A Reference Point Detection Algorithm for Top-View Finger Image Recognition

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 an algorithm for automatic reference point detection in a top-view finger image recognition system. In tests of 700 finger images, only 6 images were rejected by our algorithm. A reference point location error correction technique was developed to improve the recognition accuracy. When using the proposed algorithm, the accuracy of the top-view finger image identification system was only reduced to 93.80% compared to 96.57% when using a manually defined reference point. This shows the feasibility of using top-view finger images to increase the recognition accuracy of fingerprint identification.

1. Introduction

Fingerprint recognition is one of the most prominent biometric identification methods, partly due to its cost benefits when compared to other biometric systems (e.g. iris, retina, DNA). There have been many attempts to combine other features with fingerprint system to increase the identification accuracy. [1] These multimodal biometrics, include Jain [2] which uses speech and face features in conjunction with fingerprints, Hong [3] which employs both fingerprints and face features, and Marcialis [4] which utilizes two fingerprint images from different types of sensor. The drawback of these systems is that they require the user to carry out additional tasks.

Our paper [5] shows that top-view finger image is very possible to be used for improving the accuracy of fingerprint identification, but the system nests on a good reference point location for extracting features. A manually determined reference point is an extra user task and so has the same problem as described before.

This paper proposes an autonomous reference-point detection methodology for the top-view finger image identification system, so no additional tasks need to be carried out by the user.

The rest of the paper is organized as follows: section 2 starts with a brief overview of top-view finger image recognition. Section 3 gives details on our proposed reference-point detection algorithm. Section 4 presents the reference point location error compensation. Experimental results appear in section 5, and section 6 concludes the paper.

Figure 1. The Multimodal biometric system using finger image and fingerprint

2. Personal Identification using a Top-View Finger Image

A gray scale top-view finger image captured from a CCD camera is smoothed, binarized, and inverted, as shown in Figure 2(a). The image is rotated so it is vertically oriented. The image is skeletonized and filtered with a bank of Oriented Filters, the resulting images were extracted for features using a reference point during NailCode [5] generation. The NailCode is employed in the matching process by employing a Euclidean distance computation.
2. Reference Point Detection Algorithm

The reference point of a top-view finger image is located at the midpoint of the finger’s nail-base, as shown in Figure 3. The steps for reference point detection can be summarized as follows:

1) Let $S_{\text{left}}$ and $S_{\text{right}}$ are the inclination parameters of the left and right edge of the finger:

\[
S_{\text{left}} = \left( V_{x_{l}}, V_{y_{l}}, X_{w_{l}}, Y_{w_{l}} \right)
\]

\[
S_{\text{right}} = \left( V_{x_{r}}, V_{y_{r}}, X_{w_{r}}, Y_{w_{r}} \right).
\]

($V_{x_{l}}, V_{y_{l}}$) is a normalized vector collinear to the line, and $(X_{w_{l}}, Y_{w_{l}})$ is some point on the line.

2) Image $M_{L}$, as shown in Figure 2(c), is the same size as the gray-scale top-view input image. It is created using the following condition:

\[
M_{L}(x, y) = \begin{cases} 
255 & \text{if } \left( m_{l} < 0 \text{ and } y \geq m_{l}x + c_{l} \right) \\
0 & \text{otherwise} 
\end{cases}
\]

where $m_{l} = \frac{V_{x_{l}}}{V_{y_{l}}}$ and $c_{l} = Y_{w_{l}} - \frac{V_{x_{l}}X_{w_{l}}}{V_{y_{l}}}$.

3) Image $M_{R}$, as shown in Figure 2(b), is the same size as $M_{L}$. It is created using the following condition:

\[
M_{R}(x, y) = \begin{cases} 
255 & \text{if } \left( m_{r} < 0 \text{ and } y \leq m_{r}x + c_{r} \right) \\
0 & \text{otherwise} 
\end{cases}
\]

where $m_{r} = \frac{V_{x_{r}}}{V_{y_{r}}}$ and $c_{r} = Y_{w_{r}} - \frac{V_{x_{r}}X_{w_{r}}}{V_{y_{r}}}$.

4) Image $F_{D}$, as shown in Figure 2(a), is the top-view finger image after being thresholded and inverted using the algorithm in [5]. It is employed to create image $F_{K}$, using the following condition:

\[
F_{K} = (M_{R} \cap M_{L} \cap F_{D}) \cup (M_{R} \cap M_{L}).
\]

5) The $F_{K}$, as shown in Figure 2(d), is dilated using a square-shaped structure element of size 3*3 [6].

6) The color of the dilated image is inverted using:

\[
F_{T}(x, y) = \begin{cases} 
0 & \text{if } F_{T}(x, y) = 255 \\
255 & \text{otherwise.} 
\end{cases}
\]

7) The image is rotated to ensure that it is exactly vertically oriented, resulting in $F_{L}$.

8) $L$ is the set of contours found in the image $F_{L}$:

\[
L = \{ L_{1}, L_{2}, L_{3}, ..., L_{f} \}
\]

where $f$ is the number of contours detected in the image. The reference point location can be derived by applying the algorithm described in Figure 4 to $F_{L}$, using the following parameters in each iteration:

- $N_{i}$ = The number of white pixels in each contour $L_{i}$
- $BR_{i}$ = The bounding rectangle [7] of each contour $L_{i}$. Each rectangle contains the parameters:

\[
\{ (x_{i}^{br}, y_{i}^{br}, w_{i}^{br}, h_{i}^{br}) \}
\]

where

$y_{i}^{br} =$ y coordinate of top most rectangle corner

$x_{i}^{br} =$ x coordinate of left most rectangle corner

$w_{i}^{br} =$ width of rectangle

$h_{i}^{br} =$ height of rectangle

- $ratio_{i} = \frac{N_{i}}{w_{i}^{br} \cdot h_{i}^{br}}$
9) \( L_j \) is the selected contour derived from the algorithm shown in Figure 4. The reference point can be computed on the chosen contour \( L_j \) by using the following equation:

\[
R(x, y) = (x_j^{BR} + 0.5w_j^{BR}, y_j^{BR} + h_j^{BR})
\]  

(5)

```plaintext
no_loop=0
found_ref_pt = false
do
    {  
        find contours from image \( F_L \)
        for each contour \( L_i \)
        if (\( N_i > \) threshold and \( \text{ratio}_i > 0.6 \))
            found_ref_pt = true
            add \( L_i \) in the list
        else
            erode image \( F_L \)
            no_loop++
        if (found_ref_pt=true)
            select contour \( L_j \) from the list with the largest \( \text{ratio} \)
    }  
while (no_loop<MAX_LOOPS and found_ref_pt=false)
if (no_loop<MAX_LOOPS)
    extract reference point from contour \( L_j \)
else
    report that reference point can not be found
```

Figure 4. Automatic reference point detection algorithm.

4. Reference Point Location Error Compensation

If the derived reference point location is incorrect, then the identification accuracy will be affected. Table 1 shows that an increasing reference point location error reduces identification accuracy. The table was obtained by trying to tessellate the filtered image with a translated version of the reference point moved in eight directions with a distance value of \( \delta \).

The effect of reference point location errors are reduced in our algorithm by translating the reference point into eight directions around its original point. Nine NailCodes are generated for nine reference points.

Table 1. Identification accuracy of personal identification with different reference-point location errors.

<table>
<thead>
<tr>
<th>Reference point error (in pixels)</th>
<th>Identification accuracy (percent)</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>96.57</td>
</tr>
<tr>
<td>5</td>
<td>89.10</td>
</tr>
<tr>
<td>8</td>
<td>78.00</td>
</tr>
<tr>
<td>10</td>
<td>65.90</td>
</tr>
<tr>
<td>12</td>
<td>51.90</td>
</tr>
<tr>
<td>15</td>
<td>32.00</td>
</tr>
</tbody>
</table>

5. Modified NailCode Matching

The effects of reference point location errors are handled by a modified version of our NailCode matching process [5]:

1) Let \( R_0(x,y) \) is the reference point obtained by the algorithm described in section 3. The 8 translated versions are \( R_1-R_8 \), each of which is a distance \( \delta \) away from \( R_0 \), as shown in figure 5. The NailCode values for nine reference points are

\[
\text{Extracted Features} = \{ N_{R_0}, N_{R_1}, N_{R_2}, \ldots, N_{R_8} \}
\]

\( N_{R_i} \) is the NailCode for the reference point \( R_i \).

2) The Euclidean distance between the NailCode \( N_{R_i} \) and each finger in the database is represented by:

\[
E = \left\{ E_{R_0}^{R_1}, E_{R_0}^{R_2}, \ldots, E_{R_0}^{R_q} \right\}
\]

\[
E = \left\{ E_{R_1}^{R_1}, E_{R_1}^{R_2}, \ldots, E_{R_1}^{R_q} \right\}
\]

\[
E = \left\{ E_{R_2}^{R_1}, E_{R_2}^{R_2}, \ldots, E_{R_2}^{R_q} \right\}
\]

\[
E = \left\{ E_{R_7}^{R_1}, E_{R_7}^{R_2}, \ldots, E_{R_7}^{R_q} \right\}
\]

where

\( q \) = The number of fingers in the database

\( E_{R_i}^{R_j} \) = The Euclidean distance between the \( i^{th} \) finger in the database and the input finger tessellated using reference point \( R_j \).

3) For each \( E_{R_i}^{R_j} \), the two smallest Euclidean distances \( E_{R_i}^{R_j} \) and \( E_{R_i}^{R_j} \) are found such that:

\[
E_{R_i}^{R_j} < E_{R_j}^{R_i} < E_{R_p}^{R_i} \quad \forall(m \neq n, m \neq p \text{ and } n \neq p)
\]

\[
1 \leq m \leq q, \quad 1 \leq n \leq q, \quad 1 \leq p \leq q
\]

(6)
4) The distance between $E_{n}^{R_{i}}$ and $E_{m}^{R_{i}}$ is calculated for every $E^{R_{i}}$, resulting in $D_{n}^{R_{i}}$:

$$D_{n}^{R_{i}} = E_{n}^{R_{i}} - E_{m}^{R_{i}}$$ (7)

where $D_{n}^{R_{i}} \in \{D_{R_{0}}, D_{R_{1}}, D_{R_{2}}, \ldots, D_{R_{8}}\}$.

5) The input finger matches with the $v^{th}$ finger in the database finger if and only if:

$$D_{v}^{R_{i}} > D_{n}^{R_{i}} \text{ for all } i \neq j, 0 \leq i \leq 8, 0 \leq j \leq 8.$$ (8)

### 6. Experimental Results

A gray scale finger image was captured from a Creative VF0080 CCD camera whenever the user touched the fingerprint sensor. Our top-view finger image database consists of 800 finger image from 100 different fingers, with eight images per individual. One image from each individual was employed to enroll the system, while the other seven images were used to test. This means that there were 700 test images in the database. Only 6 test images were rejected by our reference point detection algorithm. When manual reference point were utilized, the identification accuracy was 96.57%, which dropped to 73.78% when only a single automatic reference point, $R_{0}$, was used. Table 2 shows that additional points $R_{1}$-$R_{8}$, improved the accuracy of the system dramatically, especially when $\delta$ was 9 or 10 pixels. In those cases, the identification accuracy was 93.80%.

<table>
<thead>
<tr>
<th>Reference point marking method</th>
<th>Reject</th>
<th>Accuracy</th>
</tr>
</thead>
<tbody>
<tr>
<td>Manual</td>
<td>0</td>
<td>96.57%</td>
</tr>
<tr>
<td>Automatic using R0 only</td>
<td>6</td>
<td>73.78%</td>
</tr>
<tr>
<td>Automatic using R0-R8 ($\delta=5$)</td>
<td>6</td>
<td>87.03%</td>
</tr>
<tr>
<td>Automatic using R0-R8 ($\delta=8$)</td>
<td>6</td>
<td>92.51%</td>
</tr>
<tr>
<td>Automatic using R0-R8 ($\delta=9$)</td>
<td>6</td>
<td>93.80%</td>
</tr>
<tr>
<td>Automatic using R0-R8 ($\delta=10$)</td>
<td>6</td>
<td>93.80%</td>
</tr>
</tbody>
</table>

### 7. Conclusions

This paper shows the feasibility of using finger images for increasing the recognition accuracy of fingerprint identification without the need for additional user tasks. In particular, the user does not need to manually find the finger’s reference point as required in [5]. We now plan to construct a personal identification system that uses finger images together with fingerprint data.

The average reference-point location error in our algorithm is about 10 pixels. This error is reduced by our compensation method, but processing time is increased by about 8 times compared to when only one reference point is utilized. This suggests that the algorithm is not suitable for an environment with limited CPU performance, such as embedded systems. In such an environment, reference point location accuracy must be improved to avoid the need for compensation.

### Acknowledgements

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

### References


