



## Palmprint verification using binary orientation co-occurrence vector

Zhenhua Guo<sup>a</sup>, David Zhang<sup>a,\*</sup>, Lei Zhang<sup>a</sup>, Wangmeng Zuo<sup>b</sup>

<sup>a</sup> Biometrics Research Centre, Department of Computing, The Hong Kong Polytechnic University, Hung Hom, Kowloon, Hong Kong, China

<sup>b</sup> Biocomputing Research Center, School of Computer Science and Technology, Harbin Institute of Technology, Harbin, China

### ARTICLE INFO

#### Article history:

Received 14 August 2008

Received in revised form 2 February 2009

Available online 23 May 2009

Communicated by T. Tan

#### Keywords:

Palmprint recognition

Biometrics

Orientation code

Gabor filter

### ABSTRACT

The development of accurate and robust palmprint verification algorithms is a critical issue in automatic palmprint authentication systems. Among various palmprint verification approaches, the orientation based coding methods, such as competitive code (CompCode), palmprint orientation code (POC) and robust line orientation code (RLOC), are state-of-the-art ones. They extract and code the locally dominant orientation as features and could match the input palmprint in real-time and with high accuracy. However, using only one dominant orientation to represent a local region may lose some valuable information because there are cross lines in the palmprint. In this paper, we propose a novel feature extraction algorithm, namely binary orientation co-occurrence vector (BOCV), to represent multiple orientations for a local region. The BOCV can better describe the local orientation features and it is more robust to image rotation. Our experimental results on the public palmprint database show that the proposed BOCV outperforms the CompCode, POC and RLOC by reducing the equal error rate (EER) significantly.

© 2009 Elsevier B.V. All rights reserved.

### 1. Introduction

Automatic authentication using biometric characteristics is becoming more and more popular. Biometrics is the study of methods for uniquely recognizing humans based on one or more intrinsic physical or behavioral traits (Jain et al., 1999; Zhang, 2000). As an important member of the biometric characteristics, palmprint has merits such as robustness, user-friendliness, high accuracy, and cost-effectiveness.

A palmprint image has mainly three kinds of features: principal lines (usually three dominant lines on the palm), wrinkles (weaker and more irregular lines) and crease (the ridge and valley structures like in fingerprint) (Zhang et al., 2003). The features of principal lines and wrinkle could be captured under a relatively low resolution (Zhang et al., 2003), e.g. less than 100 dpi, while the feature of crease can only be acquired under a higher resolution, e.g. 500 dpi (NIST report, 2001). Fig. 1a shows a typical palmprint image captured at 75 dpi. After image acquisition, the palmprint image will be processed to extract the region of interest (ROI) for feature extraction and matching (Zhang et al., 2003). Fig. 1b shows the extracted ROI image. For more information about ROI extraction, please refer to (Zhang et al., 2003).

Since Shu and Zhang (1998) first proposed to use palmprint as a characteristic for automatic personal identification, many algorithms have been proposed for palmprint authentication. These algorithms can be classified into three main classes: detecting

interest points or lines (Duta et al., 2002; Han et al., 2003), subspace learning (Connie et al., 2005; Hu et al., 2007; Jing et al., 2007; Kumar and Zhang, 2005; Lu et al., 2003; Ribaric and Fratric, 2005; Wang et al., 2008; Wu et al., 2003; Yao et al., 2007), wavelet based feature extraction (Zhang and Zhang, 2004; Chen and Xie, 2007) and texture-based coding (Jia et al., 2008; Kong et al., 2006; Kong and Zhang, 2004; Sun et al., 2005; Wu et al., 2005; Zhang et al., 2003). Duta et al. (2002) investigated the feasibility of personal identification by using feature points. Han et al. (2003) proposed to use Sobel and morphological operator to extract palm line-like features. Taking palmprint image as a 2D matrix, some subspace learning based methods, such as principal component analysis (Lu et al., 2003; Kumar and Zhang, 2005; Ribaric and Fratric, 2005; Yao et al., 2007), Fisher linear discriminant (Wu et al., 2003), independent component analysis (Connie et al., 2005), locality preserving projection (Hu et al., 2007; Wang et al., 2008), and discriminative common vector (Jing et al., 2007) were explored. Texture-based coding is another popular research topic in palmprint authentication. Zhang et al. (2003) used the complex Gabor filtering to code image texture information by the phase information of the filter response. Kong et al. (2006) proposed to select discriminatory information using several complex Gabor filters. Sun et al. (2005) tried to code ordinal relationship from two orthogonal directions by using Gaussian filters. Recently, some methods (Jia et al., 2008; Kong and Zhang, 2004; Wu et al., 2005) have been proposed to code line orientation information by regarding palm lines as negative lines.

Among various schemes, the orientation based coding methods (Jia et al., 2008; Kong and Zhang, 2004; Wu et al., 2005) are

\* Corresponding author. Tel.: +852 2766 7271; fax: +852 2774 0842.

E-mail address: [csdzhang@comp.polyu.edu.hk](mailto:csdzhang@comp.polyu.edu.hk) (D. Zhang).

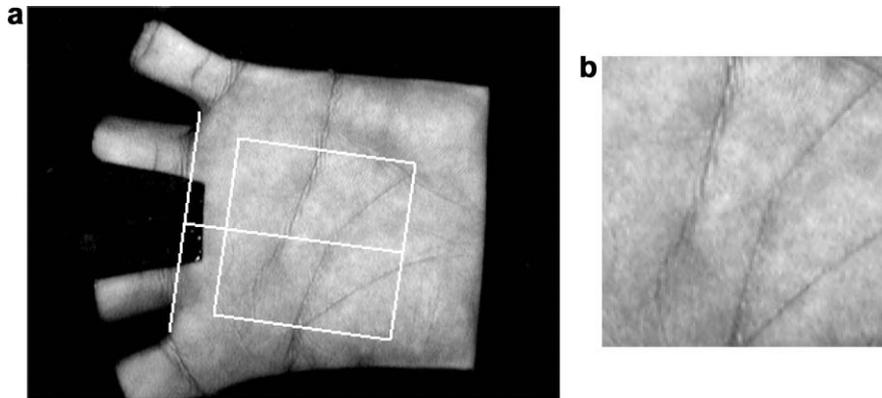


Fig. 1. (a) A sample palmprint image; and (b) its region of interest (ROI).

state-of-the-art ones and they have merits of high accuracy, robustness to illumination variation and fast implementation, etc. Since the orientation of palm lines is stable and can provide enough discriminatory information for personal identification, many palmprint coding schemes, including competitive code (CompCode) (Kong and Zhang, 2004), palmprint orientation code (POC) (Wu et al., 2005), robust line orientation code (RLOC) (Jia et al., 2008), were proposed. These algorithms use different filters or masks, such as Gabor filter (CompCode), self designed mask (POC), and modified finite Radon transform (RLOC), to estimate the orientation feature of each local region. A common rule, the “competition” rule, is shared by these algorithms: several filters or masks with different orientation were convolved with the image, and then the “dominant” orientation was determined with some criterion. By simply coding the orientation map of the palmprint, high accuracy palmprint identification could be implemented with high speed matching.

However, the line structures in palmprint image are very complex. Multiple lines may intersect at some regions, so some struc-

tural information may be lost if only one orientation is used to represent the local feature. Fig. 2a and b show an example area where two lines intersect. Fig. 2c plots the curve of Gabor filtering response (Kong and Zhang, 2004) vs. orientation for the local area in Fig. 2b. We can see two valleys, which imply two main orientations in this area. If only one orientation is kept, much valuable discriminatory information will be lost.

In addition, the extracted “dominant” orientation is sensitive to rotation. Fig. 3 shows an example. If we rotated Fig. 3a only by  $5^\circ$  counter-clockwise, the extracted orientation of the local area will change from  $120^\circ$  to  $90^\circ$ , i.e.  $30^\circ$  difference.

To circumvent the above problems in traditional orientation coding schemes, we propose in this paper a new feature representation algorithm, namely binary orientation co-occurrence vector (BOCV). Instead of extracting only one orientation from the filtering responses, we preserve all the orientation information by concatenating the responses as a vector. Then, the response vector is binarized by thresholding. There are two main advantages of BOCV over with the traditional orientation based methods. First, the dis-

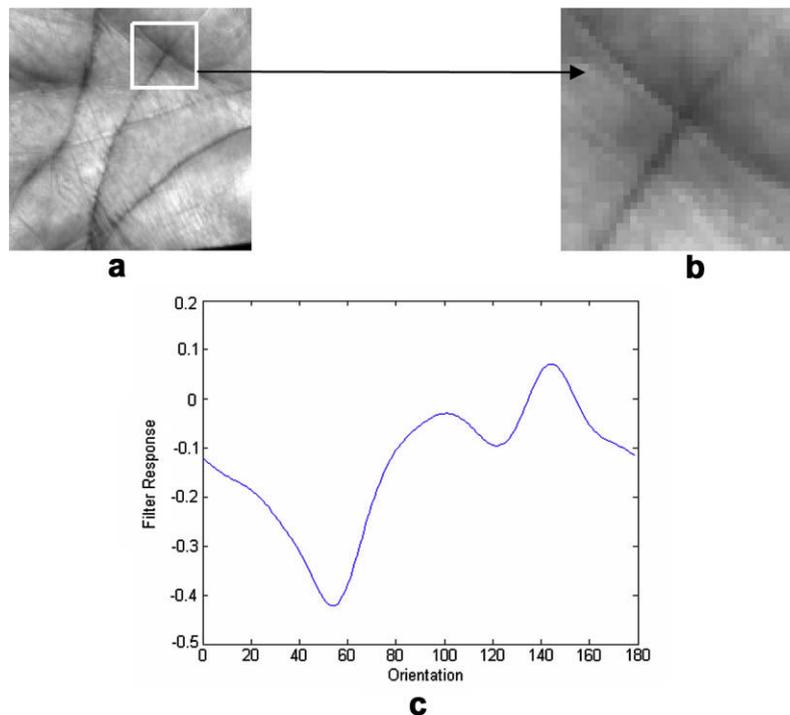
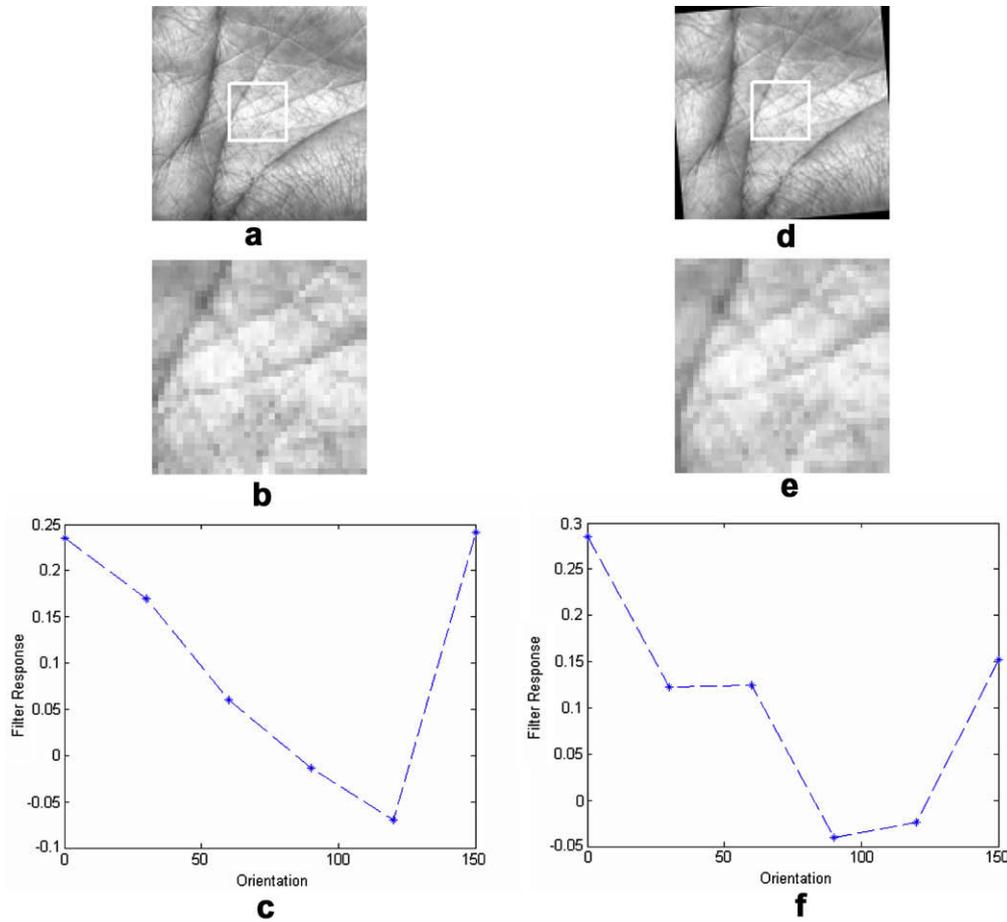


Fig. 2. (a) A palmprint image; (b) cropped and enlarged image with two intersected lines; and (c) Gabor filtering responses vs. orientation.



**Fig. 3.** (a) is a palmprint image; and (d) is the 5° rotation of it; (b) and (e) are the cropped and enlarged images of (a) and (d); (c) and (f) are the curves of Gabor filtering responses vs. six orientations for (b) and (e), respectively.

criminatory ability is enhanced because more line orientation information is preserved. Second, it is more robust to small rotation. Taking Fig. 3 as an example, we see that the “dominant” direction is very sensitive to rotation. A 5° rotation of the image will lead to a 30° change of the dominant direction (from 120° to 90°). However, if we code all the directions, it is possible that Fig. 3c and f have the same code. For instance, we code the Gabor filtering responses above 0 as “0”, and code the responses below 0 as “1”. Then the codes for both Fig. 3c and f are “000110”. They have the same representation after small rotation.

The rest of the paper is organized as follows. Section 2 briefly introduces CompCode. Section 3 shows the proposed feature extraction algorithm and its matching metric. Section 4 presents extensive experimental results to illustrate and verify the proposed method. Section 5 draws the conclusions.

## 2. Brief review of CompCode

Gabor filters are widely used as tunable filters for extracting the orientation or edge information from images. The Gabor function is usually defined as the following form:

$$\psi(x, y, \omega, \theta) = \frac{\omega}{\sqrt{2\pi\kappa}} e^{-\frac{\omega^2}{8\kappa^2}(4x^2+y^2)} \left( e^{i\omega x'} - e^{-\frac{\kappa^2}{2}} \right) \quad (1)$$

where  $x' = (x - x_0)\cos\theta + (y - y_0)\sin\theta$ ,  $y' = -(x - x_0)\sin\theta + (y - y_0)\cos\theta$  is the center of the function;  $\omega$  is the radial frequency in radians per unit length and  $\theta$  is the orientation of the Gabor functions in radians.  $\kappa$  is defined as  $\kappa = \sqrt{2\ln 2} \left( \frac{2^{\delta} + 1}{2^{\delta} - 1} \right)$ , where  $\delta$  is the half-amplitude bandwidth of the frequency response.

To extract the orientation information from palm lines, CompCode uses six real part of the neurophysiology-based Gabor filters  $\psi_R$ , with different orientations,  $\theta_j = j\pi/6$ , where  $j = \{0, 1, 2, 3, 4, 5\}$ . According to the features of palm lines, CompCode selects  $\arg \min_j (I(x, y) * \psi_R(x, y, \omega, \theta_j))$  as the orientation at position  $(x, y)$  of image  $I$ .

To implement real-time palmprint recognition, CompCode uses three bits to represent an orientation. An angular distance based on Hamming distance was used:

$$D(P, Q) = \frac{\sum_{y=1}^M \sum_{x=1}^N \sum_{i=0}^2 (P_i^b(x, y) \otimes Q_i^b(x, y)) \cap (P_M(x, y) \cap Q_M(x, y))}{3 * \sum_{y=1}^M \sum_{x=1}^N P_M(x, y) \cap Q_M(x, y)} \quad (2)$$

where  $P$  and  $Q$  are two CompCodes,  $P_i^b(Q_i^b)$  is the  $i$ th bit plane of  $P$  ( $Q$ ),  $\otimes$  is bitwise exclusive OR and  $\cap$  is bitwise AND operator.  $P_M$  and  $Q_M$  are the corresponding masks of  $P$  and  $Q$ , respectively. The masks are used to record the palmprint pixels (Zhang et al., 2003).

## 3. Binary orientation co-occurrence vector

Usually, the cross section of palm lines is Gaussian-shaped. Fig. 4c shows an example of palm line intensity value distribution, while Fig. 4d shows the real part of a Gabor filter, which has similar (but upside-down) shape to Fig. 4c. The Gabor filter can be regarded as a line detector or matched filter to detect palm lines (Van Deemter and Du Buf, 2000). If we normalize the Gabor filtering response vector to  $L_2$ -norm unity as in (3), the filter response at each orientation can be treated as a confidence measure of the feature occurring at that orientation (Varma and Zisserman, 2005).

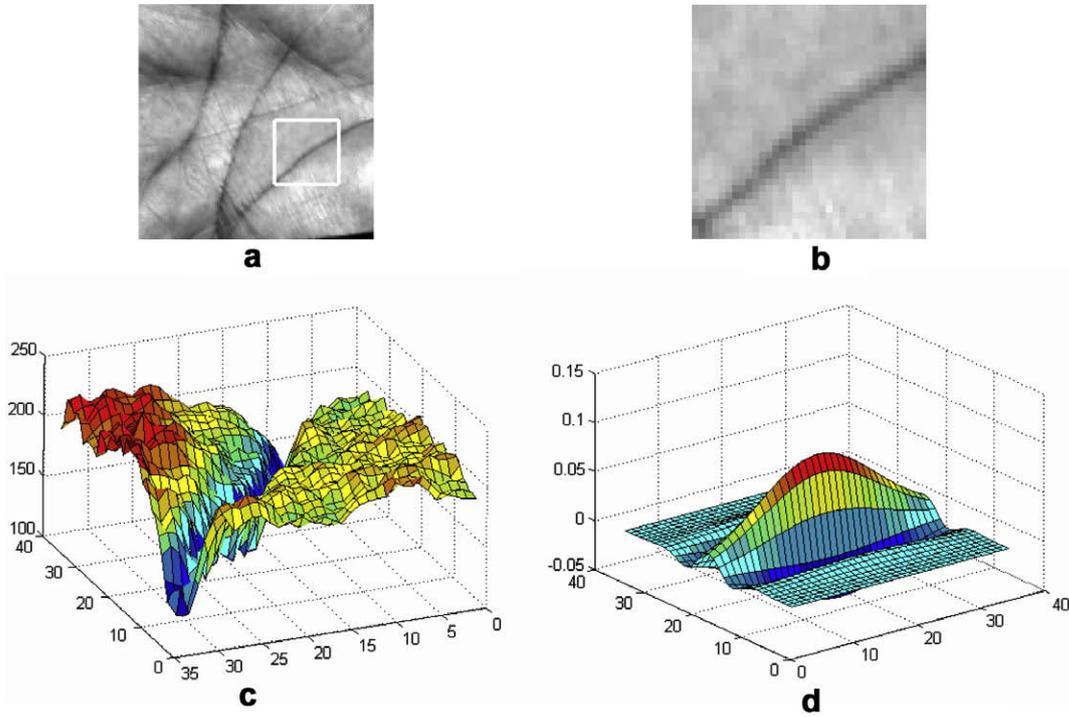


Fig. 4. (a) A palmprint image; (b) cropped and enlarged image of (a); (c) Intensity value distribution of (b); (d) Gabor filter with  $\theta = 0$ .

$$G_j(x, y) = \frac{G_j(x, y)}{\sqrt{\sum_{i=0}^5 G_i(x, y)^2}}$$

$$G'_j(x, y) = I(x, y) * \psi_R(x, y, \omega, \theta_j), \quad \theta_j = j\pi/6, \quad j = \{0, 1, 2, 3, 4, 5\} \quad (3)$$

For each local region, we can calculate a 6-dimensional vector by concatenating the normalized responses along 6 directions, namely the orientation co-occurrence vector (OCV). The distance between two OCVs,  $P$  and  $Q$ , can be computed by using the  $L_1$ -norm:

$$D(P, Q) = \frac{\sum_{y=1}^M \sum_{x=1}^N \sum_{j=0}^5 |P_j(x, y) - Q_j(x, y)| * (P_M(x, y) \cap Q_M(x, y))}{\sum_{y=1}^M \sum_{x=1}^N P_M(x, y) \cap Q_M(x, y)} \quad (4)$$

where  $P_i$  and  $Q_i$  are the  $i$ th dimension planes of  $P$  and  $Q$  respectively, while  $P_M$  and  $Q_M$  are the corresponding palmprint pixel masks (Zhang et al., 2003) of  $P$  and  $Q$ , respectively.

However, because the orientation features are represented by float numbers, it is time consuming to use OCV for dissimilarity computation. To speed up the matching time, a 6-bit binarized vector, called binary OCV (BOCV), is defined by thresholding each orientation's filter response

$$P_j^b(x, y) = \begin{cases} 1, & \text{if } G'_j(x, y) < T_j \\ 0, & \text{else} \end{cases} \quad (5)$$

The threshold  $T_j$  could be set as 0, which is simple and intuitive but could lead to good result. It can also be chosen according to the filter response distribution which could further improve the accuracy. How to set the optimal threshold will be discussed in Section 4. In the following, unless state explicitly, we set  $T_j = 0$ ,  $j = \{0, 1, 2, 3, 4, 5\}$ . Fig. 5 shows an example of the extracted BOCV.

Similar to CompCode, the widely used Hamming distance can be applied to BOCV for matching:

$$D(P^b, Q^b) = \frac{\sum_{y=1}^M \sum_{x=1}^N \sum_{j=0}^5 (P_j^b(x, y) \otimes Q_j^b(x, y)) \cap (P_M(x, y) \cap Q_M(x, y))}{6 * \sum_{y=1}^M \sum_{x=1}^N P_M(x, y) \cap Q_M(x, y)} \quad (6)$$

Obviously,  $D$  is between 0 and 1, and for a perfect matching the distance will be 0. In practice, we will shift the BOCV map along different directions in a small range to find the smallest distance between two maps. If the distance is smaller than a certain level, the two palmprints will be classified into the same class.

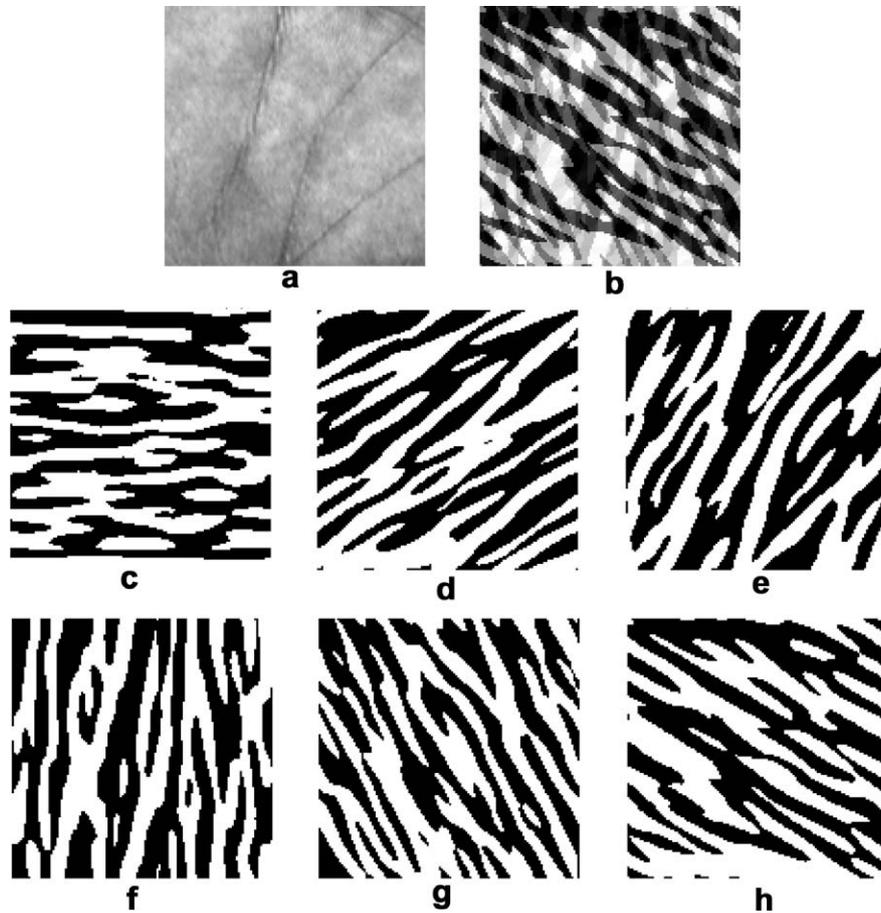
## 4. Experimental results

### 4.1. Palmprint database and test protocol

The public palmprint database (PolyU Palmprint Database, 2006) includes 7752 palmprint images from 193 individuals. The database is collected in two sessions. Each time, the subject was asked to collect around 10 palmprint images from his left and right palms. Altogether, each person provided around 40 images. The average time interval between the two sessions is 69 days.

To compute the verification accuracy in the following tests, each palmprint image is matched with all the other palmprint images in the database. A match is counted as a genuine if the two palmprint images are from the same palm; otherwise, it is counted as an impostor. The total number of matches is 30,042,876 and the number of genuine is 74,068. The Equal Error Rate (EER) (the point when false accept rate (FAR) is equal to false reject rate (FRR)) and the decidability index  $d'$  (Daugman, 2003) (the index measures how well the genuine and impostor distributions are separated) are used to evaluate the accuracy.

$$d' = \frac{|\mu_1 - \mu_2|}{\sqrt{(\sigma_1^2 + \sigma_2^2)/2}} \quad (7)$$



**Fig. 5.** A palmprint image and its BOCV features. (a) Original palmprint image; (b) BOCV feature map; (c–h) are the binarized feature maps by six Gabor filters in six directions.

A ROI extraction procedure similar to that in (Zhang et al., 2003) is used to extract the ROI of size  $128 \times 128$ . To reduce the influence of imperfect ROI extraction, we shift the feature maps vertically and horizontally in a small range for matching. The minimal distance obtained by shift matching is taken as the final distance. The shift range is set as  $[-4, 4]$  in the following experiments.

#### 4.2. Determination of the number of gabor filters

Although those orientation based coding algorithms (Jia et al., 2008; Kong and Zhang, 2004; Wu et al., 2005) have been widely used, the relationship between the number of employed directional filters (i.e. the number of quantized orientations) and the recognition accuracy has not been well discussed. CompCode (Kong and Zhang, 2004), POC (Wu et al., 2005) and RLOC (Jia et al., 2008) use 6, 4, and 6 filters respectively, but the authors did not clearly show why such numbers were used and whether the number was optimal. Intuitively, using more filters may obtain higher accuracy but increase the computational cost. Thus, it is necessary to analyze the determination of the optimal number of filters, which could result in high accuracy and fast implementation. In the following, we discuss this issue.

First, we use 2, 4, 6, 8, 10, 12, 14 and 16 Gabor filters with  $\pi/2, \pi/4, \pi/6, \pi/8, \pi/10, \pi/12, \pi/14$  and  $\pi/16$  interval of  $[0, \pi)$  to extract BOCV features and implement verification using Hamming distance. The computed EER and  $d'$  are plotted in Fig. 6. We see that the EER is relatively high when the number of Gabor filters is smaller than 6. When the number is bigger than 6, the EER is much lower but fluctuates rather than monotonically decreases with the

number of filters. Although  $d'$  increases as the number increases, the curve is flat when the number is greater than 6. Thus 6 can be regarded as the optimal number balancing between the accuracy and time consumption. This finding is also in accordance with the neuro-physiological discovery: simple cells are sensitive to specific orientation with approximate bandwidth of  $\pi/6$  (Lee, 1996).

The mainly reason that why increasing the number of filters could not further reduce the EER is the feature redundancy. To illustrate it, we can calculate the average rate of identical features between adjacent bit planes as follows:

$$Sb(P) = \frac{\sum_{y=1}^M \sum_{x=1}^N \sum_{i=0}^{n-1} (P_i^b(x, y) \otimes P_{mod(i+1, n)}^b(x, y)) \cap (P_M(x, y) \cap Q_M(x, y))}{n * \sum_{x=1}^M \sum_{y=1}^N P_M(x, y) \cap Q_M(x, y)} \quad (8)$$

where  $P^b$  is an extracted BOCV map,  $mod(x, y)$  is the modulus of  $x$  divided by  $y$ ,  $n$  is the number of Gabor filters, and “ $\otimes$ ” is a bitwise NOT operator.

The curve of average rates on the whole database vs. the number of Gabor filters is plotted in Fig. 7. We can see that as the number of Gabor filters increases, the percentage of identical bits between adjacent bit planes also increases. Assume that the binary values in each plane follow Bernoulli trials. If two planes are uncorrelated, then the percentage of identical bits should be 50%. However, Fig. 7 shows that as the number of Gabor filters increases, the correlation between adjacent bit planes increases, so that using

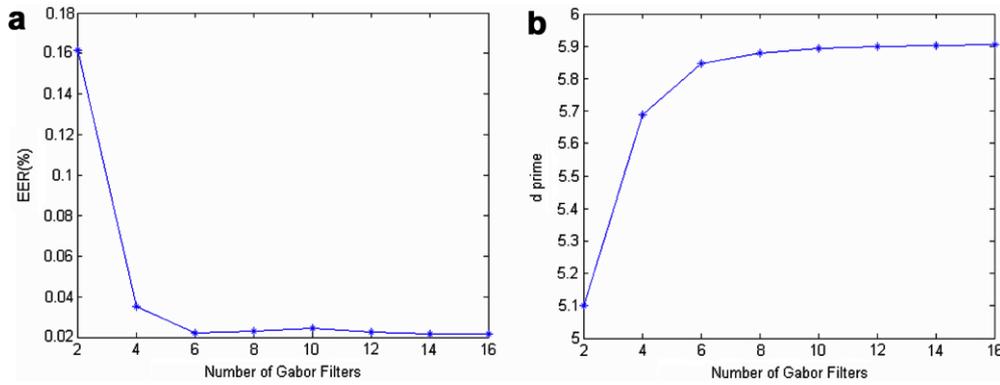


Fig. 6. EER and  $d'$  of BOCV using different number of Gabor filters. (a) EER vs. number of Gabor filters; and (b)  $d'$  vs. number of Gabor filters.

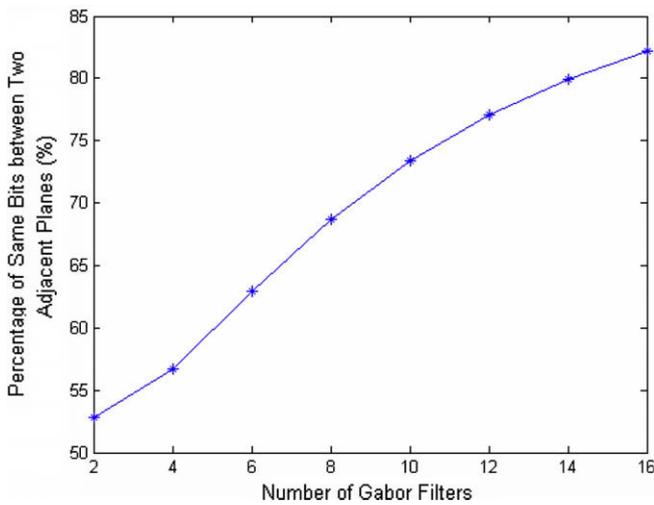


Fig. 7. Average rate of the identical features between two adjacent planes vs. the number of Gabor filters.

more Gabor filters could not increase much the discriminatory information.

Assuming the comparison (exclusive OR) between two BOCVs from two different palms follows a Bernoulli trial, the full distribu-

tion of impostor distance corresponds to a fractional binomial, whose degrees-of-freedom could be simulated as (Daugman, 2003):

$$N = p(1 - p)/\sigma^2 \tag{9}$$

where  $p$  is the mean of impostor distance distribution,  $\sigma$  is the standard deviation of impostor distance distribution. Fig. 8 shows the degrees-of-freedom using different number of filters. It shows similar trend to that of  $d'$  in Fig. 6b. The discriminatory information bits increases rapidly when the number of filters is less than 6, while it increases little when the number is greater than 6. On the other hand, finer quantization may increase the genuine distance due to noise and rotation. Fig. 9 shows an example. If six Gabor filters are used, the 6-bit codes of the two ROIs are both 000110. However, if eight filters are used, the 8-bit codes of the two ROIs are 01011110 and 00011110, respectively.

Based on the above analysis, in the all the experiments in this paper, we set the number of Gabor filters as 6.

#### 4.3. The robustness to rotation

As shown in Fig. 3, the extracted “dominant” orientation by CompCode is sensitive to small rotation, while the proposed BOCV scheme is not so sensitive. To further show that BOCV is more robust to rotation than CompCode, two experiments are performed

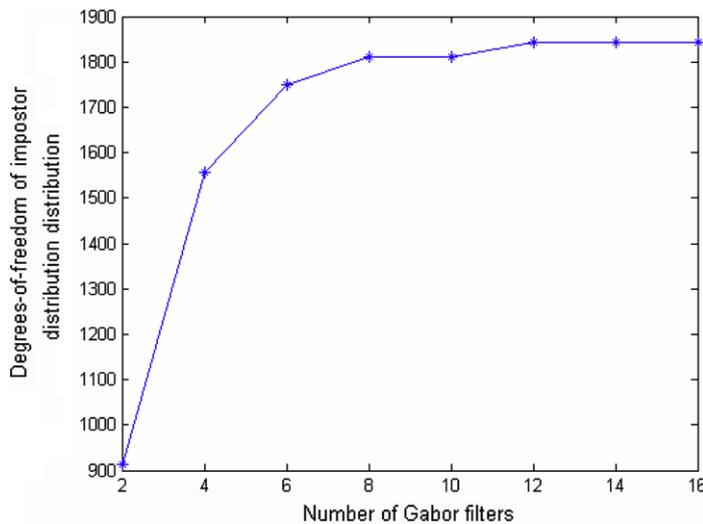


Fig. 8. Degrees-of-freedom of impostor distance distribution vs. number of Gabor filters.

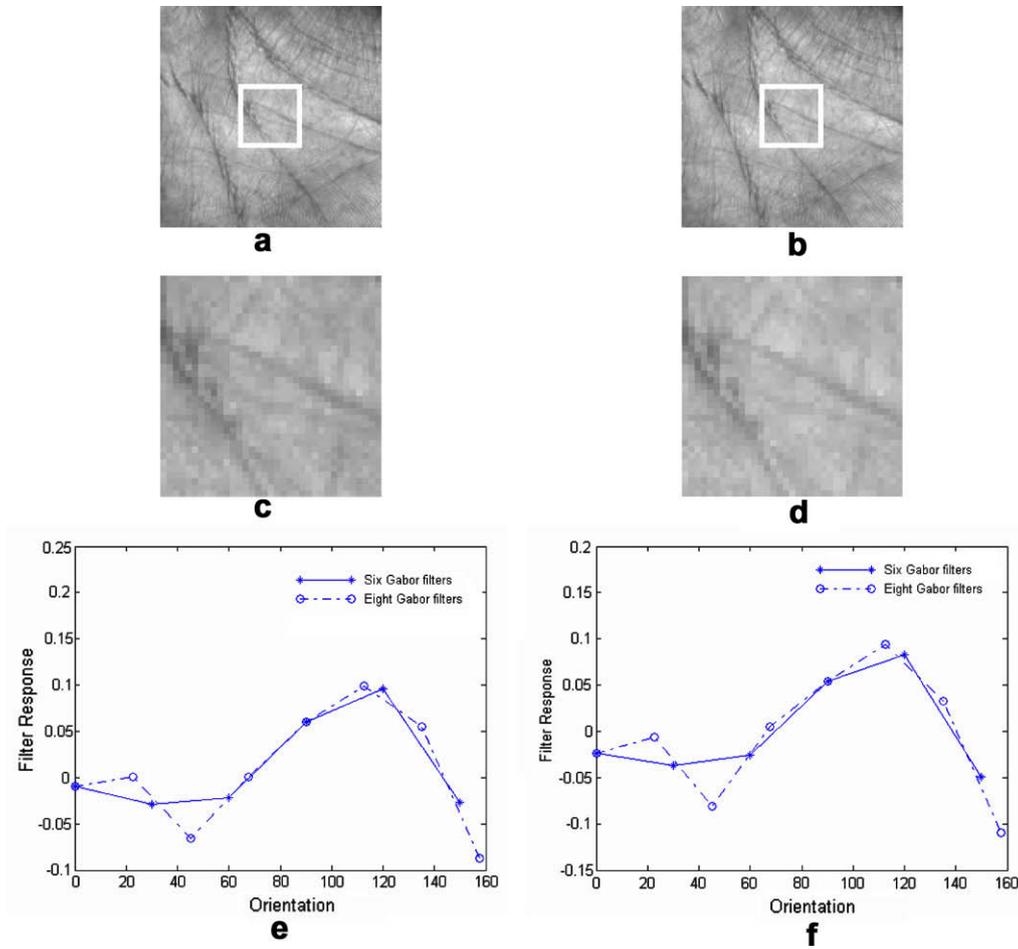


Fig. 9. An example to show finer quantization may increase genuine distance. (a and b) Two ROIs of two sample images from the same palm; (c and d) cropped and enlarged images of (a and b); (e and f) filter responses using six and eight Gabor filters.

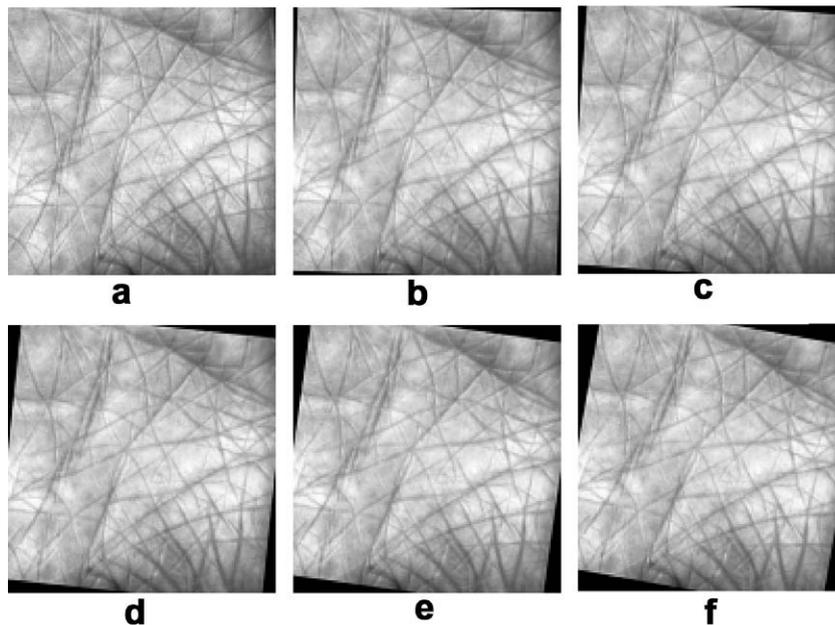


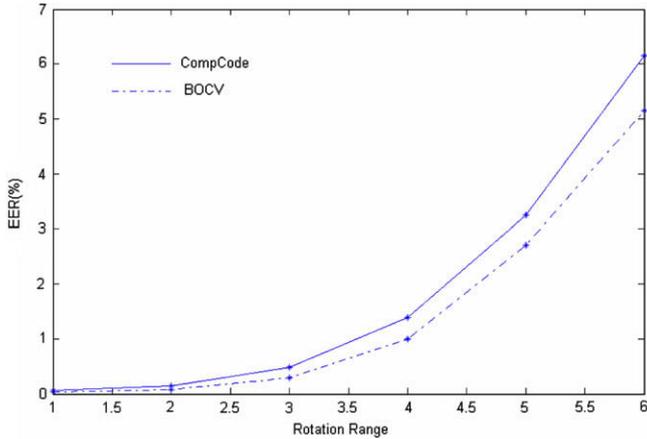
Fig. 10. An original image and its rotated images. (a) original image; (b–f) are the rotated images of (a) by 2°, 4°, 6°, 8° and 10° clockwise, respectively.

in this section. In the first experiment, we rotate a ROI image by 2°, 4°, 6°, 8° and 10° clockwise, as shown in Fig. 10. The matching distances between the images by CompCode and BOCV are listed in Table 1.

From Table 1, we can see that the matching distances by BOCV are smaller than those by CompCode. This validates that BOCV will give more robust recognition results when there are small alignment or registration errors of the palmprint images. To further

**Table 1**  
Matching distances between the images in Fig. 10.

CompCode/BOCV	(a)	(b)	(c)	(d)	(e)	(f)
(a)	0	0.1974/0.1393	0.3337/0.2635	0.4281/0.3744	0.4775/0.4475	0.4959/0.4858
(b)		0	0.1842/0.1389	0.3243/0.2699	0.4155/0.3688	0.4700/0.4449
(c)			0	0.1924/0.1398	0.3379/0.2560	0.4333/0.3579
(d)				0	0.1868/0.1232	0.3465/0.2493
(e)					0	0.1970/0.1342



**Fig. 11.** EER vs. rotation by CompCode and BOCV.

validate BOCV's robustness to small rotation, another experiment is performed. Each images in the database is rotated randomly by a degree within a range  $[-d, d]$ , where  $d = \{1, 2, 3, 4, 5, 6\}$ . By using the test protocol described in Section 4.1, the calculated EER curves are plotted in Fig. 11. We can see that the EER values by BOCV are always lower those by CompCode at all the rotation degrees.

**4.4. Palmprint verification results**

In the proposed BOVC, a set of thresholds need to be configured for binarization (referring to (5)). An intuitive and simple but effective setting is 0, which is widely used in the coding for palmprint (Kong et al., 2006; Zhang et al., 2003) and iris (Daugman, 2003) recognition. However, by using 0 as the threshold, the distribution of the binarized values may not be even. Suppose the probability of 1 in the binarized result is  $p$ , Table 2 shows the Exclusive OR outputs and their associated probabilities under different inputs. If we assume the matching score between two BOCV maps from two different palms follows a Bernoulli trial (Daugman, 2003), the distance or difference between them will be maximized if and only if  $p = 0.5$ . When  $p$  is not equal to 0.5, the probability of smaller impostor distance will be increased.

As shown in Table 3, if we use 0 as the threshold, the average rates of 1 and 0 in the six planes are not even, and this increases

**Table 2**  
Exclusive OR output and the associated probability.

Exclusive OR output/possibility		
0	$0/(1-p)^2(1-p)$	$1/p^2(1-p)$
1	$1/(1-p)^2p$	$0/(p^2p)$

**Table 3**  
The average rate of 1 and 0 among six planes by using 0 as threshold.

	Plane 0	Plane 1	Plane 2	Plane 3	Plane 4	Plane 5
Rate of 1 (%)	47.6146	47.0470	46.5781	46.2687	46.5260	46.6556
Rate of 0 (%)	52.3854	52.9530	53.4219	53.7313	53.4740	53.3444

**Table 4**  
The average percentage of 1 among six planes using tuned thresholds.

	Plane 0	Plane 1	Plane 2	Plane 3	Plane 4	Plane 5
Rate of 1 (%)	50.0186	50.0527	50.0505	50.0955	50.0326	50.0496
Rate of 0 (%)	49.9814	49.9473	49.9495	49.9045	49.9674	49.9504

the probability of smaller impostor distance. Thus, if a threshold can result in a more even distribution between 1 and 0 in the binarized plane, the impostor distance could be increased and the accuracy may be increased accordingly.

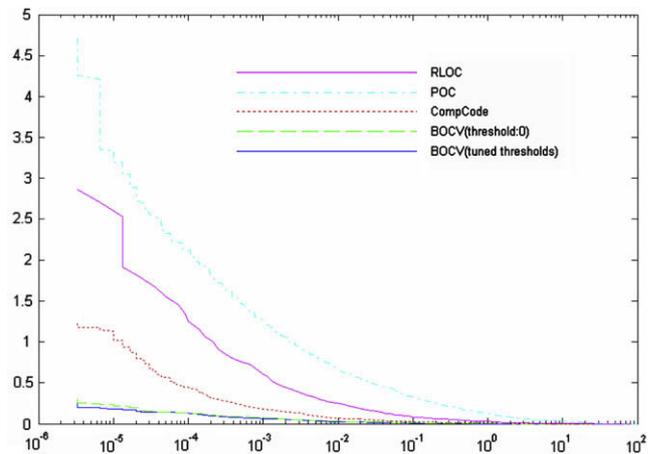
Since (4) can be rewritten as (10), which is an average distance of six independent planes, the six thresholds for six planes can be tuned based on the first 192 palms, totally 3849 images. Using the tuned thresholds, the average rates of 1 and 0 in the binarized BOCV over the whole database are listed in Table 4, which shows that the tuned thresholds could get a more even distribution for each plane.

$$D(P, Q) = \frac{\sum_{y=1}^M \sum_{x=1}^N \sum_{j=0}^5 (P_j^b(x, y) \otimes Q_j^b(x, y)) \cap (P_M(x, y) \cap Q_M(x, y))}{6 * \sum_{y=1}^M \sum_{x=1}^N P_M(x, y) \cap Q_M(x, y)}$$

$$= \frac{1}{6} \sum_{j=0}^5 D(P_j^b, Q_j^b)$$

(10)

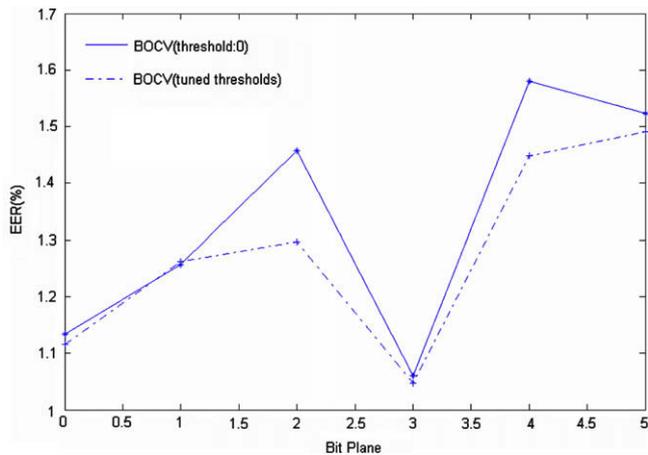
Fig. 12 plots the receiver operating characteristic (ROC) curves by different methods and Table 5 shows the accuracy rates for comparison. Some optimizations have been made on ROI extraction and matching, so the experimental results for RLOC and CompCode are better than the previous publications (Jia et al., 2008; Kong and Zhang, 2004). From Fig. 12 and Table 5, some conclusions could be made. First, because BOCV could keep more directional information than CompCode, it could get better results than CompCode. Second, by using tuned thresholds, the EER is lower than using 0 as the threshold, and lower FRR could be achieved in most



**Fig. 12.** ROC curves by different methods.

**Table 5**  
Verification accuracy by different methods.

	EER (%)	$d'$	FRR (when FAR = $3.3 \times 10^{-6}$ %)
POC	0.2366	3.4549	4.7092
RLOC	0.0905	<b>6.2768</b>	2.8594
CompCode	0.0379	5.4122	1.2273
BOCV (threshold: 0)	0.0220	5.8477	0.3011
BOCV (tuned thresholds)	<b>0.0189</b>	5.7575	<b>0.2525</b>



**Fig. 13.** EER vs. individual plane.

cases especially when  $FAR < 4 \times 10^{-5}$ %. This validates the effectiveness of tuned thresholds in increasing the impostor distance. To better illustrate the effectiveness of tuned thresholds, the curves of EER vs. individual plane are plotted in Fig. 13. We can see that a more even distribution and smaller EER values can be obtained by using tuned thresholds.

#### 4.5. Feature size vs. speed

In the proposed BOVC, 6 bits are used to represent orientations for each pixel. To speed up matching during verification, the feature is down-sampled to  $32 \times 32$ , thus the feature size is 768 bytes in total for one image, twice the CompCode. The system is implemented using Visual C++6.0 on a Windows XP, T6400 CPU (2.13GHz) and 2GB Ram PC. The execution time for ROI extraction, feature extraction and matching is about 138 ms, 40 ms, and 0.33 ms respectively. The total execution time of verification is less than 0.5 s, which is fast enough for real-time application. As the speed of matching is fast, it can be easily extended to identification system.

## 5. Conclusions

In this paper, we proposed a novel feature extraction scheme, BOCV, for palmprint verification. The BOCV scheme could keep more orientation information for complex palmprint lines and is more robust to small rotations than the conventional CompCode. We also investigated the relationship between orientation quantization and accuracy, and found that 6 is an optimal number of orientation quantization in terms of accuracy and complexity. A threshold selection based on binary value distribution was proposed for the BOCV scheme. Experimental results demonstrated the effectiveness of this scheme. Using the same Gabor filters as in CompCode, the proposed BOCV could reduce the EER from

0.0379% to 0.0189%. The proposed BOCV can be extended to other binary feature extraction algorithms, such as POC, RLOC and orthogonal line ordinal feature (OLOF) (Sun et al., 2005).

## Acknowledgements

The work is partially supported by the CERG fund from the HKSAR Government, the central fund from Hong Kong Polytechnic University, the NSFC/863 funds under Contract Nos. 60620160097, 60803090, and 2006AA01Z193 in China, and Science Foundation of Shenzhen City (CXQ 2008019). We would like to thank the anonymous reviewers for their constructive comments.

## References

- Chen, G.Y., Xie, W.F., 2007. Pattern recognition with SVM and dual-tree complex wavelets. *Image Vision Comput.* 25, 960–966.
- Connie, T., Andrew, T., Goh, K., 2005. An automated palmprint recognition system. *Image Vision Comput.* 23, 501–505.
- Daugman, J., 2003. The importance of being random: Statistical principles of iris recognition. *Pattern Recognition* 36, 279–291.
- Duta, N., Jain, A.K., Mardia, K.V., 2002. Matching of palmprint. *Pattern Recognition Lett.* 23, 477–485.
- Han, C.-C., Cheng, H.-L., Lin, C.-L., Fan, K.-C., 2003. Personal authentication using palm-print features. *Pattern Recognition* 36, 371–381.
- Hu, D., Feng, G., Zhou, Z., 2007. Two-dimensional locality preserving projections (2DLPP) with its application to palmprint recognition. *Pattern Recognition* 40, 339–342.
- Jain, A., Bolle, R., Pankanti, S., 1999. *Biometrics: Personal Identification in Network Society*. Kluwer Academic Publishers, Boston.
- Jia, W., Huang, D.-S., Zhang, D., 2008. Palmprint verification based on robust line orientation code. *Pattern Recognition* 41, 1521–1530.
- Jing, X.-Y., Yao, Y.-F., Zhang, D., Yang, J.-Y., Li, M., 2007. Face and palmprint pixel level fusion and Kernel DCV-RBF classifier for small sample biometric recognition. *Pattern Recognition* 40, 3209–3224.
- Kong, A., Zhang, D., 2004. Competitive coding scheme for palmprint verification. In: *Internat. Conf. on Pattern Recognition*, pp. 520–523.
- Kong, A., Zhang, D., Kamel, M., 2006. Palmprint identification using feature-level fusion. *Pattern Recognition* 39, 478–487.
- Kumar, A., Zhang, D., 2005. Personal authentication using multiple palmprint representation. *Pattern Recognition* 38, 1695–1704.
- Lee, T.S., 1996. Image representation using 2D Gabor wavelet. *IEEE Trans. Pattern Anal. Mach. Intell.* 18, 959–971.
- Lu, G., Zhang, D., Wang, K., 2003. Palmprint recognition using eigenpalms features. *Pattern Recognition Lett.* 24, 1463–1467.
- NIST Report on the United States Congress, 2001. Summary of NIST Standards for Biometric Accuracy, Temper, Resistance, and Interoperability.
- PolyU Palmprint Database, 2006. <<http://www.comp.polyu.edu.hk/~biometrics>>.
- Ribaric, S., Fratric, I., 2005. A biometric identification system based on eigenpalm and eigenfinger features. *IEEE Trans. Pattern Anal. Mach. Intell.* 27, 1698–1709.
- Shu, W., Zhang, D., 1998. Automated personal identification by palmprint. *Opt. Eng.* 37, 2359–2362.
- Sun, Z., Tan, T., Wang, Y., Li, S.Z., 2005. Ordinal palmprint representation for personal identification. In: *IEEE Comput. Soc. Conf. on Comput. Vision and Pattern Recognition*, pp. 279–284.
- Van Deemter, J.H., Du Buf, J.M.H., 2000. Simultaneous detection of lines and edges using compound Gabor filters. *Pattern Recognition and Artif. Intell.* 14, 757–777.
- Varma, M., Zisserman, A., 2005. A statistical approach to texture classification from single images. *Internat. J. Comp. Vision* 62, 61–81.
- Wang, J., Yau, W., Suwandy, A., Sung, E., 2008. Person recognition by fusing palmprint and palm vein images based on “Laplacianpalm” representation. *Pattern Recognition* 41, 1531–1544.
- Wu, X., Wang, K., Zhang, D., 2000. Palmprint authentication based on orientation code matching. In: *Internat. Conf. on Audio- and Video-Based Biometric Person Authentication*, pp. 555–562.
- Wu, X., Zhang, D., Wang, K., 2003. Fisherpalm based palmprint recognition. *Pattern Recognition Lett.* 24, 2829–2838.
- Yao, Y.-F., Jing, X.-Y., Wong, H.-S., 2007. Face and palmprint feature-level fusion for single sample biometrics recognition. *Neurocomputing* 70, 1582–1586.
- Zhang, D., 2000. *Automated Biometrics – Technologies and Systems*. Kluwer Academic Publishers, Boston.
- Zhang, D., Kong, W.-K., You, J., Wong, M., 2003. Online palmprint identification. *IEEE Trans. Pattern Anal. Mach. Intell.* 25, 1041–1050.
- Zhang, L., Zhang, D., 2004. Characterization of palmprints by wavelet signatures via directional context modeling. *IEEE Trans. System Man Cybernet. Part B* 34, 1335–1347.