Application of Palmprint Feature Analysis on Fate Theory

Hwei-Jen Lin
Tamkang University
hjlin@cs.tku.edu.tw

Hung-Hsuan Wu
Tamkang University
joseph5@seed.net.tw

Kuo-Chang Chen
Tamkang University
jerry@mail200.com.tw

Shwu-Huei Yen
Tamkang University
shyen@cs.tku.edu.tw

Abstract

Throughout the course of its development, palmprint features have always been studied and analyzed by experts from diverse fields, such as biometric identification, psychology, and fate theory. In this paper, we put into practice a system that analyzes palmistry. With the help of image processing techniques, we transmit the information extracted from the palmprint image into our analyzing system, and have the palmprint assayed. Further discussion about each processing step of the system and details about the experiments are also included in the paper.

Keywords: palmprint, biometric identification, thresholding, linear regression, conditional dilation.

1. Introduction

The emergence of palmistry can be dated back to an early age, and is divided into two schools: the Chinese and the Western. It is said that Western palmistry evolved at around 3000 B.C., and flourished at the same time as the Aryan Culture in India and the palmistry and physiognomy in China. But even though palmistry is divided into two schools, the Chinese and the Western, the contents of each school are in fact highly similar; or you can even say the only things different are the names.

Some experts in biological characteristics devote themselves to the research and analysis of palmprint features. Since it helps to identify people, its commercial value lies mostly in the installment of security systems, such as safes, gate-guard systems, etc. Besides this, it is employed mainly in the investigation of criminal cases, when they have to use palmprint- or fingerprint-identification techniques to identify suspects, which is an aid to the investigation and analysis of the case. Though PIN codes are simpler and easier for management, they have little defense against premeditated hackers; while using biological features for personal identification has many advantages, such as uniqueness, innateness, and convenience in carrying around.

Since most papers about palmprint features nowadays are focused on its application in personal identification, analysis on palmistry and Fate Theories are few and far between. Therefore, with the help of image processing techniques, we hope to find out the characteristics of gray palmprint images, extract the principal lines, and calculate the number of collateral lines beside the principal lines; actually creating a computer-based system for analyzing palmistry.

2. Description of Palmprint Features

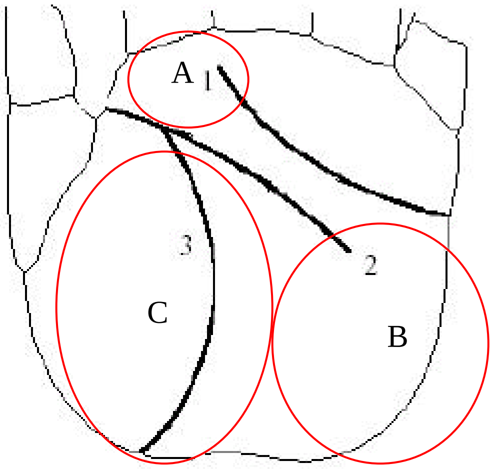
When we open our palms, most people would be able to point out with familiarity the Heart Line, the Head Line, and the Life Line; for these are the most prominent and best-known lines. Imprinted upon our hands, they are inseparable from us, accompanying us through our various lives.

The so-called palmprint refers to the countless crisscrossed long and short lines on our palm. Their number and depth depend on the individual; there might be only a few, yet there might be a lot as well. However, each line has its own name and meaning in palmistry [1]. So we will discuss the three principal lines as is necessary in this paper: the Heart Line, the Head Line, and the Life Line, as are shown in Figure 1. The introduction of each is as follows:

(1). The Heart Line: Running above the Head Line, it starts from the Mount of Jupiter (show in Figure 1 block "A") and ends below the little finger. See line No. 1 in Figure 1.

(2). The Head Line: Also called the Wisdom Line or Brain Line. It begins with the Life Line between the thumb and the index finger, and runs across the palm towards the Mount of the Moon. (shown in Figure 1 block "B") See line No. 2 in Figure 1.

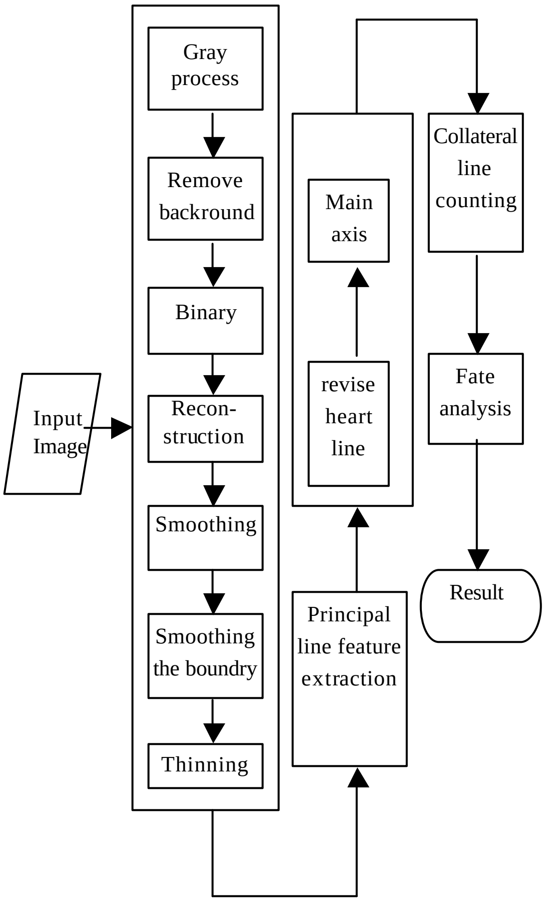
(3). The Life Line: Beginning between the thumb and the index finger and running down along the Mount of Venus; i.e. the line curving around the Mount of Venus. (shown in Figure 1 block "C") See line No. 3 in Figure 1.



【Figure 1. three principal lines and three mounts】

3. System Flow

The programming language that we use in our experiment is the Borland C++ Builder, Version 5.0. In order to simplify the running system and to make our study easier, all the palmprint image files are in the BMP format, and Figure 2 is our system architecture.



【Figure 2. system flow】

4. Image Process of Palmprints

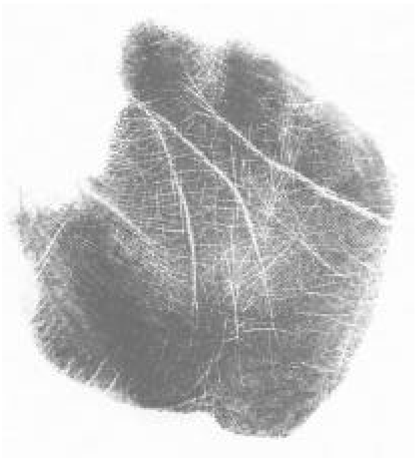
Since the purpose of our study is to provide users with a set of user-friendly software for palmprint analysis, we do our best to simplify the image-obtaining process. Therefore, after a series of experiments and analysis, we decide to use "red ink palmprint images" as the means for obtaining palmprints. In this way, users would be able to produce their own palmprint images at home and, with the help of scanners, turn them into digital images, and then feed them into the computer. This way we would be able to obtain the palmprint image easily, so that the software could analyze it.

4.1. Remove the background

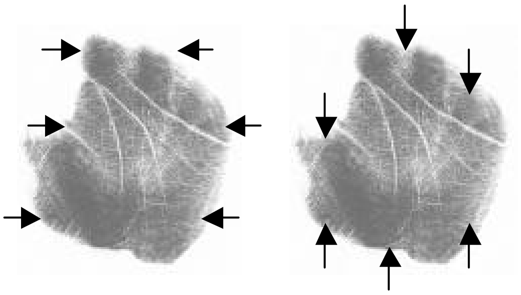
We discovered that, after being gray-level image, the gray values of the background color of the red ink palmprint image is extremely close to those of the principal lines in the palmprint (see Figure 3). If we do binary segmentation to remove the background, the actual background and the principal lines of the palmprint would be classified into the same category. Such a way of background removal would be less objective.

Therefore, we propose a solution: "to attack from four sides." In this way we will be able to separate the gray values of the actual background from those of the principal lines in the palmprint, successfully keeping the main block of the palmprint that we need.

The main idea of the "attacking from four sides" method is to press in from the boundaries on all four sides. This is helpful for defining the boundaries of the main block. We could then go on to remove the background, and to acquire the main block of the palmprint. See Figure 4.



【Figure 3. gray-level palmprint image】

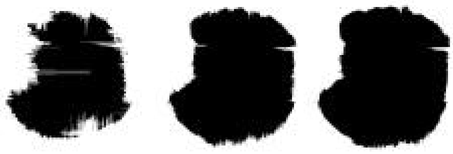


(a) attacking from left and right

(b) attacking from up and down

【Figure 4. attacking from four sides】

The most important part of this method is, we have to know how to obtain the appropriate threshold value of the surrounding boundaries, so as to aid the separation. Figure 5((a) and (b)) shows the most commonly used binary segmentation methods, such as Otsu method[5] and Mode method[8], applied to the processing of palmprint images. The results are not entirely satisfactory.



(a) Otsu method (b) Mode method (c) experiment

【Figure 5. results of thresholding】

Therefore, with manual experimentation to the current sample images and we found the optimal threshold value T for each image, see Figure 6. According to the results of these experiments, see Figure 5(c), we discovered a characteristic: the ratio, x, of the first peak's height to the second peak's height is linearly related to the ratio, y, of the difference between the first peak's gray value and its actual threshold value to the difference between the second peak's gray value and its actual threshold value. That is, point (x,y) corresponding to these images distribute around an invisible line *y* = *ax*+*b*, see Figure 7.

To find out this invisible line *y* = *ax*+*b*, or the coefficient *a* and *b*, we could use the linear regression method [7]. For the histogram of a sample image *i*, let $M1_i$ and $M2_i$ be the heights of the first and the second peaks, respectively, let $N1_i$ and $N2_i$ be the gray values of the first and the second peaks, respectively, and T_i the threshold value of the best segmentation.

Let $$x_i = \frac{M2_i}{M1_i}, y_i = \frac{N1_i - T_i}{T_i - N2_i}$$

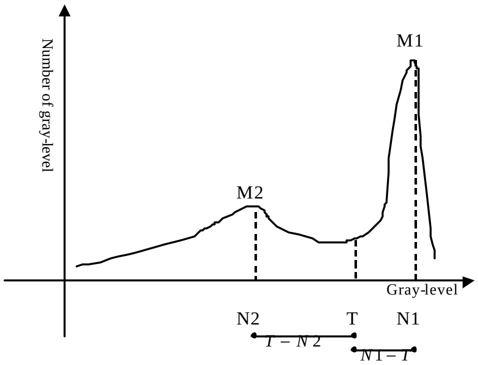
With the samples shown in Figure 7 and according to the result of the linear regression computation, we obtain the coefficients of the line *y* = *ax*+*b*:

$$a = 0.922754, \quad b = 0.075833.$$

With these coefficient values, we may automatically obtain the background-removal threshold value *T* for a test image, using the formula:

$$T = N2 + \frac{N1 - N2}{W},$$

where $$W = 1 + a\frac{M2}{M1} + b \cdot$$



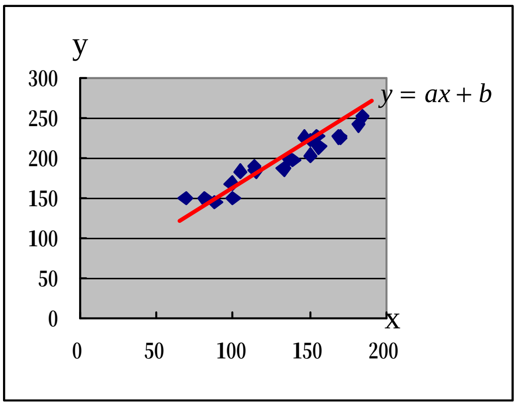
$$x = \frac{M2}{M1}, y = \frac{N1 - T}{T - N2}$$

T : threshold value

$M1$: first peak's total value, $M2$: the second peak's total value

$N1$: first peak's gray value, $N2$: the second peak's gray value

【Figure 6. histogram of the palmprint image with the optimal threshold value T】



【Figure 7. linear relation between values of x and y】

4.2. Binarization Operation

According to the results of another manual experiments, we found that the histogram of the main block of the sample palmprints also have a characteristic: the ratio, *x*, of the first peak's height to the second peak's height is linearly related to the ratio, y, of the difference between the second peak's gray value and i's optimal threshold value to the difference between the first peak's gray value and it's actual threshold value, see Figure 8 and Figure 9. For the histogram of the main block of a sample image *i*, the notations M_i1, M_i2, N_i1, N_i2 and T_i having the same definition as in the previous section. And let

$$x_i = \frac{M_i2}{M_i1}, y_i = \frac{T_i - N_i2}{N_i1 - T_i}.$$

With the samples shown in Figure 7 and according to the result of the linear regression computation, we obtain the coefficients of the line $y = ax+b$:

$$a=1.603684, \quad b=0.051251.$$

With these coefficient values, we may automatically obtain the binary threshold value T for the main block of a test image, using the formula: $T = N1 + \frac{N2 - N1}{W}$,

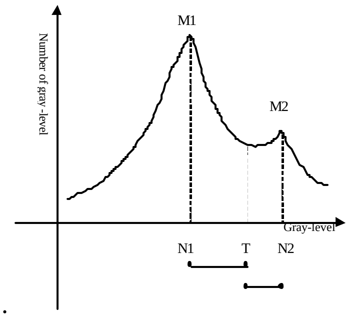
where $W = 1 + a\frac{M2}{M1} + b$

4.3. Conditional Dilation

In order to get a more complete principal line, we adopt the method of conditional dilation [6], set A as the condition, and let the Structural Element S_e acting on A until no more pixels appear. The notation for conditional dilation will be

$$A \quad \oplus \; (S_e, A')$$

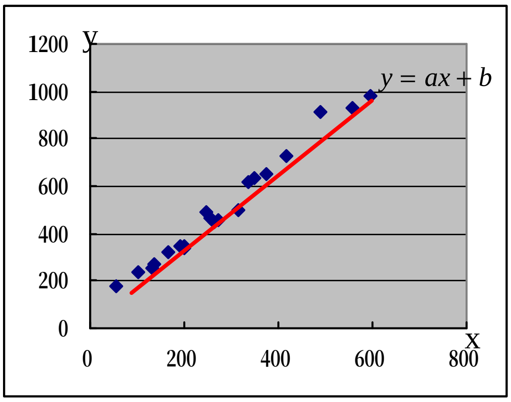
The result of the conditional dilation is shown in Figure 10, and the commonly used structural elements are shown in Figure 11. We use the structural element in (c) to do the conditional dilation.



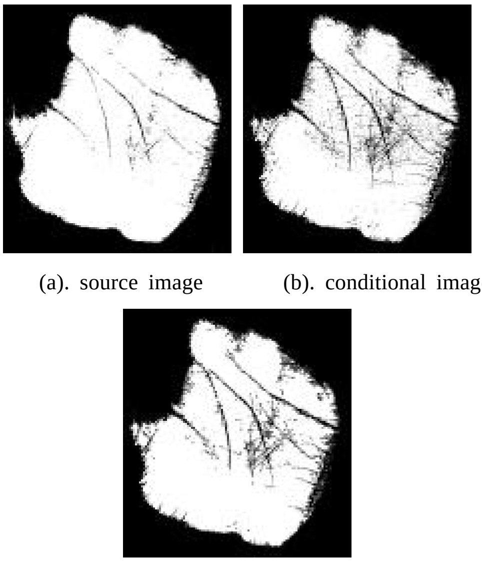
【Figure 8. histogram of the main block of palmprint image】

4.4. Smoothing

In order to smooth out the principal line so that it would be easier to extract the characteristics of a detailed image, we introduced the median filter here [3]. The median filter is defined as follows:



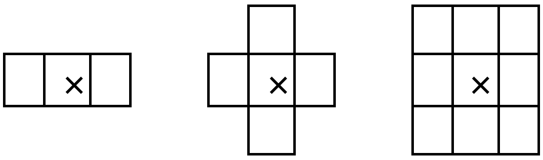
【Figure 9. linear relation between values of x and y】



(a). source image (b). conditional image

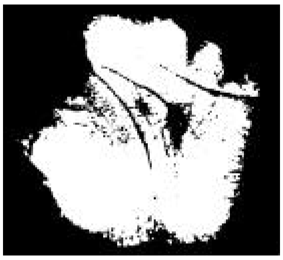
(c). resulting image

【Figure 10. the results of conditional dilation】



(a) 2-neighbor (b) 4-neighbor (c) 8-neighbor

【Figure 11. structural elements of conditional dilation】



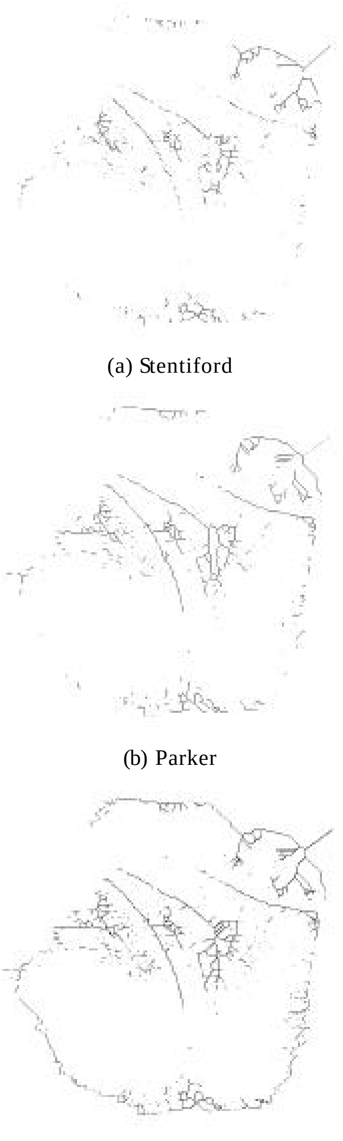
【Figure 12. smoothing result】

$$f(x, y) = median\{ f(x-k, y-l), (k,l) \in w\}$$

Here *w* is the 5x5 mask, and the result is shown in Figure 12.

4.5. Thinning

We have employed several commonly used thinning methods, such as Stentiford Method [9], Parker and Hold Method [4], and Zhang-Suen Method [10], to process our palmprint image. After the experiments, we adopted the Zhang-Suen Method. The results are shown in Figure 13.



(a) Stentiford

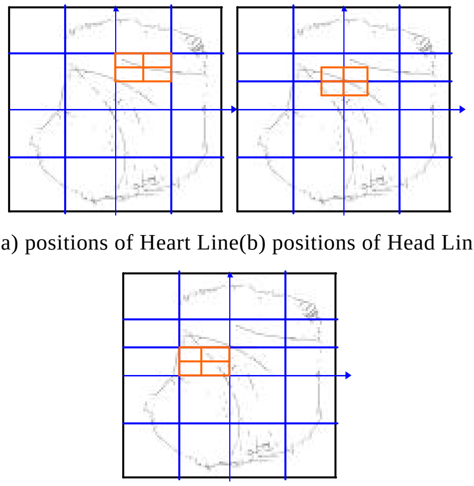
(b) Parker

(c) Zhang-Suen

【Figure 13. results of thinning】

4.6. Principal Line Extraction

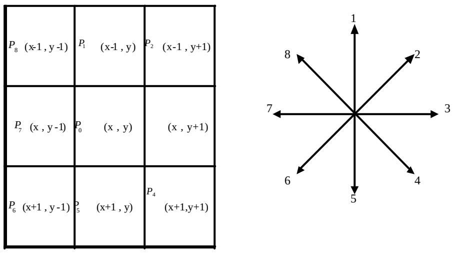
The three principal lines on a person's palm – the Heart Line, the Head Line, and the Life Line – have a special property: their relative positions on the palm is fixed, so if we start from the top, their sequence must be Heart Line, Head Line, and then Life Line, and could never be otherwise. Therefore, making use of this special property, we could use the Block Moving Method (shown in Figure 14) to find one feature point on each principal line separately and, using them as starting points, to trace each principal line with the 8 directions (see Figure 15.)



(a) positions of Heart Line(b) positions of Head Line

(c) positions of Life Line

【Figure 14. relative positions on the palm】



P_8 (x-1 , y -1)	P_1 (x-1 , y)	P_2 (x-1 , y+1)
P_7 (x , y -1)	P_0 (x , y)	(x , y+1)
P_6 (x+1 , y -1)	P_5 (x+1 , y)	P_4 (x+1,y+1)

【Figure 15. the 8 directions】

Steps for leftward tracing are as follows:

Step 1: Using P_0 as the basis, check in the three directions P_1, P_7, and P_8 to see if there's any black point. If yes, we go on to step two; if not, we skip to step six.

Step 2: Using P_0 as the basis, check in the three directions P_1, P_7, and P_8 to see if there is only one black point. If yes, we write down the co-ordinates of the point, and replace P_0 with this new point. Continue tracing by repeating step one. If the answer is no, we go on to step three.

Step 3: Now there are more than two points in the three directions P_1, P_7, and P_8, so we calculate the chain-code P_{max} of the average vector of the current track, which stands for the main trend, and go on to step four.

Step 4: Determine whether P_{max} is a black point or not. If yes, write down its coordinates, replace P_0 with this point and return to step one to continue tracing. If not, go to step five.

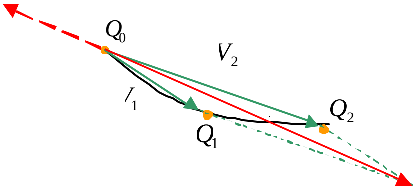
Step 5: Now, look for black points in this order P_8—P_7—P_1 , and as soon as we find one, we record its coordinates, replace P_0 with this point, and return to step one to continue tracing.

Step 6: Start marking-up processing. Rightward tracing is similar to this, except that the directions P_1, P_7, and P_8 are replaced by P_3, P_4 and P_5.

And steps for marking up are as follows:

Step 1: Suppose we've just arrived at Q_n. Move forward and take two other points Q_1 and Q_2, calculate the vectors V_1 (Q_n Q_1) and V_2 (Q_n Q_2), then calculate the mean vector V of V_1 and V_2. And then, starting from Q_n, look for dots in the opposite direction of vector V. The way to trace this vector is shown in Figure 16.

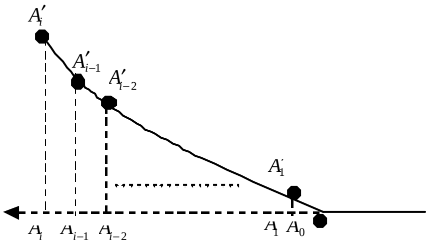
Step 2: Make a move in the opposite direction of vector V, which brings us to point A_i, and then determine whether there exists a point A'_i on or around A_i. If yes, go on to step three. If not, repeat step two until we come to the boundary.



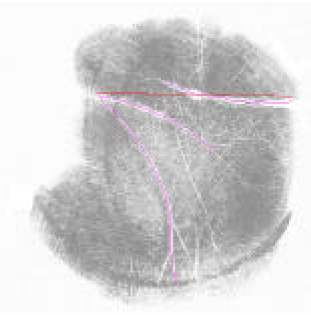
【Figure 16. tracing the vector】

Step 3: Suppose we find A'_i now, and we also know that we've moved i units so far. Calculate the ratio ΔA of the distance between A'_i and A_i to the distance i, then mark up in the forward direction until we reach point A_0, then go on to step four. The way to mark up is shown in Figure 17.

Step 4: Mark up process complete. Return to the tracing process to continue tracing. The result of tracing and marking up is show in Figure 18.



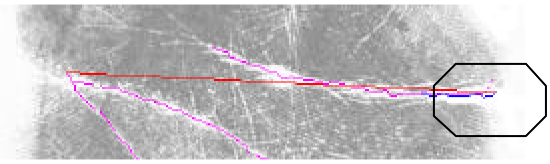
【Figure 17. method of marking-up】



【Figure 18. result of tracing and marking up】

4.7. Revising the Heart Line

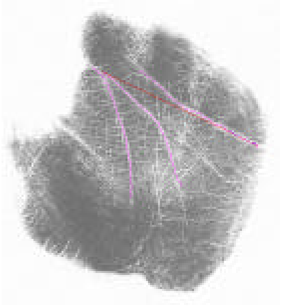
Since the tail of the Heart Line in a red ink palmprint image often to cause some tiny branches, after we trace this part, the line is prone to slant either upwards or downwards, making it easy to misjudge the curvature of the Heart Line's tail. Therefore we need to return to the gray-level palmprint image, and adjust the curvature of the tail part according to the gray values of the last section of the Heart Line. First we extract the gray values of three random points on the first three quarters of the section: T_a、T_b、T_c, get their average, have them moved for the length of d units, and then use the value T_m, which we gained from these calculations, as the threshold value to detect changes in the gray values of the palmprint. And then, using this value T_m, we start to reappraise the last quarter of the Heart Line. First we measure the upper and lower boundaries of the Heart Line block and calculate its median value, and then we can obtain the new coordinates. And according to these new coordinates, we move it one pixel to the right, and repeat the steps above, until we come up with the right side boundaries. The result is shown in Figure 19.



【Figure 19. Revising the Heart Line】

4.8. Main Axis Position

After such adjustments of the Heart Line, we would be able to get a more accurate coordinates of the point of the tail part. And then, connecting this point with the leftmost tip of the Head Line, the main axis is produced, as shown in Figure 20.



【Figure 20. Main Axis Position】

4.9. Collateral Line Counting

In order to find the number of collateral lines around the principal line efficiently, we can return to the original gray-level image. Since the features of the collateral lines' gray-level values are similar to those of the principal line, the gray-level values of collateral lines are higher. According to such a characteristic, we can use each point p_i in principal line and find the actual boundary point e_i in one side, and then, starting from e_i, select four points $z1_\mathrm{i}$, $z2_\mathrm{i}$, $z3_\mathrm{i}$ and $z4_\mathrm{i}$ along the direction perpendicular to the principal line, calculate their gray-level values $t1_\mathrm{i}$, $t2_\mathrm{i}$, $t3_\mathrm{i}$ and $t4_\mathrm{i}$, and compare these to the threshold value T_m calculated in the previous section. For each tk_i, if it exceeds T_m, then the point is possibly on the collateral line, so we increase W_k by 1(where W_k denotes the width of the cross-section of the detected collated line and it's initial value is set to 0, where $k = 1,2,3,4$). If tk_i does not exceeds T_m, then check if $W_k \geq num$ (by experience we obtain that $num = 3$): if $W_k \geq num$, then we increase the value of N_k by 1 and reset $L_k = 0$. If tk_i exceeds T_m, then just increase W_k by 1. All the detail processes are shown in algorithm 1. Finally, we select the maximum value N_{max} from N_1, N_2, N_3 and N_4, which is the estimated number of collateral lines along one side of the principal line. Figure 21 illustrates the way to estimate the number of collateral lines, and an example is shown in Figure 22. Collateral lines along the other side of the principal line can be counted in the same way.

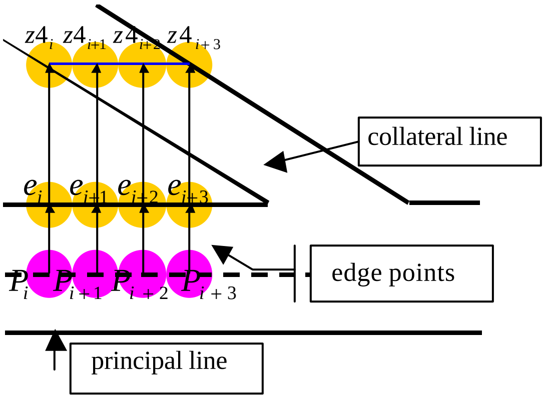
```
for $k=1$ to $4$
  $N_k = 0$ and $W_k = 0$
for $i=1$ to $n$
  find edge point $e_i$ for $P_i$ in one side
  for $k=1$ to $4$
    if $tk_\mathrm{i} > T_m$ then $W_k = W_k + 1$
    else
      if $W_k \geq num$ then $N_k = N_k + 1$
      $W_k = 0$
$N_{MAX} = \underset{k}{Max}\, N_k$
```

【Algorithm 1. Collateral Line Counting Algorithm】

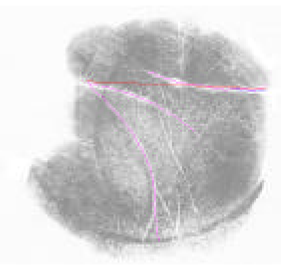
5. Analysis Results

According to the rules of Fate Theory [2], we could gather the information, apply it to the parameters obtained by image processing, and get the result of the analysis. What follows is a list of rules, and an example of fate analysis is shown in Figure 23.

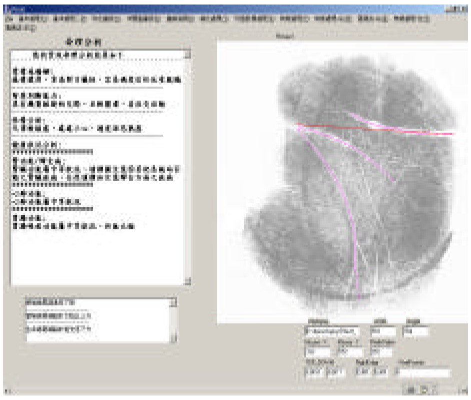
■ Information needed for the Heart Line: number of collateral lines above the Heart Line, number of collateral lines below the Heart Line, and whether the curvature of the Heart Line's tail is of the upside trend or the downside trend.



【Figure 21. Collateral Line Counting】

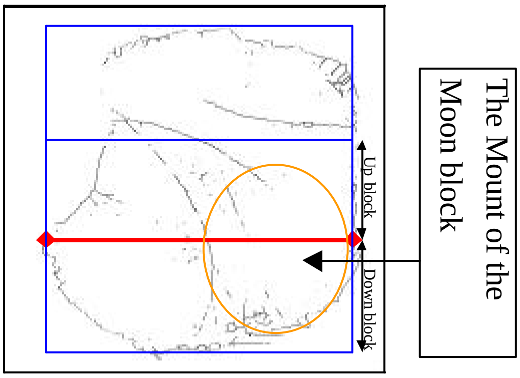
	Count numbers	
	Heart Line	upper collateral lines: 7 lower collateral lines: 8
	Head Line	upper collateral lines: 2 lower collateral lines: 8
	Life Line	right-side collateral lines: 8 left-side collateral lines: 2

【Figure 22. the example of Collateral Line Counting】



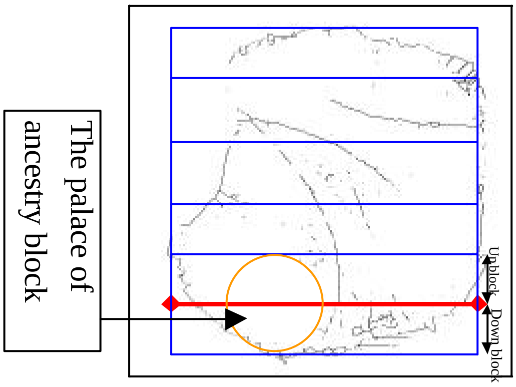
【Figure 23. Example of Fate Analysis】

■ Information needed for the Head Line: number of collateral lines above the Head Line, number of collateral lines below the Head Line, and the location of the block in which the Head Line ends (i.e. whether the block lies above or below the Mount of the Moon(**太陰丘**). According to its approximate position, we divide the Mount of the Moon into two parts: the upper part covers two-thirds of the Mount, while the lower part occupies one-third. This is how we decide whether the tail ends above or below the Mount of the Moon.) See Figure 24.



【Figure 24. the Mount of the Moon】

- Information needed for the Life Line: number of collateral lines to the right of the Life Line, number of collateral lines to the left of the Life line, and the location of the block in which the Life Line ends (i.e. whether the tail lies above or below the palace of ancestry(祖先宮). According to its approximate position, we divide the palace of ancestry into two parts: the upper part covers 5/6 of the Mount, while the lower part occupies one-sixth. This is how we decide whether the tail ends above or below the palace of ancestry.) See Figure 25.



【Figure 25. the palace of ancestry】

6. Conclusions and Future Work

So far, papers about palmprints are mostly concerned about applications like personal identification; analyses about its application in palmistry are fewer. Therefore, using image-processing techniques, we find the characteristics of gray-level palmprint images, extract the features of principal lines, calculate the number of collateral lines, and apply them to the palmistry analysis module.

Because it is difficult to get the concave part at the center of the ink palmprint, after pre-processing, there will often be a darker block at the center of the palmprint. Even though we use auxiliary methods to extract the trends when we are tracking the principal lines, we still hope to further improve this condition in future work.

There is also a lot of other information in a palmprint that could be used for further analysis in palmistry, such as color and luster. How we could effectively extract the features of this in

7. References

[1] 李鐵筆著,“手掌乾坤”,益群書局, 2001.

[2] 許紹龍著,“手相與面相”, 隆泉書局, 1986.

[3] R. Gonzalez and R. Woods (ed.), “Digital Image Processing”, Addsion-Wesley Publishing Company, 1992.

[4] C. M. Holt, A. M. Stewart Clint, and R. H. Perrott, “An Improved Parallel Thinning Algorithm”, Communications of the ACM, Vol. 30, No. 2, pp. 156-160, 1987.

[5] N. Otsu,“A Thresholding Selection Method from Gray-scale Histogram”, IEEE Transactions on System Man Cybernet, Vol. SMC-8, pp. 62-66, 1978.

[6] J. R. Parker (ed.), “Algorithms For Image Processing and Computer Vision”, John Wiley & Sonics Inc. Pub. , 1996.

[7] G. Peters (ed.), “Fuzzy Linear Regression with Fuzzy Intervals”, Fuzzy Sets and Systems, 1994.

[8] J. M. S. Prewitt and M. L. Mendelsohn, “The Analysis of Cell Images”, in Ann. New York Acad. Sci. Vol. 128, pp 1035-1053, 1966.

[9] F. W. Stentiford and R.G. Mortimer, Some New Heuristics for Thinning Bonary Hand-printed Characters for OCR”, IEEE Transactions on Systems, Man, and Cybernetics, Vol. 13, No. 1, pp 81-84, 1983.

[10] S. Zhang and K. S. Suen,“A thinning Algorithm for Discrete Binary Images”. Processing of the International Conference on Computers and Application. Beijing, China. pp. 879-886, 1984.