# A Contactless Palmprint Recognition Algorithm for Mobile Phones

Shoichiro Aoyama, Koichi Ito and Takafumi Aoki
Graduate School of Information Sciences, Tohoku University
6–6–05, Aramaki Aza Aoba, Sendai-shi 980–8579, Japan
E-mail: aoyama@aoki.ecei.tohoku.ac.jp

Haruki Ota
KDDI R&D Laboratories, Inc.
2-1-15 Ohara, Fujimino-shi 356-8502, Japan

***Abstract*—Mobile phones have become indispensable and powerful digital devices to use many services. To protect mobile devices and stored data, biometric authentication would play an important role. This paper proposes a contactless palmprint recognition algorithm for mobile phones which combines a hand segmentation algorithm using skin-color thresholding and region growing and a palmprint matching algorithm using phase-based correspondence matching with distortion correction. Experimental evaluation using palmprint image databases demonstrates efficient performance of the proposed algorithm compared with conventional algorithms. We also implement the proposed palmprint recognition algorithm on HTC Nexus One.**

***Index Terms*—palmprint recognition, contactless, mobile phones, phase-only correlation**

## I. Introduction

With the rapid growth of mobile computing technology, mobile phones have become indispensable and powerful digital devices to use many services requiring reliable user identification such as electronic account settlement in on-line shopping and banking and to store personal data and financial information. To protect mobile devices and stored data, even if mobile phones are lost or stolen, biometric authentication would play an important role. Fingerprint-based or face-based user authentication has been already embedded in some of mobile phones, but there are problems in practical use. In the case of a fingerprint, a fingerprint sensor must be embedded in a mobile phone. In the case of a face, significant computational resources are required to address facial expression changes, pose variations, etc. On the other hand, a palmprint can be easily taken by a built-in camera of mobile phone and exhibits reliable authentication as well as a fingerprint [1].

Various palmprint recognition algorithms have been proposed in the literatures [2]. Almost all algorithms assume that a hand is fixed on a system when a palmprint image is captured. Hence, a central region of a palm, which is a Region Of Interest (ROI) used in the matching process, can be easily extracted by using simple pixel-value thresholding. The palmprint recognition system for mobile devices has been proposed by Han, et al. [3]. In this system, the users must adjust the position of their hand to the frame during hand image capturing in order to capture the images with little rotation, translation and scale variation. It may result in inconvenience for users. Also, the coding-based matching algorithms [4]–[6] have been successfully applied, since extracted ROIs contain only minute displacements. In practical situation, a simple thresholding approach cannot be applied to extract ROIs, since background does not have a uniform color, and the recognition performance of matching algorithms may drop significantly, since ROIs are deformed by variations in hand pose.

In order to address the above problems and to effectively utilize the limited computational resources of mobile phones, this paper proposes a novel contactless palmprint recognition algorithm which consists of the preprocessing and matching steps. The preprocessing step extracts a hand from the input image using skin-color thresholding and region growing, detects keypoints [7], and extracts a ROI [4]. The matching step normalizes affine transformation between ROIs according to the correspondence between ROIs obtained by using phase-based correspondence matching [8] and then calculates the matching score. Experimental evaluation using palmprint image databases demonstrates efficient performance of the proposed algorithm compared with conventional algorithms. We also implement the proposed palmprint recognition algorithm on HTC Nexus One.

## II. Algorithm Overview

The proposed algorithm assumes that the users capture their left hand turned sideways with the mobile phone camera. To support users to capture their palmprint images, we prepare the guide as shown in Fig. 1. So, the users can be encouraged to roughly fit their palm within the red box. The captured images have different scale, rotation, translation and distortion due to variations in hand pose. The proposed algorithm consists of 2 stages: (i) enrollment stage and (ii) matching stage as shown in Fig. 2. In the enrollment stage, the preprocessing is performed to extract ROIs from the registered image captured by mobile phone camera and the extracted ROIs are stored in the database. In the matching stage, the preprocessing is performed to extract a ROI from the input image, and then the matching score between the input and registered ROIs is calculated by the proposed matching algorithm. The following sections describe the details of the preprocessing step and the matching step.

## III. Preprocessing

This section describes the preprocessing algorithm to extract a ROI. The proposed algorithm assumes that the users capture

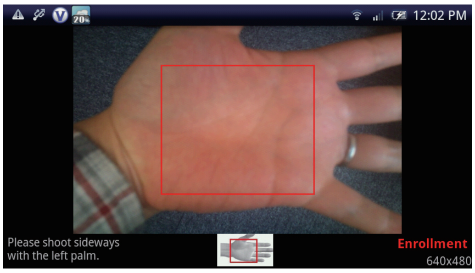


Fig. 1. Guide for capturing the palmrpint image.

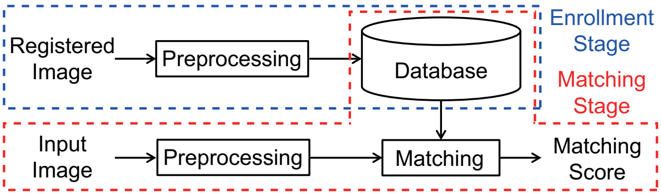


Fig. 2. Flow diagram of the proposed algorithm.

their left hand turned sideways and the size of captured images is $640 \times 480$ pixels as shown in Fig. 3 (a).

In general, the procedure for preprocessing consists of 5 steps: (i) binarizing the images, (ii) extracting contour of hand, (iii) detecting keypoints, (iv) establishing a coordinate system and (v) extracting the ROI [9]. In the case of images captured by the mobile phone camera, the background of images does not usually have a uniform color, so the accurate binarization method to segment a hand is the most important for preprocessing. Using the limited computational resources of mobile phones, we employ skin-color thresholding and region growing [10] to segment a hand from the image with complex background. The proposed preprocessing consists of 5 steps: (i) cropping and shrinking the image, (ii) binarizing the image, (iii) calculating the radial distance function, (iv) detecting keypoints and (v) extracting the ROI. Fig. 3 shows the overview of the preprocessing. The followings describe detailed process of each step.

**(i) Image cropping and shrinking**

In order to extract ROI, the keypoints located at the bottom of gaps between index and middle fingers and between ring and little fingers have to be extracted [4]. The keypoints are located at a right half of the captured image, since the user captures his/her left hand turned sideways as shown in Fig. 3 (a). To reduce the computational cost, we extract the right half of the image and shrink it by 50% as shown in Fig. 3 (b).

**(ii) Binarization**

This step is to segment a hand by using image binarization as shown in Fig. 3 (c). The proposed binarization method employs skin-color thresholding and region growing. We convert the RGB color space to the HSV color space for skin-color thresholding. Through a learning process, we make a lookup table in the H–S color space to detect pixels having skin color. In this paper, we use 373 hand images in the learning process,

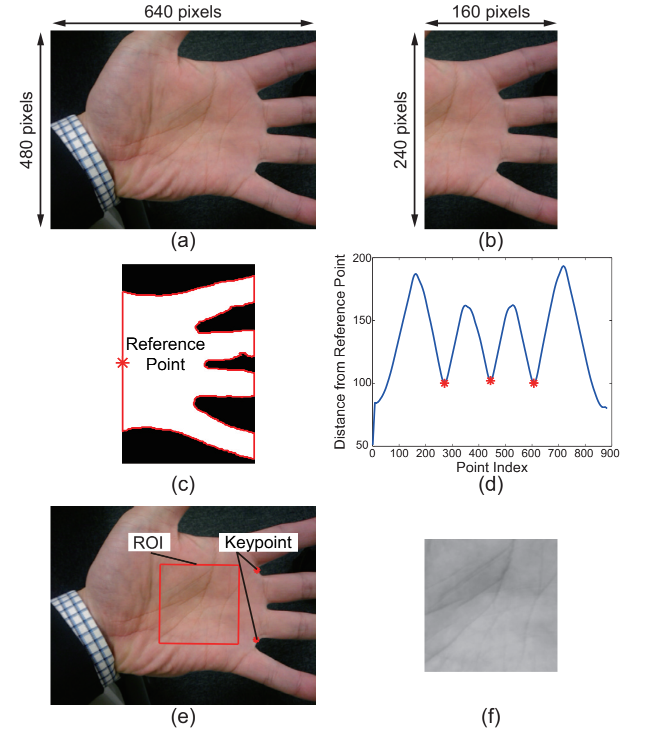


Fig. 3. Example of preprocessing: (a) image captured by mobile phone camera, (b) image used to detect keypoints, (c) binarized image and extracted contour, (d) radial distance function, (e) detected keypoints and ROI and (f) extracted ROI.

which do not include the image set in experiments. Applying the lookup table to the whole image, we cannot segment only a hand from the image due to changes in the environment such as lighting, background, etc. Therefore, the skin-color thresholding is applied only to the small region in the center of left side as shown in Fig. 4 (b). And then, we apply the region growing [10] to segment the whole hand, where the region growing starts from the extracted small region. Figs. 4 (c)–(g) show examples of the progress of region growing.

**(iii) Radial distance function**

This step is to calculate the radial distance function [7]. At first, we obtain boundaries of the binarized image using a boundary tracking algorithm. Next, we calculate the distance from the reference point located at the center of left side (Fig. 3 (c)) to all other points to obtain the radial distance function as shown in Fig. 3 (d). The 3 minima of the radial distance function, which are indicated by "$*$" in Fig. 3 (d), correspond to the bottom of gaps between fingers.

**(iv) Keypoint detection**

By finding the minima of the radial distance function, the bottom of gaps between fingers can be detected as keypoints. However, depending on the segmentation results, such as Fig. 5 (b), there are more than 3 minima including wrong keypoints as shown in Figs. 5 (c) and (d). If more than 3 minima are detected, we use the relationship among valley points between index–middle fingers, middle–ring fingers and ring–little fingers to remove wrong keypoints. We select 3

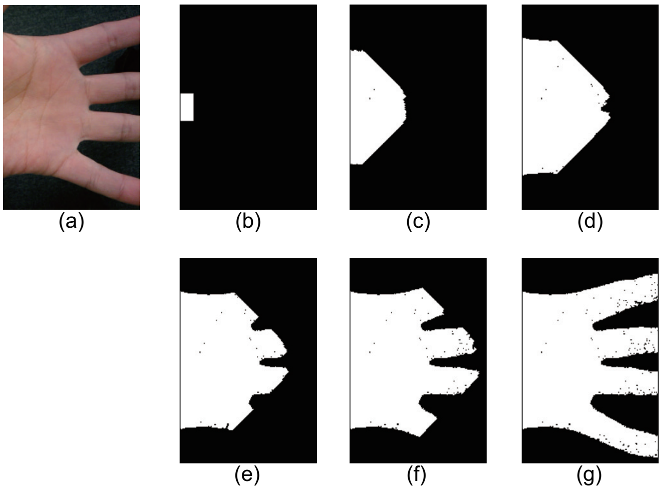


Fig. 4. Example of binarization: (a) input image for binarization, (b) initial region extraction using skin-color thresholding, and (c)–(g) binarization using region growing to segment a hand.

points from the detected minima of the radial distance function and make a triangle. If the selected 3 points correspond to correct keypoints, the triangle created by selected 3 points should be an isosceles triangle. Among all the combinations of 3 points, we select the combination which structure is the most similar to an isosceles triangle as keypoints.

**(v) ROI extraction**

This step is to extract the ROI using the position of detected keypoints. We obtain the perpendicular bisector of the line segment between valley points of index–middle fingers and ring–little fingers to determine the centroid of the ROI. We extract the ROI of fixed size as shown in Fig. 3 (e), which is centered at the centroid, to normalize the scaling of ROI. The upper and lower hems of the ROI are parallel to the perpendicular bisector of the line segment between keypoints to normalize the rotation of ROI [4]. Fig. 3 (f) shows the extracted ROI. In this paper, ROI is $160 \times 160$ pixels with grayscale color.

## IV. Matching

This section describes the matching algorithm to calculate the matching score between ROIs. One of the most successful palmprint matching algorithms is feature-based matching algorithms such as Competitive Code [5] and Ordinal Code [6]. The feature-based matching algorithms assume that scale, rotation and translation between ROIs have been already normalized by preprocessing. In the case of images captured by mobile phone camera, the recognition performance of feature-based matching algorithms may drop significantly, since ROIs are deformed by variations in hand pose. We correct the affine transformation between ROIs according to the result of phase-based correspondence matching [8] and then calculate the matching score. The proposed matching algorithm consists of 2 steps: (i) geometric correction and (ii) matching score calculation. The followings describe details of each step.

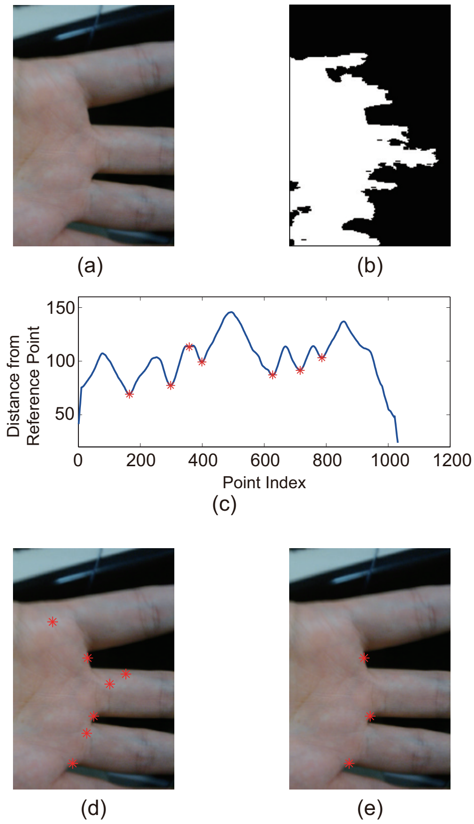


Fig. 5. Valley point detection: (a) input image, (b) binarized image, (c) radial distance function and detected minima, (d) detected minima on input image and (e) correct keypints on input image.

**(i) Geometric correction**

This step is to correct the affine transformation between ROIs. At first, we set the 16 reference points on the registered ROI and obtain the correspondence points on the input ROI using phase-based correspondence matching [8]. To estimate parameters of affine transformation, we use a set of corresponding point pairs which similarity is over threshold in Fig. 6 (a). Fig. 6 (b) shows the ROIs after geometric correction.

**(ii) Matching score calculation**

Although the global transformation between ROIs can be corrected in the previous step, nonlinear distortion between ROIs still remains. In the proposed algorithm, we employ Band-Limited Phase-Only Correlation (BLPOC) [11] to calculate the matching score taking account of nonlinear distortion. First, we extract 9 blocks with $32 \times 32$ pixels as shown in blue boxes of Fig. 6 (b) and estimate minute translations between each block pair by using BLPOC. Next, we extract local blocks of the registered ROI which centroids are displaced by the estimated minute translations as shown in red boxes of Fig. 6 (b). Then, we calculate BLPOC function between each block pair, take the average of a set of BLPOC functions and obtain the matching score as the highest peak value of the average BLPOC function.

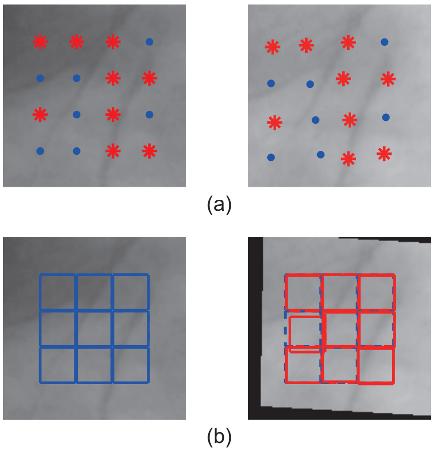


Fig. 6. Example of palmprint matching: (a) result of corresponding matching where $*$ indicates the reliable corresponding point pairs and (b) extracted local blocks for matching score calculation.

TABLE I
ACCURACY OF PREPROCESSING FOR ALGORITHMS (A) AND (B).

| Algorithm | Extracted ROIs | Correct ROIs | Stability |
|---|---|---|---|
| (A) | 556 (92.7%) | 483 (80.5%) | 86.9% |
| (B) | 559 (93.2%) | 520 (86.7%) | 93.0% |

## V. EXPERIMENTS AND DISCUSSION

This section describes experiments to evaluate performance of the proposed algorithm.

First, the performance of the preprocessing stage is evaluated using our palmprint database where the palmprint images are captured by an HTC Android DevPhone2. We do not give any instructions to subjects when subjects capture their left hand. Thus, the images in the database are captured under practical situation. Our database consists of 600 images with 30 subjects and 20 different images of each subject. The interval between first and second 10 images is about one week. To demonstrate effectiveness of the proposed algorithm, we compare 2 algorithms: (A) the preprocessing algorithm with only skin-color thresholding and (B) the preprocessing algorithm with skin-color thresholding and region growing. In the algorithm (A), we apply skin-color thresholding to the whole image to segment a hand. Fig. 7 shows examples of extracted ROIs using the algorithms (A) and (B). Table I shows the summary of experimental results for algorithms (A) and (B). "Correct ROIs" indicates the number of ROIs after manually removing wrong ROIs. "Stability" indicates the ratio of (Correct ROIs) / (Extracted ROIs). As shown in Fig. 7, the algorithm (B) exhibits more stable results than the algorithm (A). The failure cases of the algorithm (B) are mostly due to background including skin-like color such as hands of other persons, carpet, cardboard, etc.

Next, the perfomance of the matching stage is evaluated using the public palmprint databases and our database. In this experiment, we use PolyU palmprint database ver.2 [12], and CASIA palmprint database [13]. The PolyU palmprint database ver.2 (PolyU) consists of 7,752 contact palmprint images with 386 subjects and about 20 different images of each subject. The CASIA palmprint database (CASIA) consists of 5,239 contactless palmprint images with 301 subjects and about 8 different images of each left and right palm. Our database (DevPhone) consists of 520 images with 30 subjects and about 20 different images of each subject, where we use only images from which we can extract correct ROIs in the above experiment. The numbers of genuine pairs are 74,068 for PolyU, 20,584 for CASIA and 4,346 for DevPhone, while the numbers of impostor pairs are 29,968,808 for PolyU, 13,700,357 for CASIA and 130,594 for DevPhone. The performance is evaluated by the Equal Error Rate (EER) which is defined as the error rate where false reject rate and false accept rate are equal. In this experiment, we compare 4 different algorithms: CompCode [5], OrdiCode [6], BLPOC [11] and the proposed algorithm. Table II shows the summary of EERs for each database. The proposed algorithm shows highest performance under both fixed and unconstrained conditions. Therefore, the proposed matching algorithm is more suitable for practical situation than the conventional algorithms.

We implement the proposed algorithm on an HTC Nexus One (CPU: Qualcomm Snapdragon QSD8250 (1GHz), RAM: 512MB, OS: Android 2.3.4, Camera: 5M resolution with autofocus). The computation time for preprocessing and matching is about 0.6sec and about 0.4sec, respectively. It indicates that the proposed algorithm is suitable for the practical use in the mobile phones.

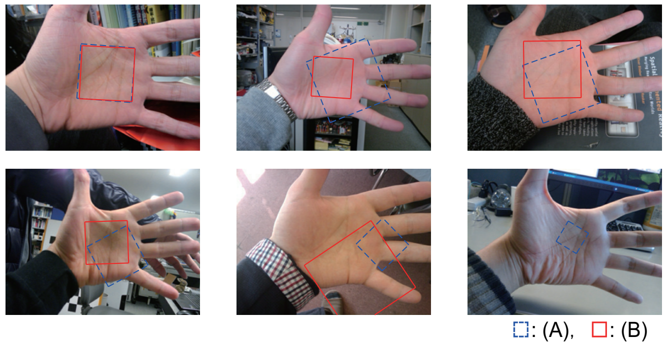


Fig. 7. Examples of extracted ROIs (the blue and red boxes indicate the results of algorithms (A) and (B), respectively.).

TABLE II
EERS [%] OF THE PALMPRINT MATCHING ALGORITHMS FOR EACH DATABASE.

| Algorithm | PolyU | CASIA | DevPhone |
|---|---|---|---|
| CompCode [5] | 0.805 | 8.102 | 8.405 |
| OrdiCode [6] | 0.622 | 5.879 | 8.169 |
| BLPOC [11] | 0.425 | 6.752 | 14.496 |
| Proposed | 0.051 | 0.488 | 4.072 |

## VI. Conclusion

This paper has proposed a contactless palmprint recognition algorithm for mobile phones. Through a set of experiments using palmprint image databases, the proposed algorithm exhibits efficient performance compared with the conventional algorithms. In future work, we will implement the proposed algorithm on various mobile phones and perform a large-scale experiment to work toward practical use of the proposed algorithm.

## References


[1] D. Zhang, *Palmprint Authentication*. Kluwer Academic Publication, 2004.

[2] D. Zhang, W. Zuo, and F. Yue, "A comparative study of palmprint recognition algorithms," *ACM Computing Surveys*, vol. 44, no. 1, Jan. 2012.

[3] Y. Han, T. Tan, Z. Sun, and Y. Hao, "Embedded palmprint recognition system on mobile devices," *Lecture Notes in Computer Science (ICB2007)*, vol. 4642, pp. 1184–1193, Aug. 2007.

[4] D. Zhang, W. Kong, J. You, and M. Wong, "Online palmprint identification," *IEEE Trans. Pattern Analysis and Machine Intelligence*, vol. 25, no. 9, pp. 1041–1050, Sep. 2003.

[5] A. Kong and D. Zhang, "Competitive coding scheme for palmprint verification," *Proc. Int'l Conf. Pattern Recognition*, vol. 1, pp. 520–523, Dec. 2004.

[6] Z. Sun, T. Tan, Y. Wang, and S. Li, "Ordinal palmprint represention for personal identification," *Proc. IEEE Computer Society Conf. Computer Vision and Pattern Recognition*, vol. 1, pp. 279–284, Jun. 2005.

[7] E. Yörük, E. Konukoğlu, B. Sankur, and J. Darbon, "Shape-based hand recognition," *IEEE Trans. Image Processing*, vol. 15, no. 7, pp. 1803–1815, Jul. 2006.

[8] K. Takita, M. A. Muquit, T. Aoki, and T. Higuchi, "A sub-pixel correspondence search technique for computer vision applications," *IEICE Trans. Fundamentals*, vol. E87-A, no. 8, pp. 1913–1923, Aug. 2004.

[9] A. Kong, D. Zhang, and M. Kamel, "A survey of palmprint recognition," *Pattern Recognition*, vol. 42, no. 7, pp. 1408–1418, Jan. 2009.

[10] R. C. Gonzalez and R. E. Woods, *Digital Image Processing*. Pearson Education, 1992.

[11] S. Iitsuka, K. Ito, and T. Aoki, "A practical palmprint recognition algorithm using phase information," *Proc. Int'l Conf. Pattern Recognition*, no. WeBCT9.18, pp. 1–4, Dec. 2008.

[12] "PolyU palmprint database," http://www4.comp.polyu.edu.hk/~biometrics/.

[13] "CASIA palmprint database," http://www.cbsr.ia.ac.cn/english/PalmprintDatabases.asp.