The Use of Top-View Finger Image for Personal Identification

Panyayot Chaikan, Montri Karnjanadecha

Department of Computer Engineering, Faculty of Engineering, Prince of Songkhla University, Thailand
{panyayot, montri}@coe.psu.ac.th

Abstract

This paper describes a feasibility study for using a top-view finger image to increase the accuracy of fingerprint recognition without adding any new user operations. A CCD camera captures a top-view finger image while the user is touching a fingerprint sensor, and the acquired gray scale image is preprocessed to enhance the edges, the skin furrows, and the nail shape before the image is filtered by a bank of Oriented-Filters. A square tessellation is applied to the filtered image to create a feature map, called a NailCode. The NailCode is employed in the matching process by employing a Euclidean distance computation. The experiment reveals that personal identification accuracy using NailCode feature is 96.57%. It is recommended that NailCode is employed in conjunction with fingerprint for multimodal biometric systems will increase the identification accuracy.

1. Introduction

Personal identification using fingerprint is ubiquitous because fingerprints of each person are unique and time invariant [2]. However, fingerprint recognition remains a complex, challenging problem, with the accuracy of fingerprint recognition having reached a limit which is difficult to improve. One approach is multimodal biometrics which combines more than one human feature for recognition purposes. For example, Hong [3] employs the face in conjunction with fingerprints, Jain [4] uses speech, face and fingerprints, and Marcialis [5] utilizes two different types of fingerprint sensor. All of these methods have the same purpose: augmenting recognition accuracy which is otherwise limited by using only fingerprint detection. Their main drawback is that the additional features increase the complexity of the user interaction.

Our approach nests on the idea that the skin wrinkles and furrows on the top of a person’s fingers are different, and that this information could be easily captured with a small camera above the fingerprint sensor, as shown in Figure 1. The additional image information should increase the accuracy of the system without adding any extra tasks to the user. The question is “How well a top-view fingers image can identify a person?”. The contribution of this paper centers on a feasibility study of using a top-view finger image only for personal identification. We will combine it with fingerprint detection in a multimodal biometric system in the future.

The rest of the paper is organized as follows: section 2 starts with a brief overview of personal identification using a top-view finger image, followed by the detail explanation of each step. Section 3 gives the detail of feature extraction. Section 4 presents the matching process. In section 5, the experimental results are given, and section 6 concludes the paper.

Figure 1. Top-view Finger Image Identification.

2. Personal Identification using a Top-View Finger Image

Our NailCode generation and matching process can be summarized in the following steps:
1) The acquired grey-scale top-view finger image, called \( F_G \), of size 326*480 pixels is smoothed using a Gaussian filter.
2) Adaptive thresholding [6] is performed on the resulting image from step 1 to get a binary image, \( F_t \). The block size of the adaptive threshold is set to 25 pixels.
3) The color of image $F_T$ is inverted using the following condition:

$$F_T(x, y) = \begin{cases} 0 & \text{if } F_T(x, y) = 255 \\ 255 & \text{otherwise.} \end{cases}$$

(1)

4) Small particles made up of white pixels less than the threshold value are deleted. The resulting image, $F_D$, is shown in Figure 2(c).

5) The background of the finger image is deleted using the algorithm described in section 2.2.

6) The image is rotated to ensure that it is aligned vertically with the x-axis of the image coordinate. This is done using the $\phi$ value derived using the algorithm described in section 2.3.

7) The rotated image is skeletonized [7], features extracted using the algorithm described in section 3.

8) The derived features are matched against a database to find the most likely matching finger using the algorithm described in section 4.

2.1 Finger Alignment Parameter

When a finger is pressed on the fingerprint sensor, it may be up to $\pm 45^\circ$ away from the assumed vertical orientation. The inclination is detected, and the image is rotated. The algorithm works as follows:

1) The Canny algorithm [8] is applied to the $F_D$ image, resulting in $F_E$.

2) $D_L$ and $D_R$ are the bottom left and right edge images (326*480 pixels) of the finger. They are obtained by copying some part of $F_E$ using the following conditions:

$$D_L(x, y) = \begin{cases} F_E(x, y) & \text{if } 0 \leq x < 180, \\ 0 & \text{otherwise} \end{cases}$$

(2)

$$D_R(x, y) = \begin{cases} F_E(x, y) & \text{if } 146 \leq x < 326, \\ 0 & \text{otherwise} \end{cases}$$

(3)

3) The parameter of the bottom-part left edge of the finger are obtained by letting $C$ be the set of contours in $D_L$, where $C = \{C_1, C_2, C_3, ..., C_k\}$ and $k = \text{number of contours}$. A line-fitting algorithm [9] is applied to each contour $C_i$ to find the straight-line parameter $S_i$ of that contour. For each $S_i$ we get:

$$S_i = \left( V_i^l, V_i^r, X_i^l, Y_i^r \right), \quad i = 1..k$$

where $\left( V_i^l, V_i^r \right)$ is a normalized vector collinear to the line and $\left( X_i^l, Y_i^r \right)$ is some point on the line.

4) The parameter $S_{left}$ of the left-edge finger is selected from the set of $S_i$ using the following condition:

$$S_{left} = S_j \text{ where } \begin{cases} \text{abs} \left( \tan^{-1} \left( \frac{V_j^r}{V_j^l} \right) \right) \leq \frac{\pi}{4} \\ N_j > N_i \text{ for all } j \neq i, \quad i = 1..k \end{cases}$$

(5)

where $N_i$ is the number of white pixels in each contour $C_i$.

5) The parameter $S_{right}$ of the right-edge finger can be derived by applying steps 3-4 to $D_R$. The following left and right edge finger parameters can be used for finger alignment correction.

$$S_{left} = \left( V_{sl}, V_{sr}, X_{wl}, Y_{wr} \right)$$

$$S_{right} = \left( V_{sr}, V_{sr}, X_{wr}, Y_{wr} \right)$$

(6) (7)

2.2 Background Subtraction

The image from the CCD camera includes the fingerprint sensor covered by the finger. The background must be removed from the image to ensure that only the finger image is processed. Background deletion is done as follows:

1) Image $M_L$, which has the same size as $F_G$, is created using the following condition:

$$M_L(x, y) = \begin{cases} 255 & \text{if } \left( m_i < 0 \text{ and } y \geq m_i x + c_i \right) \quad \text{or } \left( m_i > 0 \text{ and } y \leq m_i x + c_i \right) \\ 0 & \text{otherwise} \end{cases}$$

(8)

where $m_i = \frac{V_{sd}}{V_{sl}}$ and $c_i = Y_{wl} - \frac{V_{sd} X_{wl}}{V_{sl}}$.

2) Image $M_R$, which has the same size as $M_L$, is created using the following condition:

$$M_R(x, y) = \begin{cases} 255 & \text{if } \left( m_i < 0 \text{ and } y \geq m_i x + c_i \right) \quad \text{or } \left( m_i > 0 \text{ and } y \geq m_i x + c_i \right) \\ 0 & \text{otherwise} \end{cases}$$

(9)

where $m_i = \frac{V_{sr}}{V_{sr}}$ and $c_i = Y_{wr} - \frac{V_{sr} X_{wr}}{V_{sr}}$.

3) $F_F$ is the background-deleted image. It is created from the operation:

$$F_F = M_R \cap M_L \cap F_D.$$
2.3 Finger Alignment Correction

\( \phi \) is the angle to rotate the finger image around the origin:

\[
\phi = \begin{cases} 
0.5(\mu + \lambda) & \text{if } \mu \lambda < 0 \\
0.5(\mu + \lambda - \pi) & \text{else if } \mu \geq 0 \text{ and } \lambda \geq 0 \\
0.5(\mu + \lambda + \pi) & \text{otherwise}
\end{cases}
\]

(11)

\( \mu \) and \( \lambda \) are the angle of inclination of the left and right edges of the finger, defined as:

\[
\mu = \tan^{-1}\left(\frac{V_{\mu}}{V_{\sigma}}\right) \quad \lambda = \tan^{-1}\left(\frac{V_{\sigma}}{V_{\nu}}\right)
\]

(12)

3. Feature Extraction

To extract features from the finger image, the skeletonized image is filtered using a bank of Oriented Filters, which use different \( \theta \) values (0°, 22.5°, 45°, 67.5°, 90°, 112.5°, 135° and 157.5°) with respect to the x-axis. Figure 3 shows the kernel Oriented Filters for different \( \theta \) values.

1) An image reference point is located at the middle of the nail-base is defined, as shown in Figure 4(a). To maximize the finger-image identification, the reference point was manually defined in this paper.
2) \( Q_\theta \) is the image filtered at \( \theta \). \( Q_\theta \) is tessellated into \( H*V \) square cells of size \( W*W \) using the reference point \( R(x_r,y_r) \), as shown in Figure 4(b).
3) The variance of the pixel intensity of each tessellated cell is computed using:

\[
\sigma^2_{W^2} = \frac{1}{W^2} \sum_{x=0}^{W-1} \sum_{y=0}^{W-1} p(x,y)^2 - \left( \frac{1}{W^2} \sum_{x=0}^{W-1} \sum_{y=0}^{W-1} p(x,y) \right)^2
\]

(13)

where

\[
\sigma^2_{W^2} = \text{The variance of the pixel intensity for each tessellated cell. } h \text{ and } v \text{ define the column and row number of that cell, } h = 0..9, v = -2,-1,0,1,2; \\
p(x,y) = \text{The pixel intensity of image } Q_\theta \text{ at location } (x,y); \\
H = \text{The number of cells in each column;} \\
V = \text{The number of cells in each row;} \\
a = y_r + Wv \\
b = y_r + W(v + 1) \\
c = x_r + (h - 0.5H) \\
d = x_r + (h - 0.5H + 1)
\]

The appropriate value of \( W, H \) and \( V \) were determined empirically to be 15, 10 and 15 respectively.

The Oriented filters enhance the ridge lines along the \( \theta \) angles while blurring the lines that lie in other directions. Feature extraction is carried out as follows:
4) After applying step 3 to every filtered image, we get:

\[ \text{NailCode} = \{ V_0, V_{22.5}, V_{45}, V_{67.5}, V_{90}, V_{112.5}, V_{135}, V_{157.5} \} \quad (14) \]

where

\[
V_\theta = \begin{bmatrix}
\sigma^2_{(2,0)}, \sigma^2_{(2,1)}, \ldots, \sigma^2_{(2, H-1)} \\
\sigma^2_{(1,0)}, \sigma^2_{(1,1)}, \ldots, \sigma^2_{(1, H-1)} \\
\sigma^2_{(0,0)}, \sigma^2_{(0,1)}, \ldots, \sigma^2_{(0, H-1)} \\
\sigma^2_{(-1,0)}, \sigma^2_{(-1,1)}, \ldots, \sigma^2_{(-1, H-1)} \\
\vdots \\
\sigma^2_{(P-3,0)}, \sigma^2_{(P-3,1)}, \ldots, \sigma^2_{(P-3, H-1)}
\end{bmatrix}
\]

4. The Matching Process

The Euclidean distance is computed in order to match the extracted features from the input image with those kept in our enrolled database. \( E_0 \) is the Euclidean distance between the input image and the \( n \)th finger image stored in the database, while the database consists of \( k \) enrolled fingers. By calculating the Euclidean distance between the input finger and each enrolled finger image, we get:

\[ E = \{ E_1, E_2, E_3, \ldots, E_k \}. \quad (15) \]

The input finger is said to match with the \( i \)th database finger if and only if:

\[ E_i < E_j \quad \text{for all } i \neq j, 1 \leq i \leq k, 1 \leq j \leq k. \quad (16) \]

5. Experimental Results

The testing system consists of a Creative VF0080 CCD camera equipped with a Digital Persona UareU4000B fingerprint sensor in a light controlled environment. The system code was written in Visual C++ using OpenCV to capture the finger image whenever the fingerprint sensor was pressed. Our finger image database consists of 800 finger images from 100 different fingers with 8 images per individual. One image of each individual was employed to enroll the system, while the other 7 images were used to test the system. The reference point of each image was set manually. When assessing the accuracy of the system, we found that 676 finger images were correctly recognized while 24 finger images were misinterpreted. In other words, the accuracy of the system is 96.57%.

6. Conclusions

This paper describes a feasibility study for a new technique which increases the accuracy of fingerprint identification without requiring the user to do additional tasks. Our results show that a top-view finger image will enhance a fingerprint recognition system. One drawback is that finger image is not time invariant, but we can use runtime biometric updating [10] to overcome this problem. The reference point detection algorithm is described in another paper [11].

Acknowledgements

The authors would like to thank Dr. Andrew Davison for the English correction of the first draft of this paper.

References