FINGER VEIN RECOGNITION SYSTEM USING IMAGE PROCESSING

RASHMI R. TALLAM[1], SUPRIYA S. TEMGIRE[2], RINKU M. ZIRANGE[3]

[1,2,3]Dept. of E & TC, Bharati Vidyapeeth's College of Engineering for Women, University of Pune

Abstract- With increase in globalization and standard of living, there has been a steady increase in the use and development of consumer electronics. This calls for the need of convenient, simple, high security authentication systems to protect personal information stored in mobile devices. With increasing emphasis on security and high complexity of existing biometric systems in time or space or both, automated personal identification using finger-vein biometric is becoming a very active topic in both research and practical applications. In this paper, we have described and implemented an algorithm for finger-vein recognition system using image processing.

Index Terms - finger vein recognition, biometrics

I. INTRODUCTION

The traditional way to provide private information is by the use of either passwords or Personal Identification Numbers (PIN), magnetic swipe cards, keys and smart cards, that are easy to implement but are subjected to risk of exposure or being forgotten and are hence unreliable. As a result, biometrics that involve analysis of human biological, physical and behavioral characteristics have been developed to provide more reliable security[1].

A long list of biometric patterns is available. Many systems using these have been developed and implemented, namely face, iris, finger print, palm print, hand shape, voice, signature, gait and so on. Irrespective of this variety of biometric patterns, none of them are completely reliable and secure.

In case of fingerprint, the condition of the finger surface (e.g. dryness, sweat) and skin distortion degrades the recognition accuracy. Performance for face recognition, depends hugely on facial expressions and illuminations, which can change by occlusions or face-lifts. The biometrics like fingerprint, iris, signature, hand shape, voice, face do not necessarily provide confidentiality since the features used in the methods are exposed outside the human body. These methods are hence susceptible to forgery[2].

From the point of view of security and convenience, the finger-vein is a promising biometric pattern for personal identification[3].

The finger-vein has following advantages over other biometrics[4]:(1)Every individual has a unique pattern of veins and it is even different in case of identical twins. As the individual grows, the veins do become larger, but the position and number of veins do not change from infancy.(2)As the vein structure is underneath the skin, it is invisible to the naked eye and is very complex that it cannot easily spoof the system.(3)It is more acceptable by user because non-invasive and contactless capture of finger-vein provides convenience and hygiene.(4)It is a natural and convincing proof that the person whose finger-vein is captured is alive, since finger-vein pattern can only be taken from a live body.

II. THEORY

Edge detection is the method in which points are identified in a digital image at which the image brightness changes sharply or, more formally, has discontinuities.

The points at which image brightness changes sharply are typically organized into a set of curved line segments called edges.

The purpose of edge detection is to significantly reduce the amount of data in an image, while preserving the structural properties to be used for further image processing. Several algorithms exists for edge detection. One of them is developed by John Canny.

The algorithm runs in 4 separate steps[5]:

A. Smoothing: Blurring of the image to remove noise.

B. Finding gradients: The edges should be marked where the gradients of the image has large magnitudes.

C. Non-maximum suppression: Only local maxima should be marked as edges.

D. Hysteresis thresholding: Potential edges are determined by thresholding. Final edges are determined by suppressing all edges that are not connected to a very certain (strong) edge.

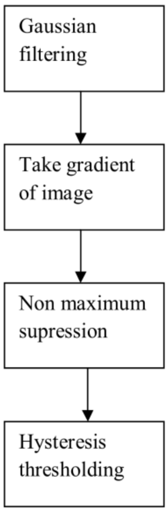


Fig. Flowchart for Canny Edge Detection

These steps are explained below briefly:

A. Smoothing: All images taken from a camera will contain some amount of noise. To prevent that noise is taken in place of edges, noise must be reduced. For this purpose the image is first smoothed by applying a Gaussian filter. The kernel of a Gaussian filter with a standard deviation of σ= 1.4 is shown in Equation (1) mentioned below

$$B = \frac{1}{159} \cdot \begin{bmatrix} 2 & 4 & 5 & 4 & 2 \\ 4 & 9 & 12 & 9 & 4 \\ 5 & 12 & 15 & 12 & 5 \\ 4 & 9 & 12 & 9 & 4 \\ 2 & 4 & 5 & 4 & 2 \end{bmatrix}$$

B. Finding gradients: The Canny algorithm finds edges where the maximum change in the grayscale intensity of the image. These areas can be found by determining gradients of the image. Gradients at each pixel in the smoothed image are determined by applying the Sobel-operator. First step in this is to approximate the gradient in the x- and y-direction respectively by applying the kernels shown in Equation (2).

$$K_{GX} = \begin{bmatrix} -1 & 0 & 1 \\ -2 & 0 & 2 \\ -1 & 0 & 1 \end{bmatrix}$$

$$K_{GY} = \begin{bmatrix} 1 & 2 & 1 \\ 0 & 0 & 0 \\ -1 & -2 & -1 \end{bmatrix} \quad (2)$$

The gradient magnitudes can be determined by applying Manhattan distance measure as shown in Equation (3) to reduce the computational complexity.

$$|G| = |G_x| + |G_y| \quad (3)$$

Where, G_x and G_y are the gradients in the x-direction and y-directions respectively.

Image of the gradient magnitudes often indicate the edges quite clearly. But the edges are typically broad and thus do not indicate exactly where the edges are. To make it possible to determine this , the direction of the edges must be determined and stored as shown in Equation (4).

$$\theta = \arctan\left(\frac{|G_y|}{|G_x|}\right) \quad (4)$$

C. Non maximum suppression: The purpose of the non maximum suppression is to convert the “blurred” edges in the image of the gradient magnitudes to “sharp” edges. Basically this is done by preserving all local maxima in the gradient image, and deleting all other part in the image.
The algorithm is for each pixel in gradient image:
1) Round the gradient direction theta to nearest 45◦, corresponding to use of an 8-connected neighbourhood.
2) Compare the edge strength of the current pixel with the edge strength of the pixel in the positive and negative gradient direction that means if the gradient direction is north (theta = 90◦), compare with the pixels to the north and south.
3) If the edge strength of the current pixel is largest then preserve the value of the edge strength. If not, suppress the value.

D. Hysteresis thresholding: The edge-pixels left after the non-maximum suppression step are (still) marked with their strength pixel-by-pixel. Many of these pixel will probably be true edges in the image, but some may be caused by noise or color variations for example due to rough surfaces. The simplest way to differentiate between these would be to use a threshold, so that only edges stronger than a certain value would be preserved. The Canny edge detection algorithm uses double thresholding method. Edge pixels stronger than the high threshold are marked as strong edges while edge pixels weaker than the low threshold are suppressed and edge pixels between the two thresholds are marked as weak edge.

Strong edges are taken as “certain edges”, and can immediately be included in the final edge image. Weak edges are included in the final edge image if and only if they are connected to strong edges. The result of this logic is that noise and other small variations are eliminated from a strong edge (with proper adjustment of the threshold levels). Hence, strong edges will (almost) be due to true edges only, in the original image. The weak edges can either be due to true edges or noise/color variations. The latter type will probably be distributed independently of edges on the entire image, and so only a small amount will be located adjacent to strong edges. Weak edges, as a result of true edges are much more likely to be connected to strong edges directly.

III. METHODOLOGY

A Finger-Vein recognition system consists of two stages - enrollment stage and verification stage. Both stages consist of following steps.

A. Image acquisition(Searching Database):
Image acquisition is of two type off-line and on-line. In image acquisition, we are using off-line images. On-line images are the images which are taken real time and off-line images means the images which are taken from already created database.

B. Image pre-processing:

1) RGB To Gray Conversion[5]: In RGB to Gray conversion, color image is converted into gray image. In RGB model, each color appears in its primary spectral component of red, green, blue. In gray image processing, gray level represents the interval number of quantization.
The formula for RGB to Gray conversion is stated below:

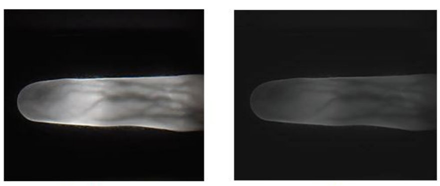
GRAY =0.299*r + 0.587*g +0.114*b

2) Image Enhancement: Image enhancement is used to improve the quality of an image. It is used to improve image contrast and brightness characteristics as well as to reduce noise contents. It highlights certain features of interest in an image.

3) Normalization: In normalization the original image is normalized into smaller size to achieve high accuracy finger-vein image. The image is normalized using following formula[6]:

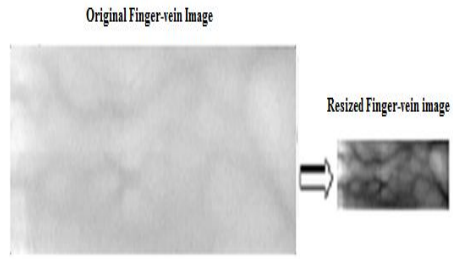
$$M = \frac{1}{M \times N} \sum_{i=0}^{M-1} \sum_{j=0}^{N-1} I(i,j)$$

$$VAR = \frac{1}{M \times N} \sum_{i=0}^{M-1} \sum_{j=0}^{N-1} (I(i,j) - M(I))^2$$

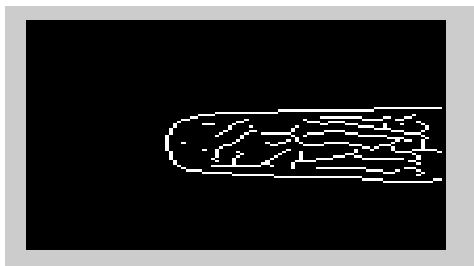


Original Image   Normalised Image

4) Resizing: By this, the dimensions of the image are changed to the required new width and height.



C. Segmentation: In image segmentation, the actual image is separated from its background and also noise is eliminated up to an extent. Image is then partitioned into multiple segments to locate objects and boundaries in the images. Segmentation is based on two basic properties of intensity values. First is discontinuity, in which image is partitioned based on sharp changes in intensity. Second is similarity, in which image is partitioned into the regions that are similar according to a set of predefined criteria. In edge detection, using edge operator, the edge point sets are extracted. There are various edge detection operators, like Sobel, Prewitt, Roberts, Laplacian of Gaussian (LoG), Zero cross and Canny, available for detection of edges. But comparison of these operators led us to the conclusion that Canny edge detection operator is most appropriate for finger-vein recognition, as we get better results as compared to the results obtained by using other operators. As a result, we have used Canny edge detection operator in our proposed algorithm.



Edge Detection Using Canny Operator

D. Matching: Euclidian distance is calculated for both the images – the image to be tested and the image from database. Following formula is used to calculate the Euclidean distance:

Euclidean distance between p and q is defined as,
D(p, q)=[(x-s)^2+(y-t)^2]^1/2

If the Euclidean distances of both the images mentioned above match each other, then the user is said to be identified or authenticated and he/she can have access to the system. On the other hand, if the distances do not match then user is said to be not-identified or not-authenticated and he/she is denied access to the system.

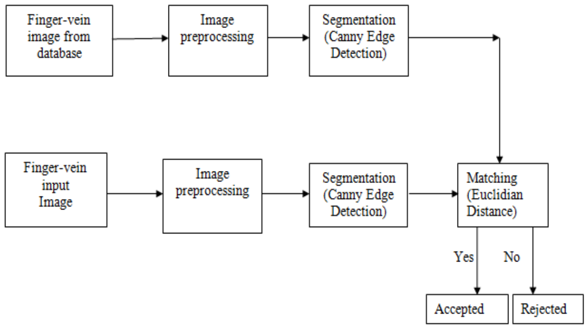


Fig. Block Diagram of Proposed System

IV. RESULT

We used SDUMLA-HMT, a finger vein database which, to the best of our knowledge, is the first open finger vein database. The device used to capture finger vein images is designed by Joint Lab for Intelligent Computing and Intelligent Systems of Wuhan University. There are two types of errors in matching results in biometric verification. The first is false rejection, which claims a genuine pair as impostor, and the second is false acceptance, which claims an impostor pair as genuine[7]. The system proposed in this paper achieves a FRR of 20.0% and FAR of 0.0% for database containing 150 images.

REFERENCE

[1] A. K. Jain, S. Pankanti, S. Prabhakar, H. Lin, and A. Ross, “Biometrics: a grand challenge”, Proceedings of the 17th International Conference on Pattern Recognition (ICPR), vol. 2, pp. 935-942, 2004.

[2] Y. Kim, J. Yoo, and K. Choi, “A motion and similarity-based fake detection method for biometric face recognition systems,” IEEE Transactions on Consumer Electronics, vol.57, no.2, pp.756-762, May 2011.

[3] D. Mulyono and S. J. Horng, “A study of finger vein biometric for personal identification”, Proceedings of the International Symposium Biometrics and Security Technologies, pp. 134-141, 2008.

[4] Z. Liu, Y. Yin, H. Wang, S. Song, and Q. Li ,“Finger vein recognition with manifold learning”, Journal of Network and Computer Applications, vol.33, no.3, pp. 275-282, 2010.

[5] Rafael C. Gonzalez, Richard E. Woods, "Digital Image Processing", Second Edition.

[6] Kejun Wang, Hui Ma, "Finger Vein Recognition", Pattern Recognition & Intelligent Systems Department, College of Automation, Harbin Engineering University, China.

[7] Zhi Liu ,Shangling Song, " An Embedded Real-Time Finger-Vein Recognition System for Mobile Devices ", IEEE Transactions on Consumer Electronics, Vol. 58, No. 2, May 2012.

[8] D. Wang , J. Li, and G. Memik, “User identification based on fingervein patterns for consumer electronics devices”, IEEE Transactions on Consumer Electronics, vol. 56, no. 2, pp. 799-804, 2010.

[9] H. Lee, S. Lee, T. Kim, and Hyokyung Bahn, “Secure user identification for consumer electronics devices,” IEEE Transactions on Consumer Electronics, vol.54, no.4, pp.1798-1802, Nov. 2008.

[10] X. Sun, C. Lin, M. Li, H. Lin, and Q. Chen, “A DSP-based finger vein authentication system”, Proceedings of the Fourth International Conference on Intelligent Computation Technology and Automation, pp.333-336, 2011.

[11] John Canny,” A Computational Approach to Edge Detection”, IEEE Transactions on Pattern Analysis And Machine Intelligence, vol. PAMI-8, No. 6, November 1986.

[12] J. F. Canny, "Finding edges and lines in images," M.I.T. Artificial Intell. Lab., Cambridge, MA, Rep. Al-TR-720, 1983

★★★