Optik 124 (2013) 3340–3344

Contents lists available at SciVerse ScienceDirect

Optik

journal homepage: www.elsevier.de/ijleo

# Palmprint recognition method based on score level fusion

Shuwen Zhang, Xuxin Gu [1]

*Bio-Computing Research Center, Shenzhen Graduate School, Harbin Institute of Technology, Shenzhen 518055, China*

ARTICLE INFO

*Article history:*
Received 18 May 2012
Accepted 17 October 2012

*Keywords:*
Gabor filter
Palmprint recognition
TPTSR
Competitive coding method
Sparse representation

ABSTRACT

Different palmprint recognition methods have different advantages. The texture- and feature-based palmprint recognition methods can well exploit the minutiae of the palmprint but are not very robust to the possible variation such as the rotation and shift of the palm. The representation-based palmprint recognition method can well take advantage of the holistic information but seems not to be able to fully exploit the minutiae of the palmprint. In this paper, we propose to fuse the competitive coding method and two-phase test sample sparse representation (TPTSR) method for palmprint recognition. As one of representation-based methods, TPTSR method takes the whole palmprint image as the input and determines the contribution of the training samples of each class in representing the test sample. TPTSR also uses the contribution to calculate the similarities between the test sample and every class. The competitive coding method is a feature-based method and is highly complementary with TPTSR. We use a weighted fusion scheme to combine the matching scores generated from TPTSR and the competitive coding method. The experimental results show that the proposed method can obtain a very high classification accuracy and outperforms both TPTSR and the competitive coding method.



## 1. Introduction

Biometrics have attracted much attention [1–8]. As one of the important biometric recognition technologies, palmprint recognition has important practical significance. Researchers have offered a broad and in-depth study of palmprint recognition and have proposed various palmprint recognition methods, such as the palmprint recognition methods based on point features and line features [1,2], texture features [3–5], subspace analysis [6–9] and hierarchical features integration [10–12].

The palmprint recognition method based on the points and lines features is the most direct palmprint recognition method and can be grouped as a feature-based palmprint recognition method. Feature-based palmprint recognition methods have been widely applied in early years. Funada J et al. [1] proposed a method to eliminate the wrinkles of the palmprint to extract the papillary ridge. However, this method only extracted the ridge line of the palmprint image and did not use the obtained line to perform palmprint recognition. Jun Chen et al. [2] estimated the wrinkled points of the palmprint by exploiting local gray-scale orientation images. They also connected the dots to form a straight line and used the straight line to match the palmprint. It seems that point features are accurate description of the palmprint images. However, point features can be only extracted from the high-resolution images. Moreover, if the number of the points is large, then the matching has a high computational cost. The line features seem to be more stable than the point features, and the feature space is smaller. A common drawback of the point features and line features is that they cannot represent the depth and intensity of the palmprint, and they are easy to be influenced by the noise.

Texture features based palmprint recognition methods not only can effectively avoid the influence of the noise, but also can simplify the pre-processing steps. When we perform palmprint recognition using the texture energy, we can also exploit the spatial location and the thickness of the palmprint. This method can better keep the distinction between classes and the compactness in classes. It should be pointed out that the competitive coding method [13–16] is one of the most widely used texture-based palmprint recognition method.

Subspace-based palmprint recognition method has a strong description ability, small computational cost and good separability, and is easy to implement. Subspace-based method can convert the sample space into the feature subspace by using a mapping transformation or matrix operations. The eigenpalms [17] and fisherpalms [18] are two typical subspace-based palmprint recognition methods.

Recently, representation-based methods have attracted extensive interests in the field of pattern recognition. For example, Y. Xu et al. [19] proposed the two-phase test sample representation (TPTSR) method. TPTSR consists of two steps. The first step represents the test sample as a linear combination of all the training

*E-mail addresses:* shuwen8219@sina.com (S. Zhang), guxuxin@gmail.com (X. Gu).
[1] Tel.: +86 15815549157.

0030-4026/$ – see front matter © 2012 Elsevier GmbH. All rights reserved.
http://dx.doi.org/10.1016/j.ijleo.2012.10.048

samples and identifies $M$ nearest neighbors of the test sample. The second step represents the test sample as a linear combination of the determined $M$ nearest neighbors and uses the representation result to perform classification. TPTSR is able to reduce the side-effect, on the classification of the test sample, of the training samples that are very dissimilar with the test sample [19]. It has been shown that TPTSR can perform very well in recognizing the palm [20]. It is noticeable that other representation-based methods have also achieved good performance in palmprint recognition and face recognition [21–24].

We note that the texture- and feature-based palmprint recognition methods can well exploit the minutiae of the palmprint but cannot well cope with the possible variation such as the rotation and shift of the palm. The representation-based palmprint recognition method can well take advantage of the holistic information but seems not to be able to fully exploit the minutiae of the palmprint. The competitive coding method is a feature-based method and is highly complementary with TPTSR. As a result, we propose to integrate TPTSR and the competitive coding method for palmprint recognition using a weighted fusion scheme. This fusion scheme exploits the weighted sum of the scores of these two methods to classify the palmprint. The proposed method has the following rationale: since these two methods work in two very different ways, the scores of TPTSR are complementary to those of the competitive coding method. The fusion scheme can exploit the minutiae of the palmprint and are very robust to the possible variation such as the rotation and shift of the palm.

The rest of the paper is organized as follows. Section 2 describes the competitive coding method. Section 3 discusses the TPTSR method. Section 4 gives the fusion scheme. Section 5 presents the experimental results and Section 6 concludes the paper.

## 2. Competitive coding method

Among many texture-based palmprint recognition methods such as Gabor filtering [25–28], Wavelet transform [29–31], Fourier transform [32–34], Local energy method, the competitive coding method [13–16] has received the most attention.

The competitive coding method is factually a multiple Gabor filters based method. It first uses six two-dimensional Gabor filters with six different orientations to extract the features of the palmprint images and then takes the most "competent" feature as the final feature. After coding the final feature, the competitive coding method uses the Hamming distance to measure the similarity between palms. In particular, this method calculates the Hamming distances between the test sample and each training sample and assigns the test sample to the class of the training sample that has the minimum Hamming distance.

The following Gabor filter is used for extracting the directions [28]:

$$\psi(x, y, x_0, y_0, \omega, \theta, k) = \frac{\omega}{\sqrt{2\pi}k} e^{-\omega^2/8k^2(4x'^2+y'^2)}(e^{i\omega x'} - e^{-k^2/2}) \quad (1)$$

where $x' = x - x_0\cos\theta + (y - y'_0)\sin\theta$, $y' = -(x - x_0)\sin\theta + (y - y'_0)\cos\theta$. $(x_0, y_0)$ is the center coordinate of the filter window. $\omega$ and $\theta$ are the radial frequency and orientation of the Gabor functions in radians respectively. $\theta$ can be assigned with $0, \pi/6, 2\pi/6, 3\pi/6, 4\pi/6, 5\pi/6$, and $k$ is a coefficient defined by $k = \sqrt{2\ln 2}\left(\frac{2^\delta+1}{2^\delta+1}\right)$, where $\delta$ is the half-amplitude bandwidth of the frequency response, $\omega = k/\sigma$. When $\delta$, $\sigma$ are fixed, then we can get $\omega$.

For directional features, after coding them to bit planes, it's very convenient to combine these bit planes for matching. The six types of orientation features can be coded to 3-bits planes as illustrated in Table 1 [15].

Tables 2–5.

**Table 1**
Directional feature representation in competitive coding method.

| Direction number | Bit 1 | Bit 2 | Bit 3 |
|---|---|---|---|
| 0 | 0 | 0 | 0 |
| 1 | 0 | 0 | 1 |
| 2 | 0 | 1 | 1 |
| 3 | 1 | 1 | 1 |
| 4 | 1 | 1 | 0 |
| 5 | 1 | 0 | 0 |

**Table 2**
The correlation coefficient of global matching score and directional matching score under different spectrum.

| G spectrum | R spectrum | B spectrum | I spectrum |
|---|---|---|---|
| 0.2445 | 0.1995 | 0.2314 | 0.1959 |

**Table 3**
The correlation coefficient of direct distance matching score and directional matching score under different spectrum.

| G spectrum | R spectrum | B spectrum | I spectrum |
|---|---|---|---|
| −0.0614 | −0.0321 | −0.0614 | 0.0355 |

**Table 4**
The correlation coefficient of LDA matching score and directional matching score under different spectrum.

| G spectrum | R spectrum | B spectrum | I spectrum |
|---|---|---|---|
| 0.0198 | −0.083 | −0.0671 | −0.0045 |

**Table 5**
The best range of M under different spectrum.

| G spectrum | R spectrum | B spectrum | I spectrum |
|---|---|---|---|
| 300–600 | 300–700 | 300–700 | 500–1000 |

For directional features, we use integers $0\sim5$ to code the six directions $0, \pi/6, 2\pi/6, 3\pi/6, 4\pi/6, 5\pi/6$, respectively. Intuitively, the Hamming distance between parallel directions should be 0, while the distance between perpendicular directions should be 3. In other cases, the distance should be 1 or 2. Let $\alpha$ and $\beta$ be the direction number. We define the Hamming distance as follows [14]:

$$F(\alpha, \beta) = \min(\left|\alpha - \beta\right|, 6 - \left|\alpha - \beta\right|) \quad \alpha, \beta \in \{0, 1, 2, 3, 4, 5\} \quad (2)$$

Obviously, the value of $F(\alpha, \beta)$ can only be 0, 1, 2 or 3 as described above.

Let $D_d$ and $D_t$ be the direction sets of test samples and the training samples, respectively. The directional matching score between them can be defined as:

$$R = \frac{1}{3nm}\sum_{i=1}^{n}\sum_{j=1}^{m} F(D_d(i, j), D_t(i, j)) \quad (3)$$

where $F$ is defined as formula (2). So we can find the directional matching scores between the test sample and every training sample. In this way, we can find the minimum score between the test sample and all the training samples. Then, we can classify the test sample into the $i$th class if the class label of the corresponding training sample is $i$.

## 3. TPTSR method

The TPTSR method [19] consists of two phases. The first phase is representing the test sample by all the training samples and finds the M nearest neighbors of the test sample. We assume that there