SCIENCE CHINA
Information Sciences

• RESEARCH PAPER •

May 2019, Vol. 62 052104:1–052104:16
https://doi.org/10.1007/s11432-018-9651-8

Anchor-based manifold binary pattern for finger vein recognition

Haiying LIU[1], Gongping YANG[1*], Lu YANG[2], Kun SU[1] & Yilong YIN[1]

[1]*School of Software, Shandong University, Jinan* 250101*, China;*
[2]*School of Computer Science and Technology, Shandong University of Finance and Economics, Jinan* 250014*, China*

Received 31 July 2018/Accepted 30 September 2018/Published online 1 April 2019

Abstract This paper proposes a novel learning method of binary local features for recognition of the finger vein. The learning methods existing in local features for image recognition intend to maximize the data variance, reduce quantitative errors, exploit the contextual information within each binary code, or utilize the label information, which all ignore the local manifold structure of the original data. The manifold structure actually plays a very important role in binary code learning, but constructing a similarity matrix for large-scale datasets involves a lot of computational and storage cost. The study attempts to learn a map, which can preserve the manifold structure between the original data and the learned binary codes for large-scale situations. To achieve this goal, we present a learning method using an anchor-based manifold binary pattern (AMBP) for finger vein recognition. Specifically, we first extract the pixel difference vectors (PDVs) in the local patches by calculating the differences between each pixel and its neighbors. Second, we construct an asymmetric graph, on which each data point can be a linear combination of its K-nearest neighbor anchors, and the anchors are randomly selected from the training samples. Third, a feature map is learned to project these PDVs into low-dimensional binary codes in an unsupervised manner, where (i) the quantization loss between the original real-valued vectors and learned binary codes is minimized and (ii) the manifold structure of the training data is maintained in the binary space. Additionally, the study fuses the discriminative binary descriptor and AMBP methods at the image representation level to further boost the performance of the recognition system. Finally, experiments using the MLA and PolyU databases show the effectiveness of our proposed methods.

Keywords finger vein recognition, feature learning, local linear embedding, fusion, manifold learning, anchor

Citation Liu H Y, Yang G P, Yang L, et al. Anchor-based manifold binary pattern for finger vein recognition. Sci China Inf Sci, 2019, 62(5): 052104, https://doi.org/10.1007/s11432-018-9651-8

1 Introduction

Recently, finger vein recognition has attracted increasing attention from the industry and research community. This is attributed to two aspects. First, finger veins have the basic properties as biometric traits, such as universality, distinctness, permanence, and collectability [1]. Second, the illegal use of forged and copied finger vein patterns is not easy. Owing to their subcutaneous presence and complex structure, finger veins can only be captured by near-infrared lights in vivo and can only be altered by surgical intervention. Therefore, finger vein recognition will be a promising alternative for personal identification schemes and it has attracted many efforts from different fields, such as data acquisition [2, 3], region of

* Corresponding author (email: gpyang@sdu.edu.cn)

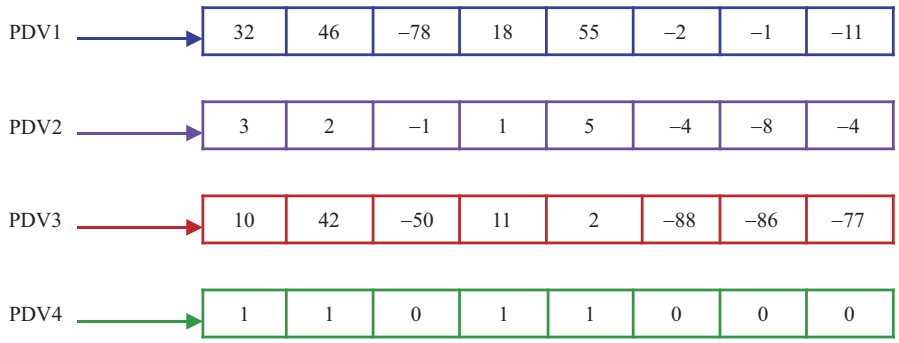

Figure 1 (Color online) Examples of LBP and various PDVs.

interest (ROI) extraction [2], image restoration [3, 4], image enhancement [5], feature extraction [6–10], and matching [11, 12].

In the above studies, feature extraction significantly affects the performance of finger vein recognition systems, because finger vein images captured in real-world environments are usually affected by many factors, such as varying finger poses, unstable lighting intensity, and occlusions. These factors reduce the similarity of finger vein images from the same finger and increase the similarity of finger vein images from different fingers, which is one of the key challenges in finger vein recognition. Therefore, it is necessary to design effective approaches of feature extraction to overcome these challenges.

Binary codes have successfully been applied to finger vein recognition [6–8,11] because they have several advantages, such as robustness to small variations caused by varying illuminations, convenience in adding prior knowledge during encoding, and low computational and storage cost owing to using the hamming distance. Despite their success, these binary features are vulnerable to image noise or degradation [13]. For example, two operations are involved in extracting local binary patterns (LBPs) [11]. One is to calculate the difference between the current pixel and its neighbors to obtain a pixel difference vector (PDV). The other is to threshold the PDV with a fixed value, ultimately forming an LBP for this pixel. The advantage of LBP is that it retains the direction information of PDV and makes the code invariant to monotonic photometric changes. Nevertheless, LBP has two disadvantages. First, the scope of extraction is limited. If the extraction range is extremely wide, the LBP will be extremely long. This is because the length of LBP is equal to the length of PDV. Although a long code can exploit the rich information from the image, the computational cost and storage requirement are increased. Second, a fixed value is used as the threshold and the amplitude information of PDV is lost. This causes different PDVs to have the same LBP code, and the intrinsic structure of all PDVs in the training set is not utilized during encoding. As shown in Figure 1, various PDVs are converted to the same exact LBP code. Specifically, PDV1, PDV2, and PDV3 in Figure 1 are significantly different, but their LBP codes are the same. However, different PDVs may contain different information; therefore, different PDVs may have different binary codes. Moreover, these problems also exist in other handcrafted binary code methods, such as local line binary pattern (LLBP) [8], polydirectional local line binary pattern (PLLBP) [6], and personalized best bit map (PBBM) [7].

To overcome the above disadvantages of handcrafted binary codes, recently, many authors use feature learning methods to extract the local binary features from images and achieve promising results in the field of recognition. Lu et al. [14] proposed a compact binary feature descriptor (CBFD) by learning a hashing map to project PDVs extracted from training images into binary codes by enforcing three constraints, namely maximizing the variance of all binary codes in the training set, making the binary codes evenly distributed, and reducing quantitative errors. Through experiments, CBFD was found to achieve a remarkable performance on several face databases. Most recently, Lu et al. [15] presented a local feature learning approach for face recognition, which simultaneously learns the projection matrix and the dictionary. Duan et al. [16] proposed a context-aware local binary feature learning for face recognition, which learns a map by utilizing the contextual information within each binary code. Liu et al. [17] proposed a discriminative binary descriptor (DBD) by using the label information to project

PDVs into binary codes by learning, and it achieved a better performance on finger vein databases.

However, the above-mentioned feature learning methods either maximize the data variance, reduce the quantitative errors, exploit the contextual information within each binary code, or utilize the label information, but none of them captures the manifold structure of the original data. However, many studies [18–20] prove that preserving the manifold structure of the original data is very important to learn the binary codes. Moreover, Ref. [21] mentioned that the features extracted from a large-scale dataset of multiple classes usually have the manifold structure in the original space and it may not be easily separated. This comment is consistent with the characteristics of PDVs in our task. This is because the PDVs in our work are extracted from images of multiple classes and many of them are large. For instance, when using three samples per class as training images on the MLA database, the number of classes is 636 and the number of PDVs is 2083536. Therefore, we doubt if preserving the manifold structure of the original dataset during the binary feature learning will improve the performance of our system.

To preserve the manifold structure of PDVs in the hamming space, we first need to build a similarity matrix of the original dataset to formulate an objective function, and then seek an effective optimization method to solve the function. However, as aforementioned, the number of PDVs in our problem is very large, and constructing a similarity matrix for them is extremely time-consuming and memory intensive. Moreover, the task of optimizing an objective function, which includes the constraint of the manifold for the large-scale dataset, is a challenge. Therefore, how to use the information of the manifold structure in large quantities of the original PDVs to obtain the binary codes is our main task.

Recently, Weiss et al. [18] proposed a spectral hashing (SH), which is aimed at keeping neighbors in the input space as neighbors in the hamming space and making the learned binary codes balanced and uncorrelated. For a larger-scale dataset, solving the problem of SH is a challenge. Weiss et al. [18] assumed that the distribution of data is uniform, and a sub-optimal solution is obtained for SH. However, this assumption is seldom satisfied in practical applications. To alleviate this problem, Liu et al. [19] designed a method called asymmetric graph hashing (AGH) to approximate the similarity between a pair of data points on a graph by leveraging a small number of anchors. Experiments showed that AGH can achieve a significant performance gain in retrieving semantically similar neighbors. However, there are two shortcomings of AGH. First, constructing the asymmetric graph uses pairwise distances that are not robust to noise and could not always accurately capture the subspace relationship among data points. Second, computing the hashing codes for unseen data is a problem, i.e., the out-of-sample problem. More recently, inspired by locally linear embedding (LLE) [21], Ji et al. [20] proposed an anchor-based discrete locality linear embedding hashing (anchor-DLLH) to preserve the manifold structure for large-scale datasets, in which a data point in the dataset is reconstructed by a linear combination of its K-nearest neighbors in the anchors. There are three merits of this method. One is that a reconstructed-based method for building the asymmetric graph is robust to noise and outliers. Second is that constructing an asymmetric graph via LLE is computationally effecient and conceptually simple. The last is that a map that is directly learned in the training stage is used to address the out-of-sample problem.

Motivated by the above studies, in this paper, we propose an anchor-based manifold binary pattern (AMBP) learning method for finger vein recognition. First, considering that PDVs are used widely and successfully in the image recognition domain [14–16], we extract the PDV for each pixel from each image in the training set. Second, to reduce the storage and computational cost of constructing the similarity matrix for a large-scale dataset, we adopt the anchor-based mode to construct the asymmetric graph, on which each data point is represented linearly by its K-nearest neighbor anchors. Specifically, the constructed graph in our approach takes $O(N \times M \times d)$ time complexity and $O(N \times M)$ storage complexity, whereas SH takes $O(N^2 \times d)$ time complexity and $O(N^2)$ storage complexity, where N is the number of data, M ($M \ll N$) is the number of anchors, and d is the dimension of data. Third, we design an objective function to learn a map that projects each PDV into a binary code. In the learning process, two main requirements are considered: (i) the quantization loss between the original PDVs and the learned binary codes are minimized and (ii) the manifold structure in the original space is maintained in the learned binary hamming space. These are necessary so that the learned binary codes would not

only retain the energy of the original data, but also maintain their similarity relationship. Moreover, it should be noted that the similarity matrix is not explicitly constructed during learning as this can largely reduce the memory. Once the map is obtained, we could transform the PDV extracted from each pixel in the image to a binary code. Obviously, the obtained binary codes preserve the proximity of original data space in the binary code space. After obtaining the binary code for each pixel in the image, considering that the bag-of-words framework has successfully been used in the field of finger vein recognition [9, 17, 22–24], we represent each finger vein image as a histogram by the bag-of-words framework. Finally, the experimental results on two public finger vein databases are used to compare the performance of the proposed method to other existing finger vein recognition methods.

In addition, considering the difference between DBD and AMBP (e.g., DBD utilizes the class information during the feature learning, whereas AMBP extracts the manifold structure of original data), we fuse DBD and AMBP at the image representation level to boost the recognition performance, and we compare the proposed fusion method with DBD and AMBP. The experimental results show that the fusion method performs better than DBD and AMBP.

The rest of this paper is organized as follows. In Section 2, we review the related work. Our proposed method is described in detail in Section 3. The experimental results that validate the proposed methods in comparison with the conventional methods are presented in Section 4. Finally, the conclusion and direction of future work are presented in Section 5.

2 Related work

In this section, we will briefly discuss two topics, namely finger vein image representation and construction of similarity matrix methods, which are closely related to our work.

2.1 Finger vein image representation

Finger vein image representation can be roughly grouped into two categories: low-level representation and high-level representation.

The methods of the low-level representation, which are directly based on the gray values in the image, include minutia-based methods [25,26], network-based methods [12,27–29], local feature methods [6,8,30], and holistic feature methods [31–33]. In general, there are several problems with the low-level representation of the finger vein image. First, the number of minutiae in the finger vein image is extremely small; thus, minutia-based methods are not suitable for finger vein image representation. Second, the quality of the captured image is usually low in the public finger vein databases. The low-quality image could affect the performance of both network-based method and local feature-method for finger vein recognition. For the network-based methods, the low-quality image might not be segmented properly; thus, the performance of recognition using vein pattern segmented from these low-quality images could be degraded. For the local feature-based methods, only original intensity comparisons are utilized; consequently, the local features extracted from these low-quality images also result in low recognition rate, but there are many methods [7, 10, 22, 23] proposed to overcome the shortcomings. Third, at present, the number of samples for each class in the public finger vein database is very small. Owing to scarcity of training samples, the performance of holistic feature methods is not satisfactory for finger vein recognition. In summary, the methods of low-level representation are primarily limited by the quality of the finger vein image and the number of available training images.

To overcome the shortcomings of low-level representation, recently, methods of high-level representation for finger vein recognition are proposed. These are usually based on the low-level finger vein representation such as superpixel-based feature [22, 23], vein textons map [24], hyperinformation feature [9], and personalized feature [7]. In general, these methods use the existing local handcrafted feature as their base feature and the semantics are extracted from the image through a certain method. For example, superpixel-based feature [22, 23], vein textons map [24], and hyperinformation feature [9] use the bag-of-words framework to extract the high-level semantic from the images, and the personalized

feature [7] draws the semantic features based on the label information of the finger vein image. Hence, the methods of high-level representation for finger vein images are suitable to low-quality images and they achieve a high performance on finger vein recognition. However, such methods are usually based on local handcrafted features; thus, the shortcomings of local handcrafted features are inherited. More recently, to tackle this issue, Liu et al. [17] proposed a local feature learning approach for finger vein recognition, which can achieve a high performance.

Therefore, in this study, we still adopt the local feature learning method to represent the finger vein image.

2.2 Constructing similarity matrix methods

Over the past two decades, a large amount of graph-based algorithms have been proposed with various applications such as dimension reduction [34,35], clustering [36–40], and hashing [18,19,41]. Building the similarity graph is a key step in these algorithms. If the graph can accurately mine the neighborhood of each data point, the performance of the algorithms will improve even when the dataset is corrupted by interference factors.

In general, the methods of constructing the similarity graph can roughly be divided into two categories: pairwise distance-based method and reconstruction-based method. The pairwise distance-based method includes the K-nearest neighbor method and ϵ ball-based method [36]. When data have noise or are coming from multiple dependent subspaces, the quality of the similarity graph constructed by these methods will degrade. The reconstruction-based method includes locally linear representation (LLR) [38], sparse representation (SR) [39,42], and lower rank representation (LRR) [40]. In LLR, each data point in the dataset can be written as a linear combination of its K-nearest neighbor data points. The advantages of LLR are as follows: it is robust to noise, suitable to nonlinear subspaces, and easy to solve. However, LLR suffers from the same disadvantage of pairwise distance-based method in that it is not always the case that a point and its K-nearest neighbors are in the same subspace. Moreover, choosing the number of nearest neighbors is challenging. These problems could be solved by SR and LRR. Different from LLR, SR is based on the idea of reconstructing a data point as a sparse linear combination of all other data points. SR is very robust to noise and the number of neighbors of a data point is automatically chosen. Nonetheless, a possible disadvantage of SR is that it can be slow because it requires solving an L_1 norm optimization problem. In addition, inaccuracy in capturing the global structures of data would be its other disadvantage because it finds the sparsest representation of each data vector individually. To overcome the last issue of SR, LRR is proposed so that every data point can be reconstructed as the lowest-rank representation of all other data. Similar to SR, the solution of LRR is slow because it is necessary to solve a nuclear norm optimization problem.

From the above analysis, LLR is more robust to noise compared to pairwise distance-based method, and it is more efficient for calculating the similarity graph and more capable of capturing the nonlinear structure of data compared to SR and LRR. Therefore, in this study, we choose LLR to construct the similarity graph.

3 Anchor-based manifold binary pattern

In general, most of the finger vein recognition methods based on learning include two major stages: training stage and testing stage. The same is true for AMBP. Figure 2 illustrates the training stage and testing stage based on AMBP. The training stage can be divided into several steps. First, PDVs are extracted from the training images (after preprocessing). Second, a feature map is learned by our AMBP from these PDVs. Third, the binary codes are obtained by projecting these PDVs using the feature map. Lastly, a codebook is learned by the k-means method from the obtained binary codes. In the testing stage, after the binary codes are extracted from an image, we pool these binary codes to the codebook as a histogram representation for this image. Furthermore, we also propose a fusion strategy to boost the performance of the finger vein recognition system.

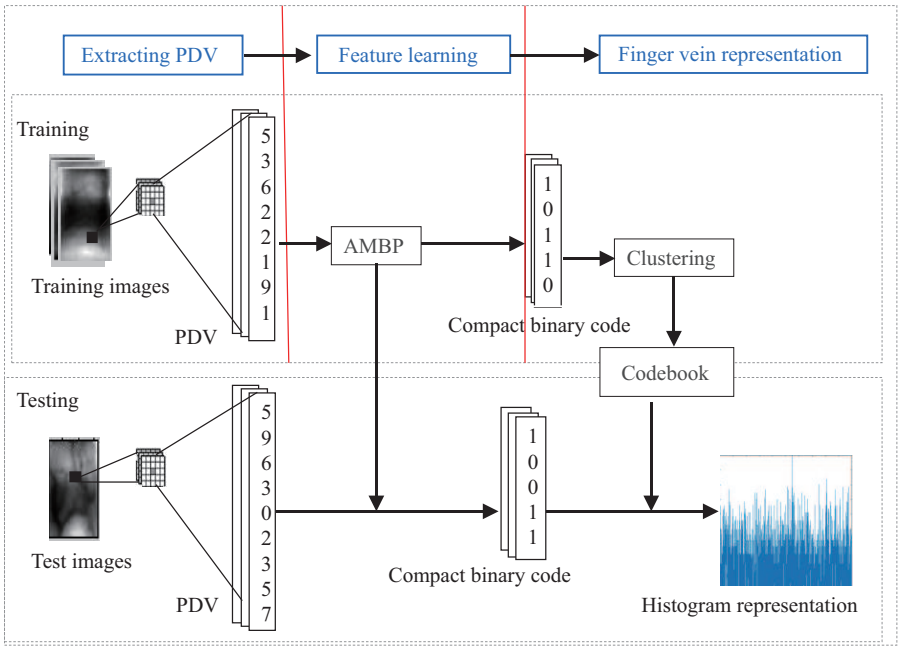

Figure 2 (Color online) Schematic of the proposed feature learning-based finger vein representation and recognition method.

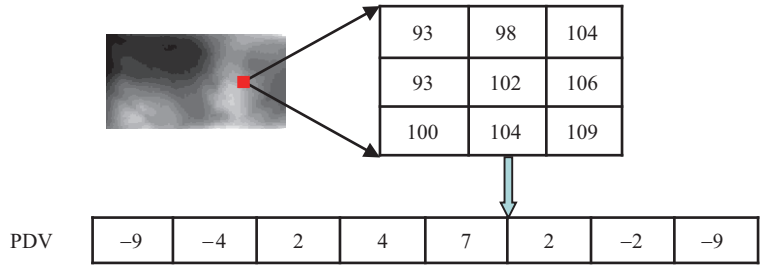

Figure 3 (Color online) Illustration of PDV operator in our method.

In the next part, we first describe AMBP, which involves extracting PDVs and feature learning. Then, we present the finger vein representation using AMBP and the fusion method, respectively.

3.1 Extracting pixel difference vectors

PDVs have been widely and successfully used in the image recognition domain [14–16]. Therefore, we select PDVs as raw data for learning in the proposed method. In the following, we will introduce how to extract PDVs from the finger vein images.

Figure 3 illustrates the extraction of PDVs from the finger vein image. Specifically, given a patch extracted at a pixel in the finger vein image, we first calculate the difference between this pixel and its neighbors. These differences are considered as PDVs, whose size is $(2D+1)\times(2D+1)$. In Figure 3, D is selected as 1, so that there are 8 neighboring pixels selected, and in this case, PDV is an 8-dimensional real-valued feature vector. While PDVs include a lot of information related to the patch, they are real-valued and high-dimensional data if D is large. For example, if D is 6, the dimension of PDV is 169. In Subsection 3.2, we will introduce how to learn a discriminative map, which can project the high-dimensional and real-valued PDVs into low-dimensional binary codes.

3.2 AMBP feature learning

Different from our previous work [17], the feature learning method in this study is unsupervised. Several important criteria are adopted in the objective function of feature learning and an efficient technique is

used to optimize these objective functions in the proposed method. In the following, we will describe the objective function and the optimization method in our finger vein recognition system.

Let $X = [x_1, x_2, \ldots, x_N] \in \mathbb{R}^{d\times N}$ be a set of N real-valued d-dimensional PDVs extracted from training images; the proposed AMBP method aims to learn K-hashing functions that map each high-dimensional real-valued PDV x_i into a low-dimensional binary code $b_i = [b_{1i}, b_{2i}, \ldots, b_{Ki}]^{\mathrm{T}}$. Let w_k denote a projection vector for the k-th hashing function; then, the binary code b_{ki} of x_i can be computed by a hashing function $b_{ki} = 0.5 \times (\mathrm{sgn}(w_k^{\mathrm{T}} x_i) + 1)$, where $\mathrm{sgn}(h)$ is equal to 1 if $h \geqslant 0$ and -1 otherwise.

In order to make the binary codes compact and preserve the intrinsic structure of the original dataset, the objective function is formulated as follows:

$$\begin{aligned} &\min\ J_1(W) = \lambda_1 \|(B - 0.5) - W^{\mathrm{T}} X\|_F^2 + \frac{1}{2}\lambda_2 \mathrm{tr}(BLB^{\mathrm{T}}) + \lambda_3 \|W\|_F^2 \\ &\text{s.t. } B \in \{0,1\}^{K\times N}, \quad W^{\mathrm{T}} W = I, \end{aligned} \tag{1}$$

where $B = [b_1, b_2, \ldots, b_N] \in \{0,1\}^{K\times N}$ is the binary code matrix; $W = [w_1, w_2, \ldots, w_K] \in \mathbb{R}^{d\times K}$ is the projection matrix; $L = D - S$ is the graph Laplacian matrix; S is an $N \times N$ similarity matrix of N PDVs, which measures the similarity between every two data points in the training set; $D = \mathrm{diag}(S\mathbf{1})$; $\mathbf{1}$ is a column vector with N entries of 1; tr denotes the matrix trace operation, and λ_1, λ_2, and λ_3 are the tradeoff parameters.

The first term in (1) is the quantization error between the original PDVs and the learned binary codes, which ensures that the learned binary codes could preserve the information of the original data. The second term in (1) preserves the manifold structure of the original dataset in the binary space. The last term is the regularization term $\|W\|_F^2$, which regulates the complexity of W and is used to avoid overfitting. Additionally, $W^{\mathrm{T}}W = I$ is the orthogonal constraint.

However, as aforementioned, constructing a similarity matrix involves a lot of computational and storage cost, e.g., constructing similarity matrix S in SH has $O(N^2)$ storage complexity and the $O(d\times N^2)$ time complexity. Fortunately, many methods of hashing and clustering use asymmetric graphs to solve this problem. The hashing methods include AGH [19] and anchor-DLLH [20], whereas the clustering methods include those described in [43, 44]. In [19, 43], the authors used pairwise distances to construct asymmetric graphs, which are sensitive to noise, and to find the neighbors for a data point that may not lie in the same space. Furthermore, this method also bring extra parameters via using a Gaussian kernel function to calculate the similarity between a data point in the graph and another data point in the anchors. To attain noise resistance in the dataset and avoid extra parameters, Nie et al. [44] obtained asymmetric graphs by learning, but pairwise distances are still used. More recently, inspired by LLE [35], Ji et al. [20] constructed the asymmetric graph by representing a data point in the graph using K-nearest neighbors in the anchors. That is they use LLR to construct the asymmetric graph. As mentioned in the related work part, LLR is robust to noise and easy to calculate, and more importantly, it can capture the nonlinear manifold structure of original data. Hence, we choose LLR to construct the asymmetric graph in AMBP.

In the following, we introduce how to use LLR to construct the asymmetric graph and we explain the process of integrating the constructed graph into our objective function.

Given M anchors $A = [a_1, a_2, \ldots, a_M]$, which are randomly selected from training set X, we compute an asymmetric similarity matrix $Z \in \mathbb{R}^{N\times M}$, where Z_{ij} reflects the similarity between data point x_i and anchor a_j. Similar to LLE [35], we seek for an asymmetric similarity matrix Z, which minimizes the reconstruction error

$$\mathrm{Err}(Z) = \min \frac{1}{2} \left\| x_i - \sum_j Z_{ij} a_j \right\|^2 \qquad \text{s.t. } \sum_j Z_{ij} = 1. \tag{2}$$

According to the Lagrange multiplier, the solution to (2) can be written as (3). For the detailed

derivation, interested readers can refer to [20].

$$Z_{ij} = \frac{\sum_k C_{ijk}^{-1}}{\sum_{lm} C_{ilm}^{-1}}, \tag{3}$$

where $C_i = [C_{ijl}]$ and $C_{ijl} = (x_i - a_j)^{\mathrm{T}}(x_i - a_l)$, whereas a_j and a_l are K-nearest neighbors of x_i. Then, the original similarity matrix can be approximated by $S = Z\Lambda^{-1}Z^{\mathrm{T}}$, where $\Lambda = \mathrm{diag}(Z^{\mathrm{T}}\mathbf{1})$. Having obtained the similarity matrix, we can construct the Laplacian matrix by the following:

$$L = D - S = I - Z\Lambda^{-1}Z^{\mathrm{T}}. \tag{4}$$

According to [43], $D = I$ in above equation can be derived by the following:

$$D_{ii} = \sum_{j=1}^{n} S_{ij} = \sum_{j=1}^{n}\sum_{l=1}^{p} \frac{Z_{il}Z_{jl}}{\Lambda_{ll}} = \sum_{l=1}^{p} Z_{il}\frac{\sum_{j=1}^{n}}{\Lambda_{ll}} = \sum_{l=1}^{p} Z_{il} = 1. \tag{5}$$

By relaxing the binary constraint and substituting (4) into (1), Eq. (1) can be reformulated as follows:

$$\begin{aligned} &\min\ J(W) = \lambda_1\|(B-0.5) - W^{\mathrm{T}}X\|_F^2 + \frac{1}{2}\lambda_2\mathrm{tr}(W^{\mathrm{T}}(XX^{\mathrm{T}} - XZ\Lambda^{-1}Z^{\mathrm{T}}X^{\mathrm{T}})W) + \lambda_3\mathrm{tr}(W^{\mathrm{T}}W) \\ &\text{s.t. } W^{\mathrm{T}}W = I. \end{aligned} \tag{6}$$

By solving the above objective function, a map that captures the manifold structure of original dataset can be obtained. Moreover, during the optimization, it is not necessary to explicitly construct the graph Laplacian matrix. As already known, the storage complexity of the graph Laplacian matrix is at least $O(N^2)$, which is unaffordable to construct for a large dataset. Thus, our objective function reduces the storage cost and at the same time captures the local manifold structure.

Note that problem (6) is not convex if W and B change simultaneously, but it is convex to one of them if the other is fixed. Based on [14], we adopt an alternate method to optimize this problem.

Fixed W and obtain B. When W is fixed, Eq. (6) can be reformulated as

$$\min\ J(B) = \|(B-0.5) - W^{\mathrm{T}}X\|_F^2. \tag{7}$$

The solution to (7) can be approximated as follows:

$$B = \frac{1}{2}(\mathrm{sgn}(W^{\mathrm{T}}X) + 1). \tag{8}$$

Fixed B and learn W. When B is fixed, we can rewrite (6) as

$$\min\ J(W) = \mathrm{tr}(W^{\mathrm{T}}QW) + \lambda_1(\mathrm{tr}(W^{\mathrm{T}}XX^{\mathrm{T}}W)) - 2\mathrm{tr}((B-0.5)\times X^{\mathrm{T}}W) \qquad \text{s.t. } W^{\mathrm{T}}W = I, \tag{9}$$

where $Q = \frac{1}{2}\lambda_2(XX^{\mathrm{T}} - XZ\Lambda^{-1}Z^{\mathrm{T}}X^{\mathrm{T}}) + \lambda_3 I$.

According to the gradient descent method with the curvilinear search algorithm in [45], we can solve this problem to obtain W. We summarize the detailed procedure of the proposed AMBP method in Algorithm 1.

3.3 Finger vein representation based on AMBP and fusion methods

After obtaining the feature map matrix W, we project all PDVs of the training dataset to the binary codes. Then, we cluster the obtained binary codes into a codebook by the k-means method. Next, the histogram representation of the finger vein image is acquired by pooling the binary codes extracted from this image to the codebook.

Partitioning the finger vein image can improve the performance of the finger vein recognition system [8, 17]; thus, we divide the finger vein image into multiple non-overlapping regions and learn the feature map matrix and codebook for each region. Therefore, a histogram representation for each region of an image

Algorithm 1 AMBP algorithm

Require: X = training dataset, t = number of iterations, λ_1, λ_2, and λ_3 = parameters, K = length of the binary code, and ϵ = the convergence parameter.

Ensure: Optimized matrix W.

1: Initialize W as the top K eigenvectors of XX^{T}, which correspond to the K largest eigenvalues; initialize t as 1;
2: **repeat**
3: Set t as $t+1$;
4: Obtain B with fixed W using (8);
5: Learn W with fixed B by solving (9);
6: **until** $|W^t - W^{t-1}| < \epsilon$ and $t > 2$;
7: **return** W.

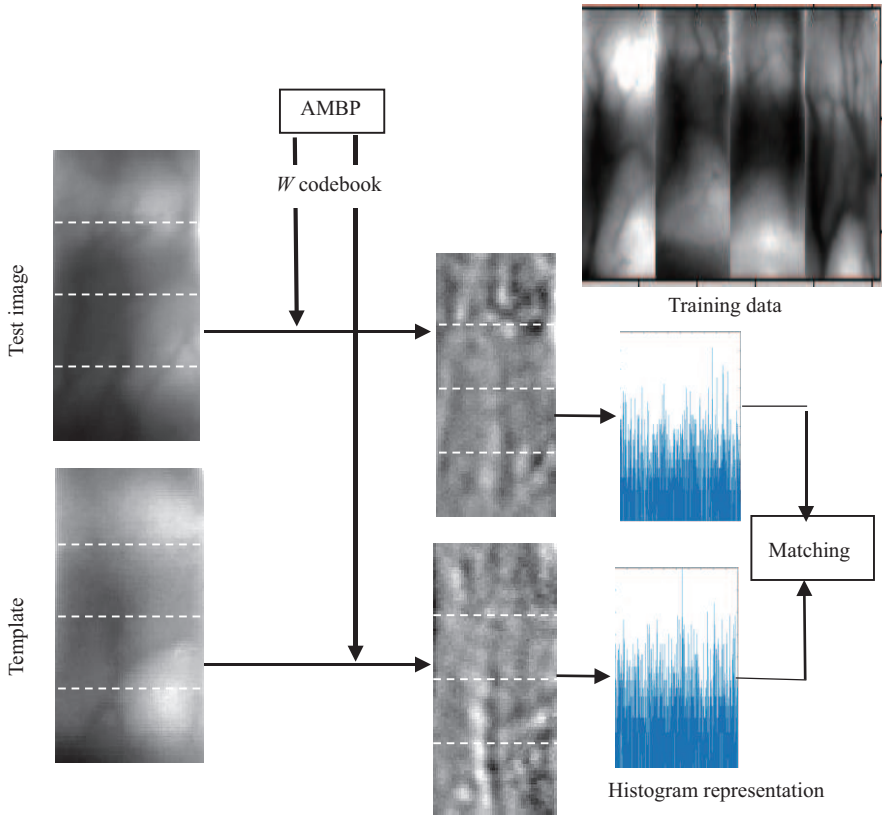

Figure 4 (Color online) Flowchart of AMBP-based finger vein representation and recognition method.

is obtained. To represent the whole image, we concatenate the histogram representations of each region in this image as the histogram feature. After this, we apply a whitened principal component analysis (WPCA) to reduce the dimension of the obtained histogram feature and compute the matching score with cosine similarity. Figure 4 shows how to use AMBP for finger vein representation.

To further enhance the performance of the recognition system, we fuse DBD and AMBP at the image representation level. Because DBD utilizes the class information during learning, the histogram representation based on it contains the semantic information of image. While AMBP preserves the local manifold structure of raw data in the binary space, the histogram representation based on it extracts the intrinsic structure of training dataset. In essence, these two representations are diverse. For example, in Figure 5, the red and blue histograms are extracted based on AMBP and DBD for a finger vein image in the left, respectively, and we can observe that there are significant differences between them. Additionally, we can see from the experimental part of this paper that DBD and AMBP both have high accuracy for finger vein recognition. Therefore, according to the principle of ensemble method [46], a good performance could be achieved by fusing these two methods. Figure 5 illustrates our fusion strategy. Given a finger vein image, we use DBD and AMBP to extract its histogram features (after WPCA), respectively. Then, these two

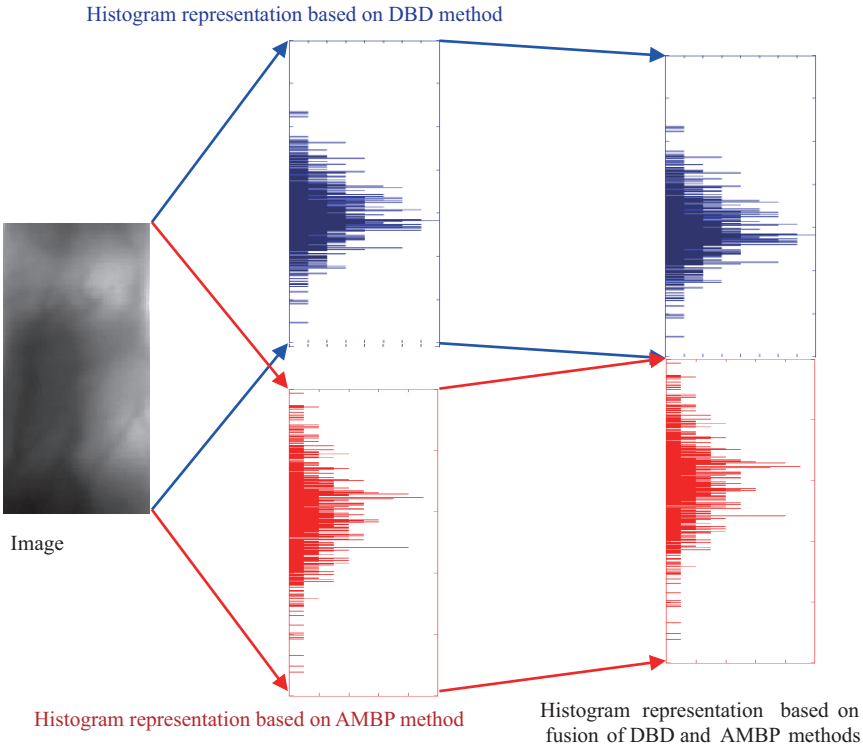

Figure 5 (Color online) Finger vein representation based on fusion of DBD and AMBP methods.

histogram features are concatenated into a final fusion representation for this image. Forming the fusion representations for other images follows the same process. After obtaining the fusion representations of images in the database, we use these fusion representations to calculate the matching score with cosine similarity. Briefly, we call this fusion method as DBD+AMBP hereafter.

4 Experiments and results

In this study, we implement the proposed methods in MATLAB R2003a and the experiments on a PC with 3.60 GHz i7-4790 CPU and 16 GB RAM. We conduct a number of experiments to evaluate our AMBP and AMBP+DBD methods on two public finger vein databases. In the following, we describe the details of the databases and experiments.

4.1 Databases

The proposed methods are verified on two public finger vein databases. One is built by the MLA Laboratory of Shandong University, named MLA database [47] and the other is constructed by Hong Kong Polytechnic University, named PolyU database [27].

MLA database. This contains 3816 finger vein images of 636 fingers. Each finger has 6 samples. The resolution of all images is 320×240 pixels.

PolyU database. This consists of 3132 finger vein images of 312 fingers. The first 210 fingers take 12 images each, and the others take 6 images each. The resolution of all images is 513×256 pixels.

Before the experiment, we preprocess each finger vein image from both databases by the methods in [27, 48] and convert each image to ROI. In both databases, the ROI size for each image is 96×64 pixels.

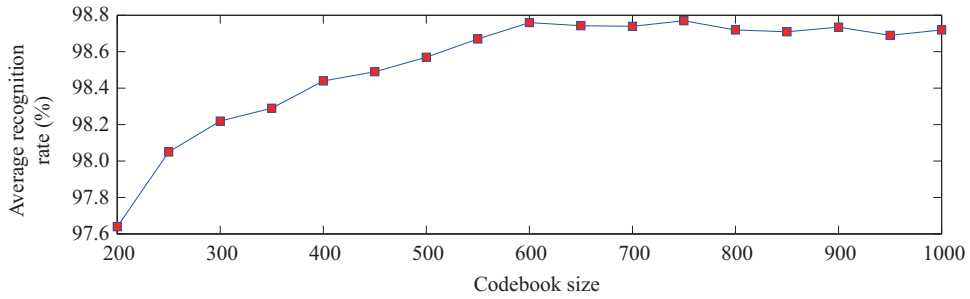

Figure 6 (Color online) Average recognition rates with various sizes of codebook on MLA database.

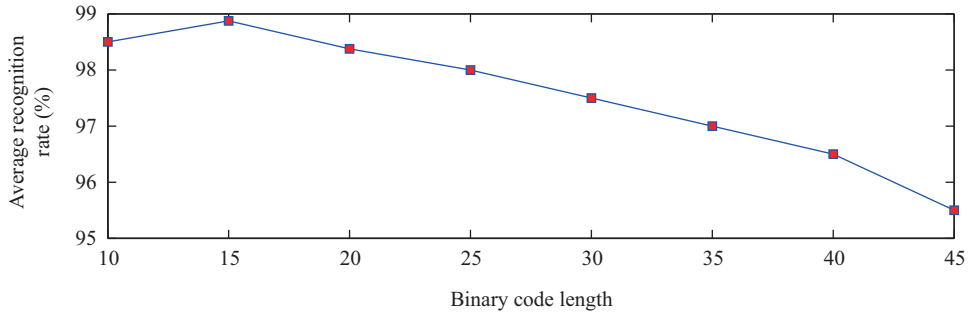

Figure 7 (Color online) Average recognition rates of binary codes with various lengths on MLA database.

4.2 Parameters determination

In this part, we investigate the best values for the codebook size, binary code length K, and number of anchors M on the MLA database, and then, we acquire their best values to be used in the experiments. Each experiment on the MLA database is repeated 10 times.

Figure 6 shows the average recognition rate of AMBP with various codebook sizes on the MLA database. We can observe that the best recognition rate can be achieved when the codebook size is 600 and 750. Therefore, considering the time consumption, we choose 600 as the codebook size.

Figure 7 shows the average recognition rate of AMBP with various binary code lengths on the MLA database. We can observe that AMBP achieves the best recognition performance when the code length is 15.

Figures 8 and 9 show the training time and the average recognition rates with various number of anchors on the MLA database, respectively. In this study, the training time comprises the time interval from inputing the training images (after preprocessing) to extracting the template representations. From Figure 8, we find that the training time shows a slight increase as the number of anchors changes from 100 to 1050 in step of 50; however, as the number of anchor points changes from 1050 to 1100, the training time shows a large change. As can be observed from Figure 9, there is a rising trend for the average recognition rate as the number of anchor points increases. Obviously, there are a few fluctuations; however, we can observe that the best average recognition rate can be obtained with the number of anchor points being 950 and 1000. Thus, in practice, we should make a tradeoff between the recognition rate and training time, and we choose 950 as the number of anchor points in the subsequent experiments.

4.3 Evaluation of AMBP and fusion methods

In this subsection, we investigate the effectiveness and efficiency of our AMBP in identification mode compared with CBFD and DBD on the MLA and PolyU databases, while the effectiveness of DBD+AMBP is evaluated on the two databases. The performance is evaluated using the average recognition rate and computational time for the key operations, which are the most common benchmarks in biometrics. The computational time includes the training time and matching time. In this study, the training time is as

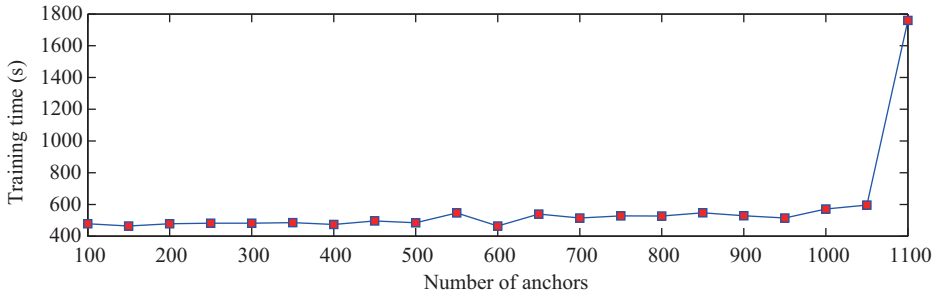

Figure 8 (Color online) Training time with various number of anchors on MLA database.

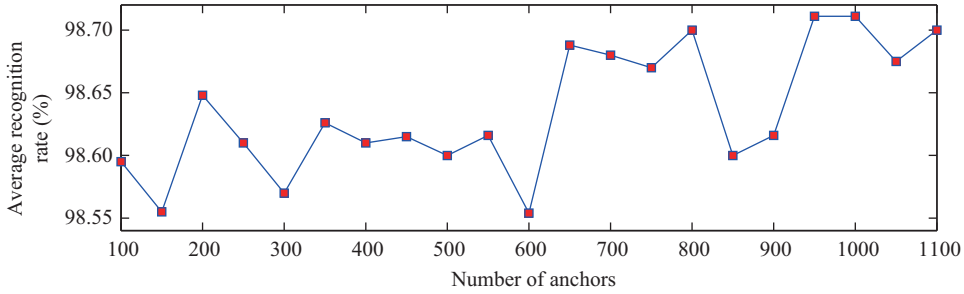

Figure 9 (Color online) Average recognition rates with various number of anchors on MLA database.

defined in Subsection 4.2, and the matching time is the interval from the input of the test image to the acquisition of the matching score.

On the MLA database, in order to show the major differences between our proposed methods against CBFD and DBD, we respectively use the first one, the first two, and the first three images of each class as training images, and we use the last three images of each class as testing images. Consequently, there are 1908 (636×3) probes, and the number of templates is 636 (636×1), 1272 (636×2), and 1908 (636×3), respectively. In order to obtain the best recognition performance for the proposed methods, we adjust the parameters through experiments. The lengths of PDV and binary code are set as 169 and 15, respectively; the parameters λ_1, λ_2, and λ_3 are set as 10^8, 10^8, and 10^7, respectively; the codebook size is 600, and the partitioning style of the finger vein image is 4×1. Therefore, we represent each finger vein image as a 2400-dimensional feature vector after using AMBP (2400=600×4×1). WPCA is used to reduce the feature dimension. Corresponding to the first one, the first two, and the first three images of each class as training images, the reduced dimensions are 300, 320, and 600, respectively. Finally, the nearest neighbor classifier with cosine similarity is applied for the finger vein matching.

On the PolyU database, we separately use the first one, the first two, and the first three images of each class as training images, and we use the last three images of each class as testing images. Consequently, there are 936 (312×3) probes, and the number of templates is 312 (312×1), 624 (312×2), and 936 (312×3), respectively. The parameter setting is the same as in the MLA database.

In all the above cases on the two databases, the probes are matched with all templates, and 10 experiments are repeated to obtain unbiased results. To achieve fairness, the optimal parameters for CBFD and DBD are obtained on the MLA database. For the sake of brevity, the computational time was calculated by only using first three samples as training set on the MLA database.

Tables 1 and 2 list the average recognition rates for the CBFD, DBD, AMBP, and DBD+AMBP methods on the MLA and PolyU databases, respectively, and Table 3 presents the computational cost on the MLA database for the CBFD, DBD, and AMBP methods.

Regarding the average recognition rates, the analysis results in Tables 1 and 2 reveal four aspects. First, the average recognition rates of the four algorithms tend to increase when the number of training set per class increases. This is because increasing the number of training samples makes the learning algorithms acquire more characteristics of samples. At the same time, the number of templates is increased as well.

Table 1 Average recognition rates of CBFD, DBD, AMBP, and DBD+AMBP on MLA database

Number of samples per class	CBFD (%)	DBD (%)	AMBP (%)	DBD+AMBP (%)
1	88.23	98.25	97.93	98.92
2	89.42	98.80	98.62	99.37
3	98.32	99.08	98.71	99.32

Table 2 Average recognition rates of CBFD, DBD, AMBP, and DBD+AMBP on PolyU database

Number of samples per class	CBFD (%)	DBD (%)	AMBP (%)	DBD+AMBP (%)
1	92.85	99.93	99.57	99.97
2	97.34	99.96	99.88	99.95
3	99.25	99.98	99.89	100

Table 3 Computational time of CBFD, DBD, and AMBP on MLA database

Method	Training time (s)	Matching time per image (ms)
CBFD	557.066	52.0
DBD	680.459	42.8
AMBP	640.735	45.0

Second, for the two unsupervised methods in Tables 1 and 2, we find that the average recognition rates of AMBP are consistently higher than those of CBFD in all three cases on the two databases. This result implies that it is important to consider the manifold structure when learning the local binary feature in the unsupervised method. Furthermore, this also proves that our assumption is correct, that is, PDVs extracted from images are located in the manifold space. Third, we observe that DBD achieves a better performance than our AMBP in all three cases on the two databases in Tables 1 and 2 because DBD is a supervised approach whereas our AMBP is unsupervised. In general, a supervised approach can extract more discriminative information from PDVs than an unsupervised one. Fourth, for our combined method, we find that DBD+AMBP obtains the best performance among all methods in all three cases on the MLA database as indicated in Table 1, and in most cases, DBD+AMBP has a higher performance than the other methods on the PolyU database in Table 2. This result proves that the fusion of DBD and AMBP at the image representation level can improve the recognition performance. This also demonstrates that the representations based on DBD and AMBP are diverse and complementary to each other.

Regarding the efficiency of AMBP, the comparison results are presented in Table 3. The analysis results indicate two aspects. First, the matching time per image for the CBFD, DBD, and AMBP methods has no obvious difference because the basic operation for these three methods during the testing stage is the same. Second, the training times for DBD and AMBP are longer than those for CBFD on the MLA database, but the difference is not very large. This is mainly because it takes a lot of time to construct the asymmetric graph for AMBP and construct the interclass scatter matrix as well as intraclass scatter matrix for DBD, while these operations are not required for CBFD. In addition, we can observe that AMBP is faster than DBD during the training. This shows that it is very efficient to construct the asymmetric matrix for large-scale datasets using AMBP.

4.4 Comparison with existing finger vein recognition methods

In this subsection, we compare the performance of the AMBP and fusion methods with that of various state-of-the-art methods in the verification mode on the MLA and PolyU databases. The methods compared are LBP [30], LLBP [8], local derivative pattern (LDP) [11], local directional code (LDC) [49], PBBM [7], superpixel-based features (SPF) [22], superpixel context features (SPCF) [23], discriminative binary code (DBC) [50], and DBD [17]. We use the equal error rate (EER) to evaluate the system performance. EER is the value at which the false acceptance rate (FAR) is equal to the false rejection rate (FRR). The experimental settings on the two databases are as follows.

On the MLA database, three images per class are treated as training samples, and another three images are used as testing samples. A three-fold cross-validation is employed in the experiment. Con-

Table 4 Comparision with different conventional methods on two databases

Method	EER (MLA database)	EER (PolyU database)
LBP [30]	0.1027	0.0744
LLBP [8]	0.1096	0.1577
LDP [11]	0.2289	0.2361
LDC [49]	0.0887	0.0331
PBBM [7]	0.0336	0.0278
SPF [22]	0.0262	0.0181
SPCF [23]	0.0194	0.0075
DBC [50]	0.0200	0.0132
DBD [17]	0.0088	0.0055
AMBP (Proposed)	0.0109	0.0042
DBD+AMBP (Proposed)	0.0055	0.0029

sequently, the intraclass and interclass comparisons need to be performed 5724 (636×3×3) and 3634740 (636×3×635×3) times, respectively. The other settings are the same as in the previous experiments on the MLA database.

On the PolyU database, the experimental procedure and parameter settings are the same as in the MLA database in this part. Because of the different number of classes between the MLA database and PolyU database, the intraclass and interclass matching on the two databases are different. On the PolyU database, there are 2808 (312×3×3) intraclass matching and 873288 (312×3×311×3) interclass matching.

Table 4 lists the EERs of the AMBP and DBD+AMBP methods in comparison with state-of-the-art methods of finger vein recognition on the two databases. From this table, the following observations can be made.

First, compared with local handcrafted feature methods such as LBP, LLBP, LDP, LDC, and PBBM, AMBP and DBD obtain a lower EER. This is attributed to two aspects. One is that the AMBP and DBD methods are both feature learning methods, which have more data adaptability than the handcrafted methods. The other is that both AMBP and DBD methods utilize the bag-of-words framework, which effectively organize the local learned features. Second, compared with the four bag-of-words methods in Table 4, the AMBP and DBD methods achieve a better performance on the two databases than the SPF and SPCF methods. This is because the AMBP and DBD methods use the learned feature as the base feature, whereas the SPF and SPCF methods use the handcrafted feature. The learned feature has a powerful ability to describe the local patches in the image than the handcrafted feature. Third, compared with the DBC [50] learning method, the AMBP and DBD methods achieve a better performance on both finger vein databases. These results are mainly because the binary features in DBC are learned holistically, whereas the AMBP and DBD methods are local feature learning methods that apply the bag-of-words framework to effectively organize the learned binary features. Therefore, the AMBP and DBD methods are more robust to local changes in finger vein images than DBC, and they achieve a lower EER than DBC on the two finger vein databases. Fourth, compared with DBD, the unsupervised AMBP method performs less well than the supervised DBD method on the MLA database. Although the AMBP outperforms DBD on the PolyU database, the difference is not significant. This result implies that the manifold structure is important to local feature learning. Lastly, the DBD+AMBP method achieves the lowest EER among the other methods on the two databases as presented in Table 4. This also shows that the features extracted by AMBP are complementary to those extracted by DBD. Hence, AMBP and DBD have different advantages and cannot be substituted for each other.

5 Conclusion and future work

This paper first proposes a local learning feature method for finger vein recognition, which is referred to as AMBP. The proposed method can improve the finger vein recognition performance by enforcing a local manifold constraint into the objective function for feature learning. Then, to further improve

the performance of the recognition system, the fusion of DBD and AMBP for finger vein recognition is proposed. In the following, we summarize the main contributions of this work. (i) This is the first attempt to learn the local binary feature using the manifold structure of PDVs. The manifold structure has been widely used for hashing, but as far as we know, it is still not explored for learning the local binary feature of images. This is partly because the dataset PDVs extracted from each pixel in the training images are large scale. Indeed, constructing the similarity matrix is the first step for using the manifold structure of dataset, but it is difficult to construct a similarity matrix for large-scale datasets. (ii) We adopt the approach of building the asymmetric graph to construct a similarity matrix for PDVs. Each data point on the asymmetric graph can be linearly represented by its K-nearest neighbor anchors. This approach of building the asymmetric graph is simple, is more robust to data noise, and has low computational and storage costs; therefore, it is suitable to large-scale datasets. (iii) We integrate the asymmetric graph into our objective function, and then design an effective and efficient algorithm to optimize it. Three benefits can be obtained from this. First, there is no need to explicitly construct the similarity matrix for large-scale data during the optimization process. Second, we adopt an alternative and iterative algorithm to solve the objective function without having to calculate the eigenvectors of the graph Laplacian matrix, as this is a time-consuming operation for large-scale data. Lastly, a map can be obtained through optimizing our objective function and we can directly use it to project the PDVs extracted from the test images. Thus, the out-of-sample problems are solved. (iv) We fuse the DBD and AMBP methods at the representation level and we achieve the best performance on two finger vein databases. In our future work, we plan to apply the proposed algorithms to other biometric modalities, such as palm vein databases.

Acknowledgements This work was supported by National Natural Science Foundation of China (Grant Nos. 61472226, 61573219, 61703235), and Key Research and Development Project of Shandong Province (Grant No. 2018GGX101032). The authors would particularly like to thank the anonymous reviewers for their helpful suggestions.

References

1 Jain A K, Ross A, Prabhakar S. An introduction to biometric recognition. IEEE Trans Circ Syst Vid, 2004, 14: 4–20
2 Yang J F, Shi Y H. Finger-vein ROI localization and vein ridge enhancement. Pattern Recogn Lett, 2012, 33: 1569–1579
3 Yang J F, Zhang B, Shi Y H. Scattering removal for finger-vein image restoration. Sensors, 2012, 12: 3627–3640
4 Lee E C, Park K R. Image restoration of skin scattering and optical blurring for finger vein recognition. Optics Lasers Eng, 2011, 49: 816–828
5 Shin K, Park Y, Nguyen D, et al. Finger-vein image enhancement using a fuzzy-based fusion method with Gabor and retinex filtering. Sensors, 2014, 14: 3095–3129
6 Lu Y, Xie S J, Yoon S, et al. Finger vein identication using polydirectional local line binary pattern. In: Proceedings of International Conference on ICT Convergence, 2013. 61–65
7 Yang G P, Xi X M, Yin Y L. Finger vein recognition based on a personalized best bit map. Sensors, 2012, 12: 1738–1757
8 Rosdi B A, Shing C W, Suandi S A. Finger vein recognition using local line binary pattern. Sensors, 2011, 11: 11357–11371
9 Xi X M, Yang G P, Yin Y L, et al. Finger vein recognition based on the hyperinformation feature. Opt Eng, 2014, 53: 013108
10 Meng X J, Xi X M, Yang G P, et al. Finger vein recognition based on deformation information. Sci China Inf Sci, 2018, 61: 052103
11 Lee E C, Jung H, Kim D. New finger biometric method using near infrared imaging. Sensors, 2011, 11: 2319–2333
12 Yang L, Yang G P, Yin Y L, et al. Finger vein recognition with anatomy structure analysis. IEEE Trans Circ Syst Video Technol, 2018, 28: 1892–1905
13 Deng W L, Hu J N, Guo J. Compressive binary patterns: designing a robust binary face descriptor with random-field eigenfilters. IEEE Trans Pattern Anal Mach Intell, 2019, 41: 758–767
14 Lu J W, Liong V E, Zhou X Z, et al. Learning compact binary face descriptor for face recognition. IEEE Trans Pattern Anal Mach Intell, 2015, 37: 2041–2056
15 Lu J W, Liong V E, Zhou J. Simultaneous local binary feature learning and encoding for homogeneous and heterogeneous face recognition. IEEE Trans Pattern Anal Mach Intell, 2018, 40: 1979–1993
16 Duan Y Q, Lu J W, Feng J J, et al. Context-aware local binary feature learning for face recognition. IEEE Trans Pattern Anal Mach Intell, 2018, 40: 1139–1153
17 Liu H Y, Yang L, Yang G P, et al. Discriminative binary descriptor for finger vein recognition. IEEE Access, 2018, 6: 5795–5804
18 Weiss Y, Torralba A, Fergus R. Spectral hashing. In: Proceedings of Advances in Neural Information Processing Systems, 2008. 1753–1760

19 Liu W, Wang J, Sanjiv K, et al. Hashing with graphs. In: Proceedings of the 28th International Conference on Machine Learning (ICML), 2011
20 Ji R R, Liu H, Cao L J, et al. Toward optimal manifold hashing via discrete locally linear embedding. IEEE Trans Image Process, 2017, 26: 5411–5420
21 Irie G, Li Z G, Wu X M, et al. Locally linear hashing for extracting non-linear manifolds. In: Proceedings of IEEE Conference on Computer Vision and Pattern Recognition, 2014. 2123–2130
22 Liu F, Yin Y L, Yang G P, et al. Finger vein recognition with superpixel-based features. In: Proceedings of IEEE International Joint Conference on Biometrics, 2014
23 Zhou L Z, Yang G P, Yin Y L, et al. Finger vein recognition based on stable and discriminative superpixels. Int J Patt Recogn Artif Intell, 2016, 30: 1650015
24 Dong L M, Yang G P, Yin Y L, et al. Finger vein verification based on a personalized best patches map. In: Proceedings of International Joint Conference Biometrics (IJCB), 2014
25 Yu C B, Qin H F, Cui Y Z, et al. Finger-vein image recognition combining modified Hausdorff distance with minutiae feature matching. Interdiscip Sci Comput Life Sci, 2009, 1: 280–289
26 Lee E C, Lee H C, Park K R. Finger vein recognition using minutia-based alignment and local binary pattern-based feature extraction. Int J Imag Syst Technol, 2009, 19: 179–186
27 Kumar A, Zhou Y B. Human identification using finger images. IEEE Trans Image Process, 2012, 21: 2228–2244
28 Song W, Kim T, Kim H C, et al. A finger-vein verification system using mean curvature. Pattern Recogn Lett, 2011, 32: 1541–1547
29 Miura N, Nagasaka A, Miyatake T. Feature extraction of finger-vein patterns based on repeated line tracking and its application to personal identification. Machine Vision Appl, 2004, 15: 194–203
30 Lee H C, Kang B J, Lee E C, et al. Finger vein recognition using weighted local binary pattern code based on a support vector machine. J Zhejiang Univ Sci C, 2010, 11: 514–524
31 Wu J D, Liu C T. Finger-vein pattern identification using principal component analysis and the neural network technique. Expert Syst Appl, 2011, 38: 5423–5427
32 Yang G P, Xi X M, Yin Y L. Finger vein recognition based on $(2D)^2$ PCA and metric learning. J Biomed Biotech, 2012, 2012: 1–9
33 Guan F X, Wang K J, Liu J Y, et al. Bi-direction weighted $(2D)^2$ PCA with eigenvalue normalization one for finger vein recognition. Pattern Recogn Art Intell, 2011, 24: 417–424
34 Li Y Y, Lu R Q. Locality preserving projection on SPD matrix Lie group: algorithm and analysis. Sci China Inf Sci, 2018, 61: 092104
35 Roweis S T. Nonlinear dimensionality reduction by locally linear embedding. Science, 2000, 290: 2323–2326
36 Luxburg U V. A tutorial on spectral clustering. Stat Comput, 2007, 17: 395–416
37 Elhamifar E, Vidal R. Sparse subspace clustering. In: Proceedings of IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2009. 2790–2797
38 Goh A, Vidal R. Segmenting motions of different types by unsupervised manifold clustering. In: Proceedings of IEEE Conference on Computer Vision and Pattern Recognition, 2007
39 Eldar Y C, Mishali M. Robust recovery of signals from a structured union of subspaces. IEEE Trans Inf Theory, 2009, 55: 5302–5316
40 Liu G C, Lin Z C, Yu Y. Robust subspace segmentation by low-rank representation. In: Proceedings of the 27th International Conference on Machine Learning (ICML), 2010. 663–670
41 Liu W, Mu C, Kumar S, et al. Discrete graph hashing. In: Proceedings of the 27th International Conference on Neural Information Processing Systems, 2014. 3419–3427
42 Wright J, Yang A, Ganesh A, et al. Robust face recognition via sparse representation. IEEE Trans Pattern Anal Mach Intell, 2008, 31: 210–227
43 Cai D, Chen X L. Large scale spectral clustering via landmark-based sparse representation. IEEE Trans Cybern, 2015, 45: 1669–1680
44 Nie F P, Zhu W, Li X L. Unsupervised large graph embedding. In: Proceedings of the 31st AAAI Conference on Artificial Intelligence, 2017. 2422–2428
45 Wen Z W, Yin W T. A feasible method for optimization with orthogonality constraints. Math Program, 2013, 142: 397–434
46 Zhou Z H. Ensemble Methods: Foundations and Algorithms. Boca Raton: Chapman and Hall/CRC, 2012
47 Yin Y L, Liu L L, Sun X W. Sdumla-hmt: a multimodal biometric database. In: Proceedings of Chinese Conference on Biometric Recognition, 2011. 260–268
48 Yang L, Yang G P, Yin Y L, et al. Sliding window-based region of interest extraction for finger vein images. Sensors, 2013, 13: 3799–3815
49 Meng X J, Yang G P, Yin Y L, et al. Finger vein recognition based on local directional code. Sensors, 2012, 12: 14937–14952
50 Xi X M, Yang L, Yin Y L. Learning discriminative binary codes for finger vein recognition. Pattern Recogn, 2017, 66: 26–33