A Fast Palmprint Feature Extraction Method Based on Embedded Device

Jiaqiang Wang¹,²
¹School of Electronics and Information Engineering, Hebei University of Technology, Tianjin300401, China
²Big Data Research Center, Beijing Institute of New Technology Applications, Beijing100094, China

Ming Yu*
School of Electronics and Information Engineering, Hebei University of Technology, Tianjin300401, China
*Corresponding author(E-mail: yueeming@126.com)

Hanbing Qu
Big Data Research Center, Beijing Institute of New Technology Applications, Beijing100094, China

Abstract
This paper proposes an optimized competitive coding method for palmprint identification. A recursive implementation Gabor filter is used instead of the Gabor function based neurophysiology. The recursive implementation Gabor filter decomposes a 2D Gabor filter into two non-orthogonal 1D Gabor filters, x-direction and φ-direction, and then calculate the point and the three points before and after in each direction. Experimental results show that this method, compared with the competitive coding method, EER increased by only 0.0004%. But the speed increased significantly, image with size of 128 x 128 feature extraction time reduced from 330ms to 160ms in the ARM9 platform and reduced from 63ms to 30ms in the Cortex-A8 platform, so this method is fully applicable to embedded devices, especially for high real-time requirements.

Key words: Competitive Coding, Recursive Implementation, Gabor Filter, Embedded Device.

1. INTRODUCTION
Palmprint identification has been in the stage of vigorous development in recent decades. According to the research content classification, both the palmprint identification system (Zhang and Zuo, 2012), also preprocessing (Aykut and Ekinci, 2013), feature extraction (Luo and Zhao, 2016; Fei and Xu, 2016), matching (Leng and Teoh, 2015; Dai and Feng, 2012). According to the application environment, both based on data mining (Bao and Guo, 2016), but also on the application of embedded devices (Choraś and Kozik, 2012).

Palmprint feature extraction is a very important stage in palmprint identification system. The feature precision and form directly affect the performance of palmprint identification system. So far, the coding-based method has high identification accuracy and fast computation speed at the same time, so this paper mainly discusses the feature extraction method based on coding. Initially, the method based on coding is originated from iris recognition. Later, researchers optimized their application in palmprint identification. Kong (Kong and Zhang, 2004) using six directions Gabor filter to extraction palmprint features, the filter results are encoded and then identified.

Due to poor computing power, there is a high demand for the complexity of algorithms running on embedded devices, especially in the high real-time application environment. This paper studies the method of improving the speed of feature extraction while keeping the accuracy of competitive coding method.

2. COMPETITIVE CODE
Kong (Kong and Zhang, 2004) proposed a palmprint feature extraction method called Competitive Code, which uses a Gabor function based Neurophysiology to filter palmprint ROI region image at six directions (0, π / 6, π / 3, π / 2, 2π / 3, 5π / 6). The Gabor filter expression as the following form:

$$
\psi(x, y, \omega, \theta) = \frac{\omega}{\sqrt{2\pi\kappa}} e^{i\omega \theta} e^{-\frac{(x^2 + y^2)}{2\kappa}} \left( e^{i\omega \theta} - e^{\frac{\omega^2}{2}} \right)
$$

(1)
where $\omega$ is the radial frequency in radians per unit length, $\theta$ is the orientation of the Gabor functions in radians, $\kappa$ is defined by $\kappa = \sqrt{2 \ln 2 \left( \frac{2^\gamma + 1}{2^\gamma - 1} \right)}$, $\delta$ is the half-amplitude bandwidth of the frequency response. 

$x' = (x - x_0) \cos \theta + (y - y_0) \sin \theta$, $y' = -(x - x_0) \sin \theta + (y - y_0) \cos \theta$ is the center of the Gabor function.

The orientation information of palmprint lines is extracted by the real part of the filter function, and then the values of $0 \sim 5$ are used respectively. Then the Winner-take-all criterion is used to record the orientation value when the filter response amplitude is the smallest. Filter result is given by Figure 1.

![Figure 1 Competitive Code](attachment:image1.png)

(a) Original image (b) Filter response image

In Figure 1, $\sigma_x = 7.5$, $\sigma_y = 3$, (a) is the preprocessed image, (b) is filter response image for the minimum amplitude of the six directions.

The orientation of the filter using the coding rules shown in Table 1 as a three-layer feature, which are shown in Figure 2.

### Table 1 Bit representation of the Competitive Code

<table>
<thead>
<tr>
<th>Original values</th>
<th>Bit 1</th>
<th>Bit 2</th>
<th>Bit 3</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>1</td>
<td>0</td>
<td>0</td>
<td>1</td>
</tr>
<tr>
<td>2</td>
<td>0</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>3</td>
<td>1</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>4</td>
<td>1</td>
<td>1</td>
<td>0</td>
</tr>
<tr>
<td>5</td>
<td>1</td>
<td>0</td>
<td>0</td>
</tr>
</tbody>
</table>

![Figure 2 Bit image of palmprint](attachment:image2.png)

(a) Bit1 (b) Bit2 (c) Bit3

Gabor filter has been widely used in image processing, pattern recognition and computer vision, which is easy to obtain image edge information convolution of image. However, the efficiency of convolution on the embedded system is very low. Such as on the Samsung ARM9 (2440) development board, we convolution the convolution with Gabor filter in Kong mentioned, in the use of fixed-point operations and the case of reference, it still takes about 330ms or so, and is not suitable for real-time requirements. Therefore, the Gabor filter realized by the traditional convolution method can meet the needs of biometrics in precision, but the speed still needs to be improved.
3. RECURSIVE GABOR FILTER

The general case of an oriented anisotropic Gaussian filter in two dimensions is given by.

\[ g(x, y; \sigma_x, \sigma_y, \theta) = \frac{1}{2\pi\sigma_x\sigma_y} \exp \left( -\frac{1}{2} \left( \frac{(x\cos\theta + y\sin\theta)^2}{\sigma_x^2} + \frac{(-x\sin\theta + y\cos\theta)^2}{\sigma_y^2} \right) \right) \] (2)

where \( \theta \) is the direction, \( (u, v) = \begin{bmatrix} \cos\theta & \sin\theta \\ -\sin\theta & \cos\theta \end{bmatrix} \) \( x \), \( (x, y) \) is the original coordinate system coordinate, \( (u, v) \) is the coordinates after rotation \( \theta \) radians.

Geusebroek (Geusebroek and Smeulders, 2003) decompose the 2D Gaussian filter shown in Equation 2 into two non-orthogonal 1D Gassian filters in the time domain, first decompose a 1D Gaussian filter along the x-direction, and then decompose another nonorthogonal direction \( \phi \). The recursive implementation of 1D Gaussian filter is divided into two steps: forward recursive and backward recursive. It needs to be combined with three pixels before and after the direction of \( x \) or \( \phi \). Forward and backward recursive along the direction \( x \) are respectively.

\[ g_{1}^{f}(x, y) = a_{0}f(x, y) - a_{1}g_{2}^{f}(x-1, y) - a_{2}g_{3}^{f}(x-2, y) - a_{3}g_{1}^{f}(x-3, y) \] (3)

\[ g_{1}^{b}(x, y) = a_{0}g_{2}^{b}(x, y) - a_{1}g_{3}^{b}(x+1, y) - a_{2}g_{4}^{b}(x+2, y) - a_{3}g_{1}^{b}(x+3, y) \] (4)

\( a_{0}, a_{1}, a_{2}, a_{3} \) are the recursive coefficients.

Gabor filter can be understood as a Gaussian filter with a Gaussian kernel, Young (Young and Van Vliet, 2002) transformed the 1D Gaussian filter into the \( Z \) domain.

\[ G(z, \omega) = \frac{1}{b_{0} + b_{1}e^{j\omega}z^{-1} + b_{2}e^{2j\omega}z^{-2} + b_{3}e^{3j\omega}z^{-3}} \] (5)

where \( q = \begin{cases} -0.2568 + 0.5784\sigma_{x}^{2}, & \sigma_{x} < 3.556 \\ 2.5091 + 0.9804(\sigma_{x}^{2} - 3.556), & \sigma_{x} \geq 3.556 \end{cases} \), \( m_{0} = 1.1668, m_{1} = 1.0783, m_{2} = 1.40586, b_{0} = 1, scale = (m_{0} + q)(m_{1}^{2} + m_{2}^{2} + 2m_{1}q + q^{2}) \), \( b_{1} = -q(2m_{0}m_{1} + m_{2}^{2} + 2m_{0}m_{1}q + 3q^{2})/scale \), \( b_{2} = q^{2}(m_{0} + 2m_{1} + 3q)/scale \), \( b_{3} = -q^{3}/scale \).

Triggs (Triggs and Sdika, 2006) present that the use of the forward recursive coefficient in the backward recursive will produce significant amplitude and phase (geometric position) distortion for all points within about 3.5 standard deviations of the boundary.

\[ M = \begin{bmatrix} M_{11} & M_{12} & M_{13} \\ M_{21} & M_{22} & M_{23} \\ M_{31} & M_{32} & M_{33} \end{bmatrix} \] (6)

Where, \( scale \ b = 1/((1-b_{1} + b_{2} - b_{3})* (1+b_{1} + b_{2} + b_{3})* (1-b_{1} - (b_{2} - b_{3})*b_{1})) \),

\[ M_{11} = scale \ b^{*}(-b_{3}b_{1} + 1 - b_{2}^{2} + b_{2}) \], \[ M_{12} = scale \ b^{*}((b_{1} + b_{2})(-b_{2} + b_{3}b_{1})) \],

\[ M_{13} = scale \ b^{*}(b_{2}(-b_{1} + b_{2}b_{3})) \], \[ M_{21} = scale \ b^{*}(-b_{3}b_{2} + 1 - b_{2}^{2} - b_{2}) \], \[ M_{22} = scale \ b^{*}((b_{1} + 1)(-b_{2} + b_{3}b_{1})) \], \[ M_{23} = scale \ b^{*}((b_{1} + b_{2})b_{3} - b_{2}^{2} - b_{2}) \], \[ M_{31} = scale \ b^{*}(b_{1}b_{2} - b_{2}^{2} + b_{2}b_{3}^{2} - b_{2}^{2} - b_{2}b_{3}) \], \[ M_{32} = scale \ b^{*}(b_{2}(-b_{1} + b_{2}b_{3})) \], \[ M_{33} = scale \ b^{*}((b_{3} - (b_{1} + b_{2}b_{3})) \).

Using the traditional 2D elliptical Gabor filter with the recursive implementation replace the Gabor function based Neurophysiology in competitive code. The results are shown in Figure 3.
4. CONCLUSIONS

Improvements to algorithms are generally focused on improving the both performance: running time and accuracy. Greatly reducing the running time while accuracy loss is within the tolerable range, is a variety of optimization algorithms unremitting pursuit, our experiment will also be this two aspects, accuracy and running time.

(1) Accuracy experiment

We uses the Hong Kong Polytechnic University Palms Open Library (PolyU Palmprint Palmprint Database, 2009), which contains 193 people, 386 hands images. We select 10 palm images each person. After the preprocessing is complete, the subimage is $128 \times 128$ for the palm ROI region. extraction their features respectively, match the interclass False Acceptance Rate(FAR), Figure 4(a) is the comparison, where x-axis is threshold, y-axis is FAR. The red curve is the result of our algorithm, and the blue curve is the competition code. It can be seen from the image that the two methods have the slight difference except for very few thresholds, The maximum difference between the two errors is 0.01%. In most cases, the accuracy of the two algorithms is not obvious.

(2) Running time experiment

This experiment was tested on Samsung ARM9 (2440) and Cortex-A8, respectively. The test results are shown in Table 2.

<table>
<thead>
<tr>
<th>Name</th>
<th>ARM9(ms)</th>
<th>Cortex-A8(ms)</th>
</tr>
</thead>
<tbody>
<tr>
<td>This method</td>
<td>160</td>
<td>30</td>
</tr>
<tr>
<td>Kong</td>
<td>330</td>
<td>63</td>
</tr>
</tbody>
</table>

Figure 4(b) is the comparison of False Rejection Rate(FRR). The FRR of this method is higher than competitive code when the threshold is small. We think the reason is that recursive error caused in $\phi$-direction, and the smoothness is not as good as the traditional convolution Gabor filter.

The Equal Error Rate(EER) of our method is 0.038% and the competitive code is 0.0376%.

Table 2 Running time comparision
The speed of our method increased significantly, image with size of 128×128 feature extraction time reduced from 330ms to 160ms in the ARM9 platform and reduced from 63ms to 30ms in the Cortex-A8 platform. So this method is fully applicable to embedded devices, especially for high real-time requirements.

ACKNOWLEDGEMENTS

This work was supported by Youth Core Plan of Beijing Academy of Science and Technology (No. 2014-30) and Supported by Innovation Group Plan of Beijing Academy of Science and Technology (No.IG201506C2)

REFERENCES


Zhang, D., Zuo, W., Yue, F. (2012)“A comparative study of palmprint recognition algorithms” ACM Computing Surveys (CSUR), 44(1), Article No. 2.