PALMPRINT RECOGNITION USING CREASE

Jun Chen, Changshui Zhang, Gang Rong

State key Laboratory of Intelligent Technology and Systems
Department of Automation, Tsinghua University
Beijing China, 100084

ABSTRACT

Palmprint has one special salient feature which is not salient in fingerprint. That is crease. In palmprint the creases are large in number and comparatively easy to extract. Creases are also approximatively stable in a person's whole life, which quaifys themselves as features in palmprint recognition. In this paper, we give creases an accurate definition which is fit for algorithm implement. We devised a rather exquisite algorithm to extract all the creases in a palmprint whose success is mainly from a new different direction computing method and a thorough local analysis and a robust search algorithm. Based on the extracted creases, we devised a robust palmprint matching algorithm which is rotation and translation invariant. The crease extraction results and palmprint matching results show that crease can be extracted successfully and crease-based palmprint matching is robust and accurate.

1. INTRODUCTION

In recent years, a series of automated biometric-based identification methods have emerged, which include fingerprint, palmprint, iris pattern, voice identifications and etc. In these biometrics, palmprint has great similarity to fingerprint whose identification is rather mature[1]. But palmprint has one special salient feature which is not salient in fingerprint, that is crease. Fingerprint has creases too, but the number of creases are scare and to extract them is rather difficult in comparison to the minutiae detection[2]. In contrast, palmprint has a large number of creases which are very salient and comparatively easy and accurate to extract.

At the same time, the crease of a person only changes slowly in a person's life, which makes it appropriate as a feature for identification[4]. But nobody can assert how it is fit for identification because no practical crease extraction algorithm exists.

On the other side, some crease patterns are related to some diseases of people, especially the chronics[5, 7]. It's very meaningful to foretell people's health problem[6]. But nobody knows the exact relation between them. The automated crease extraction will help greatly to reveal the association between some diseases and crease patterns because it can give people a lot of accurate data to analyze.

We haven't seen any record on total crease extraction or recognition based on crease in the bibliography.[3] deals with how to eliminate the creases in a palmprint.[4] focus on the extraction of the three main crease in palmprint, ignoring the large number of other creases which are rather useful and different. In our paper, we devised an exquisite algorithm to extract all the creases of a palmprint image. Because the crease is a comparatively stable feature of a person, we think it is valuable in palmprint recogniton and devise a crease-based palmprint matching algorithm to see how the crease is fit for palmprint recognition. The matching result is exciting.

This paper is arranged as, section 2, the crease extraction alogrithm, section 3, the crease-based palmprint matching algorithm, section 4, the result, section 5, the summary and future work.

2. CREASE EXTRACTION ALGORITHM

There are several difficult points in crease extraction. First, no appropriate definition for crease exists, either in mathematical form or in algorithm form. We point out that a point in the crease has either feature of the following:

1. Its direction has a large deviation from the neighbour pixels
2. It's located at a long and narrow region whose width has a large deviation from the neighbours'

These two types are illustrated in the figure 1. To get crease points, we devised an algorithm which takes use of a new direction map computing method(in 2.1) and performs a two-step local analysis(in 2.2). Second, creases are curves in nature, not regions. We should present the curves in the form of line segments, not a number of pixels. We devised a robust search algorithm to fulfill this intention(in 2.3) which can filter noise and even connect broken crease region. The postprocessing and multiscale framework is in 2.4 and 2.5.

0-7803-6725-1/01/$10.00

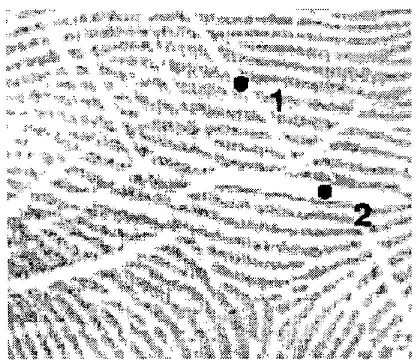

Fig. 1. The number 1 indicates a crease point of type 1, the number 2 type 2.

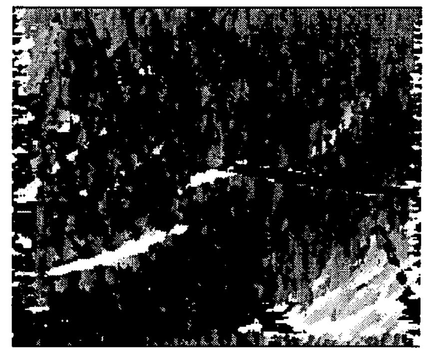

Fig. 2. The direction map. The gray value of a pixel shows its direction, with value 0 as 180 degree, value 255 as 0 degree. The original image is figure 1.

2.1. Direction Map

The direction computing is an useful method in fingerprint and palmprint analysis because the direction of a point in such stria-pattern image is a critical local feature. We divised a new different direction computing algorithm which is the most fit for crease extraction.

Denote current pixel $g = g(x, y)$, in the direction θ,$(0 \leq \theta < \pi)$, we choose pixels $g_i = g(x_{\theta_i}, y_{\theta_i}), (-L \leq i \leq L)$, according to

$$\begin{cases} x_{\theta_i} = i + x, y_{\theta_i} = i\tan(\theta) + y & \theta < \frac{1}{4}\pi, \theta > \frac{3}{4}\pi \\ x_{\theta_i} = -i\tan(\theta - \frac{1}{2}\pi) + x, y_{\theta_i} = i + y & \frac{1}{4}\pi \leq \theta \leq \frac{3}{4}\pi \end{cases} \quad (1)$$

And we compute a nonlinear and non-symmetrical σ_θ, $\sigma_\theta = \min(\alpha_\theta, \beta_\theta)$

$$\alpha_\theta = \sum_{i=1}^{i=L} e_i \cdot f(g_i, g), \beta_\theta = \sum_{i=-L}^{i=-1} e_i \cdot f(g_i, g)$$

$$f(a,b) = \begin{cases} 1 & H_1 < |a-b| \\ 1/2 & H_2 \leq |a-b| \leq H_1 \\ 0 & |a-b| < H_2 \end{cases}$$

and $e_i$is the coefficient based on Gaussian function.

The direction of $g(x, y)$ is the θ having minimum σ_θ.

H_1 and H_2 are two thresholds, selected based on the noise level and gray value distribution in the image. The experiment results show that our σ_θ definition gives out a sharp and accurate estimation of the local direction(see figure 2).

Some pixels' direction are ambiguous. We consider σ_θ, $\theta(0 \leq \theta < \pi)$ as a one-dimensional function curve(θ is x-axis, σ_θ is y-axis), and decide the salience of the direction based on the curve shape. The following algorithm is based on the pixels with salient direction.

2.2. Crease Point Estimation

Now, we can give every pixel with salient direction an estimation if it is in the crease or not using criterion mentioned in the above.

Denote $g(x, y)$ be the current pixel with direction d. Along direction $\theta = d + \pi/2$, we select M points $g(x_{\theta_i}, y_{\theta_i}), 1 \leq i \leq M$ in the same way as (1). For each point, if it has a salient direction d_i we compute a deviation quantity as $dev_i = \min(\pi/2 - |d_i - d|, |d_i - d|)$. The sum of all the deviations are denoted td_1 and the number of points with salient direction is denoted t_1. Similarly, along direction $\theta = d + 3\pi/2$, we get the sum of deviations td_2 and number t_2.

If $t_1 > TH_t$ and $t_2 > TH_t$
    If $td_1 > t_1 \cdot TH_d$ and $td_2 >$
$t_2 \cdot TH_d$
        $g(x,y)$ is a crease point
    End if
Else if $t_1 > TH_t$ and $td_1 > t_1 \cdot TH_d$
    $g(x,y)$ is a crease point
Else if $t_2 > TH_t$ and $td_2 > t_2 \cdot TH_d$
    $g(x,y)$ is a crease point
End if

If current pixel is not decided to be crease point we continue to the second step. Consider all the $2M + 1$ points (including the current pixel) defined above as a one-dimensional curve(fig. 3). The curve is divided into several segments using the valleys in the curve. If the width of the segment including the current pixel has a large difference from the average width of all the segments, we decide the current pixel to be a crease point.

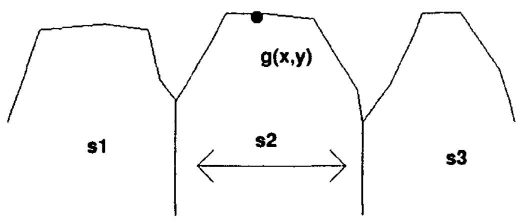

Fig. 3. The curve formed by $2M + 1$ points in the direction perpendicular to the current pixel's direction.

2.3. From Crease Point to Crease

Because of noise and complex realities, crease points may be missed or false. It's important to take use of global information to take back the lost points and reject the false one. And more, to ease the advanced palmprint analysis, the crease should be described by line segments, not a mess of point. We devised a search algorithm to form creases from crease points.

First, select a suitable start crease point, which is a center of a crease point region with the almost same direction. And use the average direction as the inital direction. Such deliberate start point and direction choosing helps much to both less duplicating and less miss. From the start point, we extend a line segment along the initial direction until either too many processed points or too many points with large direction difference or too many points with small gray value are found. The robustness of the extending process is guaranteed by using hysteresis. Three quantities are used corresponding the above three types. They can be reset when enough crease points encountered in the extending process. Otherwise they are accumulated until one of them is large enough.

Then we move the line segment in the direction perpendicular to itself up and down until regions with low gray value or different direction are reached. The region covered by this movement is considered to be the a crease part. The middle axis of the region is used to represent the whole crease part. The region end's middle point and average direction are used as the start point and direction for the next extending process. When no more extending is possible, the search process for current crease ends. All the middle axis segments form a crease.

2.4. Crease Filtering and Connecting

Because of computation cost concern, the above algorithm is one-pass, which may break some creases and give out partial duplicates of some creases. The noise also has such bad effect. We connect those creases with close direction and close end points. This process is carried out iteratively until no more creases possible to be connected. At last, we eliminate those creases that can be covered by other creases and filter out those short creases.

2.5. Multiscale framework

Our original source image is very large(864x864). The large size image is very time consuming. And more, the large width of creases frustrates the search algorithm. So we devised a multiscale framework which perform the above algorithm with scale-adjusted parameters on images with different scales. In large scale, the creases with large width are accurately extracted and they are excluded from consideration in small scale. While in small scale, those creases with small width are accurately extracted as well.

3. PALMPRINT RECOGNITION ALGORITHM BASED ON CREASE

In our crease-based palmprint recognition, the palmprint is treated as a set of line segments. We do not directly compare the creases in different images, instead, we compare the line segments comprising the creases for simplicity. The matching algorithm is of three steps. First, adjust the image so that translation and rotation transforms are counteracted. Then generate gray images representing the crease line segments. At last, compare the gray images. The matching result is a similarity coefficient.

3.1. Adjust the Line segments

First we made an assumption that different images have the same scale because we use the same hardware to scan the palmprint image. To adjust an image properly, just to counteract the translation and rotation transforms. Denote two images I_a and I_b, I_b is the one to be adjusted. We select two line segments L_0, L_1 in the image I_a. They should have enough lengths and proper angle difference. Select two line segments L_2, L_3 with enough lengths and proper angle difference in image I_b, we assume they correspond to L_0, L_1. Then we can compute the translation using the points of intersection of each pair. And also, we can compute the rotation using the angle difference between corresponding line segments. Now we can transform each line segment in image I_b using the computed translation and rotation. But we don't know the assumed correspondence is right or not. So we try all the line segment pairs in image I_b, and choose the one which make the transformed image most similar to image I_a as the right correspondence. Sometimes, the lines L_0, L_1 may be missed in image I_b, so we try all the line segment pairs in image I_a and choose the largest similarity as the similarity between the image I_a and I_b.

3.2. Generate Gray Image Representing Creases

After adjustment, if we compare the line segments and count the the number of matching segments the result is not very satisfying. It's hard to decide two line segments matches or not. Sometimes, one segment matches part of the other. Sometimes, two segments have a mutual matching sub-segment. To deal with such complexity, we generate a gray image from the crease line segments. Every pixel in the gray image is assigned a direction α(in the range from 0 to 180) if it's near a line segment with direction α. Otherwise, it's assigned 0xff.

3.3. Gray Image Matching

A pixel by pixel matching is performed on the generated gray images. If two corresponding pixels have close directions, it's treated as a hit. The sum of hits of two images show how similar the two images are. Finally we get a similarity coefficient of an image I_b to the image $I_a$by dividing the sum by the total length of crease line segments in I_b. The image with largest similarity coefficient is treated as the corresponding palmprint image.

4. RESULT

The original palmprint image is a scanned image of paper on which the palm is printed with ink. The figure 4 is the crease extraction result of a clip of the original image. No multiscale framework is utilized. It's used to show our algorithm's rationality. Both of the two types of creases are extracted.

We have used our crease extraction algorithm on over 100 palmprint images with size 864x864. Most of the ridges are extracted successfully. The results show that the performance in complicated situations in real world are very good. To save space, the representative result is omitted.

The table 1 is the result of palmprint matching. From the above images we select randomly 20 palmprints which has 2 samples each. The results show that 19 pairs match correctly. The similarity coefficient of the palmprint image in these pairs and the image belonging the same palmprint is the largest. The largest similarity coefficient is normalized to 1. The second largest similarity coefficient of each palmprint is shown in the table. These coefficients show the discernability of palmprint images.

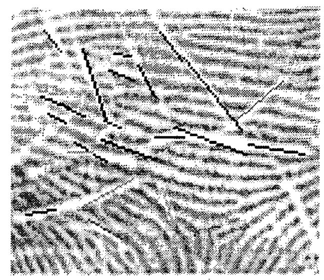

Fig. 4. The crease extraction result. Original image is the same as figure 1 . Color segments are creases.

No.	1	2	3	4	5	6	7	8
C.	.64	.84	.74	.78	.48	.18	.52	.80
No.	9	10	11	12	13	14	15	16
C.	.81	.64	.45	.91	.68	.95	.88	.69
No.	17	18	19	20				
C.	.93	fail	.55	.65				

Table 1. The palmprint matching result. C. row is the 2nd largest similarity coefficient.

5. SUMMARY AND FUTURE WORK

Our research on palmprint crease shows that creases can be extracted correctly, and more, based on the creases, the palmprint matching algorithm has great result. The crease will play an important role in palmprint analysis.

On the other side, the research on palmprint is just a start. Our methods can be improved in many aspects, especially in adaptive threshold selecting in the crease extraction algorithm. There's also work to do to accelerate the palmprint matching algorithm.

6. REFERENCES

[1] Jain A K, Hong L, Pankanti S et al. An identity-authentication system using fingerprints. Proc. of IEEE, 1997, 85(9):1365-1388

[2] Miao D, Maltoni D. Direct gray-scale minutiae detection in fingerprints. IEEE Trans. PAMI, 1997, 19(1):27-39

[3] J.Funada, etc., Feature extraction method for plamprint considering elimination of creases, Proceedings of ICPR'98,1849-1854

[4] Shu,Wei, Research on Automatic Palmprint Recognition, PhD thesis,Tsinghua Univ.,1999

[5] Wang, Chengxia, Mordern Palmprint Diagnosis, Lanzhou, Gansu Folk Press, 1995

[6] Wang, Dayou, Atlas of Palmprint Diagnosis, Beijing, Beijing Science and Technology Press, 1995

[7] H. Cummins and C. Midlo, Finger Prints, Palms and Soles, London:Dover Publications, 1943