kinds of noise , so we did some researches on the noise resistance of the extracted characteristic. Select 50 medium-quality fingerprint images obtained in laboratory , then pollute them with two kinds of Gaussian noise of different intensity : one has a mean *M* = 0 and variance *V* = 0. 05 and the other mean *M* = 0 and variance *V* = 0. 10. To the original image and the polluted

Table 1 **The relationship between characteristics and noise**

Image Type	True characteristics number	False characteristics number	Missed characteristics number	Error rate (%)
Original image	1890	381	29	21.7
Polluted image(V = 0. 05)	1462	674	155	56.7
Polluted image(V = 0. 10)	943	1219	147	144.9

Fig .8 Relationship between characteristics and noise

(a),(b),(c) Traditional characteristics (minutiae) extracted by methods in Ref . [5] under the condition no noise , V = 0. 05 , and V= 0. 10 respectively ;

(d),(e),(f) Characteristics (denoted as ò) extracted by the proposed algorithm under the same conditions as (a) , (b) and (c) respectively.

image, extract their traditional characteristics (including termination and bifurcation) with the algorithm introduced in Ref. [5] ; at the same time , as a comparison , extract characteristics with the proposed algorithm. Table 1 shows the relationship between noise intensity and the error rate of traditional characteristics extracted by algorithm introduced in Ref . [5]. The error rate p is calculated by the formula : p =(M1 + M2)PM0 . Where M1 , is the number of erroneous characteristics which is composed of two parts : false characteristics and characteristics of erroneous kind ; M2 , is the number of characteristics missed ; and M0 , is the number of true characteristics extracted. From the table 1 , we can see the traditional characteristics are sensitive to noise. With the increase of noise intensity the error rate of the traditional characteristics increases quickly which finally induces the failure of fingerprint verification. Fig. 8 shows the executive results of our algorithm. Characteristic is nearly not affected by the noise with the variation 0. 05. For the heavier intensity noise with the variation 0.10 some characteristics changed , but the position after changed is still on the tracks line of feature points (the white line in the Fig. 8) . The cause of this is that , with the increasing of noise intensity , the position of termination and bifurcation in the shinned fingerprint ridges is changing greatly. But the tendency of fingerprint ridges changes slightly. So the characteristic extracted by the proposed algorithm still well shows the macroscopic curve property of fingerprint ridge strips.

The experimental results show that the characteristic extracted proposed algorithm clearly describes the curve feature of fingerprint ridge , and compared with the traditional characteristics this characteristic is robust to noise, it can be used for fingerprint recognition and verification.

6. References

[1] Hong L , Jain A. Integrating Faces and Fingerprints for Personal Identification. *IEEE Trans Pattern Analysis and Machine Intelligence* , 1998 , 20(12) :129521307.

[2] Kawagoe M, Tojo A. Fingerprint Pattern Classification. *Pattern Recognition* , 1984 , 17(3) : 2952303.

[3] Karu K, Jain A K. Fingerprint Classification. *Pattern Recognition* , 1996 , 29(3) :3892404.

[4] Senior A. A Combination Fingerprint Classifier. *IEEE Trans Pattern Analysis and Machine Itelligence* ,2001 , 23(10) : 11652 1174.

[5] Jain A , Hong L , Bolle R. On-Line Fingerprint Verification. *IEEE Trans Pattern Analysis and Machine Intelligence* , 1997 ,19 (4) :3022314.