# Palmprint Recognition Based on Line and Slope Orientation Features

MIN-KI KIM
*Research Institute of Computer and Information Communication*
*Department of Computer Science Education*
*Gyeongsang National University, Jinju, Gyeongnam 660-701, Korea*
*E-mail: mkkim@gnu.ac.kr*

In the field of palmprint recognition, the orientation information of the principal lines and winkles has long been considered the most dominant and reliable feature. Numerous studies have tried to extract the line orientation information. Among them, the orientation based coding methods such as robust line orientation code (RLOC) and binary orientation co-occurrence vector (BOCV) showed highly promising results. However, the orientation information of pixels that are not located on the palm lines could be greatly affected by the lighting conditions. To solve this problem, this paper proposes a new combined approach using both the line orientation and the slope orientation. When a palm image is hypothetically considered as a 3-D terrain, the principal lines and winkles are deep and shallow valleys on a palm landscape. If the previous line-based approaches focus only on the direction of valleys to investigate the palm landscape, this study focuses on the slope direction of local plains as well as the direction of valleys. The proposed method extracts two different orientation features according to the location of pixels and computes the feature distance between two images by a pixel-to-area matching method. Experimental results show that the proposed approach is superior to several state-of-the-art methods based on the line orientation coding.

***Keywords:*** biometrics, palmprint recognition, line and slope orientation, pixel-to-area matching, orientation coding

## 1. INTRODUCTION

Biometrics has been receiving attention as the substitute or complement to traditional token or key based personal identification methods. Biometrics comprises various methods for uniquely recognizing individuals by using one or more intrinsic physical or behavioral traits such as fingerprint, face, iris, hand geometry, palmprint, typing rhythm, voice, and gait. Among these traits, palmprint has a relatively short history and has received increasing interest in recent years. Initial palmprint researches focused on

high-resolution images (400 dpi or more), but recently almost all researches have focused on low resolution images (150 dpi or less) for civil and commercial applications [1].

During the past decades, various techniques have been proposed for palmprint recognition [1-12]. These techniques can be mainly classified into three categories: line-based, subspace-based, and statistical approaches. Line-based approaches propose methods of extracting palm lines such as principal lines and wrinkles [2-5]. These lines are usually represented in other formats for matching. Subspace-based approaches use holistic appearance features such as principal component analysis (PCA) [6, 7], linear discriminant analysis (LDA) [8] and independent component analysis (ICA) [9]. Statistical approaches transform an input image into another domain and divide the transformed image into small regions. After that, they extract statistical features from each small region [10, 11].

In these three categories, the line-based approaches are the most appealing for the human vision and recently several studies based on orientation information from palm lines have reported promising results. To extract the line orientation information, various approaches have been proposed. Kong and Zhang [2] used six 2-D Gabor filters with different directions and extracted the dominant orientation information by using the winner-takes-all rule. Wu et al. [3] devised four directional templates to define the orientation of each pixel. Jia et al. [4] devised other templates based on modified finite Radon transform (MFRAT) and proposed the robust line orientation code (RLOC). The magnitude in a direction is computed by the convolution operation between a template and its corresponding region of image. Since the gray-level of a pixel on the palm lines is lower than that of the surrounding pixels, the direction having minimum magnitude is selected as the dominant orientation of the pixel. Guo et al. [5] raised two problems that can occur by using only the dominant orientation information. First, the line structures in palmprint image are very complex and multiple lines may intersect in some regions, so some structural information may be lost if only one orientation is used to represent the local feature. Second, the extracted dominant orientation is sensitive to rotation. To circumvent these two problems, they proposed a new feature named as binary orientation co-occurrence vector (BOCV), which preserves all the orientation information.

All the above line-based approaches have a common problem. They extract the line orientation from every pixel in a palm image regardless of whether the pixels are on the

palm lines or not. Though the orientation information of the pixels which are on the palm lines is robust to the lighting conditions, the orientation information of the other pixels could be vulnerable to the change of illumination. In order to solve this problem, this paper proposes a new combined approach using both the line orientation and the slope orientation. When a palm image is hypothetically considered as a 3-D terrain, the principal lines and winkles are deep and shallow valleys on a palm landscape. If the previous line-based approaches focus only on the direction of valleys to investigate the palm landscape, this study focus on the slope direction of local plains as well as the direction of valleys.

The rest of this paper is organized as follows: Section 2 describes the preprocessing method for acquiring a region of interest (ROI) and presents the method of orientation feature extraction based on the MFRAT; Section 3 presents the coding scheme of the line and slope orientation features and the pixel-to-area matching method; Section 4 reports the experimental results and analyzes the results with two perspectives of identification and verification. In section 5, conclusions and suggestions for future work are presented.

## 2. ORIENTATION FEATURE EXTRACTION

### 2.1 Palmprint Preprocessing

When palmprint images are captured, some variation may occur such as translation and rotation. Hence palmprint images should be aligned in position and orientation before feature extraction can take place. The central part of a palm, which should be 128×128 pixels in size, is cropped and rotated by using an approach similar to that in [12]. The five main steps (see Fig. 1) for this processing are:

Step 1. Convert the original image to a binary image by using a threshold.

Step 2. Trace the boundary of palm, and then smooth the boundary by Gaussian filter.

Step 3. Find two reference points ($P_1$ and $P_2$): Three equidistant point $p_a$, $p_b$, and $p_c$ on the boundary make a turning angle as shown in Fig 1-(e). The three points move together around the boundary clockwise. The turning angle varies according to the relative

positions of the three points. If the direction of a vector between two neighbor points turns counterclockwise, the turning angle has a negative value. Otherwise it has a positive value. $P_1$ is the point $p_b$, where the turning angle has the smallest negative value in the top left part of the boundary. Similarly, $P_2$ can be found in the bottom left part of the boundary.

Step 4. Find the central point of a region of interest (ROI): Line up $P_1$ and $P_2$ to get the $Y$-axis of the palmprint coordinate system, and use a line passing through the midpoint ($P_m$) of these two points, which is perpendicular to the $Y$-axis, to determine the central point ($P_c$) of the ROI. The ROI is defined as a 128×128 sized square.

Step 5. Crop and rotate the ROI.

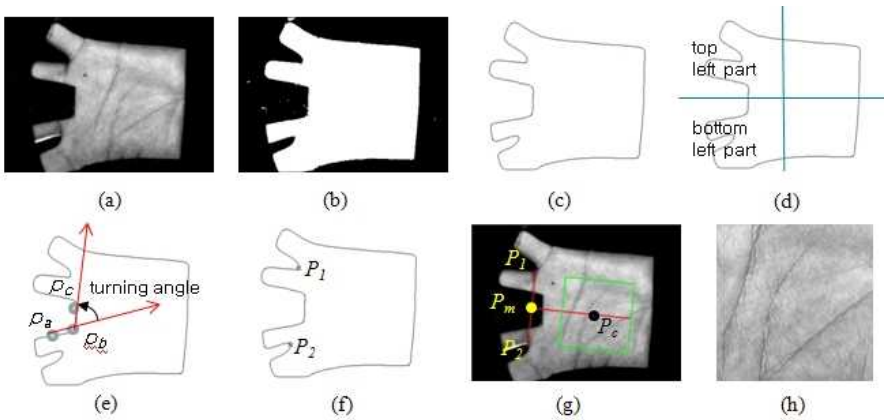



Fig. 1. The main steps of preprocessing (a) Original image, (b) binary image, (c) smoothed boundary, (d) top left part and bottom left part, (e) turning angle, (f) two reference points, (g) a central point and its ROI, and (g) preprocessed result.

### 2.2 Feature Extraction

The extraction of the line orientation feature is based on the MFRAT, which was proposed by Jia et al. [4]. The MFRAT of real function $f(x, y)$ on the finite grid $Z_p^2$, $Z_p = \{0, 1, \ldots, p\text{-}1\}$, where $p$ is a positive integer, is defined as

$$r[L_k] = \mathrm{MFRAT}_f(k) = \sum_{(i,j)\in L_k} f(i,j), \tag{1}$$

where $f(i, j)$ represents the gray level of the pixel at $(i, j)$ and $L_k$, which is defined in Eq. (2), denotes the set of points that make up a line on the lattice $Z_p^2$.

$$L_k = \{(i,j): j = k(i - i_0) + j_0,\ \ i \in Z_p\}, \tag{2}$$

where $(i_0, j_0)$ denotes the center point of the lattice $Z_p^2$ and $k$ is the corresponding slope of $L_k$. Therefore, for any given slope $k$, the summation of only one line which passes through the center point of $Z_p^2$ is calculated. Fig. 2 shows an example of the set $L_k$ with six different directions, where $Z_p^2$ is 15×15 and the line is 3-pixels wide. Due to space limitation, the remaining lines at the directions of 3π/6 and 4π/6 are not depicted here.

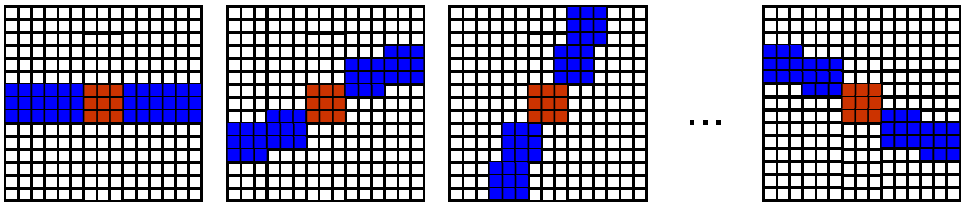


Fig. 2. The $L_k$ at the direction of 0°, π/6, 2π/6,…, and 5π/6

Since the gray-levels of the pixels on the palm lines are lower than those of the surrounding pixels, the line orientation and the line energy of the pixel at $(i, j)$ is defined as Eq. (3) and Eq. (4) respectively, where $m$ is the number of direction.

$$L_{ori}\,(i,j) = arg\,\left(\min_k(r[L_k])\right), k = 0, 1, 2, \dots, m-1. \tag{3}$$

$$L_{erg}\,(i,j) = \left|\min\left(r[L_k] - \sum_{i=0}^{m-1}\frac{r[L_k]}{m}\right)\right|, k = 0, 1, 2, \dots, m-1. \tag{4}$$

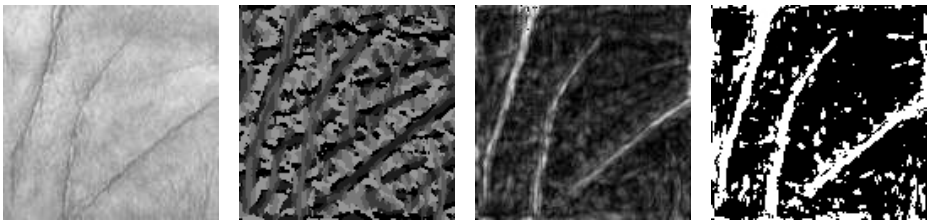
In Eq. (4), the minimum $r[L_k]$ is always smaller than the average. Therefore, Eq. (4) can be simplified as the following equation. As a result, $L_{erg}(i, j)$ represents the gap between the average depth and the deepest one.

$$L_{erg}\,(i,j) = \sum_{i=0}^{m-1}\frac{r[L_k]}{m} - \min(r[L_k]), k = 0, 1, 2, \dots, m-1. \tag{5}$$

Most of the line-based approaches have used only the orientation information. However, this study uses not only the orientation information but also the corresponding energy information. The line energy of a pixel indicates the prominence of a palm line. Lower line energy represents less prominent or less distinct palm lines or no palm lines at all. A less prominent palm line or no palm line will have a line orientation that is more susceptible to variation caused by changes in illumination and focus. The pixels on the palm lines have the high line energy but most of the other pixels have the low line energy.

Therefore, more stable features can be selectively used according to the line energy of a pixel.

The line energy of the pixel at $(i, j)$, calculated by Eq. (5), represents the magnitude of the line orientation feature. Namely, the energy of a pixel on a palm line increases as the overlapped area between the palm line and $L_k$ increase. So, the energy can be used as the criteria for classifying each pixel whether it is on the palm lines or not. If the directional energy of a pixel is greater than a predefined threshold value, the pixel is classified as a valley-type pixel which is on the palm lines. Otherwise the pixel is classified as a plain-type pixel. Fig. 3 shows a ROI image and its line orientation and energy images acquired by using the $L_k$ shown in Fig. 2



(a) (b) (c) (d)

Fig. 3. ROI image and its feature images: (a) ROI image, (b) line orientation image, (c) line energy image, and (d) valley-type pixels (white) and plain-type pixels (black)

According to the type of pixel, line or slope orientation is extracted from the ROI image. The line orientation is computed from each valley-type pixel by Eq. (3) and the slope orientation is acquired from each plain-type pixel by the following equations, where $n$ is the number of slope orientation and $R_d$ is the region to be probed for each direction.

$$S_{ori}(i,j) = arg\left(\min_d(r[R_d])\right), d = 0, 1, 2, \ldots, n-1. \quad (6)$$

$$r[R_d] = \sum_{(i,j)\in R_d} f(i,j) \quad (7)$$

$S_{ori}(i, j)$ represents the steepest descent slope direction in the local area defined on the lattice $Z_p^2$. Fig. 4 shows an example of the set $R_d$ with eight different directions whose lattice size is 15×15. Due to space limitation, the $R_d$ at the directions of 3π/4, 4π/4, 5π/4, and 6π/4 are not depicted.

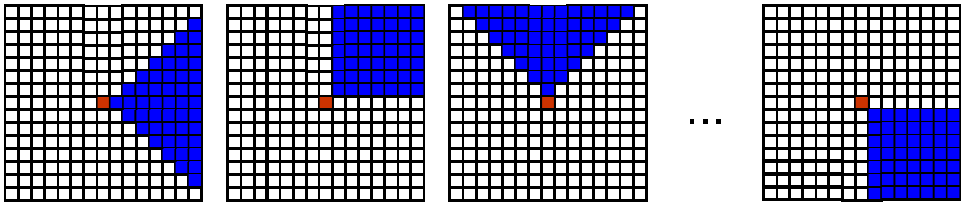


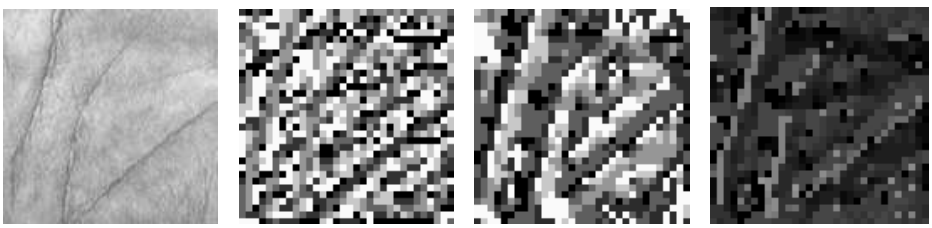
Fig. 4. The $R_d$ at the direction of 0°,π/4, 2π/4,…, and 7π/4

## 3. CODING SCHEME AND MATCHING METHOD

The size of the cropped ROI image is 128×128. For the ROI image, if all the pixels are replaced by their index of the dominant orientation, a new feature image $O(i, j)$ called the orientation map is created, the size of which is also 128×128. However, calculating the orientation of all the pixels requires much processing time, and it creates redundant information. The computation time and the amount of redundant information can be reduced by down sampling. If every fourth pixel is selected, the size of the orientation map is reduced to 32×32. Fig. 5 shows a ROI image and its feature images. The line orientation map and the slope orientation map are acquired from the orientation of line and slope of each pixel, whereas the combined orientation map is created by combining the two orientations. The combined orientation map $P_O$ is defined as

$$P_O(i,j) = \begin{cases} L_{ori}(i,j), & \text{if } p(i,j) \text{ is a vally-type pixel} \\ S_{ori}(i,j), & \text{if } p(i,j) \text{ is a plain-type pixel} \end{cases}, \tag{8}$$

where $p(i, j)$ represents the pixel at $(i, j)$.



(a) (b) (c) (d)

Fig. 5. ROI image and its orientation maps: (a) ROI image, (b) line orientation map, (c) slope orientation map, and (d) combined orientation map

The basic strategy for calculating the difference between two feature images is pixel-to-pixel matching. The feature distance between two corresponding pixels $p_1$ and $p_2$ is

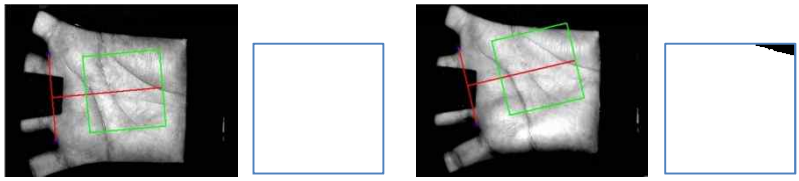
defined as Eq. (9), where $f_m$ and $f_n$ represent the modular-$m$ and modular-$n$ distance respectively. The modular distance means the distance on the circle (i.e., the modular-6 distance between 0 and 5 is 1, not 5). The function $max(m, n)$ returns a larger value, and $min(m, n)$ returns a smaller value.

$$f_{m,n}(p_1, p_2) = \begin{cases} f_m(p_1, p_2) \times W_{m,n} & \text{, if } p_1, p_2 \in \text{valley-type pixels} \\ f_n(p_1, p_2) & \text{, if } p_1, p_2 \in \text{plain-type pixels} \\ \max(m, n)/2 & \text{, otherwise.} \end{cases} \tag{9}$$

If the types of two corresponding pixels are equal, the modular distance between two orientations represents the feature distance. The weight $W_{m,n}$ is defined by the rate of $\max(m, n)/\min(m, n)$, it keeps the balance between two different numbers of orientations. If the types of the two pixels are not equal, the maximum modular distance is assigned. In this study, six orientations of line and eight orientations of slope are considered, so $m$=6 and $n$=8, and $W_{m,n}$=8/6. Based on the distance function $f_{m,n}(p_1, p_2)$, the distance of two images $P$ and $Q$ of size $r \times c$ can be described as

$$d(P, Q) = \frac{\sum_{i=1}^{r} \sum_{j=1}^{c} f_{m,n}\big(P_O(i,j), Q_O(i,j)\big) \times \big(P_M(i,j) \wedge Q_M(i,j)\big)}{\frac{\max(m,n)}{2} \times \sum_{i=1}^{r} \sum_{j=1}^{c} P_M(i,j) \wedge Q_M(i,j)} \tag{10}$$

where $P_O$ and $P_M$ represent the combined orientation map and masking map of an image $P$, $Q_O$ and $Q_M$ are similarly notated; the symbol $\wedge$ is the logical AND operator. The masking maps $P_M$ and $Q_M$ automatically created by a thresholding procedure, which generate a mask to identify the location of the non-palmprint pixels. If the gray level of a pixel is higher than a predefined threshold, the corresponding pixel at the masking map has 1. Otherwise, the pixel has 0. The masking maps are used to alleviate the mismatch problem caused by the misplacement of the palm during data sampling. Fig. 6 shows examples of two palms captured from the same person and their ROIs and masking maps.



(a) (b)

Fig. 6. Palm images captured from the same person: (a) example of normal placement of a palm and its ROI and masking map, (b) example of misplacement of a palm and its ROI and masking map

A translated or rotated palm image is mostly aligned in the preprocessing step that extracts the ROI of a palm. However, the remaining variation due to imperfect preprocessing can weaken the performance of the pixel-to-pixel matching method. So, the pixel-to-area comparison method used in [4, 12] is adopted to improve the robustness of matching. The distance function described in Eq. (10) is modified as

$$\vec{d}\,(P,Q) = \frac{\sum_{i=1}^{r}\sum_{j=1}^{c}\min\left(f_{m,n}\left(P_O(i,j),Q_O(i',j')\right)\right)\times\left(P_M(i,j)\wedge Q_M(i,j)\right)}{\frac{\max(m,n)}{2}\times\sum_{i=1}^{r}\sum_{j=1}^{c}P_M(i,j)\wedge Q_M(i,j)} \tag{11}$$

where $(i',j')\in\{(i\text{-}1,j),(i,j\text{-}1),(i,j),(i,j\text{+}1),(i\text{+}1,j)\}$. The function $\vec{d}(P,Q)$ represents the distance from $P$ to $Q$. In a similar way, the distance from $Q$ to $P$ can also be described as $\vec{d}(Q,P)$. Finally, the distance function of two images $P$ and $Q$ can be described as

$$D(P,Q) = \min\left(\vec{d}(P,Q),\vec{d}(Q,P)\right) \tag{12}$$

The $k$-nearest neighbor algorithm ($k$-NN) is used for palmprint identification. The neighbors are taken from a training set for which the correct classification is known. $K$-NN is a method for classifying objects based on closest training examples in the feature space. In this paper, $k$ was set as 1. Therefore, a palmprint is simply assigned to the class of its nearest neighbor. Training is performed by simply converting each palmprint image in training samples into the combined orientation map and masking map and storing them in the known class. Testing is done by comparing the feature maps extracted from an input image with them in each class.

## 4. EXPERIMENTAL RESULTS

In this experiment, the widely used public database, PolyU-II [13], was used. The database contains a total of 7,752 images from 386 different palms. The palmprint images were collected through two sessions. In each session, around 10 samples for each palm were captured. The average interval between the first and the second session was about two months. In addition, the light source and the focus of the CCD camera were changed so that the images collected in the first and second sessions could be regarded as being captured by two different devices [12]. LED was used in the first session while incandescent lamp was chosen in the second session. Also, the lenses were a little different in the two sessions and the focus in the first session is a little longer. Some samples in this database are shown in Fig. 7, in which the four samples in the top row were captured in the first session and those in the bottom row were captured in the second session. The two images in the same column were captured from the same palm at different sessions.

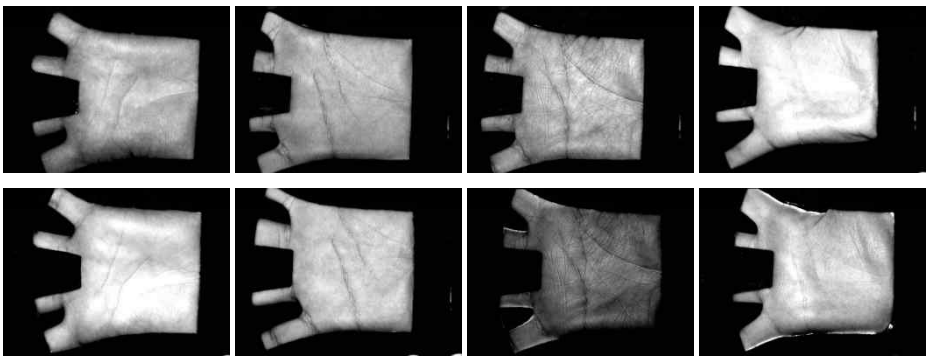


Fig. 7. Some samples in the PolyU Palmprint Database

The palm images captured in the first session were selected for training and those captured in the second session were used for testing of identification and verification. This experimental design is more realistic than that of the study [4], where the samples captured in the same session had been used both as the training set and as the test set.

### 4.1 Selection of the Window Size

The window size defining the lattice $Z_p^2$ affects the quality of the orientation feature. The combination of three different $L_k$ and three different $R_d$ were tested to select the suitable window size for the proposed method. For this experiment, 100 different palms were used. Only the first sample of each palm captured in the first session was used to make template, and all the corresponding samples captured in the second session were used as a test set. Table 1 shows the recognition rate when the nearest neighbor classifier was used for identification. The number in the parentheses represents the number of incorrect recognitions. The combination of the 2-pixels wide $L_k$ and the 19×19 $R_d$ shows the best result, but the recognition performance is not largely affected by the window size.

**Table 1. Recognition rate for each combination of $L_k$ and $R_d$**

| $R_d$ \ $L_k$ | **11×11** | **15×15** | **19×19** |
|---|---|---|---|
| 2-pixels wide line (14×14) | 97.70 (23) | 98.90 (11) | 99.00 (10) |
| 3-pixels wide line (15×15) | 97.50 (25) | 98.60 (14) | 98.80 (12) |
| 4-pixels wide line (16×16) | 98.00 (20) | 98.80 (12) | 98.70 (13) |

### 4.2 Identification and Verification

To evaluate the recognition accuracy and the scalability of the proposed method, four registration databases ($N$=100, 200, 300, and 386) were prepared. Each database contains $N$ different palms captured in the first session, and all the corresponding samples captured in the second session were used as the test data. The recognition accuracy can be evaluated with two different perspectives of identification and verification. Identification is a one-to-many comparison for finding the owner of a palmprint, whereas verification is a one-to-one comparison for deciding whether the palmprint accords with a stored template or not.

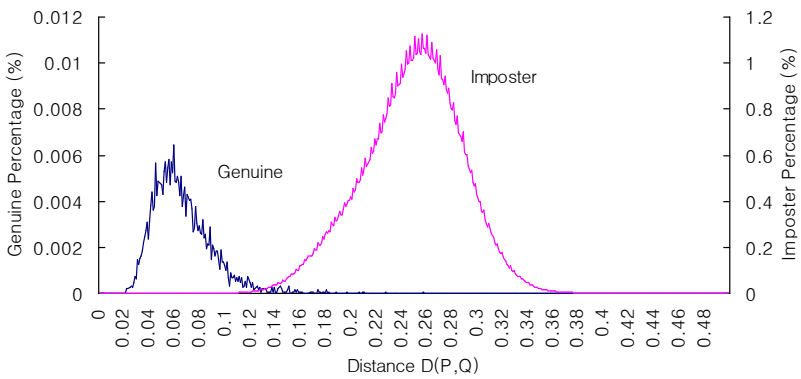
The nearest neighbor classifier was used for identification, and the combination of the 2-pixels wide $L_k$ and the 19×19 $R_d$ was used for this experiment. Table 2 shows the correct recognition rate (CRR) for $N$ palms with a different number of registered templates for each class. The value in the parentheses in the top row is the total number of

samples in the test set. The CRR decreases with increasing *N*, but the declining rate is no larger than 2% in the case of a single template. When the registration databases were extended so as to include two additional samples (i.e., the first three samples are used to make templates), it showed a high recognition rate of 99.4% with *N*=100 and keeps the recognition rate over 98.8% with *N*=386.

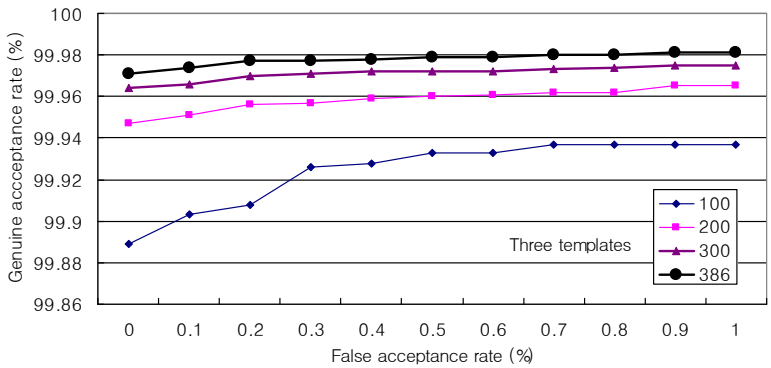
**Table 2. Correct recognition rate for the four registration databases**

| *N* / Training | 100 (1,000) | 200 (1,995) | 300 (3,001) | 386 (3,863) |
|---|---|---|---|---|
| **Single template** | 99.000 | 98.596 | 97.800 | 97.256 |
| **Three templates** | 99.400 | 99.499 | 99.067 | 98.809 |

To examine the verification accuracy, the probability distributions for a genuine match and an imposter are analyzed, which is shown in Fig. 8-(a). There are two distinct peaks in the distributions of the matching distance. One peak located around 0.06 corresponds to genuine matching distance while the other peak located around 0.26 corresponds to the matching distance of imposter. When the false rejection rate (FRR) is zero, the corresponding threshold that separates the genuine group and the imposter group is 0.094. Fig. 8-(b) shows the receiver operating characteristic (ROC) curve for the four databases. Genuine acceptance rate (GAR) is an overall accuracy measurement of a biometric system. It is calculated by 1 – FRR at a specific false acceptance rate (FAR). Contrary to the recognition rate shown in Table 2, the larger *N* shows the higher GAR.




(a)




(b)

Fig. 8. Verification results: (a) genuine and imposter distributions (N=386), (b) receiver operator characteristic curve (N=100, 200, 300, and 386)

Lastly, in order to compare the performance of the proposed and comparison methods, the three line-based coding approaches have been implemented: palmprint orientation code (POC) [3], RLOC [4], BOCV [5]. To compare the features under the same conditions, the preprocessing routine is shared and the same strategy for pixel-to-area matching is used. Three different windows were tested to select a suitable window size for RLOC, as described in section 4.1. The 4-pixel wide line $L_k$ was used for RLOC because it showed the best result. However, something additional for improving the performance such as the enlarging the training set as in [4] was not included. In general, enlarging a training set improves the recognition rate regardless of the methods. Therefore, enlarging a training set only in RLOC is not fair. BOCV-T in Table 3 indicates that the tuned thresholds were applied in BOCV.

The CRR based on the nearest neighbor classifier is used as the metric of the identification accuracy. Table 3 shows the identification results for each method. The number in the parentheses represents the number of misclassification errors. All the methods used three templates for each class. LSOC is the acronym of line and slope orientation code, which is the combined feature proposed in this paper. The proposed method gives the best result. In the last column ($N$=386), the proposed method shows a high improvement in identification performance when compared with the RLOC which uses only line orientation. It reduced the number of misclassification errors to 51% and 44% in the cases of a single template and three templates, respectively.

**Table 3. Comparison of recognition rate (%): (a) single template, (b) three templates**

| | 100 | 200 | 300 | 386 |
|---|---|---|---|---|
| POC | 92.30 (77) | 93.88 (122) | 92.60 (222) | 91.64 (323) |
| RLOC | **99.10 (9)** | 96.04 ( 79) | 95.33 (140) | 94.43 (215) |
| BOCV | 96.90 (31) | 96.34 ( 73) | 95.07 (148) | 94.33 (219) |
| BOCV-T | 97.70 (23) | 96.84 (63) | 95.90 (123) | 95.21 (185) |
| LSOC | 99.00 (10) | **98.60 ( 28)** | **97.80 (66)** | **97.26 (106)** |

(a)

| | 100 | 200 | 300 | 386 |
|---|---|---|---|---|
| POC | 94.40 (56) | 96.19 (76) | 95.77 (127) | 95.78 (163) |
| RLOC | 98.60 (14) | 98.85 (23) | 98.10 (57) | 97.88 (82) |
| BOCV | 98.70 (13) | 99.15 (17) | 98.40 (48) | 98.06 (75) |
| BOCV-T | 98.70 (13) | 99.15 (17) | 98.20 (54) | 98.60 (54) |
| LSOC | **99.40 (6)** | **99.50 (10)** | **99.07 (28)** | **98.81 (46)** |

(b)

There are several performance metrics for biometric systems. FAR and FRR can be adjusted by a matching threshold according to the level of security needed. If you make a system more difficult to enter for an imposter (reducing FAR), you also make the system more difficult to enter for a valid person (i.e., FRR raised). GAR is also dependent on the threshold because it is calculated by 1-FRR at a specific FAR. However, equal error rate (EER) is independent of the threshold because it is the rate where FAR is equal to FRR. Therefore, EER can be used as an application-independent metric. The decidability index *d'* [14] does not use the threshold but uses the statistics of two groups: genuine group and imposter group. So, it is also an application-independent metric. For this reason, EER and the decidability index *d'* were used to compare verification accuracy. EER is the rate where FAR is equal to FRR. For two-choice decision tasks (e.g. genuine versus imposter), the decidability index *d'* measures how well separated the two distributions are [14]. If their two means are $\mu_1$ and $\mu_2$, and their two standard deviations are $\sigma_1$ and $\sigma_2$, then *d'* is defined as

$$d' = \frac{|\mu_1 - \mu_2|}{\sqrt{(\sigma_1^2 + \sigma_2^2)/2}} \tag{13}$$

The comparison results of verification accuracy are described in Table 4. The superiority of verification accuracy is slightly different according to the number of templates

per class. When a single template is used, the decidability index *d'* has the highest scores in the proposed method but the EERs are lowest in BOCV-T. In the case of three templates, BOCV-T shows the best quality. However, the gaps are minor. The differences of EERs between the proposed method and BOCV-T are within the bound of 0.02% when *N*=386.

**Table 4. Comparison of verification accuracy: (a) single template, (b) three templates**

| | **100** | | **200** | | **300** | | **386** | |
|---|---|---|---|---|---|---|---|---|
| | ***d'*** | **EER (%)** | ***d'*** | **EER (%)** | ***d'*** | **EER (%)** | ***d'*** | **EER (%)** |
| **POC** | 4.152 | 0.727 | 4.262 | 0.362 | 4.218 | 0.243 | 4.181 | 0.186 |
| **RLOC** | 4.826 | 0.113 | 4.721 | 0.081 | 4.719 | 0.057 | 4.616 | 0.066 |
| **BOCV** | 4.761 | 0.128 | 4.710 | 0.065 | 4.601 | 0.090 | 4.469 | 0.102 |
| **BOCV-T** | 4.790 | **0.079** | 4.754 | **0.049** | 4.725 | **0.038** | 4.542 | **0.034** |
| **LSOC** | **5.260** | 0.120 | **5.259** | 0.090 | **5.148** | 0.062 | **5.030** | 0.052 |

(a)

| | **100** | | **200** | | **300** | | **386** | |
|---|---|---|---|---|---|---|---|---|
| | ***d'*** | **EER (%)** | ***d'*** | **EER (%)** | ***d'*** | **EER (%)** | ***d'*** | **EER (%)** |
| **POC** | 4.297 | 0.578 | 4.395 | 0.356 | 4.421 | 0.235 | 4.407 | 0.183 |
| **RLOC** | 5.686 | 0.059 | 5.897 | 0.032 | 5.781 | 0.033 | 5.695 | 0.031 |
| **BOCV** | 5.932 | 0.093 | 6.354 | 0.037 | 6.005 | 0.075 | 5.849 | 0.071 |
| **BOCV-T** | **5.935** | **0.023** | **6.390** | **0.013** | **6.181** | **0.013** | **5.995** | **0.012** |
| **LSOC** | 5.624 | 0.111 | 5.756 | 0.051 | 5.641 | 0.035 | 5.580 | 0.029 |

(b)

### 4.3 Speed

The experiments were conducted on a personal computer with an Intel Core 2 Duo CPU (2.2GHz) and 3.5GB RAM. The processing time for preprocessing, feature extraction and matching are listed in Table 5. All the methods share the same preprocessing routine. The proposed method spends 21.17ms and 1.31ms for feature extraction and matching, which is about two times slower than the fastest RLOC method. If the proposed method is applied to the database of *N*=100 with a single template per class, the total processing time for identification is 166.54 ms, which is calculated as 14.37ms + 21.17ms + 1.31ms×100. If *N*=386, 541.2 ms is required for identification. It shows that the speed of the proposed method is fast enough for developing an application. However, if multiple

templates per class are used, some additional efforts for the speedup are required.

**Table 5. Comparison of execution time (unit: ms)**

| | Preprocessing | Feature extraction | Feature matching |
|---|---|---|---|
| **POC** | 14.37 | 16.90 | 0.75 |
| **RLOC** | | 10.53 | 0.75 |
| **BOCV** | | 110.10 | 1.72 |
| **LSOC** | | 21.17 | 1.31 |

## 5. CONCLUSIONS

This study has proposed a new combined approach for palmprint recognition. Unlike the previous orientation-based methods, it uses not only the line orientation information but also the slope orientation information in a local area. Through the experiments using a public database, it is proven that the proposed method is very effective at palmprint recognition. The proposed method shows the best identification results among the several compared methods without sacrificing the verification accuracy. In the case of a single template, it greatly improved the recognition rate and got the best decidability *d'*. Compared with the RLOC which uses only line orientation, it showed superior recognition results. It is fast enough for developing an application and five times faster than the method of BOCV. The proposed approach can be extended to combining other features. Further work will be dedicated to finding the more effective features and combining them.

## ACKNOWLEDGEMENTS

The author would like to thank D. Zhang and his members at the biometric research centre in the Hong Kong Polytechnic University for providing their public palmprint image database.

## REFERENCES

1. A. Kong, D. Zhang, and M. Kamel, "A survey of palmprint recognition." *Pattern Recognition*, Vol. 42, 2009, pp. 1408-1418.

2 A. Kong and D. Zhang, “Competitive Coding Scheme for Palmprint Verification,” in *Proceedings of the 17th International Conference on Pattern Recognition* (*ICPR’04*), Cambridge, UK, August 2004, pp. 520-523.

3. X. Wu, K. Wang, and D. Zhang, “Palmprint Authentication Based on Orientation Code Matching,” *Audio- and Video-Based Biometric Person Authentication* (*ACB-PA’05*), New York, USA, LNCS 3546, 2005, pp. 555-562.

4. W. Jia, D.-S. Huang, and D. Zhang, “Palmprint verification based on robust line orientation code,” *Pattern Recognition*, Vol. 41, 2008, pp. 1504-1513.

5. Z. Guo, D. Zhang, L. Zhang, and W. Zuo, “Palmprint verification using binary orientation co-occurrence vector,” *Pattern Recognition Letters*, Vol. 30, 2009, pp. 1219-1227.

6. G. Lu, D. Zhang, and K. Wang, “Palmprint recognition using eigenpalms features,” *Pattern Recognition Letters*, Vol. 24, 2003, pp. 1463-1467.

7. M. Ekinci and M. Aykut, “Gabor-based kernel PCA for palmprint recognition,” *Electronics Letters*, Vol. 43(20), 2007, pp. 1077-1079.

8. X. Wu, D. Zhang, and K. Wang, “Fisherpalms based palmprint recognition,” *Pattern Recognition Letters*, Vol. 24, 2003, pp. 2829-2838.

9. G.-M. Lu, K.-Q. Wang, and D. Zhang, “Wavelet Based Independent Component Analysis for Palmprint Identification,” in *Proceedings of the 3rd International Conference on Machine Learning and Cybernetics* (*ICMLC*), Alberta, Canada, July 2004, pp. 3547-3550.

10. A. Kumar and H.C. Shen, “Palmprint Identification using PalmCodes,” in *Proceedings of the third International Conference on Image and Graphics* (*ICIG*), Hong Kong, China, December 2004, pp. 258-261.

11. X. Pan and Q.-Q. Ruan, “Palmprint recognition using Gabor-based local invariant features,” *Neurocomputing*, Vol. 72, 2009, pp. 2040-2045.

12. D. Zhang, W.-K. Kong, J. You and M. Wong, “Online Palmprint Identification," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, Vol. 25(9), 2003, pp. 1041-1050.

13. PolyU Palmprint database: available at http://www4.comp.polyu.edu.hk/biometrics/

14. J. Daugman, “The importance of being random: statistical principles of iris recognition,” *Pattern Recognition*, Vol. 36, 2003, pp. 279-291.