Technology and Health Care 23 (2015) S343–S353
DOI 10.3233/THC-150971
IOS Press

An effective hand vein feature extraction method

Haigang Li*, Qian Zhang and Chengdong Li
School of Information and Electrical Engineering, China University of Mining and Technology, Xuzhou, Jiangsu, China

Abstract.
BACKGROUND: As a new authentication method developed years ago, vein recognition technology features the unique advantage of bioassay.
OBJECTIVE: This paper studies the specific procedure for the extraction of hand back vein characteristics.
METHODS: There are different positions used in the collecting process, so that a suitable intravenous regional orientation method is put forward, allowing the positioning area to be the same for all hand positions. In addition, to eliminate the pseudo vein area, the valley regional shape extraction operator can be improved and combined with multiple segmentation algorithms. The images should be segmented step by step, making the vein texture to appear clear and accurate. Lastly, the segmented images should be filtered, eroded, and refined. This process helps to filter the most of the pseudo vein information.
RESULTS: Finally, a clear vein skeleton diagram is obtained, demonstrating the effectiveness of the algorithm.
CONCLUSION: This paper presents a hand back vein region location method. This makes it possible to rotate and correct the image by working out the inclination degree of contour at the side of hand back.

Keywords: Vein recognition, image processing, region segmentation, feature extraction

1. Introduction

Biometric identification technology, as one of the hottest areas of information technology at present, has been widely used in finance, information security, manufacturing, and e-commerce [1]. The current common biological areas for biometric identification technology fall into physiological categories like fingerprints, palm prints, faces, veins, and irises, as well as behavioral characteristics like handwriting and voices. As a new authentication method developed years ago, vein recognition technology features the unique advantage of bioassay. This method is remarkably superior to identification technologies based on fingerprints or other biological characteristics in the areas of anti-counterfeiting and anti-interference properties [2,3]. Because direct contact with equipment is not required, vein recognition is handier and more acceptable.

Hemoglobin is absorbed in venous blood when near an infrared ray, meaning that an infrared camera is used to capture the vein distribution graphs of the back of the hand, the fingers, and the palm. Through pre-treatment steps as normalization, de-noising and the determination of region of interest, the distribution graphs are converted into a standard image. After enhanced filtering, intravenous lines

*Corresponding author: Haigang Li, School of Information and Electrical Engineering, China University of Mining and Technology, Xuzhou, Jiangsu, China. Tel.: +86 13952194041; E-mail: haigangli@cumt.edu.cn.

Fig. 1. The original vein image.

segmentation, detailing and restoration, vein characteristics are extracted and then compared with vein images in database in order to identify individuals [4,5].

This paper studies the specific procedure for the extraction of hand back vein characteristics. Since the hand back of the user may be subject to movement and rotation during the process of acquisition, there is a need for a method for accurate positioning of vein region. This method is called ROI positioning, also known as the positioning of the region of interest. Upon the determination of the region of interest, the size of vein image obtained needs to be normalized for convenient subsequent treatment. A detailed vein sketch will then be achieved through filtering, erosion, and other morphological operations. This eventually leads to the development of a vein skeleton diagram.

2. ROI location

Since the original vein image varies significantly depending on the user's posture, ROI location shall be realized by the dedicated processing algorithm. As for the way we acquire intravenous data, the location method may vary based on the acquisition equipment. For example, Concordia University developed a palm vein acquisition system, which detects the junction between the forefinger and middle finger and the one between the ring finger and little finger after the acquisition of palm vein image. It then draws a linear segment and describes the perpendicular bisector at mid-point, before figuring out the designed region along perpendicular bisector [6]. The vein image captured for this paper is shown in Fig. 1, according to which, clear contours at the right and front parts are available for the backs of hands. Since it is observed that the contour remains substantially unchanged, even when the position of the hand changes, this paper takes the contour at right part of hand back and the raised area of middle finger as location reference.

2.1. Common edge detection algorithms

There are two kinds of image edge detection: look-up based detection and zero crossing based detection. For the first technique, the detection edge is determined by seeking the maximum and minimum values of the first-order derivative in the image. As for the second technique, the boundary is determined by seeking the zero-crossing point through the calculation of the second-order derivative of the image, where the Laplacian operator is used as a classic second-order operator. The second technique is based on the second-order difference, which may result in an enhanced impact from noise. In contrast, commonly used first-order operators include the Roberts operator, the Sobel operator, the Prewitt

operator and the Canny operation, each of which suit the requirements of different applications. If the edge of the vein image is dark, the Roberts, Sobel and Prewitt operators cannot effectively extract the boundaries. Despite image interference, the accurate positioning of the vein region is not available due to an incomplete contour. Edge detection performed using second-order Laplacian operator gives a relatively complete external contour, around which there are several noise points. These bring about poor anti-jamming capacity against intravenous regional location.

2.2. Canny algorithm for edge detection

The Canny operator for edge detection is a multi-stage edge detection algorithm developed by John in 1986 [8]. He put forward three criteria for evaluation of the merits of detection operator: high SNR criterion, accurate positioning criterion, and single-edge sole response criterion. The Canny algorithm for edge detection has three steps, de-noising, seeking brightness gradient, and following the edge. Considering the dense calculation for the traditional algorithm, the processing procedure is simplified as follows:

Step 1: Since the original image comprises noise, direct edge detection may impair the performance of the edge detection algorithm. Thus, it is important to perform Gaussian smoothing filtration of the original image, so that the step noise of a single pixel would not impact edge detection. In practice, eight pixels near the filtration point are averaged to obtain a smooth image.

Step 2: Sobel operators in different directions are used with points to be detected (point 8 in neighborhood) to determine the gradients in 45°, 135°, horizontal, and vertical directions. The maximum of the four gradients is taken as the gradient for this point, while the maximum gradient direction is recorded, thereby obtaining a brightness gradient map including direction information.

Step 3: Use high and low thresholds for edge detection. In the brightness gradient map, the point with a higher gradient value is likely to belong to the edge, but there is no clear way to select the threshold. For this reason, the Canny algorithm employs two thresholds, the high threshold and the low threshold. First, a higher threshold is used to detect the obtained brightness gradient to get a reliable boundary contour with poor coherence. The low threshold is then used for tracking the search along the contour edge determined by the high threshold, thus identifying the boundary of the fuzzy region.

As shown in Fig. 2, the Canny detection algorithm works well for the segmentation of dark regions in vein image and brings about no Laplacian operator-induced noise points around the region of the back of the hand. Above all, the contour is continuous. Therefore, this algorithm enables a reliable hand back segmentation contour.

2.3. Regional location

Upon the determination of the contour of the vein, it is essential to figure out the rotation degree of hand geometry and perform the appropriate corrections. In this paper, we performed the least square fit at ten points at the right boundary of contour to create a regression line. In this way, we were able to figure out the hand geometry rotation angle and perform the reverse rotation of the image.

Upon the correction of the image rotation, the Canny algorithm is used again to figure out its contour, detect the median area projection, and record its nearest point from the top of image. Then, the fitted straight line is used to determine the shear region as shown in Fig. 3, where, “w” represents 20 pixel

Fig. 2. Operator Canny edge detection.

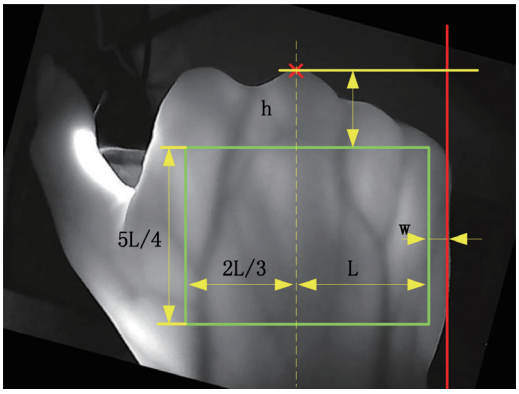


Fig. 3. Vein cutting area location.

distance; "h" is 60 pixel distance; "L" is obtained by subtracting the distance between the acme of the middle finger projection and fitting straight line from "w"; the other region thereafter takes 2L/3; the height of the shear region is determined as 5L/4. Finally, it's possible to get a relatively fixed shear region that may vary depending on hand size.

3. Vein image normalization

Since the vein images obtained after making up do not match in size and the gray-scale value's distribution area is relatively narrow, the images have to be normalized in the first instance. This allows the output images to be consistent with the same criteria to assure the consistency of vein images during the extraction.

3.1. Scale normalization

Vein information is distributed in an image in a relatively dispersed fashion, so the direct treatment thereof requires heavy load of calculation. It is thus necessary to adjust the dimensions to reduce the required calculations and normalize the images to the same size. Size transformation is performed through bilinear interpolation, which ensures the minimization of the loss of image information.

The pixel point of this original image is $[x, y]$; the position of the pixel point after the transformation is $[x_1, y_1]$; the proportions of change in x, y directions are k_1 and k_2, respectively, whereupon:

$$\begin{bmatrix} x_1 \\ y_1 \end{bmatrix} = \begin{bmatrix} k_1 & 0 \\ 0 & k_2 \end{bmatrix} \begin{bmatrix} x \\ y \end{bmatrix} = \begin{bmatrix} \dfrac{1}{k_1} & 0 \\ 0 & \dfrac{1}{k_2} \end{bmatrix} \begin{bmatrix} x_1 \\ y_1 \end{bmatrix} \tag{1}$$

According to Eq. (1), when $1/k_1$ and $1/k_2$ are integers, it is possible to work out the accurate position of original image pixel point by introducing the pixel position of the converted image into the equation. However, since $1/k_1$ and $1/k_2$ are not integers normally, it is impossible to accurately determine the pixel position of original image. This means it is advisable to perform an approximate calculation of the value of the converted pixel point through bilinear transformation.

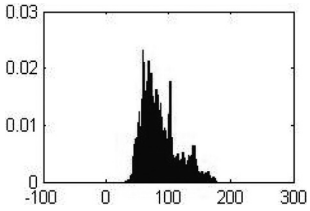
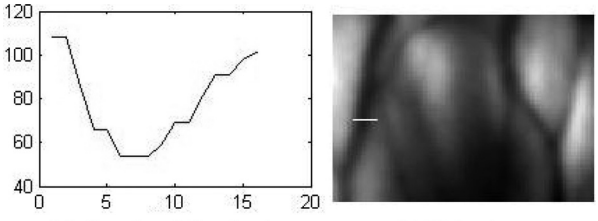


Fig. 4. Gray histogram of vein image.

Fig. 5. Vein image normalization.



(a) Gray level distribution (b) Veins image

Fig. 6. Veins gray level distribution.

Assuming that $[\widetilde{x}, \widetilde{y}]$ is the approximate integer value of $[x, y]$, which is the original image position obtained through Eq. (1), that $p(x, y)$ represents the pixel value of the $[x, y]$ position of original image, and that $p'(x_1, y_1)$ is the pixel value of the converted image at $[x_1, y_1]$, the value of $p'(x_1, y_1)$ could approximately be expressed by Eq. (2).

$$\begin{aligned} p'(x_1, y_1) = {} & (\tilde{y} + 1 - y) \times ((x - \tilde{x}) \times p(\tilde{x} + 1, y) + (\tilde{x} + 1 - x) \times p(\tilde{x}, \tilde{y})) \\ & + (y - \tilde{y}) \times ((x - \tilde{x}) \times p(\tilde{x} + 1, \tilde{y} + 1) + (\tilde{x} + 1 - x) \times p(\tilde{x}, \tilde{y} + 1)) \end{aligned} \tag{2}$$

The three points selected around the $[\widetilde{x}, \widetilde{y}]$ position of the original image compose a 2×2 square. Assuming that the distance between two pixels is 1, we can work out the post-transformation pixel value through the position of the point $[x, y]$ in the square using its proportion value [9].

3.2. Gray-scale transformation

As shown in Fig. 4, the gray-scale information of the image is relatively concentrated after making-up. According to the figure, the gray-scale information is concentrated in region 50–180. In order to get a clear image, a gray-scale transformation could be performed for the image, extending the gray-scale value to 0–255, thereby improving the image contrast.

Gray-scale transformation is divided into linear gray-scale transformation, piecewise linear gray-scale transformation, and non-linear gray-scale transformation. Piecewise linear gray-scale transformation and non-linear gray-scale transformation are intended to highlight the image details needed, as well as compress certain immaterial gray information. This is enough to increase the contrast of the image.

Assuming that the gray-scale range of original image $f(x, y)$ is $[a, b]$, if the gray-scale range of image $g(x, y)$ after transformation is [0,255], $g(x, y) = \frac{255}{b-a}[f(x, y) - a]$ would be true. The final vein image after the scale normalization and gray-scale transformation is shown in Fig. 5, from which it is observed that the vein information is more conspicuous.

4. Vein segmentation

4.1. Algorithm introduction

Upon the normalization of the image, it is necessary to extract the vein skeleton by dividing the image into background and foreground sections through binary segmentation. Commonly used fixed threshold segmentation methods include the averaging method, the OTSU method, and the NiBlack method. However, these image segmentation methods are not ideal for vein imaging since it's easy to

$$\begin{bmatrix} 0&0&0&0&0&0&0&0&0&1&1\\ 0&0&0&0&0&0&0&0&1&1&1\\ 0&0&0&0&0&0&0&1&2&1&0\\ 0&0&0&0&0&0&1&3&1&0&0\\ 0&0&0&0&0&-4&-5&1&0&0&0\\ 0&0&0&0&-4&-6&-4&3&0&0&0\\ 0&0&0&1&-5&-4&1&0&0&0&0\\ 0&0&1&3&1&0&0&0&0&0&0\\ 0&1&2&1&0&0&0&0&0&0&0\\ 1&1&1&0&0&0&0&0&0&0&0\\ 1&1&0&0&0&0&0&0&0&0&0 \end{bmatrix} \quad \begin{bmatrix} 1&1&0&0&0&0&0&0&0&0&0\\ 1&1&1&0&0&0&0&0&0&0&0\\ 0&1&2&1&0&0&0&0&0&0&0\\ 0&0&1&3&1&0&0&0&0&0&0\\ 0&0&0&1&-5&-4&0&0&0&0&0\\ 0&0&0&0&-4&-6&-4&0&0&0&0\\ 0&0&0&0&0&-4&-5&1&0&0&0\\ 0&0&0&0&0&0&1&3&1&0&0\\ 0&0&0&0&0&0&0&1&2&1&0\\ 0&0&0&0&0&0&0&0&1&1&1\\ 0&0&0&0&0&0&0&0&0&1&1 \end{bmatrix}$$

45° direction operator 135° direction operator

$$\begin{bmatrix} 0&0&0&0&0&0&0&0&0&0&0\\ 0&0&0&0&0&0&0&0&0&0&0\\ 0&0&0&0&0&0&0&0&0&0&0\\ 0&0&0&0&0&0&0&0&0&0&0\\ 0&1&1&1&1&-8&1&1&1&1&0\\ 1&1&2&3&-5&-6&-5&3&2&1&1\\ 0&1&1&1&1&-8&1&1&1&1&0\\ 0&0&0&0&0&0&0&0&0&0&0\\ 0&0&0&0&0&0&0&0&0&0&0\\ 0&0&0&0&0&0&0&0&0&0&0\\ 0&0&0&0&0&0&0&0&0&0&0 \end{bmatrix} \quad \begin{bmatrix} 0&0&0&0&0&1&0&0&0&0&0\\ 0&0&0&0&1&1&1&0&0&0&0\\ 0&0&0&0&1&2&1&0&0&0&0\\ 0&0&0&0&1&3&1&0&0&0&0\\ 0&0&0&0&1&-5&1&0&0&0&0\\ 0&0&0&0&-8&-6&-8&0&0&0&0\\ 0&0&0&0&1&-5&1&0&0&0&0\\ 0&0&0&0&1&3&1&0&0&0&0\\ 0&0&0&0&1&2&1&0&0&0&0\\ 0&0&0&0&1&1&1&0&0&0&0\\ 0&0&0&0&0&1&0&0&0&0&0 \end{bmatrix}$$

horizontal direction operator vertical direction operator

Fig. 7. Valley shaped area enhancement operator.

generate non-vein information and pseudo-vein information. Some shaded regions exist from time to time in obtained vein image, but the magnitude of the change is not significant. The shaded regions are significantly different from the vein region. These disturbances must be eliminated, or the accuracy of identification may be jeopardized. Because the pixel values in the vein region are valley-shaped, the gradient changes sharply as shown in Fig. 6, while the change in gradient is slow or rigged in other regions. Therefore, it is advisable to enhance the vein region in the image by taking into account the gradient operator in the contour extraction, and to filter out the ridged and relatively flat regions, thus facilitating the extraction of intravenous lines.

By referring to the valley-shaped region enhancement algorithm and Canny gradient operator described in Reference [10], we present a valley-shaped region enhancement operator suitable for processing. The window size is intended to be 11×11. In order to reduce the computation load, this paper selects operators in 4 directions, i.e. 45°, 135°, horizontal and vertical directions, of which the specific parameters are shown in Fig. 7.

When seeking the convolution with the valley-shaped region enhancement operator and the pixel gray scale in 11×11 neighborhood around each pixel, the convolution value obtained is less than 0. This is because the gray-scale values of the central pixels at the ridge line are large while of surrounding pixels are small. The convolution value is greater than zero in valley-shaped regions; the convolution value for level region is close to zero. It can thus be seen that the valley-shaped region detection eliminates the regions that do not belong to the vein and retains the vein region and pseudo-vein region. Since the vein region is remarkably enhanced, it could be segmented by the averaging method, while the pseudo-vein region could be re-segmented by the NiBlack segmentation method. This achieves an accurate vein binary image.

The calculation procedure is as follows.

Step 1: Slide a template into the image to enhance valley-shaped regions. Convolution is performed between the valley enhancement operators in 4 directions and the gray-scale value in 11×11 neighborhood around the pixel at $[x, y]$, respectively. Record the maximum values and generate a new image, expressed as $G(x, y)$.

Step 2: Eliminate the regions that y do not belong to the vein in image $G(x, y)$, and the image is renewed as $G'(x, y)$.

Step 3: Perform the initial segmentation of image $G'(x, y)$ by the mean value method. Work out the number of non-zero pixels in image $G'(x, y)$, and express the "number" as *Num*. Calculate the mean value of non-zero pixels as:

$$Tavg = sum(G'(x, y))/Num \tag{3}$$

Step 4: Set the pixel that is greater than *Tavg* to zero, obtain the image $G''(x,y)$ containing pseudo-vein information, create a two-dimensional null matrix $G_{BW1}(x,y)$ in the same size as original image, and set the points at the same position to 1. Save them in matrix $G_{BW1}(x,y)$, i.e.

$$G_{BW1}(x,y) = \begin{cases} 1 & G'(x,y) \geqslant Tavg; \\ 0 & else; \end{cases} \tag{4}$$

Step 5: Slide the window in size 11×11 in image $G''(x,y)$, and work out the mean value and standard deviation of the current pixels and surrounding neighborhood as follows:

$$Avg(x,y) = \frac{1}{11 \times 11} \sum_{i=x-5}^{x+5} \sum_{j=y-5}^{y+5} G''(i,j) \tag{5}$$

$$\sigma(x,y) = \frac{1}{11} \sqrt{\sum_{i=x-5}^{x+5} \sum_{j=y-5}^{y+5} [G''(i,j) - Avg(x,y)]^2} \tag{6}$$

Figure out the mean value and the mean square of pixels in sliding window with Eqs (5) and (6), and determine the threshold for each pixel using the NiBlack method:

$$T(x,y) = Avg(x,y) + \alpha \times \sigma(x,y) \tag{7}$$

Upon that, work out the threshold for each pixel through Eq. (7), and mark the newly obtained vein image in two-dimensional null matrix $G_{BW2}(x,y)$ in the same size as the original image as shown in the equation below:

$$G_{BW2}(x,y) = \begin{cases} 1 & G''(x,y) > T(x,y); \\ 0 & else; \end{cases} \tag{8}$$

Step 6: Two binary vein images $G_{BW1}(x,y)$ and $G_{BW2}(x,y)$ obtained respectively by the averaging method and the NiBlack method could be achieved through Eqs (4) and (8). The first segmentation offers relatively obvious vein information, while the second segmentation is performed to separate vein information from pseudo-vein information and non-vein information, so the obtained complete binary image of vein should be:

$$G_{BW}(x,y) = G_{BW1}(x,y) + G_{BW2}(x,y) \tag{9}$$

Step 7: Most of the wrongly detected vein regions in the obtained binary image $G_{BW}(x,y)$ of the vein are small and independent. In order to obtain an accurate vein image, it's advisable to calculate the area of connected region for primary filtering to eliminate the regions in small size, thus generating an accurate vein segmentation binary image.

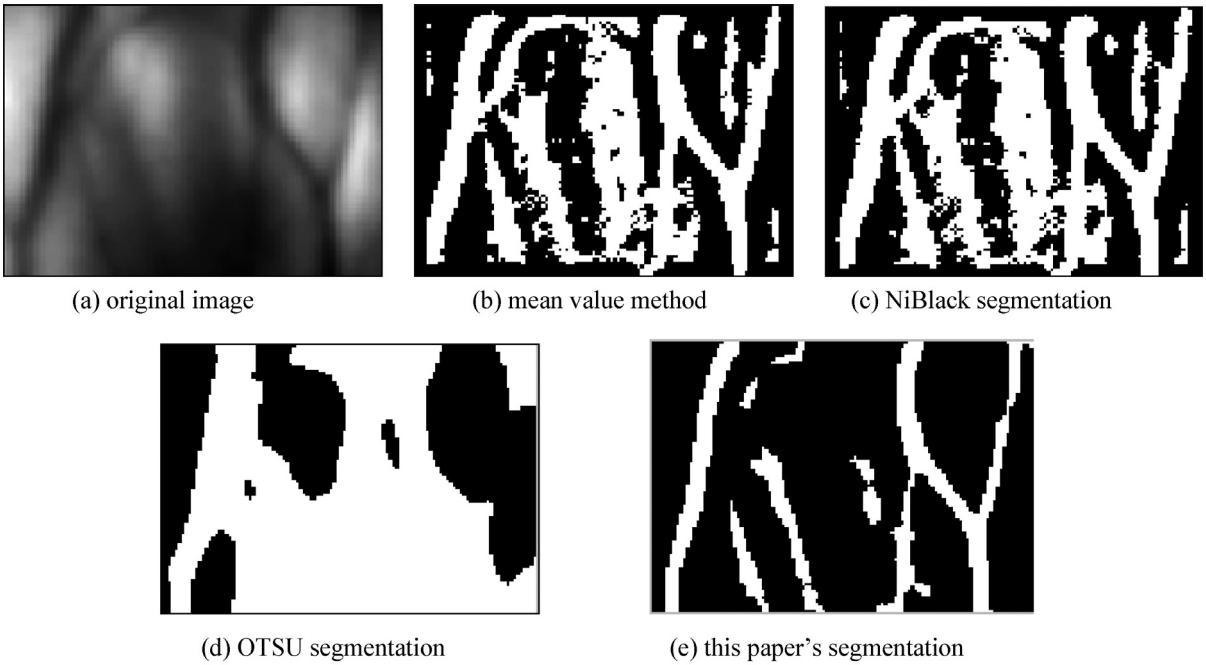


Fig. 8. Vein segmentation algorithm comparison.

4.2. Algorithm comparisons

Refer to Fig. 8 for the comparison between the vein image segmentation method presented in this paper and the single threshold segmentation algorithm. It is observed that the averaging method and the NiBlack method have certain effects on the segmentation of hand back vein image, despite all of the erroneous information. The OTSU segmentation algorithm is not suitable for the segmentation of this image; the valley enhancement and the combination of various segmentation algorithms in this paper bring about relatively satisfactory result by achieving high accuracy through the reservation of sufficient vein information and the elimination of most error information.

Table 1 shows the time complexity of above segmentation algorithms. We can see from Table 1 that, the run time in this paper's is the shortest.

In order to have objective and scientific comparison results, hypothesis testing is used on the experimental results. Let the variable X_1, X_2, X_3, X_4 denote the segmentation error rate of this paper's segmentation, OTSU, NiBlack and mean value algorithms, respectively. Since the value of X_1, X_2, X_3, X_4 is subject to many random factors, we assume that they submit to normal distribution, $X_i \sim N(\mu_i, \sigma_i^2)$, $i = 1, 2, 3, 4$. Now, we compare the random variable mean of these algorithms, μ_i ($i = 1, 2, 3, 4$). The smaller the μ_i is, the lower the expected classification error rate is, and the higher the efficiency is. Because the sample variance is the unbiased estimation of the overall variance, the sample variance value is used as an estimate of the generality variance. In this experiment the significance level α sets as 0.01. Table 2 shows the comparison process on μ_i and other parameters. Correct segmentation has high accuracy through the reservation of sufficient vein information and the elimination of most error information.

We can see from Table 2 that, the expectations of segmentation error rate in this paper's segmentation is far below than other algorithms.

Table 1
Comparison of computational complexity

Number	Methods	Time(s)
1	Mean value segmentation	1.573
2	NiBlack segmentation	1.284
3	OTSU segmentation	0.913
4	This paper's segmentation	0.768

P9	P2	P3
P8	P1	P4
P7	P6	P5

Fig. 9. Pixel labeling relation.

Table 2
Hypothesis testing for experimental results

Hypothesis	$H_0 : \mu_1 \geqslant \mu_2$ $H_1 : \mu_1 < \mu_2$	$H_0 : \mu_1 \geqslant \mu_3$ $H_1 : \mu_1 < \mu_3$	$H_0 : \mu_1 \geqslant \mu_4$ $H_1 : \mu_1 < \mu_4$
Statistics	$U_2 = \dfrac{\bar{X}_1 - \bar{X}_2}{\sqrt{\dfrac{\sigma_1^2}{n_1} + \dfrac{\sigma_2^2}{n_2}}}$	$U_2 = \dfrac{\bar{X}_1 - \bar{X}_3}{\sqrt{\dfrac{\sigma_1^2}{n_1} + \dfrac{\sigma_3^2}{n_3}}}$	$U_3 = \dfrac{\bar{X}_1 - \bar{X}_4}{\sqrt{\dfrac{\sigma_1^2}{n_1} + \dfrac{\sigma_4^2}{n_4}}}$
Rejection region	$U_1 \leqslant -Z_\alpha = -2.475$	$U_1 \leqslant -Z_\alpha = -2.475$	$U_1 \leqslant -Z_\alpha = -2.475$
Value of the statistic	$U_1 = -68.37$	$U_2 = -108.95$	$U_3 = -124.29$
Conclusion	$H_1 : \mu_1 < \mu_2$	$H_1 : \mu_1 < \mu_3$	$H_1 : \mu_1 < \mu_4$

5. Vein refinement

Despite the clear vein texture in the image after segmentation, the vein thickness detected may vary depending on the definition of the acquired image. In order to express vein characteristics more accurately, this paper is intended to refine the veins, extract the refined texture, and perform identification using more accurate geometrical information. For vein refinement, the edge pixels of three vein textures share the same pixel width. This assures the composite characteristic information like the length, the number of terminals, and the number of cross points of the vein becomes more compressed for efficient identification. Furthermore, vein refinement plays a certain role in de-noising the pseudo-vein region, where there are many branches and void regions.

The refinement algorithm used for this paper is as follows:

Assuming that all pixels in the binary image and in its neighborhood are marked as shown in Fig. 9, if the current pixel and the pixel in its neighborhood satisfy Eq. (10), the said pixel would be considered a boundary point that should be deleted. As shown in the equation, $N(P1)$ means the number of non-zero pixels in surrounding neighborhood of current pixel, while $T(P1)$ represents the number of rising edges (from 0 to 1) from pixel $P2$ to $P9$.

$$\left.\begin{array}{l} 2 \leqslant N(P1) \leqslant 6 \\ T(P1) = 1 \\ P2 \cdot P4 \cdot P6 = 0 \\ P4 \cdot P6 \cdot P8 = 0 \end{array}\right\} \tag{10}$$

In the next step, Eq. (11) should be used to check if there are any boundary points that can be deleted. Repeat the above two steps until there is no dispensable point in the image. The binary image of the vein

(a) vein original drawing (b) vein texture after refining (c) vein texture after deburring

Fig. 10. Vein refinement and deburring.

obtained after the processing and segmentation by this refinement algorithm is shown in Fig. 10(b).

$$\left.\begin{array}{l} 2 \leqslant N(P1) \leqslant 6 \\ T(P1) = 1 \\ P2 \cdot P4 \cdot P8 = 0 \\ P2 \cdot P6 \cdot P8 = 0 \end{array}\right\} \tag{11}$$

It is observed that the image directly processed by refinement algorithm contains many burrs. To smooth the vein texture, it is advisable to mark the terminal coordinates in the image. Search forward one by one along the vein texture until the next terminal or junction, and record its length. When the length is less than a certain set value, the image could be trimmed to get an image without burr as shown in Fig. 10(c), at which point the extraction of the vein characteristics is finished.

6. Conclusion

This paper presents a hand back vein region location method. This makes it possible to rotate and correct the image by working out the inclination degree of contour at the side of hand back, detecting the acme of the median projection, and then developing a trimming method suitable for various gestures based on the distance between this point and the contour line at the side of hand back. The feasibility of this method has been demonstrated by the experiment. Furthermore, this paper offers a synthetic algorithm for the extraction of the veins of the back of the hand through the combination of the merits of various segmentation algorithms. First, the self-developed valley-shaped region enhancement operator was used to obtain an enhanced vein image; then, a relatively clear and accurate vein texture was achieved through a number of segmentations by averaging method and NiBlcak method. Finally, a clear vein skeleton diagram was obtained through refinement for eliminating the pseudo-vein region in an extracted image, which demonstrated the reliability of this algorithm.

Acknowledgement

This work was supported by the Fundamental Research Funds for the Central Universities under Grant No. 2013XK09.

References

[1] S.J. Lu. A survey of biometric identification technology. Computer Security. 2013; (1): 63-67.
[2] L.Y. Wang, G. Leedham, Near and far infrared imaging for vein pattern biometrics. In: The IEEE International Conference on Video and Signal Based Surveillance, Sydney, Australia. 2006; 52-52.
[3] S. Crisan, I.G. Tarnovan, T.E. Crisan. A low cost vein detection system using near infrared radiation. In: The IEEE Sensors Applications Symposium, San Diego, CA, USA. 2007: 1-6.
[4] S. Lu, L. LI. Hand vein recognition based on improved FCM algorithm. Computer Engineering. 2010; 36(16): 153-156.
[5] Y.J. Zheng, X.D. Gu. Method for palm-dorsal vein recognition based on gabor phase encoding. Journal of Data Acquisition and Processing. 2010; 25(4): 516-520.
[6] Y.B. Zhang, Q. Li. Palm vein extraction and matching for personal authentication. Springer-Verlag Berlin Heidelberg. 2007; 154-164.
[7] L.J. Zhao, C.L. Jia, G.Y. Kuang. Overview of edge detection in SAR images. Journal of Image and Graphics. 2007; 12(12): 2042-2049.
[8] J. Zhao, W. Li, Z.P. Hao. Image edge detection based on improved canny operator and image morphology. Microcomputer and Its Applications. 2011; 30(10): 44-47.
[9] H.T. Huo, Digital image processing. Beijing Institute of Technology Press, 2002.
[10] C.B. Yu, H.X. Qin. The finger vein recognition technology. Tsinghua University Press, 2008.