Palmprint Recognition System with Double-assistant-point on iOS Mobile Devices

Fumeng Gao[1, 2]
Lu Leng[1, 2]
leng@nchu.edu.cn
Jiexian Zeng[1, 2]

[1]School of Software, Nanchang Hangkong University, Nanchang, 330063, P. R. China
[2]Key Laboratory of Jiangxi Province for Image Processing and Pattern Recognition, Nanchang Hangkong University, Nanchang, 330063, P. R. China

Abstract

The security of current authentication technologies on mobile devices, such as smartphone, are not ideal. Palmprint recognition has been developed rapidly thanks to its multi-fold strengths. However, due to the various severe interference factors in contactless collection, palmprint recognition on smartphone has not yet reached the requirements of practical application. In this paper, a palmprint recognition system with the assistance acquisition and localization of "double-assistant-point" (DAP) was developed on iOS mobile devices, like iPhone, iPad, iPod, etc. DAP helped users place their hands at appropriate position with correct gesture during acquisition. The palmprint image was rotated so that the line connecting two valley points was horizontal, and then the coordinate system was established by virtue of two valley points between fingers. The region of interest was cropped for feature extraction and matching. The developed system has good real-time property and strong robustness of palmprint preprocessing against interference factors.

1 Introduction

The security of current authorization technologies on mobile devices are not completely satisfactory. Both numeric passwords and graphics passwords are easy to be observed and stolen. Moreover, simply and short passwords are easily memorized, but are also easily cracked. On the contrary, complex passwords are likely forgotten.

Biometrics, which has high reliability and security, identifies or verifies users based on their physiological or behavioral characteristics directly. Unlike possessions and passwords, biometric features are hardly lost, forgotten and cracked, so biometrics overcomes the defects of traditional authentication methods.

As one signification biometric modality, palmprint recognition has been more development potential thanks to high accuracy, low cost, high user acceptance and so on. Online palmprint recognition can be categorized into contact and contactless modes [1]. In contact mode, users' hands and equipment surface are contacted. On the contrary, in contactless mode, users' hands do not need to contact any equipment surface.

The main task of preprocessing for palmprint recognition is to crop Region of Interest (ROI) from original image. In most contact palmprint preprocessing schemes, two valley points are considered as key points, which are between index finger and middle finger as well as ring finger and little finger. Coordinate system is established according to the two key points and a square region of palm is cropped as ROI [2,3].

Although contact palmprint recognition systems can achieve high accuracy performance, their practical applications incur several problems, including personal hygiene, lack of acquisition flexibility, surface contamination, resistance of traditional cultures, etc [4]. Therefore, contactless palmprint recognition technology has attracted more and more researchers . Unfortunately, it is impractical to transplant the contact preprocessing methods onto contactless palmprint systems directly due to the severe challenges, including uncontrollable hand location and gesture, complex background, variant illumination , etc.

In 2008, Michael et *al.* proposed a competitive hand valley detection method [5]. Gaussian skin color model was used to segment palm region. Then the palm boundary was detected and all the boundary points were judged according to some rules. Finally, four valley points were found. This method heavily depends on the segmentation results of skin color. In 2011, Tang et *al.* also designed a contactless palmprint key-point localization method that are suitable to both open fingers and close fingers [6]. In 2014, Leng et *al.* proposed triple-perpendicular-translation residuals for key-point localization with low computational complexity [7]. In 2015, Aykut and Ekinci applied Active Appearance Model for palm detection [8]; however the computation complexity is high. In 2016, Javidnia developed an efficient light normalization algorithm to overcome illumination disturbance [9].

Palmprint images can be captured directly with the built-in cameras of mobile device, palms do not need to contact the equipment surface, so mobile mode can be considered as one special case of contactless mode [10].

iOS is the mobile operating system created and developed by Apple Inc. exclusively for its hardware. It is the operating system that presently powers many mobile devices of Apple company, including iPhone, iPad, iPod Touch, etc. It is the second most popular mobile operating system globally after Android.

Compared with the current biometric modalities used on iOS mobile devices, such as fingerprint, face, palmprint has several previously mentioned specific advantages; unfortunately, due to the various severe interference factors, palmprint recognition on the mobile devices with iOS platform has not yet reached the requirements of practical application. Owing to the particularity of the acquisition on mobile devices, palmprint recognition suffers from more severe technical challenges, including uncontrollable hand location and gesture, extremely complex background, remarkably variant illumination, relatively limited computation capacity and storage space, etc.

We developed a palmprint recognition system with “double-assistant-point” (DAP) assistant scheme on iOS mobile devices. All kinds of classes and threads were programmed in Object C++ language. Finally the software and hardware were jointly debugged. Assistant points help users place their hands at appropriate location with correct gesture during palmprint image acquisition. The restriction and constraint drastically enhance the robustness of palmprint preprocessing against the severe interference factors. Moreover, real-time ability is remarkably improved. The sufficient test results confirm the effectiveness and practicability of the developed palmprint verification system.

2 Methodology

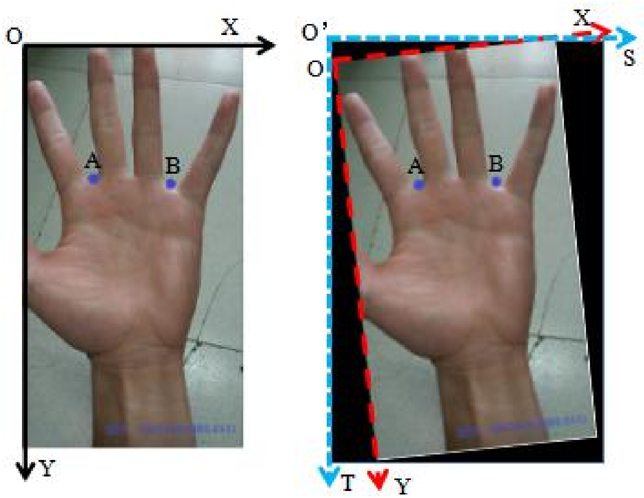
2.1 Dual-assistant-point technique

The centers of the two crosshairs in the preview screen form the two assistant points. The forward directions of X-axis and Y-axis in the image coordinate are rightward and downward, respectively. The screen height and width are *h* and *w*. For example, assume *W*<*H* on iPhone, then the positions of the left and right assistant points A and B for left hand are:

$$(x_A, y_A)=(w/3, h/3). \quad \textbf{(1)}$$

$$(x_B, y_B)=(2w/3, h/3+\tan\theta\times w/3). \quad \textbf{(2)}$$

Similarly, the assistant points for right hand are the vertical mirror of the points for left hand. When palmprint images are captured, the users should try to align their two key points to the assistant points, as shown in Fig. 1(a) for left hand where $\theta=\pi/36$. Actually, the effective distance for checking palmprint is determined if the two key points are placed on the assistant points.



(a) (b)

Figure 1: Palmprint image acquisition and rotation: (a) acquisition interface; (b) rotated image.

2.2 Rotation

The palmprint image is anticlockwise rotated around the center of the image, so that the line connecting the two valley points A and B is horizontal, as shown in Fig. 1(b). The coordinate system with S-axis and T-axis is established by virtue of the two valley points. The height and width of the rotated image are *H* and *W*, respectively.

$$H=h\cos\theta+w\sin\theta \quad \textbf{(3)}$$

$$W=w\cos\theta+h\sin\theta \quad \textbf{(4)}$$

The coordinate transformation formulas are:

$$s\text{-}W/2=(x\text{-}w/2)\cos\theta\text{-}(y\text{-}h/2)\sin\theta \quad (5)$$

$$s\text{-}H/2=(x\text{-}w/2)\sin\theta+(y\text{-}h/2)\mathrm{con}\theta \quad (6)$$

The positions of A and B in SO'T coordinate system are:

$$\begin{aligned} s_A &= \left(x_A - w/2\right)\cos\theta - \left(y_A - h/2\right)\sin\theta + W/2 \\ &= \left(w/3 - w/2\right)\cos\theta - \left(h/3 - h/2\right)\sin\theta + W/2 \\ &= -\cos\theta \bullet w/6 + \sin\theta \bullet h/6 + \cos\theta \bullet w/2 + \sin\theta \bullet h/2 \\ &= \cos\theta \bullet w/3 + \sin\theta \bullet 2h/3 \end{aligned} \quad (7)$$

$$\begin{aligned} t_A &= \left(x_A - w/2\right)\sin\theta + \left(y_A - h/2\right)\cos\theta + H/2 \\ &= \left(w/3 - w/2\right)\sin\theta + \left(h/3 - h/2\right)\cos\theta + H/2 \\ &= -\sin\theta \bullet w/6 - \cos\theta \bullet h/6 + \cos\theta \bullet h/2 + \sin\theta \bullet w/2 \\ &= \sin\theta \bullet h/3 + \cos\theta \bullet h/3 \end{aligned} \quad (8)$$

$$\begin{aligned} s_B &= \left(x_B - w/2\right)\cos\theta - \left(y_B - h/2\right)\sin\theta + W/2 \\ &= \left(2w/3 - w/2\right)\cos\theta \\ &\qquad - \left(h/3 - \tan\theta \bullet w/3 - h/2\right)\sin\theta + W/2 \\ &= \cos\theta \bullet w/6 + \sin\theta \bullet h/6 + \tan\theta \sin\theta \bullet w/3 \\ &\qquad + \cos\theta \bullet w/2 + \sin\theta \bullet h/2 \\ &= \cos\theta \bullet 2w/3 + \sin\theta \bullet 2h/3 + \tan\theta \sin\theta \bullet w/3 \end{aligned} \quad (9)$$

$$\begin{aligned} t_B &= \left(x_B - w/2\right)\sin\theta + \left(y_B - h/2\right)\cos\theta + H/2 \\ &= \left(2w/3 - w/2\right)\sin\theta \\ &\qquad + \left(h/3 - \tan\theta \bullet w/3 - h/2\right)\cos\theta + H/2 \\ &= \sin\theta \bullet w/6 - \cos\theta \bullet h/6 - \sin\theta \bullet w/3 \\ &\qquad + \cos\theta \bullet h/2 + \sin\theta \bullet w/2 \\ &= \sin\theta \bullet h/3 + \cos\theta \bullet h/3 \end{aligned} \quad (10)$$

2.3 ROI cropping

As shown in Fig. 2, C is the midpoint of AB. CD is perpendicular to AB. CD=$\alpha\times$AB. The square is ROI, and D is the midpoint of one edge of the square. The length of the edge is $\beta\times$AB. The ROI is resized to 128×128. α and β are the coefficients to control the length of CD and the ROI edge, respectively.

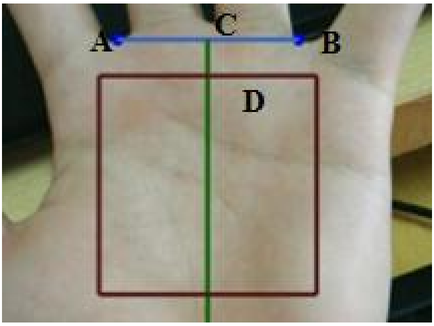


Figure 2: ROI cropping.

2.4 Feature extraction and matching

Many alternatives, including Palm Code, Fusion Code, Competitive Code, Ordinal Code, Robust Line Orientation Code, Binary Orientation Co-occurrence Vector (BOCV), Extended Binary Orientation Co-occurrence Vector, etc., are classical palmprint texture codes for verification. The computational complexity of Palm Code is low, but its accuracy is not high. BOCV fuses the matching scores along six orientations, so it can achieve high accuracy compared with the previous methods including Palm Code. Speed and accuracy should be both considered in the choice of palmprint texture codes. The dissimilarity between two templates is measured with normalized distance. The feature extraction and matching of Palm Code and BOCV are popular and well know, more details can be found in [11].

3 System design and implementation

3.1 Performance requirements

Table 1 summarizes the critical development environments.

Item	Description
Operating	Apple Mac OS X El Capitan (v12.11)
IDE	Xcode 10.3
Programming	Objective-C, C++, C
Framework	Cocoa Touch, AVFoundation, CoreGrahics,
Development	MacBook Pro (13-inch, Mid 2015)
Test device	iPhone 6S-Plus, iPad Air 2

Table 1: Development environments.

3.2 Class description

Modular and Model-View-Controller (MVC) are employed for the implementation of the iOS platform. The functions are implemented by the following classes:

(1) PPCameraDelegate: Similar to inheritance, the Swift protocol defines the methods of an instance of a class that conforms to the protocol. Protocols only define the methods and do not implement them, but classes that conform to the protocol must implement the methods defined in the protocol.

(2) PPCameraCapture: As an interface of other classes accessing to cameras, this camera control class conforms to the PPCameraDelegate protocol. Through the API provided by the Apple AV Foundation framework, we implement the functions controlling camera, including switch between camera on and off, switch of flash status, switch between front and rear cameras, and the mandatory proxy methods specified in the custom PPCameraDelegate protocol.

(3) ROICropper: It is a class for ROI cropping. With the series of APIs provided by the CoreGraphics framework, we realize the functions of rotating, cropping, scaling.

(4) PalmprintVerificater: It is a class for palmprint matching. The two inputs of the instance of this class are a stored gallery palmprint template and a query palmprint template. The similarity/dissimilarity between them are computed.

(5) mainController: The view control class corresponding to the main application interface is responsible for responding to all interactions of the user on the main interface.

(6) CameraBaseViewController: This virtual base class defines the basic functions of two view controller classes for palmprint acquisition and palmprint matching, including view control, calling the camera via PPCameraCapture, and responding to user interaction.

(7) CaptureViewController: This view controller class of enrollment interface, inherited from CameraBaseViewController, is fine-tuned for acquisition enrollment.

(8) VerificationViewController: The view controller class of verification interface, inherits from CameraBaseViewController, is fine-tuned for verification.

(9) PopoverController: This view controller class of pop-up window is used to display help information about how to use this palmprint recognition system.

Fig. 3 shows the call relation and inheritance of class.

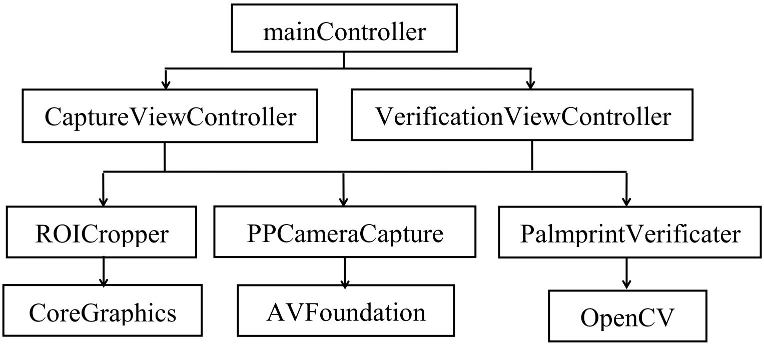


Figure 3: Call relation of class.

4 Experimental results and discussions

Fig. 4 shows the verification results of genuine matching and imposter matching.

(a) (b)

Figure 4: Verification results: (a) genuine matching; (b) imposter matching.

The palmprint images of 900 palms of 450 users were captured with iPhone6s Plus and iPad Air2. We acquired 5 images of each palm are , so there are totally 4500 images in the database. PalmCode along different orientations and BOCV are tested on the developed system. Table 2, the recognition accuracy of equal error rate (EER), shows that BOCV outperform PalmCode along single orientation.

PalmCode							BOCV
orientations	0	$\pi/6$	$\pi/3$	$\pi/2$	$2\pi/3$	$5\pi/6$	
EER(%)	2.8	2.5	2.7	3.0	2.6	2.5	2.4

Table 2: Recognition accuracy.

5 Conclusions and future works

A palmprint recognition system with "double-assistant-point" (DAP) was developed on iOS mobile devices. which helped users place their hands at appropriate position with correct gesture during acquisition. This assistance scheme for acquisition and localization enhances real-time property and strong robustness of preprocessing against interference factors. However, assistance techniques heavily rely on the exactitude and proficiency of user operations. In future works, we will design accurately key-point detection algorithm to further improve the recognition accuracy and relieve the restriction and constraint, so the users' comfort will be promoted. In addition, the feature extraction and matching results will be improved by combining several methods.

References

[1] W. Jia, B. Zhang, J. T. Lu, Y. H. Zhu, Y. Zhao, W. M. Zuo, and H. B. Ling. Palmprint recognition based on complete direction representation. *IEEE Transactions on Image Processing*, 26(9):4483–4498, 2017.

[2] L. Zhang, L. D. Li, A. Q. Yang, Y. Shen, and M. Yang. Towards contactless palmprint recognition: a novel device, a new benchmark, and a collaborative representation based identification approach. *Pattern Recognition*, 69:199–212, 2017.

[3] L. Leng, M. Li, C. S. Kim, and X Bi. Dual-source discrimination power analysis for multi-instance contactless palmprint recognition. *Multimedia Tools and Applications*, 76(1):333–354, 2017.

[4] L. Leng, M. Li, and A. B. J. Teoh. Conjugate 2DPalmHash code for secure palm-print-vein verification. *6th International Congress on Image and Signal Processing*:1694–1699, 2013.

[5] M. G. K. Ong, T. Connie, and A. B. J. Teoh. Touch-less palmprint biometrics: novel design and implementation. *Image and Vision Computing*, 26(12):1551–1560, 2008.

[6] Y. B. Tang, W. Bu, X. Q. Wu, and K. Q. Wang. Key points localization methods of contactless palmprint recognition technology. *Intelligent Computer and Applications*, 1(1):62–65, 2011.

[7] L. Leng, G. Liu, M. Li, and M. K. Khan. Logical conjunction of triple-perpendicular-directional translation residual for contactless palmprint preprocessing. *11th International Conference on Information Technology: New Generations*:523–528, 2014.

[8] M. Aykut and M. Ekinci. Developing a contactless palmprint authentication system by introducing a novel ROI extraction method. *Image and Vision Computing*, 40:65–74, 2015.

[9] H. Javidnia, A. Ungureanu, C. Costache, and P. Corcoran. Palmprint as a smartphone biometric. *IEEE International Conference on Consumer Electronics*:463–466, 2016.

[10] L. Leng, F. M. Gao, Q. Chen, and C. S. Kim. Palmprint recognition system on mobile devices with double-line-single-point assistance. *Personal and Ubiquitous Computing*, 22(1)93–104, 2018.

[11] L. Leng, A. B. J. Teoh, and M. Li. Simplified 2DPalmHash code for secure palmprint verification. *Multimedia Tools and Applications*, 76(6)8373–8398, 2017.