Finger Vein Verification Based on a Personalized Best Patches Map

Lumei Dong, Gongping Yang*, Yilong Yin, Fei Liu, and Xiaoming Xi

School of Computer Science and Technology, Shandong University, Jinan 250101, P.R. China

dongcomeon@gmail.com, gpyang@sdu.edu.cn, ylyin@sdu.edu.cn, lf_ff@sina.com, fyzq10@126.com

Abstract

Finger vein pattern has become one of the most promising biometric identifiers. In this paper, we propose a robust finger vein verification method based on a personalized best patches map (PBPM). Firstly, some robust and discriminative visual words of finger vein are learned from traditional base feature such as local binary pattern (LBP). These visual words are named as finger vein textons (FVTs), which can well represent the visual primitives of finger vein. Secondly, we represent the finger vein image as a finger vein textons map (FVTM) by mapping each patch of the image into the closest FVT. Thirdly, by rejecting inconsistent patches, the PBPM of a certain individual is learned from these FVTMs which are extracted from the training samples of the same finger. Finally, the matched best patch ratio is used to measure similarity between the extracted FVTM of the input finger and the PBPM of a certain individual. Experimental results show that our method achieves satisfactory performance on the open PolyU database. In addition, it also has strong robustness and high accuracy on the self-built rotation and translation databases.

1. Introduction

Finger vein verification, as one of the most promising biometric techniques, has received considerable attention from researchers due to its advantages over other biometric techniques [1,2]: (1) non-contact; (2) live-body identification; (3) high security; (4) small device size. Now, it is playing a more and more important role in many mission-critical applications such as access control, personal identification, E-passport, etc.

Generally speaking, a typical finger vein verification process includes the following four steps, namely, image capturing, preprocessing, feature extraction and matching. Among these steps, feature extraction is very important. Many researchers are dedicated to studying effective feature extraction methods in finger vein verification. According to whether finger vein network is segmented or not, these feature extraction methods can be divided into two categories. The first kind of method is based on the segmented finger vein network including the repeated line tracking method [3], the mean curvature [4], the maximum curvature point method [5], the Gabor filter [6] and so on. These methods obtain the geometric shape, topological structure or other information from the segmented blood vessel network [3-8]. However, if the quality of captured images is low, the network may not be segmented properly. Consequently, the extracted features based on improper network will make the performance of verification degrade dramatically.

To alleviate the above problem, feature extraction methods without network segmentation [9-14] have been proposed. Some local pattern-based methods such as local binary pattern (LBP) [9], local line binary pattern (LLBP) [10] and personalized weight maps (PWM) [11] were applied to extract different effective features respectively. Among these methods, LBP can reflect the texture of the finger vein very well. As a variant of LBP, LLBP demonstrates a better accuracy than LBP. PWM is rooted in LBP and trains the personalized weight values for each individual according to the stabilities of different bits. Since extracted features of these methods are based on a single pixel, they are sensitive to the variation of finger vein image. In addition to local pattern-based methods, some methods like principal component analysis (PCA) [12], linear discriminant analysis (LDA) [13] and two-dimensional principal component analysis ($(2D)^2$PCA) [14] were used to extract global finger vein feature. Nevertheless, for these methods, if the finger vein database is very large, the training process of projection matrix will be complicated.

Though many feature extraction methods have been studied and proposed, the rotation and translation problem of images is still a big challenge in finger vein verification. Since the Bag-of-Words (BoW) method has good resistance to occlusions, geometric deformations, rotation and illumination variations [15], which has been successfully applied in texture analysis and visual classification [15-19], the BoW method is an intuitively good candidate for finger vein feature extraction. And several researchers also have applied it in other biometric recognition (e.g., face, action, and iris) [20-23].

*Corresponding author

Meanwhile, we also observed that the finger vein images exhibit important regularity. Some local finger vein patterns are similar and appear repeatedly in the image. Inspired by the BoW method which can learn a small number of visual words as repeating appearance primitives, we represent these similar and repeated finger vein patterns as visual words, which are called finger vein textons (FVTs). Figure 1 shows some FVTs intuitively.

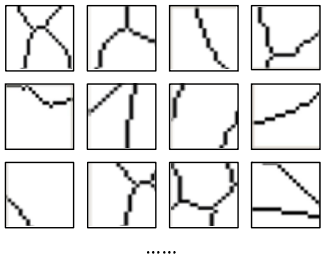

Figure 1: Description of some FVTs.

Based on the above analysis, we use these typical FVTs to demonstrate finger vein images. After learning a number of informative and discriminative FVTs (i.e., clusters in the finger vein feature space) from partitioned patches of training samples, each finger vein image is represented as a FVTs map (FVTM) by mapping each patch into the closest FVT. The FVTM can describe the finger vein effectively on a patch level. However, not all of the patches in a FVTM are equally useful. Generally speaking, there are many samples captured from the same finger. And we can extract a FVTM from each sample. Comparing each patch of the FVTMs from the same individual, we find that some patches are consistent. In other words, for two FVTMs extracted from two samples of the same finger, some FVTs are identical in the same patch of the two FVTMs. These patches are called Best Patches. Instead, there are also some inconsistent patches possibly caused by the rotation or translation of finger vein, which tends to magnify an intra-class matching distance. These phenomena motivated us to use these consistent patches only for personalized matching. According to this consideration, we propose a finger vein verification method based on a personalized best patches map (PBPM). Experimental results show that PBPM can significantly improve verification performance. In addition, related results also demonstrate that the PBPM is robust to rotation and translation variations of finger vein images, which is very crucial to a practical finger vein verification system.

The rest of this paper is organized as follows. Section 2 describes the details of our proposed method. Section 3 presents and discusses the experimental results. Finally, Section 4 concludes this paper.

2. The proposed method

In this section, we describe the framework of finger vein verification based on PBPM. It mainly involves two stages: training stage and testing stage. The training stage includes preprocessing, feature extraction and PBPM generation. At the testing stage, after preprocessing the testing sample and extracting the FVTM, we compute the similarity between the FVTM and the PBPM of a certain individual. With a given threshold, the testing sample will be accepted or rejected by the verification system. The framework of the proposed method is shown in Figure 2.

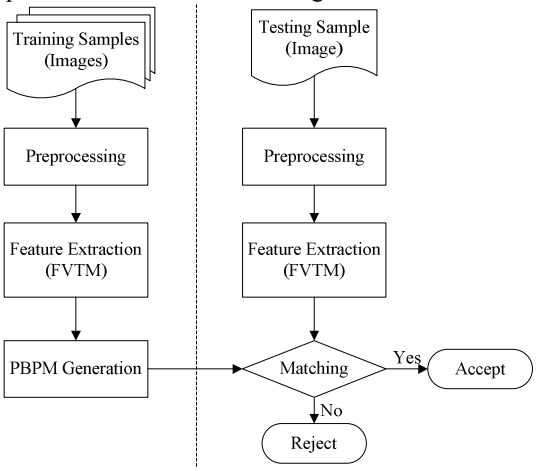

Figure 2: Framework of the proposed method.

2.1. Image preprocessing

For obtaining efficient features, image preprocessing is necessary. The main preprocessing steps involve ROI extraction, size normalization, and gray normalization. More details can be found in [24] [11]. The size of the normalized ROI is set to be 96×64.

2.2. Feature extraction

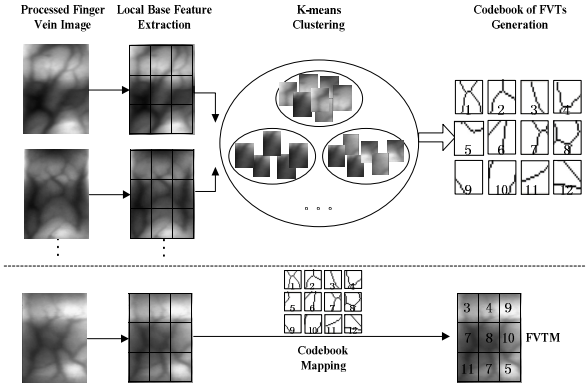

Figure 3: Flowchart of finger vein feature extraction.

In our method, feature extraction process includes three steps as shown in Figure 3. Firstly, local base features are extracted from the partitioned image patches. Secondly, a small codebook of visual words is learned from some collected patches using k-means clustering algorithm. These visual words of finger vein are called finger vein

textons (FVTs). Thirdly, finger vein textons map (FVTM) is used as feature to represent the characteristics of the finger vein image.

2.2.1 Local base feature extraction As a typical feature descriptor, LBP has attained some attention in finger vein recognition [9]. Due to its simplicity and excellent performance, we choose the LBP as base feature. An LBP can be described as an ordered set of binary values determined by comparing the gray values of a center pixel and its 3×3-neighborhood pixels [25,26]. All binary codes can be concatenated together and converted to a decimal, which is expressed as follows [9]:

$$LBP(x_c, y_c) = \sum_{n=0}^{7} s(i_n - i_c)2^n \tag{1}$$

where i_c and i_n corresponds to the gray value of the center pixel (x_c, y_c) and its eight neighboring pixels respectively. The function $s(x)$ is defined as [25]:

$$s(x) = \begin{cases} 1 \; if \; x \geq 0 \\ 0 \; if \; x < 0 \end{cases} \tag{2}$$

To extract local finger vein feature, we partition a finger vein image into R non-overlapping patches of the same size. For each patch, the histogram of LBP (i.e., $H(r)$ $r \in \{1,2,...,R\}$) is calculated as follows [27]:

$$H(r) = \left\{ \sum_{x,y \in patch(r)} I(LBP(x,y) = i) | i = 0,...,N-1 \right\} \tag{3}$$

where $N = 256$ is the number of different values calculated by the LBP operator and I is the indicator function, as shown in Equation (4):

$$I\{A\} = \begin{cases} 1 \; if \; A \; is \; true \\ 0 \; if \; A \; is \; false \end{cases} \tag{4}$$

This histogram contains the distribution of finger vein texture information on a patch level. Consequently, each patch has a 256-dimensional LBP histogram vector. These histogram vectors of LBP in corresponding patch are used as local base features in this paper.

2.2.2 Codebook of FVTs generation There are many different clustering techniques for the generation of codebook [15]. In this paper, codebook is calculated using the k-means clustering algorithm. A schematic diagram illustrating the steps of learning the codebook of FVTs is shown in the top of Figure 3. Firstly, we choose some finger vein images randomly and partition these images into regular patches. Secondly, local base features (i.e., $H(r)$ $r \in \{1,2,...,R\}$) are extracted based on these patches. And then k-means clustering algorithm is performed on these local base features to obtain the most informative centers which consist of the codebook of FVTs. As the center of a learned cluster, a FVT can reflect certain characteristic of finger vein and represent all finger vein features in this cluster. To make a trade-off between performance and computational complexity, we choose 100 finger vein images from the database randomly to learn the codebook of FVTs in our experiment.

2.2.3 FVTM extraction After learning the codebook of FVTs, we can use these FVTs to represent the finger vein image. The procedure of extracting the map of FVTs (FVTM) is shown in the bottom of Figure 3. The key step is to map extracted local base feature (i.e., $H(r)$ $r \in \{1,2,...,R\}$) of a patch into the closest FVT according to the similarity between them. Euclidean distance is used as the similarity measure. Consequently, each patch has a FVT number and each finger vein image can be represented by a FVTM, which is calculated as follows:

$$FVTM(r) = \arg\min_k \{ \| H(r) - FVT(k) \| \} \; k = 1,...K \tag{5}$$

where K is the number of FVTs in codebook, $FVT(k)$ is the k-th FVT in the codebook.

As a map of FVTs which represent some typical finger vein primitives, the FVTM can well represent the finger vein on a patch level. In addition, codebook mapping is beneficial and robust to finger vein verification because it can make an intractable number of distinct possible features categorized into a manageable number of FVTs, reducing the influence of outliers [28]. Besides, for a patch r, a 256-dimensional LBP histogram vector $H(r)$ is mapped into a 1-dimensional $FVTM(r)$, which dramatically reduce the computation complexity and storage requirement.

2.3. PBPM generation

As stated, we can get a FVTM from each finger vein image. In an ideal situation, without considering the interference factors such as rotation and translation, for a certain individual, the FVT in the same patch of different FVTMs should be all identical. However, in practice, finger vein image are typically variant under the influence of captured conditions. For all FVTMs of the same individual, if the same patch doesn't have the same FVT, it is considered as a noisy patch, which may have a negative effect on verification performance. According to this consideration, we try to eliminate the noisy patches and use the best patches only for matching when verifying a testing sample. By rejecting noisy patches, we can obtain better performance in verification and less time consumption and storage requirements than comparing all patches.

Provided that individual A has m samples (i.e., $Image_1$, $Image_2$…$Image_m$) and the corresponding features are

denoted by $FVTM_1, FVTM_2 ... FVTM_m$. For a patch r, $num(k)$ ($\sum_{k=1}^{K} num(k) = m$) denotes the number of the k-th FVT among $FVTM_1(r)$, $FVTM_2(r) \ldots FVTMm(r)$. If the max number of $num(k)$ ($k = 1,..., K$) is larger than two, patch r is regarded as Best Patch. Certainly, all Best Patches of a certain individual make up his Best Patches Map (BPM). Because different individual has different BPM, the BPM for each individual is named as the Personalized Best Patches Map (PBPM). The pseudo-code of generating the PBPM of a certain individual is summarized in Algorithm 1.

Algorithm 1. Generating PBPM

INPUT: $FVTM_1, FVTM_2 ... FVTM_m$ extracted from m samples of a certain individual.
OUTPUT: PBPM of the certain individual.
Algorithm:

```
for r =1 to R // R is the number of patches in FVTM
  for j =1 to m // m is the number of samples
    for k =1 to K // K is the size of codebook
      num(k) = 0 //Initialize the number of consistent
                 // FVT for each patch
      if FVTM_j(r) == k
        num(k) = num(k)+1
      end if
    end for
  end for
  set max_n = max{num(k)} and
    max_k = arg max_k {num(k)} where k = 1,..., K
// max_n is the max number of consistent FVT for a
//patch
  if max_n >= 2 then
    PBPM(r) = max_k //best patch
  end if
end for
return PBPM
```

2.4. Matching

Now every finger vein image is represented by FVTM and each individual has its own PBPM. Here, we use the Matched Best Patch Ratio (MBPR) to measure similarity between the extracted FVTM of the input finger and the PBPM of a certain individual. The MBPR is the ratio of the number of the matching patches to the total number of the best patches in the PBPM. It is calculated as follows:

$$MBPR = \frac{\sum_{i \in BP} f(FVTM(i) - PBPM(i))}{\|PBPM\|} \quad (6)$$

In Equation (6), BP is defined as the set of Best Patch number for PBPM. It means that not every patch of FVTM but only the patches which have the same location in the PBPM (i.e., best patches) are compared by the function $f(x)$. The $FVTM(i)$ and the $PBPM(i)$ denote the FVT number of patch i in FVTM and PBPM respectively. The $\|PBPM\|$ is the number of the Best Patches of a certain individual. The function $f(x)$ is defined as:

$$f(x) = \begin{cases} 1 \; if \; x = 0 \\ 0 \; if \; x \neq 0 \end{cases} \quad (7)$$

3. Database and experiment

3.1. The experimental database

The experiments were conducted on the open database (i.e., PolyU database) constructed by Hong Kong Polytechnic University [6].The database consists of 6264 images collected from 156 volunteers over a period of 11 months (April 2009–March 2010). The finger images were acquired in two separate sessions. In each session, each of the volunteer provided six image samples from the index finger to the middle finger, respectively. As only 105 volunteers turned up for the imaging during the second session, we use the images acquired in the first session in our experiment. In addition, each sample consists of one finger vein image and one finger texture image, but we only use the finger vein image. Consequently, there are totally 1872 (312 fingers×6 samples) finger vein images. In this paper, we consider each finger as an individual. Samples collected from the same individual belong to the same class. Therefore, there are 312 classes in the PolyU database.

3.2. The experimental settings

All the experiments are implemented in MATLAB, and conducted on a PC with 2.9GHz CPU and 4.0GB memory. In this paper, four experiments are designed to evaluate the proposed method: (a) Experiment 1 evaluates the performance of the PBPM and FVTM by comparing with corresponding base feature. (b) Experiment 2 evaluates the performances of PBPM in comparison with some state-of-the-art finger vein verification methods. (c) Experiment 3 is designed to verify the robustness of the PBPM. (d) Experiment 4 discusses the influence of codebook size, patch size and the number of samples to generate the PBPM. (e) Experiment 5 measurers the average processing time of our proposed method.

3.3. Experiment 1

In this section, we evaluate the performance of the PBPM and the FVTM in comparison with traditional base feature (i.e., LBP) on the PolyU database.

In the experiments, we use the first four samples of each class in the PolyU database to generate the PBPM and use the other two as testing samples. For FVTM and LBP methods, we randomly choose one from the first four samples of each class as training sample and use the other two as testing samples. Therefore, there are 624 (312×2) intraclass matchings and 194064 (312×2×311) interclass matchings in total. In this paper, the performance of the system is evaluated by Equal Error Rate (EER), False Rejection Rate (FRR) at 0.01 False Acceptance Rate (FAR) and FAR at 0.01 FRR. The EER is defined as the error rate when the FRR is equivalent to the FAR. It is suitable to measure the overall performance of biometrics systems because the FRR and FAR are treated equally.

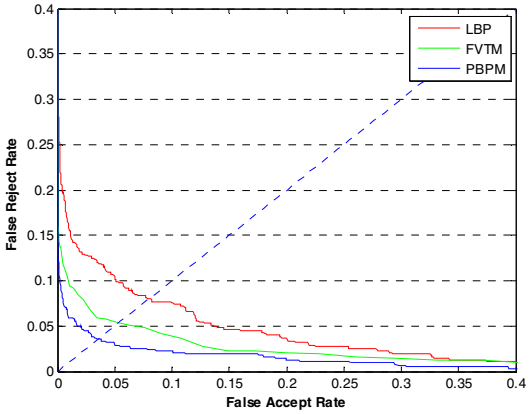

Figure 4: ROC curves by different methods

Table 1. The performance by different methods

Method	EER	FAR at 0.01 FRR	FRR at 0.01 FAR
LBP	0.0802	0.4455	0.1554
FVTM	0.0530	0.3365	0.0946
PBPM	**0.0353**	**0.2537**	**0.0593**

The ROC curves of different methods are shown in Figure 4. The EER, FRR at 0.01 FAR and FAR at 0.01 FRR values are listed in Table 1. The results from Figure 4 and Table 1 show that PBPM and FVTM methods achieve much lower EER than the LBP-based method. This indicates that the extracted feature can well represent the finger vein. Because both PBPM and FVTM use the most informative and representative features (FVTs) learned from traditional base feature (i.e., LBP) for verification, which reduces the influence of outliers at certain degree. In addition, the EER of PBPM method on the PolyU database is 0.0353, outperforming the FVTM method. This demonstrates that each PBPM can truly reflect characteristics of a certain individual because the best patches are different for different individuals. Besides, by removing the noisy patches, the performance of finger vein verification is largely improved.

In order to further verify the generalization of our method with different base features. We also take mean curvature [4] as base feature to evaluate the performance of FVTM and PBPM. The mean curvature method [4] views the vein image as a geometric shape and finds the valley-like structures with negative mean curvatures, which has achieved good performance in finger vein verification. The experimental setting is identical to taking LBP as base feature. The EERs of PBPM and FVTM using mean curvature as base feature are 0.1304 and 0.1731, while the EER of mean curvature is 0.2195. These results show that PBPM and FVTM have better performance than the base feature based method (i.e., mean curvature) and the performance of PBPM is the best of all. It indicates that our method is also effective for other base features.

3.4. Experiment 2

In this section, we evaluate the performances of the PBPM for finger vein verification in comparison with some state-of-the-art finger vein verification methods: mean curvature (MeanC) [4], LBP [9], $(2D)^2PCA$ [14] and PWM [11]. The ROC curves of different methods are shown in Figure 5 and the ERR, FRR at 0.01 FAR and FAR at 0.01 FRR are listed in Table 2.

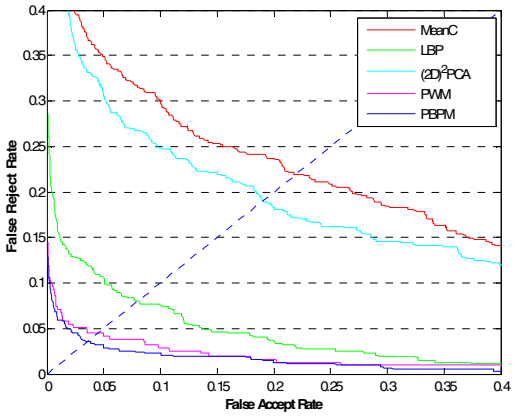

Figure 5: ROC curves by different methods

Table 2. The performance by different methods

Method	EER	FAR at 0.01 FRR	FRR at 0.01 FAR
MeanC	0.2195	0.9675	0.4471
LBP	0.0802	0.4455	0.1554
$(2D)^2PCA$	0.1907	0.9223	0.4359
PWM	0.0449	0.2586	0.0705
PBPM	**0.0353**	**0.2537**	**0.0593**

Table 3. The performances by different methods.

	PolyU Database	Rotation Database		Translation Database	
	EER1	EER2	Difference1	EER3	Difference2
MeanC	0.1304	0.1891	0.0587	0.2708	0.1404
LBP	0.0802	0.0913	0.0111	0.1144	0.0342
$(2D)^2PCA$	0.1907	0.3190	0.1283	0.2676	0.0769
PWM	0.0449	0.0545	0.0096	0.0678	0.0229
PBPM	**0.0353**	**0.0430**	**0.0077**	**0.0465**	**0.0112**

From Figure 5 and Table 2, it can be seen that the PBPM method achieves the best performance. This can be explained by the following four reasons: (i) the FVT can well represent visual primitives of finger vein; (ii) the FVTM feature extracted by our method in section 2.2.3 can maintain the spatial layout information; (iii) the personalized differences among different individuals are very useful, but unfortunately they are often ignored. However, our method makes full use of the differences among individuals and trains a specified PBPM for each individual which can best discriminate this particular individual from others; (iv) comparing to other personalized map methods (e.g., PWM) which are based on some bits of a pixel, our method is based on best patches which are a bigger granularity representation of the image and can be more robust to the noise in pixel level.

3.5. Experiment 3

In order to verify the robustness of our method, we construct a rotation database and a translation database. The rotation database is built by rotating each preprocessed finger vein image in the PolyU database at a random degree which is in the scope of [-5°, +5°]. Similarly, the translation database is built by translating each preprocessed finger vein image in the PolyU database at a random pixel in the scope of [-3 3] in the vertical direction and [-2 2] in the horizontal direction. Consequently, the rotation and translation databases both contain 1872 (312classes × 6samples) finger vein images.

We evaluate the performance of different methods on the rotation and translation databases. The experimental settings are the same as Experiment 1. The EER on PolyU database named EER1, the EER on rotation database named EER2, the EER on translation database named EER3, the difference between EER1 and EER2 named Difference1 and the difference between EER1 and EER3 named Difference2 are listed in Table 3. From Table 3, we can see that PBPM achieves the lowest EER on rotation and translation databases when comparing with the state-of-the-art methods. In addition, from Difference1 and Difference2, we also note that the performance of PBPM decreases 0.0077 and 0.0112 respectively, which decreases less than the state-of-the-art methods. This indicates that our method has stronger robustness to the images with rotation and translation. The main reasons are as follows. Firstly, when extracting FVTM, codebook mapping has good resistance because it could map two slightly different patches which are affected by rotation or translation into the same FVT. For example, for two finger vein image patches which have a little difference from the same finger, traditional feature like LBP about the two patches may be different but our proposed method will map them into the same FVT. Figure 6 intuitively shows that codebook mapping is robust to rotated finger vein image. Secondly, we just use the Best Patches of an individual for matching and remove these patches affected by the interference factors such as rotation and translation, which significantly improves the robustness of finger vein verification.

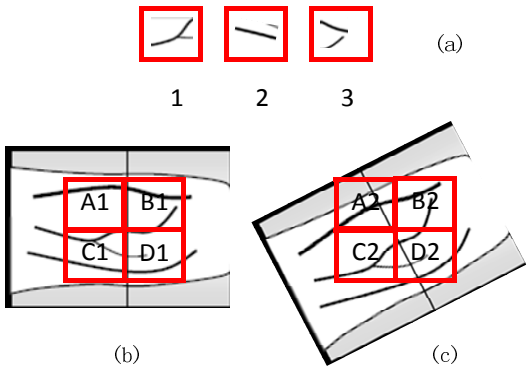

Figure 6: Toy example of representing the robustness of codebook mapping for rotated finger vein image. (a) denotes three FVTs. (b) and (c) are the original and rotated finger vein image respectively. Every image is partitioned into four patches. The codebook mapping result of (b) is (A1, 2) (B1, 3) (C1, 1) (D1, 3). For rotated image (c), the mapping result is (A2, 2) (B2, 3) (C2, 1) (D2, 3). Therefore, (b) and (c) can be classified into the same class.

3.6. Experiment 4

In this section, we discuss the influence of the different codebook size, patch size and the number of training samples.

Firstly, we evaluate the influence of different patch size and codebook size when extracting a FVTM. Here, we just

fix one parameter and vary the other. Table 4 shows the performance with different patch sizes when extracting local base feature (i.e., local LBP histogram). We can see that the patch size of 5×5 pixels works best and the performance is not good enough when the patch size is too small or too large. This is because a small patch contains poorer finger vein information so that the generated codebook of FVTs cannot be very discriminative. Conversely, if the patch size becomes large, the patch may contain richer finger vein information, but the number of patches will become small. With small number of patchs, the spatial layout information will be insufficient and the number of best patches will decrease, which will have a negative effect on the performance of finger vein verification. Therefore, the discriminability of large patch size is also unsatisfactory. The experimental results when varying the size of codebook are shown in Table 5. From Table 5, we discover that the best performance is achieved when the codebook size reaches 20 and that the performance is unsatisfactory when the codebook is small. This can be explained that, with small size of codebook, the FVTs are not very discriminative and some dissimilar finger vein features may be mapped into the same FVT. From the trend in Table 5, we can see that the performance tends to drop when the codebook size becomes large. The reasons may be that, with large codebook size, the FVTs become more discriminative to distinguish irrelevant variations (i.e., noise) and map similar finger vein features to different FVTs. Therefore, the performance of large codebook size is also unsatisfactory.

Secondly, we test the effect on verification performance when using different numbers of training samples to generate a PBPM. To be fair, the number of testing samples is fixed and the number of training samples is adjusted to generate the corresponding PBPMs. For each class in the database, we set the last two as testing samples, so that the number of training samples is between one and four. It should be pointed out that when the number of training samples is one, each patch of FVTM is regarded as the Best Patch. Therefore, the FVTM can be taken as a special case of the PBPM. The EER values are listed in Table 6. From Table 6, we can see that the performance gets better when the number of training samples increases. The lowest EER is achieved when the number of training samples reaches 4. It should be pointed out that, since the number of samples of each class in PolyU database is small, we cannot well verify whether the PBPM is convergent with the increasing number of training samples.

Table 4. The performance with different patch size (p-size)

p- size	3×3	4×4	**5×5**	6×6	7×7
EER	0.0480	0.0417	**0.0353**	0.0399	0.0419

Table 5. The performance with different codebook size (c-size)

c-size	10	**20**	30	40	50
EER	0.0457	**0.0353**	0.0401	0.0433	0.0468

Table 6. The performance with different number of training samples (num-training)

num-training	1	2	3	**4**
EER	0.0530	0.0448	0.0401	**0.0353**

3.7. Experiment 5

To verify that our method can be used in real-time applications, we measure the average processing time of key steps in our method. Table 7 shows the average time required for preprocessing, codebook training, feature extraction, PBPM training and matching. The average preprocessing time per image was 53ms; The average time of training codebook from random 100 finger vein images was 51s; The average feature extraction (i.e., FVTM generation) time per image was 32ms; The average PBPM training time for 4 samples was 0.043 ms, and the average matching time between the PBPM and the FVTM was 0.022ms. Although the average codebook training time is time-consuming, the process can be done off-line. In a word, from Table 7, we can conclude that the proposed method can be used in real-time applications.

Table 7. The average processing time

Prepro-cessing	Codebook training	Feature extraction	PBPM training	Matching
53ms	51s	32ms	0.043ms	0.022 ms

4. Conclusion

This paper proposes a novel finger vein verification method based on PBPM with vein textons. Experimental results show that our method greatly improves the performance of finger vein verification, especially under rotation conditions. The main contribution of this paper can be summarized as follows: (1) According to the regularity that some local finger vein patterns are similar and appear repeatedly in the image, we represent these similar and repeated finger vein patterns as FVTs, which can be learned from some finger vein images. So the map of FVTs (FVTM) can well represent the finger vein image. (2) To further use the differences among individuals, a specified PBPM is learned for each individual, which significantly improves the performance of finger vein verification.

It should be pointed out that the method is a general framework for finger vein verification. Besides LBP, we also used mean curvature as base feature which achieved better performance than base feature based method. In the future, we plan to test and verify the effectiveness of other base features such as LDP [29], LLBP. In addition, every FVT in the codebook is treated as equally important, so

considering the weight of FVT is also our future work.

Acknowledgement

The work is supported by National Science Foundation of China under Grant No. 61173069 and Shandong Natural Science Funds for Distinguished Young Scholar under Grant No. JQ201316. The authors would like to thank the anonymous reviewers for their helpful suggestions.

References

[1] Z. Liu, Y. L. Yin, H. J. Wang, S. L. Song, and Q. L. Li. Finger vein recognition with manifold learning. Journal of Network and Computer Applications, 33(3): 275-282, 2010.

[2] J. D. Wu, and S. H. Ye. Driver identification using finger-vein patterns with radon transform and neural network. Expert Systems with Applications, 36(3): 5793-5799, 2009.

[3] N. Miura, A. Nagasaka, and T. Miyatake. Feature extraction of finger-vein patterns based on repeated line tracking and its application to personal identification. Machine Vision and Applications, 15(4): 194-203, 2004.

[4] W. Song, T. Kim, H. C. Kim, J. H. Choi, H. J. Kong, and S. Lee. A finger-vein verification system using mean curvature. Pattern Recognition Letters, 32(11):1541-1547, 2011.

[5] N. Miura, A. Nagasaka, and T. Miyatake. Extraction of finger-vein patterns using maximum curvature points in image profiles. IEICE Transactions on Information and Systems, 90(8): 1185-1194, 2007.

[6] A. Kumar, and Y. Zhou. Human identification using finger images. IEEE Transactions on Image Processing, 21(4): 2228-2244, 2012.

[7] B. N. Huang, Y. G. Dai, and R. F. Li. Finger-vein authentication based on wide line detector and pattern normalization. In: Proceedings of 20th International Conference on Pattern Recognition, 1269-1272, 2010.

[8] Z. B. Zhang, S. L. Ma, and X. Han. Multiscale feature extraction of finger-vein patterns based on curvelets and local interconnection structure neural network. In: Proceedings of 18th International Conference on Pattern Recognition. 2006, 145-148.

[9] E. C. Lee, H. C. Lee, and K. R. Park. Finger vein recognition using minutia-based alignment and local binary pattern-based feature extraction. International Journal of Imaging Systems and Technology, 19(3): 179-186, 2009.

[10] B. A. Rosdi, C. W. Shing, and S. A. Suandi. Finger vein recognition using local line binary pattern. Sensors, 11(12): 11357-11371, 2011.

[11] G. P. Yang, R. Y. Xiao, Y. L. Yin and L Yang. Finger vein recognition based on personalized weight maps. Sensors, 13(9): 12093-12112, 2013.

[12] J. D. Wu, and C. T. Liu. Finger-vein pattern identification using principal component analysis and the neural network technique. Expert Systems with Applications, 38(5): 5423-5427, 2011.

[13] J. D. Wu, and C. T. Liu. Finger-vein pattern identification using SVM and neural network technique. Expert Systems with Applications, 38(11): 14284-14289, 2011.

[14] G. P. Yang, X. M. Xi, and Y. L. Yin. Finger vein recognition based on $(2D)^2$PCA and metric learning. Journal of Biomedicine and Biotechnology. 2012(2012).

[15] F. Jurie, and B. Triggs. Creating efficient codebooks for visual recognition. In: Proceedings of 10th IEEE International Conference on Computer Vision, 604-610, 2005.

[16] S. Lazebnik, C. Schmid, and J. Ponce. Beyond bags of features: spatial pyramid matching for recognizing natural scene categories. In: Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2196-2178, 2006.

[17] F.F. Li, and P. Perona. A Bayesian hierarchical model for learning natural scene categories. In: Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition, 524-531, 2005.

[18] S. Tong, and D. Koller. Support vector machine active learning with applications to text classification. The Journal of Machine Learning Research, 2: 45-66, 2002.

[19] T. Leung, and J. Malik. Representing and recognizing the visual appearance of materials using three-dimensional textons. International Journal of Computer Vision, 43(1): 29-44, 2001.

[20] Z. S. Li, J. I. Imai, and M. Kaneko. Face and expression recognition based on bag of words method considering holistic and local image features. In: Proceedings of the 10th International Symposium on Communications and Information Technologies, 1-6, 2010.

[21] N. H. Dardas, and N. D. Georganas. Real-time hand gesture detection and recognition using bag-of-features and support vector machine techniques. IEEE Transactions on Instrumentation and Measurement, 60(11): 3592-3607, 2011.

[22] P. Scovanner, S. Ali, and M. Shah. A 3-dimensional SIFT descriptor and its application to action recognition. In: Proceedings of the 15th international conference on Multimedia, 357-360, 2007.

[23] X. C. Qiu, Z. N. Sun, and T. N. Tan. Coarse iris classification by learned visual dictionary. In: Proceedings of the Second International Conference on Biometrics, 770-779, 2007.

[24] L. Yang, P. Yang, Y. L. Yin, R. Y. Xiao. Sliding window-based region of interest extraction for finger vein images. Sensors, 13(3):3799-3815, 2013.

[25] T. Ojala, M. Pietikainen, and D. Harwood. A comparative study of texture measures with classification based on feature distributions. Pattern Recognition, 29(1): 51-59, 1996.

[26] T. Ojala, M. Pietikainen, and T. Maenpaa. Multiresolution gray-scale and rotation invariant texture classification with Local Binary Patterns. IEEE Trans. IEEE Transactions on Pattern Analysis and Machine Intelligence, 24(7): 971-987, 2002.

[27] T. Ahonen, A. Hadid, and M. Pietikäinen. Face recognition with local binary patterns. In: Proceedings of the 8th European Conference on Computer Vision, 469-481, 2004.

[28] A. J. Chavez. Image classification with dense SIFT sampling: an exploration of optimal parameters. PhD thesis. Kansas State University, USA, 2012.

[29] E. C. Lee, H. Jung, and D. Kim. New finger biometric method using near infrared imaging. Sensors, 11(3): 2319-2333, 2011.