Image and Vision Computing 26 (2008) 1551–1560

Contents lists available at ScienceDirect

Image and Vision Computing

journal homepage: www.elsevier.com/locate/imavis

Review article

Touch-less palm print biometrics: Novel design and implementation

Goh Kah Ong Michael [a], Tee Connie [a,*], Andrew Beng Jin Teoh [b]

[a] Multimedia University, Faculty of Information Science and Technology, Jalan Ayer Keroh Lama, 75450 Melaka, Malaysia
[b] School of Electrical and Electronics Engineering, College of Engineering, Yonsei University, Seoul, Republic of Korea

ARTICLE INFO

Article history:
Received 4 October 2007
Received in revised form 17 June 2008
Accepted 29 June 2008

Keywords:
Palm print recognition
Touch-less biometrics
Local binary pattern (LBP)
Gradient operator
Probabilistic neural networks (PNN)

ABSTRACT

In this paper, we propose an innovative touch-less palm print recognition system. This project is motivated by the public's demand for non-invasive and hygienic biometric technology. For various reasons, users are concerned about touching the biometric scanners. Therefore, we propose to use a low-resolution web camera to capture the user's hand at a distance for recognition. The users do not need to touch any device for their palm print to be acquired. A novel hand tracking and palm print region of interest (ROI) extraction technique are used to track and capture the user's palm in real-time video stream. The discriminative palm print features are extracted based on a new method that applies local binary pattern (LBP) texture descriptor on the palm print directional gradient responses. Experiments show promising result using the proposed method. Performance can be further improved when a modified probabilistic neural network (PNN) is used for feature matching. Verification can be performed in less than one second in the proposed system.

1. Introduction

Palm print recognition is a biometric technology which recognizes a person based on his/her palm print pattern. Palm print serves as a reliable human identifier because the print patterns are not duplicated in other people, even in monozygotic twins. More importantly, the details of these ridges are permanent. The ridge structures are formed at about 13th weeks of the human embryonic development and are completed by about 18th week [1]. The formation remains unchanged from that time on throughout life except for size. After death, decomposition of the skin is last to occur in the area of the palm print. Compared with the other physical biometric characteristics, palm print authentication has several advantages: low-resolution imaging, low-intrusiveness, stable line features, and low-cost capturing device. Palm print covers wider area than fingerprint and it contains abundance of useful information for recognition. Apart from the friction ridges, the three principal lines (the dominant lines on the palm) and wrinkles (the weaker and more irregular lines) on the palm can also be used for recognition. In addition, palm print system does not require very high resolution capturing device as the principal lines and wrinkles can be observed under low-resolution images (for example, 100 dpi or lower).

Currently, most of the palm print biometrics utilize scanner or CCD camera as the input sensor. The users must touch the sensor for their hand images to be acquired. In public areas, like the hospital especially, the sanitary issue is of utmost importance. People are concerned about placing their fingers or hands on the same sensor where countless others have also placed theirs. This problem is particularly exacerbated in some Asian countries at the height of the SARS epidemic. Besides, latent palm prints which remain on the sensor's surface could be copied for illegitimate uses. Aside from that, the surface will get contaminated easily if not used right, especially in harsh, dirty, and outdoor environments. Additionally, some conservative nations may resist placing their hands after a user of the opposite sex has touched the sensor. Therefore, there is pressing need for a biometric technology which is flexible enough to capture the users' hand images without having the users to touch the platform of the sensor.

1.1. Related work

Although palm print is relatively new as compared to the other biometrics like face and fingerprint recognition systems, a number of interesting methods in this area have been reported in the literature. In general, there are two main approaches to palm print recognition: structural and statistical. In the former approach, researchers analyzed the structure of the palm print and tried to find the "best" lines, creases or minutia-like point features to represent the palm pattern. Funada et al. [2] devised a minutiae extraction method which extracted ridges from the palm print by eliminating the creases. On the other hand, Zhang and Shu [3] determined the datum points derived from the principal lines by using the directional projection algorithm. They found that the

* Corresponding author. Tel.: +60 6 2523592; fax: +60 6 2318840.
E-mail addresses: michael.goh@mmu.edu.my (G.K. Ong Michael), tee.connie@mmu.edu.my (T. Connie), bjteoh@yonsei.ac.kr (A.B. Jin Teoh).

0262-8856/$ - see front matter © 2008 Elsevier B.V. All rights reserved.
doi:10.1016/j.imavis.2008.06.010

curvatures of the principal lines were small enough to be represented by several straight line segments, and these lines could be used to represent a palm print. As opposed to the work proposed by Zhang and Shu [3], Duta et al. [4] did not explicitly extract palm lines, but used only isolated points that lied along the palm lines. The feature points, together with their orientation, characterized the palm print. In general, it is reported that the structural based techniques could extract most ridges correctly. However, they are restricted by the complication in determining the primitives and placements of the line structures. Besides, more computational power is usually required to match the line segments with the templates stored in the database. Apart from that, as high quality images (as large as 864 pixels $\times$ 764 pixels) are needed to extract the fine palm print details, these algorithms may not be practical solutions to mass real-time application.

Another research direction towards palm print recognition is to investigate palm print from the statistical aspect. Lu et al. [5] extracted palm print feature by using principal component analysis (PCA). The concept was based on Karhunen–Loeve (K–L) transform in which the original palm print images were projected to a relatively lower dimensional space called Eigenpalms. A similar subspace projection technique was employed by Wu et al. [6]. They deployed Fisher linear discriminant (FLD) to project palmprint images into a reduced dimensional feature space called "Fisherpalm space" for representation. Another approach discussed in [7] deployed independent component analysis (ICA) to obtain the palm print feature in higher order basis. PCA, FLD, and ICA can be classified as subspace analysis techniques because they seek a linear representation of the images such that the basis satisfying a certain fitting criterion are learnt to encode the variables. These methods are more computational effective and are more straightforward to implement. Nevertheless, they suffer from a limitation in which training procedure is required to construct the subspaces. When a new subject is enrolled into the system, the subspaces need to be retrained and this learning procedure impedes the usefulness of these methods in real-time palm print recognition applications.

There are other approaches presented in the literature which include Gabor filters [8–10], morphological techniques [11], and wavelet transform [12]. Zhang et al. [8] proposed to use 2D Gabor filter to extract palm print feature called PalmCode. This method was improved by [9] which fused several PalmCodes in varying directions and the resulting feature was called FusionCode. Kong et al. [10] further enhanced the technique by encoding the orientation information using bit string representation which was named as competitive code. Other researchers like Han et al. [11] and Zhang and Zhang [12] analyzed palm print features by using morphological operators (for example, erosion and dilation) and wavelet transformation. These techniques are more flexible as no training procedure is required. Recently, Sun et al. [13] proposed to fuse the best-performing methods like FusionCode and competitive code by using ordinal features. They claimed that the algorithm served as a general framework for the state-of-the-art palm print representation techniques.

Although extensive research have been conducted in finding effective ways to represent the palm print feature, not much detail of how the palm print images were acquired was discussed in the literature. In fact the acquisition process is one of the key considerations in developing a fast and robust online recognition system. In earlier study, inked-based palm print images [2–4] were used. The palm prints were inked to paper and digitized using scanner. The two-step process was slow and was not suitable for online system. Recently, various input sensor technology like flatbed scanner, CCD camera, CMOS camera, and infrared sensor have been introduced for more straight-forward palm print acquisition. Among the technology, scanner and CCD camera are the commonly used input devices ([5–7,11,12]). Scanner and CCD camera are able to provide very high quality images with little loss of information. However, the process of scanning a palm image requires some time (a few seconds) and the delay cannot cope with the requirement of an online system. The work in [8–10] proposed the use of CCD camera in semi-closed environment for online palm print acquisition and good results had been reported by using this approach.

1.2. Purpose and contribution of this paper

Palm print recognition is an emerging biometric technology which offers efficient and reliable personal authentication. There is high demand for touch-less biometrics due to various social and sanitary issues. However, the design and development of a touch-less system is not easy. The challenges in developing a touch-less palm print recognition system are highlighted as follow:

- *Distance between the hand and input sensor:* Since the user's hand is not touching any platform, the distance of the hand from the input sensor may vary. If the hand is placed too far away from the input sensor, the palm print details will be lost. On the other hand, if the hand is positioned too near to the input sensor, the sensor may not be able to capture the entire hand image and some area of the palm print maybe missing. Thus, a system which allows flexible range of distance between the hand and the input sensor should be designed.
- *Hand position and orientation:* As no guidance peg is used to constraint the user's hand, the user may place his/her hand in various directions and positions. The system must be able to cope with changes in position and orientation of the user's hand in a less restrictive environment.
- *Lighting changes:* Variation in lighting can have significant effect on the ability of the system to recognize individuals. Thus, the system must be capable of generalizing the palm print images across lighting changes.

In this paper, we have endeavored to develop an online touch-less palm print recognition system that attempts to confront the challenges above. The main contributions of this paper are summarized as follows:

- A touch-less palm print recognition system is designed by using low-resolution CMOS web camera to acquire real-time palm print images. The system is designed in such a way that the user positions his/her hand about 40 cm above the sensor, and the web camera captures the palm print images in continuous video stream.
- A novel hand tracking algorithm is developed to automatically track and detect the region of interest (ROI) of the palm print. When a hand object is detected in the video stream, the algorithm tracks and locates the palm print region immediately. The algorithm can continuously track and follow the ROI of the palm print when the hand is moved across the sensor.
- A pre-processing step is proposed to correct the illumination and orientation in the image. As edges (principal lines, wrinkles, and ridges) capture the most important aspects of the palm print images, an algorithm is developed to preserve and enhance the line structures under varying illumination and pose changes.
- A new feature extraction method is proposed to extract the distinguishing palm print feature for representation. A simple gradient operator is applied to obtain the directional responses of the palm print and the local binary pattern (LBP) method is used to obtain the texture description of the palm pattern in different directions. The simplicity of LBP allows very fast feature extraction.

- A real-time feature matching method by using modified probabilistic neural networks (PNN) is devised. PNN has a simple learning rule and it requires only single pass through the training data. Besides, new data can be added anytime without the need to retrain the entire network. These characteristics make PNN very suitable for the proposed real-time recognition system.

The rest of this paper is organized as follows: Section 2 provides the details of the proposed touch-less palm print recognition system. Section 3 describes the experimental setup in the proposed system. The experimental results are presented and discussed in Section 4, and this is followed by some concluding remarks in Section 5.

2. Proposed system

In this paper, we describe a touch-less online palm print recognition system. No guidance peg is used to restrict the user's hand during the input process. We propose a flexible hand tracking and ROI locator to detect and extract the palm print in real-time video stream. The algorithm works under typical office lighting and daylight conditions. Fig. 1 shows the framework of the proposed system.

2.1. Hand tracking and ROI extraction

The hand tracking and ROI extraction steps consist of three stages. First, we segment the hand image from the background by using the skin-color thresholding method. After that, a valley detection algorithm is used to find the valleys of the fingers. These valleys serve as the base points to locate the palm print region. The detailed processing steps are provided in the following sections.

2.1.1. Skin-color thresholding

In order to segment human hand from the background, the skin-color modal proposed by Chang and Robles[14] is used. The human skin color can be modeled as a Gaussian distribution, $N(\mu,\sigma)$, in the chromatic color space. The chromatic color space can remove luminance from the color representation. To segment the hand from the background, the likelihood of the skin color, L, can be computed using $L = \exp[-0.5(x-\mu)^{\mathrm{T}}\sigma^{-1}(x-\mu)]$ where μ and σ are the mean and covariance of the skin-color distribution. We use samples from 1005 skin-color images to determine the values of μ and σ. After the skin likelihood value is determined, the hand is segmented from the background by using the thresholding method (Fig. 2).

2.1.2. Hand valley detection algorithm

We propose a novel competitive hand valley detection (CHVD) algorithm to locate the ROI of the palm. We trace along the contour of the hand to find possible valley locations. A pixel is considered a valley if it has some neighboring points lying in the non-hand region while the majority neighboring points are in the hand region. If a line is directed outwards from the pixel, the line must not cross any hand region along the way. Based on these assumptions, four conditions are formulated to test the existence of a valley. A pixel must satisfy all the four conditions to be qualified as a valley location. If it fails one of the conditions, the pixel will be disregarded and the algorithm proceeds to check for valley existence in the next pixel. Rather than scanning the entire hand image for valley location, the CHVD method discards any "disqualified" candidate upon failure-to-meet-condition, which greatly speeds up the valley detection process.

The four conditions to check the current pixel for valley existence are:

- *Condition 1:* Four checking-points with equal distance are placed around the current pixel (Fig. 3(a)). The four points are placed β pixels away from the current pixel. If one of the points falls in the non-hand region (pixel value = 1), while the remaining within the hand region (pixel values = 0), this pixel is considered

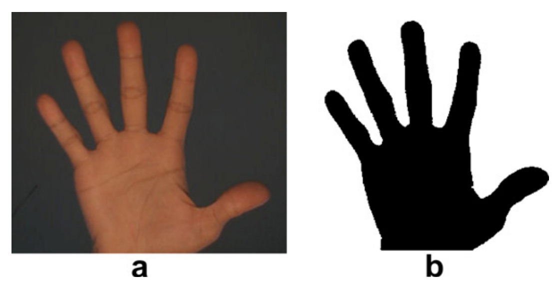


Fig. 2. Skin-color thresholding: (a) the original hand image; (b) segmented hand image in binary form.

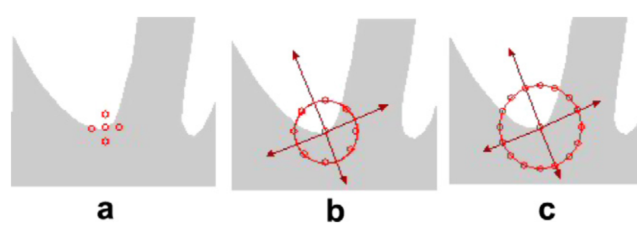


Fig. 3. Competitive valley detection algorithm: (a) four points are placed around the current pixel. If one of the points falls in the non-hand region, Condition 1 is satisfied; (b) eight points are placed around the current pixel. If $1 \leqslant$ points $\leqslant 4$ fall in the non-hand region, Condition 2 is satisfied; (c) 16 points are placed around the current pixel. If $1 \leqslant$ points $\leqslant 6$ fall in the non-hand region, Condition 3 is satisfied. A line is drawn outwards in the non-hand region. If it does not cross any hand region, the pixel is considered a valley location.

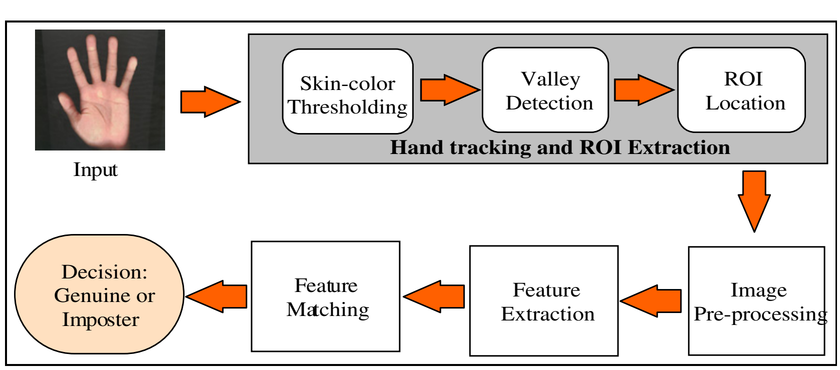


Fig. 1. The proposed touch-less palm print recognition system.

a candidate for valley and we proceed to check for Condition 2. Otherwise, the test stops and the algorithm continue to check the next pixel.

- *Condition 2:* The distance of the checking-points from the current pixel is increased to $\beta + \alpha$ pixels, and the number of checking-points is increased to eight (Fig. 3(b)). If there is at least one and not more than four consecutive neighboring points falling in the non-hand region, while the remaining within the hand region, this pixel satisfies the second condition and we proceed to the next condition.
- *Condition 3:* The number of checking-points is increased to 16. The distance of the points from the current pixel is $\beta + \alpha + \mu$ pixels (Fig. 3(c)). If there is at least one and not more than seven points falling in the non-hand region, while the remaining points within the hand region, this pixel is considered a candidate for valley and the test proceeds to the last condition.
- *Condition 4:* To complete the test, a line is drawn from the current pixel towards the non-hand region (Fig. 3(c)). This is to avoid erroneous detection of a gap/loop-hole in the hand as valley. If this line does not pass through any hand region along the way, the current pixel is asserted as a valley point.

In this research, the values of β, α, and μ are set to 10. We set the range of the number of checking-points in the non-hand region in the three conditions to be 1, $1 \leqslant$ points < 4, and $1 \leqslant$ points < 7, respectively. This is based on the assumption that nobody can stretch his/her finger apart beyond 120°. For example, the angle between the two fingers illustrated in Fig. 3(c) is approximately 90° estimated based on the sectors of the circle between the fingers (each sector = 22.5°).

2.1.3. ROI location

After obtaining the valleys of the finger, P_1, P_2, P_3, and P_4 (Fig. 4(a)), a line is formed between P_2 and P_4. After that, a square is drawn below the line as shown in Fig. 4(b). The square represents the region of interest (ROI) of the palm. Based on the experiment, the average time taken to detect and locate the ROI was less than 1 ms.

The proposed system provides the flexibility to allow the user to use both hands for recognition. The left and right palms are stored separately in the database. This speeds up the recognition process as only half of the database needs to be searched by knowing which side of the hand is used. The following rules are applied to determine the right and left hands:

- Right-hand determination:

$$\begin{aligned} &y_1 > y_2 \quad \text{AND} \quad y_1 > y_3 \quad \text{AND} \quad y_1 > y_4 \quad \text{AND} \\ &x_1 < x_2 \quad \text{AND} \quad x_1 < x_3 \quad \text{AND} \quad x_1 < x_4 \quad \text{AND} \\ &x_4 > x_2 \quad \text{AND} \quad x_4 > x_3 \end{aligned} \tag{1}$$

- Left-hand determination:

$$\begin{aligned} &y_4 > y_1 \quad \text{AND} \quad y_4 > y_2 \quad \text{AND} \quad y_4 > y_2 \quad \text{AND} \\ &x_4 > x_1 \quad \text{AND} \quad x_4 > x_2 \quad \text{AND} \quad x_4 > x_3 \quad \text{AND} \\ &x_1 < x_2 \quad \text{AND} \quad x_1 < x_3 \end{aligned} \tag{2}$$

2.2. Image pre-processing

As the ROIs are of different sizes and orientations, the pre-processing job is performed to align all the ROIs into the same location. First, the images are rotated to the right-angle position by using the vertical axis as the rotation-reference axis. After that, as the size of the ROIs vary from hand to hand (depending on the sizes of the palms), they are resized to a standard image size by using *bicubic* interpolation. In this research, the images are resized to 150 pixels × 150 pixels.

We enhance the contrast and sharpness of the palm print images so that the dominant palm print features like principal lines and ridges can be highlighted and become disguisable from the skin surface. The Laplacian isotropic derivative operator is used for this purpose. After that, the Gaussian low-pass filter is applied to smooth the palm print images and bridge some small gaps in the lines. Fig. 5(a) shows the original palm print image and Fig. 5(b) depicts the result of applying the image enhancement operators. The detail in the enhanced image is clearer and sharper in which fine details like the ridges become more visible. Experiment result in Section 4.2 shows the gain in accuracy by applying the proposed image enhancement techniques.

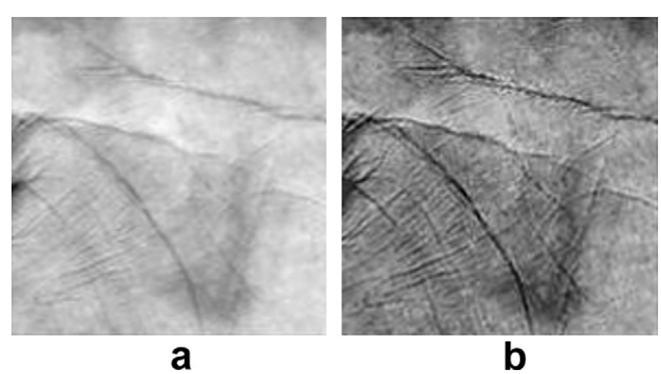


Fig. 5. (a) The original palm print; (b) palm print after the contrast adjustment and smoothing effect.

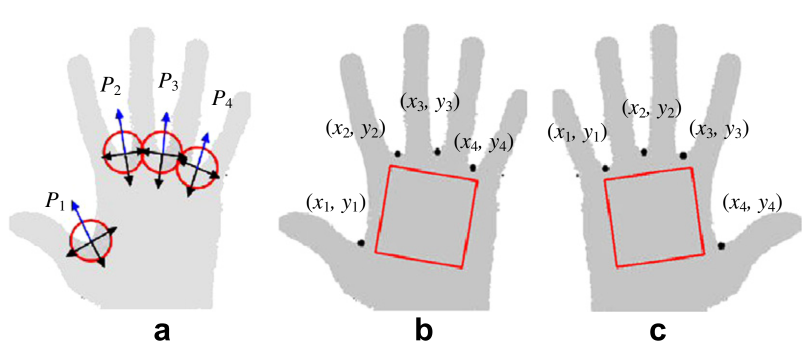


Fig. 4. The ROI location technique: (a) locations of the four valleys; (b) a line is drawn to connect P_2 and P_4. A square is drawn from the line. This square forms the ROI of the palm; (c) the ROI detected in the other side of the hand.

2.3. Feature extraction and matching

We propose a new way to apply the local binary patterns (LBP) texture descriptor [15] on the directional responses of gradient operator. Unlike fingerprint which flows in uniform structure with alternating ridges and furrows, the texture of palm print is irregular and the lines and ridges can flow in various directions. This motivates us to decompose the line patterns into four directions and study them separately. LBP texture descriptor is used to analyze and extract the texture of the palm print in the various directions.

2.3.1. Sobel directional operator

Since palm print contains edges that flow in various directions, it is worthwhile to investigate its texture in different angles. The Sobel and Prewitt operators are well-known filters that can be used to detect discrete gradient in different orientations. Prewitt masks are simpler to implement than the Sobel masks. However, Sobel masks have better noise suppression ability. The comparative study of the performance of Prewitt and Sobel operators is provided in Section 4.2. For better recognition reason, the Sobel operator is used in this work to find the palm print responses along the horizontal, vertical and diagonal in plus and minus 45° directions. The Sobel masks used in this paper are illustrated in Fig. 6.

For computational efficiency and noise reduction purposes, we first decompose the palm print image into lower resolution images by using wavelet transformation before applying the Sobel operator. The detail of applying wavelet transformation on palm print images could be found in [7]. Fig. 7 shows the components of the palm print in four directions by applying Sobel operator.

-1	-2	-1
0	0	0
1	2	1

a

-1	0	1
-2	0	2
-1	0	1

b

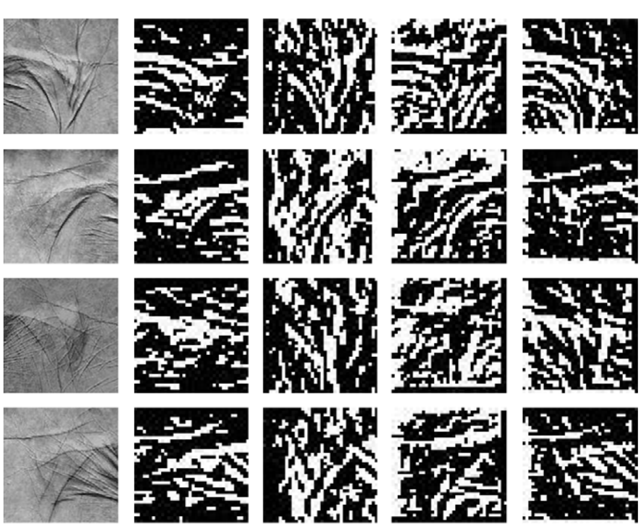
0	1	2
-1	0	1
-2	-1	0

c

-2	-1	0
-1	0	1
0	1	2

d

Fig. 6. The Sobel masks used to detect the palm print (a) horizontally; (b) vertically; (c) diagonally at positive 45°; and (d) diagonally at negative 45°.



a b c d e

Fig. 7. Examples of directional responses derived using Sobel operator. (a) Original palm print images; (b–e) components of the images in the horizontal, vertical, positive 45°, and negative 45° directions.

2.3.2. Local binary patterns

The LBP operator [15] is a simple yet powerful texture descriptor that has been used in various applications. Its high discrimination ability and simplicity in computation have made it very suitable for online recognition system. LBP operator labels every pixel in an image by thresholding its neighboring pixels with the center value. Fig. 8 illustrates an example how the binary label for a pixel value is obtained by thresholding the value against the eight neighborhood pixels.

It is found that certain fundamental patterns in the bitstring account for most of the information in the texture [15]. These fundamental patterns are termed as "uniform" patterns and they are bitstrings with at most 2 bitwise transitions from 0 to 1 and vice versa. Examples of uniform patterns include 00000000, 11110000, and 00001100. A label is given to each of the uniform patterns, and the other "non-uniform" patterns are assigned to a single label. After the labels have been determined, a histogram, H, of the labels is constructed as:

$$H_l = \sum_{i,j} \{L(i,j) = l\}, \quad l = 0, \ldots, n-1 \tag{3}$$

where n is the number of different labels produced by the LBP operator, while i and j refer to the pixel location. The histogram of the labels is used as the texture descriptor and it contains information about the local configuration of the image.

Several parameters can be tuned to optimize the performance of LBP. The parameters include the number of labels, n (varies according to the number of neighborhood pixels selected), the number of sub-windows (local regions), m, and also the size of the sub-windows (w pixels $\times$ h pixels). LBP operator is a flexible method in which it allows incorporation of global information of the image based on a set of low-dimensional local features. This can be achieved by dividing the palm print into several local regions, $R_1, R_2, \ldots, R_m$, and extracting the texture information from each region independently. The local texture descriptors are then concatenated to form a global descriptor of the image. It is intuitive that dividing the image into more sub-windows provides more spatial information. However, having more sub-windows produces longer feature length that would hamper the processing speed. Fortunately, finding in [16] shows that LBP operator is a robust algorithm in which the changes in parameters will not have significantly impact on the overall performance. Changes in parameters only cause differences in the length of the feature vector, but not the discriminative power of LBP. As tradeoff between speed and performance, we divide the palm print image into nine equally sized sub-windows, and an overlapping window in the center. The size of the nine equally spaced sub-windows is set to 13 pixels $\times$ 13 pixels, while the size of the overlapping central window is adjusted to 26 pixels $\times$ 26 pixels. The reason we form a window in the center is because we believe this region encodes important information of the edge flow of the three principal lines. An example of the sub-windows formed on a palm print image is illustrated in Fig. 9. We apply uniform LBP in the $(8,1)$ neighborhood on each of the sub-windows to extract the local texture feature, and concatenate the local features to get a global descriptor of the palm

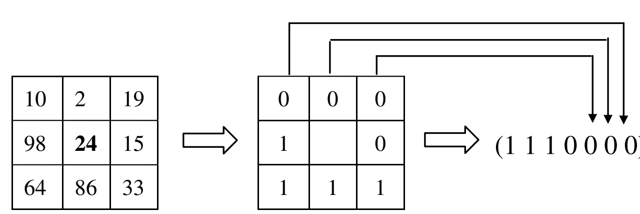


Fig. 8. Example to calculate the binary label in LBP.

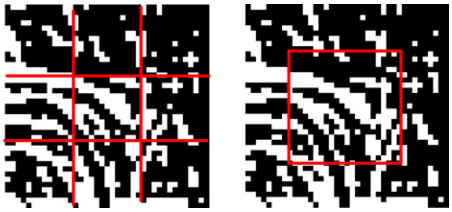


Fig. 9. A palm print image is divided into nine equally sized rectangular sub-windows and an overlapping sub-window in the center. (We show the sub-windows in two separate images for clearer illustration.)

print. This process is performed on the palm print components in four directions (horizontal, vertical, diagonal +45° and −45°). Thus, the texture descriptor for a given palm print will have a size of n (the number of labels) $\times$ m (the number of sub-windows) $\times$ 4 (the components of palm print in four orientations).

2.3.3. Feature matching

In this research, the χ_2 measure is deployed as the feature matching tool:

$$\chi^2(P,G) = \sum_{i=0}^{n} \frac{(P_i - G_i)^2}{P_i + G_i} \tag{4}$$

where n is the number of length of the feature descriptor, P is the probe set, and G denotes the gallery set. We have also deployed a modified probabilistic neural network (PNN) to classify the palm print texture descriptors using the neural networks approach. The motivations of using PNN are driven by its good generalization property and its ability to classify dataset in just one training epoch.

PNN is a kind of radial basis network primarily based on the Bayes–Parzen classification. Besides the input layer, it contains a pattern, summation and output layers [17]. The pattern layer consists of one neuron for each input vector in the training set, while the summation layer contains one neuron for each class to be recognized. The output layer merely holds the maximum value of the summation neurons to yield the final outcome (probability score). To tailor the specific requirement of the proposed online palm print recognition system, the formula to calculate the outcome of the pattern layer is modified to become $\mathrm{out}_j = \exp\left(-\sum_{i=1}^{n}((\boldsymbol{P}_i - \omega_{ij})^2/(\boldsymbol{P}_i + \omega_{ij}))/\sigma\right)$. In this case, out_j is the output of neuron j in pattern layer; $\boldsymbol{P}_i$ refers to the probe set of user i, and ω_{ij} denotes the weight between ith neuron of the input layer and jth neuron in the pattern layer. σ is the smoothing parameter of the Gaussian kernel and is also the only parameter dependent on the user's choice. In this paper, the value of σ is set to 0.1 [17].

3. Experiment setup

In this experiment, a standard PC with Intel Pentium 4 HT processor (3.4 GHz) and 1024 MB random access memory was used. The program was developed using Visual Studio.NET 2005. The image acquisition, as well as the hand tracking and ROI extraction modules, were written in Visual C#.net. On the other hand, the image pre-processing, feature extraction, and feature matching modules were developed using Matlab functions. Our capturing device was a 1.3 mega pixel web camera. The palm print was detected in real-time video sequence at 25 fps. The image resolution was 640 pixels $\times$ 480 pixels, with color output type in 256 RGB (8 bits-per-channel). The ROI of the palm print was captured and stored as bitmap format from the video sequence. The interval between capturing the next ROI was 2 s. The exposure parameter of the web-cam was set low to reduce the effect of background light as the background light might disrupt the quality of the palm print image. We placed a 9 W warm-white light bulb beside the camera. The bulb emitted yellowish light source that enhanced the lines and ridges of the palm. A black cardboard was placed around the web-cam and light bulb to set up a semi-controlled environment as shown in Fig. 10. The black cardboard absorbed some reflectance from the light bulb so that the palm image did not appear too bright.

The proposed methodology was tested on a database containing palm images from 320 individuals. One hundred and forty seven of them are females, 236 of them are less than 30 years old, and 15 of them are more than 50 years old. The testing subjects come from different ethnic groups: 136 Chinese, followed by 125 Malays, 45 Indians, 6 Arabians, 2 Indonesians, 2 Pakistanis, a Africans, a Mongolian, a Sudanese, and a Punjabi. Most of them are students and lecturers from Multimedia University. To investigate how well the system can identify unclear or worn palm prints due to laborious work, we had also invited 10 cleaners to contribute their palm print images to our system.

The users were requested to stretch their fingers during the image capturing process. They were allowed to wear rings and other ornaments. Besides, users with long finger nails could also be detected by the system. The system was able to locate 20 images from both hands in less than 15 s. Some palm print images acquired by the system are shown in Fig. 11.

4. Results and discussion

In this section, we had conducted comprehensive experiments to evaluate the effectiveness and robustness of the proposed system. We first carried out palm print tracking in dynamic environment to validate the robustness of the proposed hand tracking technique. After that, we performed off-line and on-line experiments to assess the accuracy and performance of the proposed system.

4.1. Online palm print tracking

We presented an experiment to evaluate the robustness of the proposed palm print tracking algorithm. The first experiment was conducted in a semi-controlled environment shown in Fig. 10. A user was asked to present his hand above the web camera and slowly rotate his hand to the left and right directions. The user was also asked to move his hand closer and gradually away from the web-cam. Some tracking results of the palm print region are shown in Fig. 12.

The proposed palm print tracking method performed quite well as the ROI of the palm print could be located regardless of changes in size and direction. We further assessed the effectiveness of the algorithm in dynamic environment. In this video sequence, the user had continuous body movements, and the image might be

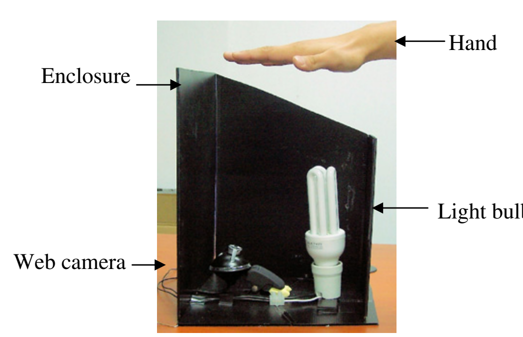


Fig. 10. The experiment setup.

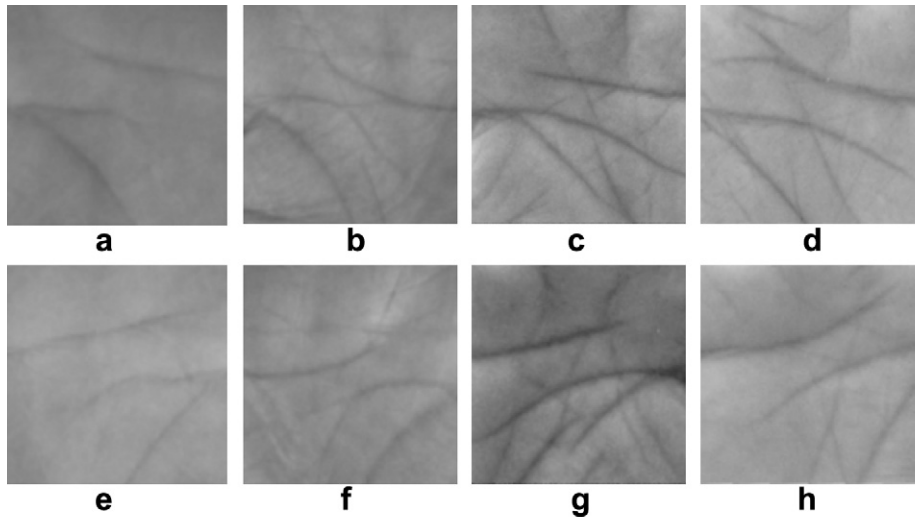


Fig. 11. Sample palm print images acquired by the system. (a) Through (d) images from right hand. (e) Through (h) images from left hands.

disrupted by other background objects and varying illumination conditions. Fig. 13 displays the test sequence to locate the palm print in dynamic environment.

Based on the tracking result, the proposed algorithm performed well in the dynamic environment. The images in the top row, for example, contained other background objects like calendar, white-board, computer, and even the face of the user. The algorithm was able to locate the palm print region among the cluttered background. When both hands were present in the image (for example, the first image from the right in the first row), the algorithm detected one of the palm prints. We designed the system in such a way that only one hand was required to access the application. Therefore, the first palm detected in the video sequence was used for further analysis. Besides, we tried to spoof the algorithm by presenting a fake hand made from Manila paper. Some lines were drawn on the fake image to make it more "palm-like". Nevertheless, the algorithm still managed to recognize the real palm based on the color cue. Apart from that, we wanted to investigate how well the tracking algorithm performs under adverse lighting condition. When the palm was placed under a bright light exposure (the first image from the right in the last row), the algorithm could locate the palm print region accurately.

4.2. Off-line verification

We performed off-line experiments to evaluate the performance and accuracy of the proposed system. The experiment was conducted based on the palm print images from 320 users in our database. Among the 20 images provided by each user for each hand, 10 images were used as gallery set while the others as probe set. Equal error rate (ERR) was used as the evaluation criteria in the experiment. EER is the average value of the two closest error rates: false acceptance rate (FAR) and false rejection rate (FRR).

First, we wanted to investigate which gradient operators, Prewitt or Sobel, yielded better result. We obtained the directional responses from Prewitt and Sobel operators, and applied uniform LBP in the (8,1) neighborhood on the responses to get the texture descriptor. The χ^2 distance was used as the dissimilarity measure. The results of applying the two operators are shown in Table 1. As expected, Sobel performed slightly better than Prewitt operator

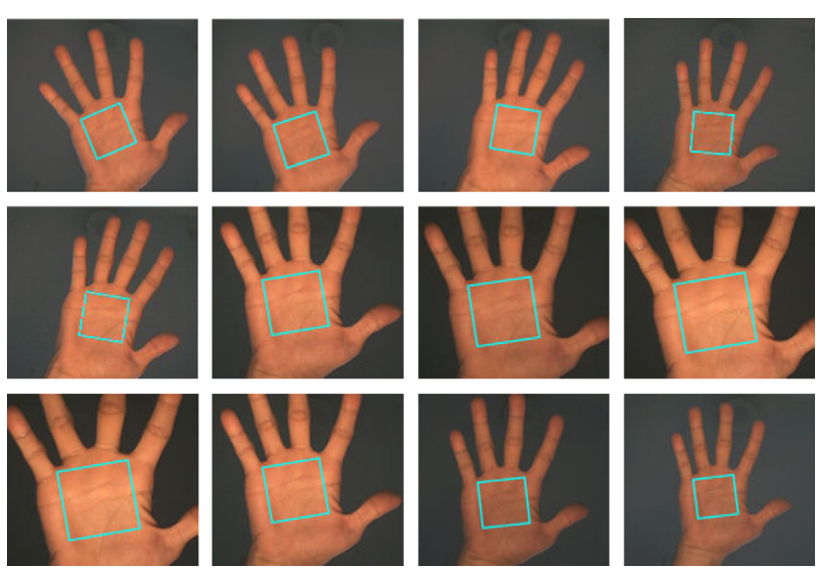


Fig. 12. Some tracking results of the proposed palm print tracking algorithm in semi-controlled environment.

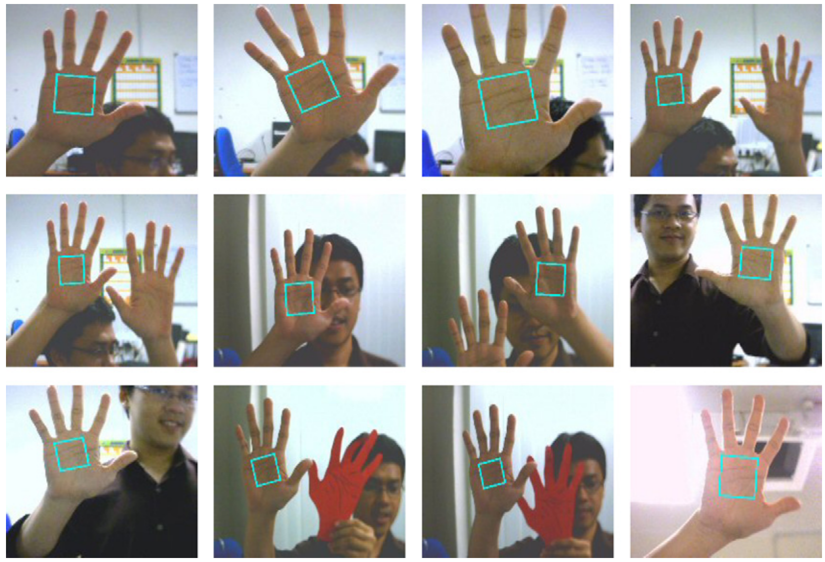


Fig. 13. Some tracking result of the proposed palm print tracking algorithm in dynamic environment.

Table 1
Comparative study between Prewitt and Sobel operators

Gradient operators	EER (%)	Correct recognition rate (%)
Prewitt operator	1.58	98.62
Sobel operator	1.52	98.68

and gave EER equaled 1.52%. The result was inline with the hypothesis that Sobel outperforms Prewitt operator due to its better ability to suppress noise. Hence, we used Sobel operator in the subsequent experiments.

We had also conducted an experiment to validate the usefulness of the image enhancement techniques discussed in Section 2.2. The setting used in this experiment was the same as the first experiment. Table 2 shows the result of applying Sobel and LBP operators on the palm print images with and without the contrast adjustment and smoothing techniques. An improvement gain of about 0.6% in correct recognition rate was obtained with the proposed image enhancement methods.

The next experiment was conducted to investigate how well the proposed gradient operator and LBP method complemented each other. To distinguish one method from the others, we adopted the following naming convention:

- DG – directional gradient operator (Sobel).
- LBP – local binary pattern.
- DGLBP – directional gradient with LBP.

We first performed the experiment by using Sobel operator on the palm print dataset. The Sobel operator was used to extract the binary images of the palm print edges in four different directions. Feature matching was performed by using Hamming distance. The images were translated and rotated slightly when they were matched. After that, we tested LBP on the palm print images without Sobel operator. The original images in 256 grayscale color space were used and χ^2 distance measure was deployed. Next, we combined Sobel operator and LBP in which the histogram of LBP (texture descriptor) was extracted from the directional responses of the palm print edges obtained using Sobel masks. The histogram of each image was then matched using χ^2 distance. The result of the three experiments is depicted in Fig. 14.

The experiment showed that DGLBP indeed performed better than that of DG and LBP. Sobel operator removed the noise in the palm print images and extracted the edges in different orientations. This process helped to reduce the intra-class variability among the palm print dataset. When LBP was applied, the intriguing texture description of the palm print that flowed in various directions could be vividly extracted and represented. Thus, it could be shown that Sobel operator and LBP complemented each other very well.

As there is no open database for palm print images captured using continuous video streams, we tested some other algorithms by using the dataset we collected based on the setting described in Section 3. We had applied some representative techniques in palm print recognition which include PCA [5], competitive code [9] and ordinal code [13] to corroborate the proposed system. Euclidean distance was used as the similarity measure for PCA, Hamming distance was applied for competitive code and ordinal code, whereas

Table 2
Comparison of using the image enhancement techniques (contrast adjustment and smoothing)

Methods	EER (%)	Correct recognition rate (%)
Without image enhancement	2.10	98.10
With image enhancement	1.52	98.68

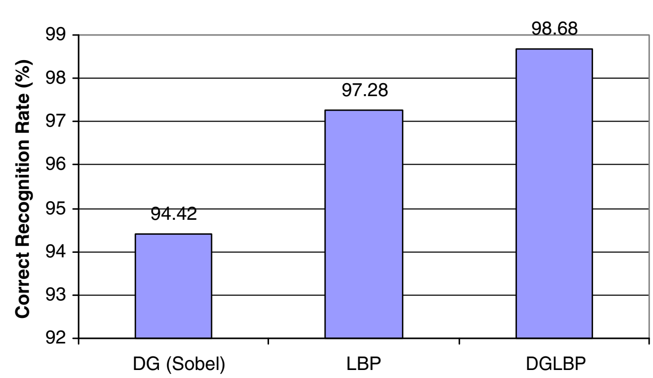


Fig. 14. Correct recognition rates obtained using different operations on the palm print images.

χ^2 distance was deployed for DGLBP in the experiment. The comparative result is illustrated in Fig. 15. It can be observed that the performance of DGLBP is comparable to that of competitive code and ordinal code. Apart from the promising result, DGLBP has another big advantage over other methods because of its simplicity in computation. LBP operator only requires time complexity of $O(2^n)$, where n equals the number of neighborhoods, to generate the labels once. Depending on the number of sub-regions formed in an image, the time complexity required to produce the LBP descriptor is $O(mhw)$, where m denotes the number of sub-regions, while, h and w refer to the height and width of a sub-region, respectively. The complexity of the algorithm can be reduced to $O(mn)$ if the sizes of h and w are small. In this research, the average time taken to extract the feature of a palm print using DGLBP was only 0.07 s. The time taken for the other methods is recorded in Table 3.

Apart from assessing the performance of DGLBP, a study was also conducted to determine a suitable feature matching tool for the proposed method. We compared the result of applying χ^2 measure and modified PNN with DGLBP, and the comparison is provided in Table 4. Three samples were used to train the modified PNN in this experiment. Modified PNN demonstrated superior performance as compared to χ^2 measure as PNN possessed better generalization property. When modified PNN was used, DGLBP outperformed the other methods (PCA, competitive code and ordinal code) where EER equaled 0.74% was achieved. However, the speed of training was achieved at the cost of increase in complexity and computational/memory requirements. The time complexity for training by using PNN is $O(kp)$, where k denotes the input vector dimension and p is the number of training samples. The time recorded in Table 4 is the speed taken for PNN and χ^2 measure to run the verification test using 20 palm print samples. It can be observed that modified PNN indeed took longer time than χ^2 measure. However, the gain in performance is noteworthy as the EER could be reduced from 1.52% to 0.74%. Therefore, modified PNN is still favored over χ^2 measure if accuracy is the main concern.

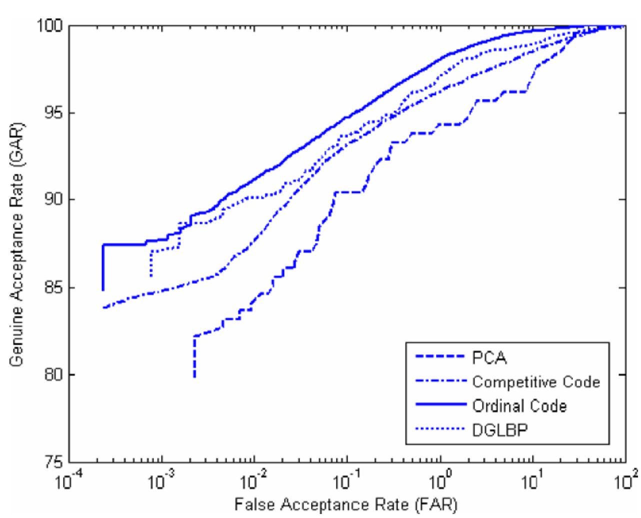


Fig. 15. ROC which compares the performance of Sobel and Prewitt operator.

Table 3
Accuracy of the methods and average time taken for feature extraction

Methods	EER (%)	Average time (s)
PCA	3.14	0.330
Competitive code	2.68	0.406
Ordinal code	1.18	0.204
DGLBP	1.52	0.073

Table 4
EER and speed taken for verification using χ^2 and modified PNN

Classifiers	EER (%)	Average time (s)
Modified PNN	0.74	0.73
χ^2 measure	1.52	0.22

4.3. On-line palm print recognition

Following the off-line experiments, we carried out on-line tests to evaluate the efficiency and robustness the proposed method. The first experiment was conducted to assess the speed of the proposed system for on-line recognition. There are two modes of recognition: (i) verification and (ii) identification. Verification is a one-to-one comparison in which the biometric sample provided by the user is compared with the previously stored template. If the two samples match, the system confirms the identity of the user. On the other hand, identification is one-to-many comparisons in which the system recognizes the user's identity by comparing the presented sample against the entire database to find a possible match. Table 5 shows the time used to verify and identify a palm data. The recorded results correspond to the time taken from hand tracking, ROI location, image pre-processing, feature extraction to palm print matching. It just took 0.22 s to verify a palm print, and 1.27 s to identify a person (using the simple χ^2 measure) in a database containing 320 individuals. The efficiency demonstrated by the proposed system shows that it has promising potential to be implemented in real-time application.

The subsequent experiments were conducted to test the robustness of the system against varying image sizes and qualities. We randomly selected 50 users in the database for the tests. As no peg or other tool was used in the system, the users may place their hands at different heights above the web camera. The palm image appeared large and clear when the palm was placed near the web-cam. Many detailed line features and ridges could be captured at near distance. As the hand moved away from the web-cam, the focus faded and many prints information were lost. Therefore, we wanted to find out the optimal height for the users to place their hands. In the experiment, the users were asked to position their hands at certain distances away from the web-cam. Verification was performed for each user by using threshold value equaled 2.5. The verification tests were performed 10 times for each user at each distance. The same hand was used for all the testing. Fig. 16 depicts the relationship between the distance (between the hand and the input sensor) and performance (in terms of average correct recognition rate). In the experiment, the distances 40–45 cm away from the web-cam raised the best recognition result. The performance dropped when the distance increased because of the degradation in image quality. On the other hand, smaller distances yielded poorer result because of the over-exposure effect. The range from 40 to 45 cm gave the sharpest image quality in this experiment.

The next experiment was conducted to verify the stability of the system over time. The users were asked to test the system in six occasions, for the intervals of 5, 10, 15, 30, and 60 days from the first test. The subjects were asked to perform 10 verifications in

Table 5
EER and speed taken for online verification and identification

Recognition mode	Time (s) using χ^2 measure	Time (s) using modified PNN
Verification	0.22	0.73
Identification	1.27	2.57

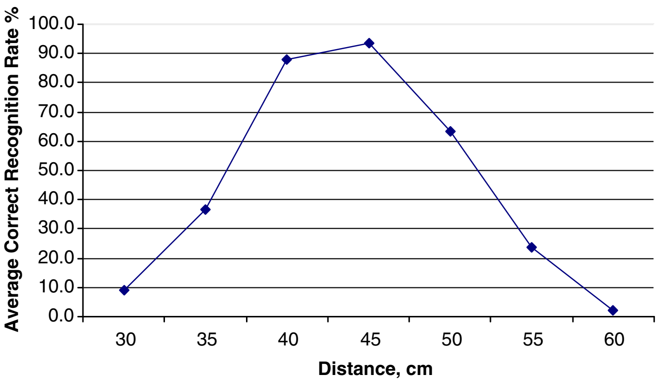


Fig. 16. The average correct recognition rate vs. distances of the palm from the web-cam (cm).

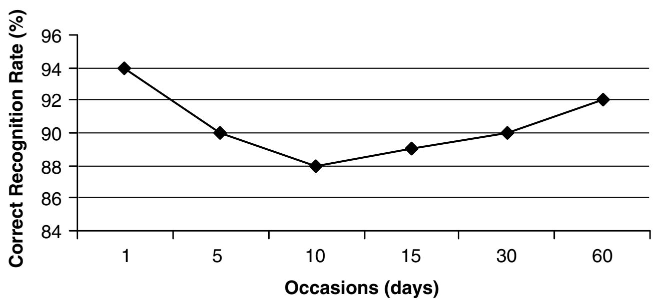


Fig. 17. Average correct recognition rate vs. time length.

each occasion. The average correct recognition rate for each occasion is displayed in Fig. 17. It can be observed that the performance of the system dropped after the first few tests, and gradually stabilized after that. We conjecture that this may be due to the user's unfamiliarity of using the system for the first few attempts. They may position their hands at awkward directions that their palms could not be recognized correctly by the system. As the user used the system more frequently, they started to get accustomed to place their hands at the correct location for recognition.

5. Conclusions

This paper presents an innovative touch-less palm print recognition. The proposed touch-less palm print recognition system offers several advantages like flexibility and user-friendliness. We proposed a novel palm print tracking algorithm to automatically detect and locate the ROI of the palm. The proposed algorithm works well under dynamic environment with cluttered background and varying illumination. As the user's hand can be rotated and placed at different directions when captured, we applied a pre-processing step to correct the illumination effect and orientation of the hand image. The pre-processing step also helps to enhance and amplify the dominant line edges in the palm print image. A new feature extraction technique, DGLBP, is presented which analyzes the palm print texture in different directions. The proposed method reduces noise and increases the discriminatory power of the system. The major advantage of the proposed scheme is its efficiency in computation. Besides, we also introduced a new feature matching tool for online recognition by using modified PNN. This method possesses good generalization ability with little training cost.

Extensive experiments have been conducted to evaluate the performance of the system in both on-line and off-line environments. Experiments show that the proposed system is able to produce promising result. The proposed touch-less palm print system could perform very fast in real-time application. It takes less than one second to capture, process and verify a palm print image. Besides, the proposed system is able to cope with real-time recognition challenges such as hand movement, lighting change and variation in hand position and orientation.

References

[1] C. Harold, M. Charles, Finger Prints Palms and Soles an Introduction to Dermatoglyphics, The Blakiston Company, Philadelphia, 1943.
[2] J. Funada, N. Ohta, M. Mizoguchi, T. Temma, K. Nakanishi, A. Murai, T. Sugiushi, T. Wakabayashi, Y. Yamada, Feature extraction method for palmprint considering elimination of creases, in: Proceedings of the 14th International Conference on Pattern Recognition, vol. 2, 1998, pp. 1849–1854.
[3] D. Zhang, W. Shu, Two novel characteristics in palmprint verification: datum point invariance and line feature matching, Pattern Recognition 32 (1999) 691–702.
[4] D. Duta, A.K. Jain, K.V. Mardia, Matching of palmprint, Pattern Recognition Letters 23 (2002) 477–485.
[5] G. Lu, D. Zhang, K. Wang, Palmprint recognition using Eigenpalms features, Pattern Recognition Letters 24 (9–10) (2003) 1473–1477.
[6] X. Wu, D. Zhang, K. Wang, Fisherpalms based palmprint recognition, Pattern Recognition Letters (24) (2003) 2829–2838.
[7] T. Connie, T. Andrew, K.O. Goh, An automated palmprint recognition system, Image and Vision Computing 23 (5) (2005) 501–515.
[8] D. Zhang, W. Kong, J. You, M. Wong, On-line palmprint identification, IEEE Transaction on PAMI 25 (9) (2003) 1041–1050.
[9] W.K. Kong, D. Zhang, Competitive coding scheme for palm print verification, in: Proceedings of the 17th International Conference on Pattern Recognition, 2004, pp. 520–523.
[10] A. Kong, D. Zhang, M. Kamel, Palmprint identification using feature-level fusion, Pattern Recognition 39 (2006) 478–487.
[11] C.C. Han, H.L. Cheng, K.C. Fan, C.L. Lin, Personal authentication using palmprint features, Pattern Recognition 36 (2) (2003) 371–381.
[12] L. Zhang, D. Zhang, Characterization of palmprint by wavelet signatures via directional context modeling, IEEE Transactions on SMC – B 34 (3) (2004) 1335–1347.
[13] Z. Sun, T. Tan, Y. Wang, S.Z. Li, Ordinal palmprint representation for personal identification, in: Proceedings of the 2005 IEEE Computer Society Conference on Computer Vision and Pattern Recognition, 2005, pp. 279–284.
[14] H. Chang, U. Robles, Face Detection, May 2000. Available from: <http://www-cs-students.stanford.edu/~robles/ee368/main.html>.
[15] T. Ojala, M. Pietikäinen, T. Mäenpää, Multiresolution gray-scale and rotation invariant texture classification with local binary patterns, IEEE Transaction on Pattern Analysis and Machine Intelligence 24 (7) (2002) 469–481.
[16] T. Ahonen, A. Hadid, M. Pietikäinen, Face recognition with local descriptors, in: Proceedings of the Eighth European Conference Computer Vision, 2004, pp. 145–150.
[17] T. Andrew, T. Connie, N. David, Remarks on BioHash and its mathematical foundation, Information Processing Letters 100 (4) (2006) 145–150.