Binary Search Path of Vocabulary Tree Based Finger Vein Image Retrieval

Kuikui Wang, Lu Yang, Kun Su, Gongping Yang*, Yilong Yin
School of Computer Science and Technology, Shandong University, Jinan, 250101, P.R. China
*Corresponding author:gpyang@sdu.edu.cn

Abstract

Many related studies have reported promising results in finger vein recognition, but it is still challenging to perform robust image retrieval, especially in the application scenarios with large scale populations. With the purpose in consideration, this paper presents a binary search path of hierarchical vocabulary tree based finger vein image retrieval method. In detail, a vocabulary tree is built based on the local finger vein textons by the hierarchical k-means method. Each image patch is represented by the binary path in the search of its most similar leaf node, and the value of each bit in the path is labeled as 1 or 0 according to whether the corresponding node is passed or skipped in search. The similarity of two images is defined as the number of overlapped bits in all involved path pairs. And, the enrolled images with top t scores in the sorted score vector will be selected as candidates to narrow the search space. Experimental results on five finger vein databases confirm that the proposed method can improve the retrieval performance on both accuracy and efficiency.

1. Introduction

Finger vein recognition uses the internal vein pattern in finger palmar sider to perform identity authentication. In finger vein image capturing, the near infrared light with wavelength between 700 and 1000 nanometers is always used, as the light can be intensively absorbed by the hemoglobin in vein, but easily transmit other finger tissues [4]. Compared with other biometrics, finger vein has some unique advantages [20, 12], which mainly focus on four points: (1) There is no need to worry about the duplication of finger vein by touching somewhere, as finger vein is under the skin; (2) Vein pattern can only be captured in a living finger, so vein pattern based recognition has high security; (3) The size of image capturing device is quite small and portable; and (4) For most of us, everyone can get 10 finger vein patterns at most and all of them are discriminative, hence it is alternative and flexible to perform identity authentication. With these advantages, finger vein recognition attracts lots of research interests and it is regarded as one promising biometric technique.

Although many accomplishments have been achieved in finger vein recognition, one ignored but existing challenge is how to improve the matching efficiency. This problem is extremely tough in the application scenarios with large scale populations. Fortunately, image classification and image retrieval are two effective solutions to the problem. In the first solution, images are categorized into several classes, and the probe image is only matched with the enrolled images in one specific class to reduce the matching cost. There are some related examples, *i.e.*, local ridge distribution models based fingerprint classification [5], directional pattern and singularity feature based fingerprint classification [3] and hierarchical visual codebook based iris classification [14]. There are also several finger vein image related studies [17, 13, 15]. The hierarchal classification framework [17] is a representative example. In the framework, image quality features (*i.e.*, gradient, image contrast and information entropy) and image content features (*i.e.*, moment invariant feature, wavelet coefficient and Gabor filters) are respectively used in the first and second layer. However, quality features will cause the imbalance problem between different finger vein categories, as the number of high quality images is much larger than it of low quality images. So, image quality based classification cannot thoroughly overcome the matching efficiency problem.

Compared with image classification, in which the probe image will be matched with a part of the enrolled images, image retrieval is more available to narrow down the matching space. The reason is that only top t candidates will be selected and further matched with the probe image by image retrieval. The retrieval techniques has been investigated in many biometric traits, for instance, fingerprint geometric hashing [19] and local feature based iris retrieval [9]. Recently, locality sensitive hashing (LSH) was derived to conduct finger vein image retrieval [18], which is only one existing image retrieval related study in finger vein recognition. In the method, the vein pattern was extracted by the repeated line tracking [10], and it was further encoded according to whether the number of vein point in an image

978-1-5090-1869-7/16/$31.00 ©2016 IEEE

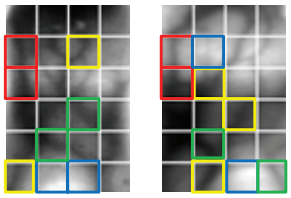


Figure 1. Examples of similar local vein patterns. These two images are from different fingers, but some local vein patterns in the rectangle with same color are very similar.

patch is above the predefined threshold or not. Based on the encoded vector, the candidates are returned by LSH principle. However, there are several limitations in the method: (1) About base feature extraction, the repeated line tracking method is quite time-consuming, especially for large size images; and (2) About the LSH, the random variation of hash function may have big influence on the retrieval performance, and the additional memory is required to store the hash tables.

This paper tries to employ the intrinsic characteristics of finger vein in image retrieval to further improve the retrieval accuracy and efficiency. In finger vein images, some local vein patterns are similar and appear repeatedly (as shown in Fig. 1), which are viewed as vein textons in Ref. [2]. And the paper has proved that vein textons were very effective in finger vein image representation. As the textons can express the local characteristics of finger vein, *i.e.*, the shape, the growth direction and so on, we also use vein textons to represent finger vein image. In addition, to do fast retrieval, it is better to encode the representation into binary pattern. Considering the advantages of tree based retrieval in speed and capacity [11], we use a hierarchical vocabulary tree with vein textons as nodes to conduct the encoding process. In one word, this paper attempts to use the binary pattern representation of finger vein image to perform image retrieval by integrating vein textons and the hierarchical vocabulary tree.

More specifically, the binary search path in the hierarchical vocabulary tree of finger vein image is employed in image retrieval in this paper. The hierarchical vocabulary tree is built based on the local features of image patches by multilayer k-means. All the clustering centers in k-means are viewed as finger vein textons, and will be used as the nodes in the vocabulary tree. For each patch in the probe image, its most similar leaf node in the built tree is searched from root node, and all the tree nodes are labeled as 1 or 0 according to whether the node is passed or skipped in search. The labeled binary search path of all patches will be used as the retrieval feature of one image. The similarity scores between the sparse binary paths of the probe and enrolled images are measured by the number of overlapped bits, and sorted to return the top t candidates.

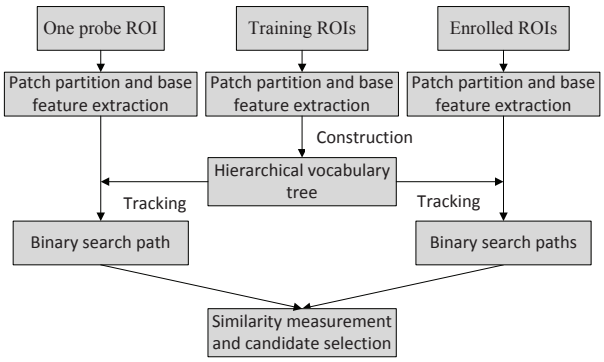


Figure 2. Flow diagram of proposed retrieval method.

The rest of the paper is organized as follows. Section 2 describes the proposed retrieval method in detail. Section 3 reports the experimental results and gives a comprehensive analysis. Finally we draw the conclusion in section 4.

2. Proposed Retrieval Method

To perform finger vein image retrieval, we firstly construct the hierarchical vocabulary tree based on the local feature of image patches by iterating k-means. Then, we represent each patch in one image by the binary search path of the patch's most similar leaf node. Last, the similarities of the probe image and all enrolled images are defined as the number of overlapped bits in all patch pairs, and based on the similarity scores the candidates will be selected. The flow diagram of the proposed binary search path based finger vein image retrieval method is shown in Fig. 2. In the following, we will detail the related three steps, mainly including hierarchical vocabulary tree construction, binary search path tracking, and candidate selection.

2.1. Finger Vein Texton based Hierarchical Vocabulary Tree Construction

We exploit finger vein textons to construct the hierarchical vocabulary tree. Finger vein textons have two advantages in image representation: (1) These textons carry the intrinsic characteristics of finger vein, for example, the local shape and growth direction of vein branch; and (2) It is very effective to represent finger vein images by the vein textons. Considering these advantages, the finger vein textons are learned and further used as tree nodes to represent finger vein image.

The process of the tree construction can be regarded as the hierarchical k-means clustering of image patches. Similar to the existing study [2], k-means is used in this paper for finger vein texton leaning. But the difference is that we employ the iteration of k-means to construct the hierarchical vocabulary tree. The training images are firstly divided into non-overlapped image patches, and the local features are

Algorithm 1 Finger vein texton based hierarchical vocabulary tree construction

Input: Training database P , Layer number L , Cluster number in each k-means k.
Output: Hierarchical Vocabulary Tree C.

$C_0 = [c_{01}] = P$; %Root node of the tree
for i=1 to L **do** %Layer number of the tree
  for j=1 to k^{i-1} **do** %Branch number in each layer
    $c_{i\ (j-1)*k+1,...,j*k} = kmeans(c_{i-1\ j})$; %Children nodes of node $c_{i-1\ j}$
    $c_{i\ (j-1)*k+1,...,j*k} = [c_{i\ (j-1)*k+1}, c_{i\ (j-1)*k+2}, ..., c_{i\ (j-1)*k+(k-1)}, c_{i\ j*k}]$;
  end for
  $C_i = [c_{i1}, c_{i2}, ..., c_{ik}, ..., c_{i\ (j-1)*k+1}, c_{i\ (j-1)*k+2}, ..., c_{i\ j*k}, ..., c_{i\ k^i-(k-1)}, c_{i\ k^i-(k-2)}, ..., c_{i\ k^i}]$;
  %All nodes in ith layer
end for
$C = [C_1; C_2; ...; C_L]$; %All layers in the tree
Return: C.

extracted from these image patches. And then, based on the local features, we perform the hierarchical k-means clustering to construct the tree. In detail, all the cluster centers in repeated clustering can be seen as finger vein textons, and further used as nodes of the tree. The number of iterations determines the depth of tree, and the number of clusters in each iteration fixes the width of tree (*i.e.*, the number of branches). More detail, in the initial k-means clustering, all training patches are classified into k groups, and the center vectors in all groups will be used as k nodes in the first layer of the tree. And in the following clustering, the image patches in each group are clustered again by k-means to obtain new subgroups, and the center vectors are used as k children nodes of each node in the first layer, which can be seen as the second iteration. (Note that there may be several k-means clusterings in each iteration and the number of k-means clusterings depends on it of nodes in the upper layer of tree.) There are totally k^2 nodes in the second layer of the tree. In this way, the iteration may be continued more times.

Assuming there are D training images, and each image are partitioned into $M * N$ patches. For all patches, local feature is firstly extracted. Then a training local feature dataset is constructed, denoted by $P = [P_1, P_2, ..., P_d, ..., P_D]$, where P_d is the local features of the dth images. For this image, $P_d = [p_{11}, p_{12}, ..., p_{mn}, ..., p_{MN}]$, where p_{mn} is the local feature of the patch on the mth row and nth column. Based on the training dataset, the tree construction begins. The k cluster centers in first iteration are denoted by $C_1 = [c_{11}, c_{12}, ..., c_{1k}]$, in which $c_{11}, c_{12}, ..., c_{1k}$ will be used as the k nodes in the first layer of the vocabulary tree. Then the images in each cluster of the first iteration will be further grouped into k subclusters in the second iteration. Take c_{11} as an example. In the second iteration, c_{11} will be grouped into k subclusters denoted $c_{11} = [c_{21}, c_{22}, ..., c_{2k}]$. All obtained new cluster centers can be represented by

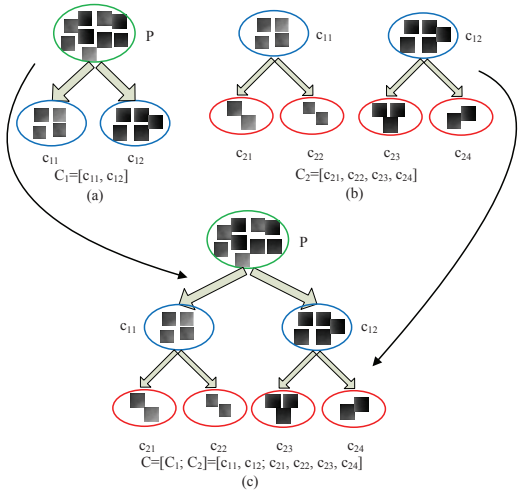


Figure 3. Construction of a tree with 2 layers and 2 branches.

$C_2 = [c_{21}, c_{22}, ..., c_{2k}, c_{2\ k+1}, c_{2\ k+2}, ..., c_{2\ k^2}]$, in which $c_{21}, c_{22}, ..., c_{2\ k^2}$ will be used as the k^2 nodes in the second layer of the tree. Like this, the iteration is continued and new nodes in the next layer of the tree will be produced. If there are L layers, the tree can be denoted:

$$\begin{aligned} C &= [C_1; C_2; ...C_L] \\ &= [c_{11}, c_{12}, ..., c_{1k}; \\ &\quad c_{21}, c_{22}, ..., c_{2k}, c_{2\ k^2-(k-1)}, c_{2\ k^2-(k-2)}, ..., c_{2\ k^2}; \\ &\quad ....; \\ &\quad c_{L1}, c_{L2}, ..., c_{Lk}, ..., c_{L\ k^L-(k-1)}, c_{L\ k^L-(k-2)}, ..., c_{L\ k^L}]. \end{aligned} \tag{1}$$

The construction process is summarized in Algorithm 1. And the construction process of a tree with 2 layers and 2 branches is shown in Fig. 3.

To construct the hierarchical vocabulary tree, we need to deal with two relevant issues ahead, which are the depth and width of tree. Although the precise values of depth and

width are determined by the retrieval performance, two related points are intelligible. The first one is that the more layers and branches of the tree, the larger time cost in the tree construction and the following retrieval feature extraction. Moreover, as the increase of the number of iterations, the cluster may not be classified, which is the second one. This phenomenon is caused by the facts that the patches in one cluster are very similar and the number of patches in one cluster is very small. These two points can give an alternative range for the determination of the depth and width of the tree.

2.2. Binary Search Path Tracking

Here, we extract the retrieval feature, *i.e.*, the binary search path, for the finger vein image. As the vocabulary tree is built by the local feature of image patch, in path tracking, the image is also divided into many patches and the local feature is extracted from these patches. For each patch in one image, we will search its most similar leaf node in the constructed vocabulary tree. The search process starts from the root node, and ends at the most similar leaf node. In the process, the node in each layer of the tree, which is more similar to the input patch than other nodes in the same layer, will be passed, and the node is labeled as 1 in the path vector. Other nodes in the corresponding layer will be skipped and are labeled as 0. The encoded binary search path will be used as the retrieval feature of the image.

The binary search path of a patch p_{mn} in one image can be represented by

$$b_{mn} = \{C(p_{mn})\} = \{v_1, v_2, ..., v_{k+k^2+...+k^L}\} \quad (2)$$

where the dimensionality of b_{mn} is equal to the number of nodes in the tree, and the passed nodes and the skipped nodes in the search of nearest leaf node is separately labeled by 1 and 0. For one image, the binary retrieval feature is $B = [b_{11}, ..., b_{mn}, ..., b_{MN}]$. We give an example to explain how to calculate the binary search path. For one probe image patch p_{mn}, its nearest leaf node is c_{23} in the Fig. 4. The tacking path of the probe image patch is shown as two bold black lines with arrows in the figure. In the tracking of the nearest leaf node, nodes c_{12} and c_{23} are passed, and labeled as 1. But other nodes are skipped in the tracking, and labeled as 0. So, the encoded binary search path is {0, 1, 0, 0, 1, 0}.

We use an example to quickly illustrate why we use the binary search path to denote one image patch, not the traditional nearest leaf node. As shown in Fig. 5, there are two enrolled image patches A, B and one probe image patches C, and their nearest neighbors are separately the leaf nodes with numbers 3, 5, and 4. If the nearest leaf node is used to label the image patch, the absolute difference between A and C is equal to it between B and C. However, if we use

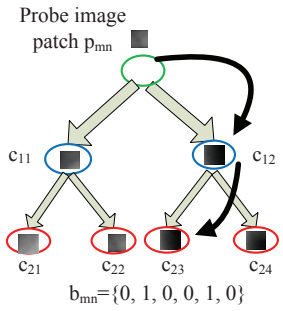


Figure 4. An example of binary search path tracking.

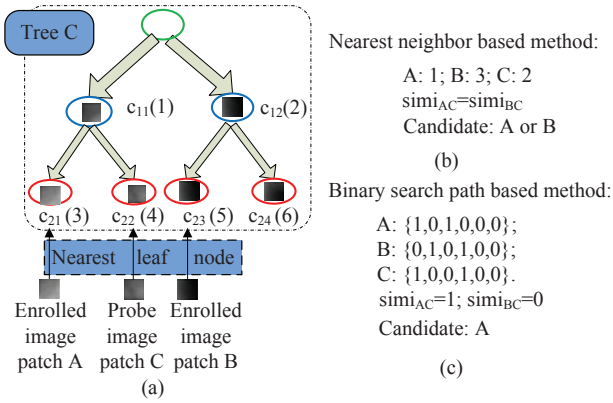


Figure 5. Various ways of image patch representation. (a) Image patches and their nearest leaf nodes; (b) Nearest leaf node based image patch representation; and (c) Binary search path based image patch representation.

the binary search path to denote the image patch, the number of overlapped bits between A and C is 1, which is larger than it between B and C, *i.e.*, 0. In the first case, A and B both can be returned as candidates for C, but in the second case, A, not B, is the candidate. Obviously, the candidate in second case is right, which implies the binary search path is more discriminative than the single nearest leaf node.

Assuming there are $M*N$ patches in one image, and one patch is denoted by a $k + k^2 + ... + k^L$ dimensional binary path, one image will be represented by a binary vector with $M * N * (k + k^2 + ... + k^L)$ bits. In other words, the binary paths can distinguish $2^{M*N*(k+k^2+...+k^L)}$ images. If there are 4 patches in one image, and the dimensionality of the binary path is 6, the binary paths can represent 2^{24} (about 16, 000, 000) different images. Hence, it is reasonable to believe that the binary path is powerful even in the scenarios with large scale populations.

2.3. Candidate Selection

As the binary search path was tracked in the former section, here we will measure the similarity of image pair and

selected the candidates according to the sorted similarity scores. We use the number of the overlapped bits in binary search path to measure the similarity of patch pair. In Ref. [11], the term frequency inverse document frequency (TF-IDF) value of each bit is used as a weight to label the importance of the bit in similarity measurement. For simpleness, we give the equal weight for each bit to replace the TF-IDF value. At same time, the spatial layout information of the patch is also considered in similarity measurement. In other words, only if two patched separately from two images are located at same position, the similarity is valid. The similarity score of image pair is the total score between all involved image patch pairs. The probe image will be matched with all enrolled images, and the similarity scores will be sorted in descending order. The enrolled images with top t ranks will be picked as candidates.

Denote the binary research paths of one probe image I^p and the sth enrolled image I^s separately as $B^p = [b^p_{11}, b^p_{mn}, ..., b^p_{MN}]$ and $B^s = [b^s_{11}, b^s_{mn}, ..., b^s_{MN}]$. The similarity score of these two images $sim(I^p, I^s)$ is defined as the total similarity score of all patch pairs:

$$\begin{aligned} sim(I^p, I^s) &= sim(B^p, B^s) \\ &= \sum_{m=1,...,M;n=1,...,N} f(b^p_{mn}, b^s_{mn}) \end{aligned} \quad (3)$$

where m and n are the location indexes of one image patch. Only if the probe image patch b^p and the enrolled image patch b^s are at the same location, the similarity score is calculated. The matching function f of two patches can be further expressed:

$$f(b^p_{mn}, b^s_{mn}) = \sum_{v^p_j \in b^p_{mn}, v^s_j \in d^s_{mn}} w_j f_v(v^p_j, v^s_j) \quad (4)$$

where w_j is the equal weight for bits in binary path to replace the TF-IDF value. We define as:

$$\begin{aligned} f_v(v^p_j, v^s_j) &= \begin{cases} 1, if v^p_j = v^s_j = 1, \\ 0, \text{otherwise}, \end{cases} \\ &\text{subject} \quad \text{to} \quad j = 1, ..., k + k^2 + ... + k^L \end{aligned} \quad (5)$$

As there are S enrolled images , we will obtain S similarity scores:

$$SIM = [sim(I^p, I^1), sim(I^p, I^2), ..., sim(I^p, I^S)] \quad (6)$$

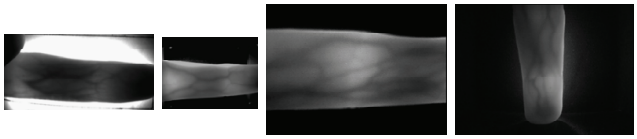
The scores will be sorted and the enrolled images with top t ranks will be selected as candidates.

3. Experiments

In this section, we compare the proposed retrieval method with the single layer k-means based method and the state-of-the-art to demonstrate our performance. And, we also investigate the influence of the depth and width of the tree to the retrieval accuracy.

Database	Finger number	Image number per finger	Image size(pixels)
POLYU [6]	312	6/12	513*256
SDUMLA [22]	636	6	320*240
MMCBNU [8]	492	12	640*480
FVUSM [1]	600	12	640*480
FUSIOND	2040	6	96*64

Table 1. Five finger vein databases we used in experiments



(a) (b) (c) (d)

Figure 6. Typical finger vein images. (a)-(d) are separately from four public databases.

3.1. Database and Experiment Settings

To comprehensively evaluate the performance of the proposed retrieval method, five finger vein databases are used in experiments, including four public finger vein databases, *i.e.*, POLYU [6], SDUMLA [22], MMCBNU [8] and FVUSM [1], and one ourselves-built fusion database. The details of these five databases are given in Table 1. Some typical images are illustrated in Fig. 6. In the public databases, the region of interest (ROI) image with 96*64 pixels is used in the experiments, which is extracted from the original image by the methods in Ref. [21]. And the fusion database is built by integrating these four public databases to simulate the scenario with large scale of populations. As the sizes of original images from different databases are different, the integration of these databases is based on the extracted ROI images. In different databases, the number of images per finger is also varied, so we select the first 6 images of each finger on each database to construct the fusion database.

In all experiments, six-fold cross validation is used to evaluate the influence of the randomness of k-means to the retrieval performance. In other words, the experiments are conducted six times, and every time we take one image per finger as a probe sample and the remainders as the enrolled ones. The average value and variance value of retrieval accuracies in six times are reported. All experiments are conducted in MATLAB on a PC with Intel i7-4790 3.60GHz CPU and 8G memory.

Additionally, in the proposed method, ROI image is partitioned into 12*8 patches each with 8*8 pixels, and local binary pattern (LBP) is used as based feature to extract the local feature of image patch [7]. An example of ROI image and its base feature are shown in Fig. 7.

Database	POLYU	SDUMLA	MMCBNU	FVUSM	FUSIOND
Top 5	0.9899±2.03e-04	0.9845±2.41e-04	0.9925±5.43e-05	0.9963±4.05e-05	0.9902±3.28e-05
Top 10	0.9941±7.95e-05	0.9864±1.87e-04	0.9941±4.16e-05	0.9973±2.42e-05	0.9918±2.30e-05
Top 20	0.9957±3.14e-05	0.9890±1.16e-04	0.9961±1.97e-05	0.9976±1.73e-05	0.9936±1.74e-05
Top 30	0.9963±2.62e-05	0.9895±1.02e-04	0.9961±1.97e-05	0.9976±1.73e-05	0.9952±1.09e-05
Top 50	0.9963±2.62e-05	0.9921±5.83e-05	0.9967±1.44e-05	0.9976±1.73e-05	0.9964±8.39e-06
Top 100	0.9973±9.89e-06	0.9942±3.42e-05	0.9975±9.61e-06	0.9987±6.05e-06	0.9971±5.28e-06

Table 2. Retrieval accuracy (average±variance) of proposed method on multiple databases

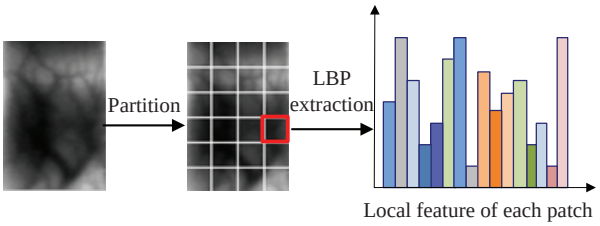


Figure 7. ROI image partition and local feature extraction of each patch.

Method	k-means	Proposed method
POLYU	0.8568±0.28e-02	0.9899±2.03e-04
SDUMLA	0.9387±4.71e-04	0.9845±2.41e-04
MMCBNU	0.9175±0.19e-02	0.9925±5.43e-05
FVUSM	0.8625±0.13e-02	0.9963±4.05e-05
FUSIOND	0.8690±9.23e-04	0.9902±3.28e-05

Table 3. Comparison of the hierarchical vocabulary tree and single layer k-mean based methods

3.2. Performance of Proposed Method

We conduct this experiment on all five databases to examine the performance of the proposed method. The hierarchical vocabulary tree we used has three layers and three branches. We vary the number of selected top candidates *i.e.*, t, and give all experimental results with different numbers. The average value and variance value of retrieval accuracies in six-fold cross validation are given in Table 2.

From Table 2, we can find that the proposed method achieves satisfying retrieval performance on all databases. There are two plausible reasons: (1) The learned finger vein textons are very effective in image patch representation; and (2) As our analysis in Section 2.2, the multidimensional binary search path has high discrimination. It can also be told that the stability of the proposed method is acceptable by the variance value of the accuracies in six-fold cross validation. Owing to the randomness of the cluster centers in k-means clustering, the nodes of the tree in multiple cross validation may be varied to some extent, but the binary search paths of corresponding patches in genuine images in each validation are unvaried. So, the randomness of k-means has limited influence to the retrieval performance. In addition, with the increment of t from 5 to 100, the retrieval accuracy has improved from about 99% to 99.70%.

3.3. Comparison with Single Layer k-means based Method

As k-means is iteratively used in the hierarchical vocabulary tree construction, the tree based retrieval method (*i.e.*, binary search path as retrieval feature) is compared with the single layer k-means based one (*i.e.*, nearest leaf node as retrieval feature) in this section. As there are 27 leaf nodes in the hierarchical vocabulary tree with three layer and three branches, we built 27 clusters in the single layer k-means as the leaf nodes. For the single layer k-means, the frequency numbers of all leaf nodes in one image are used to represent the image, and the histogram intersection function [16] is used to measure the similarity of image pair. The retrieval accuracies with top 5 candidates of these two methods are shown in Table 3.

It can be told from the results that the proposed hierarchical vocabulary tree based method outperforms the single layer k-means based one on all databases. The better performance is mainly attributed to the higher discrimination of all tree nodes involved binary search path than the nearest leaf node in image representation. In detail, although the number of clusters in the single layer k-means based method is same to it of leaf nodes in our method, the former only uses the nearest leaf node to represent one image patch, but our method uses the sparse binary codes of all nodes in searching of the nearest leaf node to represent the patch. It is evident that the multidimensional binary codes have larger discrimination capacity. And the variance of the single layer k-means based method is always higher than it of our method, which implies our method is more stable than the single layer k-means based one on the retrieval performance. It can be easily understood that the multidimensional binary search path is more robust than one nearest leaf node.

3.4. Comparison with LSH based Method

We compare our method with the recent state-of-the art [18] about the retrieval accuracy and efficiency on the fusion database. The accuracies and time costs of two methods are given in Table 4. Note that: (1)The accuracy is based on top 5 candidates; and (2) The candidate selection

Method	Accuracy	Time cost (s)			
		Base feature extraction	Binary pattern encoding	Candidate selection	Total time
LSH based method [18]	0.9587±2.45e-04	0.9616	3.77e-04	0.0328	0.9944
Proposed method	0.9902±3.28e-05	0.0158	0.0175	0.1016	0.1349

Table 4. Comparison between our method and the LSH based method on fusion database

includes the similarity measurement of one probe image and all enrolled images and the sorting of the measured similarity scores. And the time costs on all steps are the average values per image on the fusion database.

The results in Table 4 show that the proposed method achieves better performance on both retrieval accuracy and time cost. The inferior retrieval accuracy of LSH based method is mostly caused by the number of vein point based binary pattern encoding. In other words, the binarization of image patch based on the number of vein point in the patch makes finger veins intrinsic characteristics (*e.g.*, the shape and growth direction of vein branch) seriously lost, which leads to the limited discrimination of the encoding. For example, although two patches in different images are both labeled as 1, the numbers of vein points in these two patches may have big gap, and even if the numbers are almost equal, the shape and growth direction of vein branch may be varied greatly. However, the encoding of the probe image patch is conducted based on its comparison with the tree nodes (*i.e.*, finger vein textons) in our method, in which the intrinsic characteristics of finger vein can kept.

Furthermore, the total time spent by LSH based method is more than it spent by our method. For LSH based methods, the time is mainly cost in base feature extraction, *i.e.*, repeated line tracking based vein pattern extraction. In the method, the more tracking times, the more spent time and the better extracted vein pattern. In order to extract plentiful vein pattern, the tracking times is always very large, which causes too much time cost. But the hash table based candidate selection in LSH based method only spends 0.0328, which is less than ours.

In addition, the variation tendencies of retrieval accuracy with the increment of selected top candidates are illustrated in Fig. 8 for these two methods. The figure shows the proposed method outperforms LSH based method no matter how many candidates are selected, which further proves the advantage of the proposed method.

3.5. Selection of the Depth and Width of the Hierarchical Vocabulary Tree

Here, we study the influence of the tree depth and width to the performance of proposed method. The experiments are conducted on two databases, *i.e.*, POLYU and SDUMLA. For the depth, the trees with three layers and four layers are considered, and for the width, the branch number

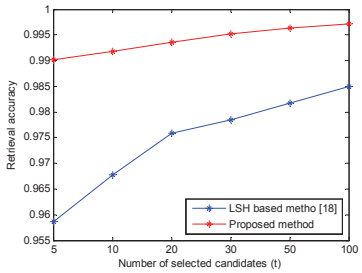


Figure 8. Retrieval accuracies with different numbers of selected candidates.

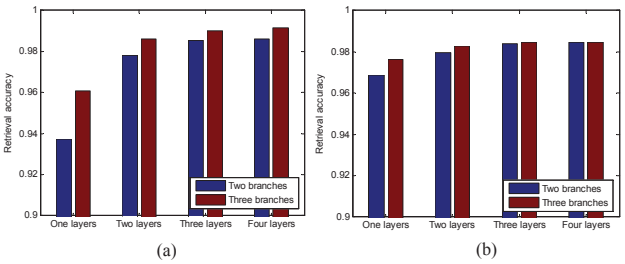


Figure 9. Retrieval accuracies of the vocabulary tree with different layers and branches. (a) Retrieval accuracies on POLYU database; (b) Retrieval accuracies on SDUMLA database.

of each parent nodes varies from two to three. Like experiment 3.3 and 3.4, the average retrieval accuracy in cross validations is also based on top 5 candidates. The accuracies with different layers and branches on two databases are illustrated in Fig. 9.

It can be clearly noted from the figure that, the retrieval accuracy of the proposed method is gradually improved with the increment of the layer and the branch. But the improvement between three layers and four layers is very limited. And in tree construction and binary search path tracking, the time cost with four layers is larger than that with three layers. By the trade-off between the retrieval accuracy and retrieval efficiency, the tree with three layers and three branches is believed to be the best choice.

4. Conclusion

In this paper, we employ the binary search path of a hierarchical vocabulary tree to perform finger vein image re-

trieval. The finger vein textons are learned and further used as the nodes of the tree, which can express the intrinsic characteristics of finger vein, such as the local shape and the growth direction of finger vein. And all the tree nodes are used to sparsely and comprehensively represent the image patch in the binary search path. The tree nodes based binary search path has two superiorities: (1) All nodes involved search path is more discriminative than the nearest leaf node in image patch representation; and (2) The binary search path based similarity measurement is quicker than other pattern feature based ones, and it is more practicable in the scenarios with large scale populations. In total, finger vein texton based tree nodes and the binary search path based image patch representation work together to accelerate the retrieval efficiency and improve the retrieval accuracy.

At this work, we treat all tree nodes (*i.e*., all bits in binary search path) equally, no matter the middle nodes or the leaf nodes. It may be meaningful to explore the discrimination of different nodes, and give different weight values to these nodes according to their discrimination, like the TF-IDF value in Ref. [11].

5. Acknowledgements

The authors would like to thank the anonymous reviewers for their helpful suggestions. The work is supported by National Science Foundation of China under Grant No.61173069, 61472226, 61573219 and Shandong Natural Science Funds for Distinguished Young Scholar under Grant No. JQ201316.

References

[1] M. S. M. Asaari, S. A. Suandi, and B. A. Rosdi. Fusion of band limited phase only correlation and width centroid contour distance for finger based biometrics. *Expert Systems with Applications*, 41(7):3367–3382, 2014.

[2] L. Dong, G. Yang, Y. Yin, X. Xi, L. Yang, and F. Liu. Finger vein verification with vein textons. *International Journal of Pattern Recognition and Artificial Intelligence*, page 1556003, 2015.

[3] K. Dorasamy, L. Webb, J. Tapamo, and N. P. Khanyile. Fingerprint classification using a simplified rule-set based on directional patterns and singularity features. In *International Conference on Biometrics (ICB)*, pages 400–407. IEEE, 2015.

[4] J. Hashimoto. Finger vein authentication technology and its future. In *Symposium on VLSI Circuits*, pages 5–8. IEEE, 2006.

[5] H.-W. Jung and J.-H. Lee. Noisy and incomplete fingerprint classification using local ridge distribution models. *Pattern Recognition*, 48(2):473–484, 2015.

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

[7] F. Liu, Y. Yin, G. Yang, L. Dong, and X. Xi. Finger vein recognition with superpixel-based features. In *IEEE International Joint Conference on Biometrics (IJCB)*, pages 1–8. IEEE, 2014.

[8] Y. Lu, S. J. Xie, S. Yoon, Z. Wang, and D. S. Park. An available database for the research of finger vein recognition. In *6th International Congress on Image and Signal Processing (CISP)*, volume 1, pages 410–415. IEEE, 2013.

[9] H. Mehrotra and B. Majhi. Local feature based retrieval approach for iris biometrics. *Frontiers of Computer Science*, 7(5):767–781, 2013.

[10] 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.

[11] D. Nister and H. Stewenius. Scalable recognition with a vocabulary tree. In *IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, volume 2, pages 2161–2168. IEEE, 2006.

[12] R. Raghavendra, K. B. Raja, J. Surbiryala, and C. Busch. A low-cost multimodal biometric sensor to capture finger vein and fingerprint. In *IEEE International Joint Conference on Biometrics (IJCB)*, pages 1–7. IEEE, 2014.

[13] R. Raghavendra, J. Surbiryala, and C. Busch. An efficient finger vein indexing scheme based on unsupervised clustering. In *IEEE International Conference on Identity, Security and Behavior Analysis (ISBA)*, pages 1–8. IEEE, 2015.

[14] Z. Sun, H. Zhang, T. Tan, and J. Wang. Iris image classification based on hierarchical visual codebook. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 36(6):1120–1133, 2014.

[15] J. Surbiryala, R. Raghavendra, and C. Busch. Finger vein indexing based on binary features. In *Colour and Visual Computing Symposium (CVCS), 2015*, pages 1–6. IEEE, 2015.

[16] M. J. Swain and D. H. Ballard. Color indexing. *International journal of computer vision*, 7(1):11–32, 1991.

[17] D. Tan, J. Yang, Y. Shi, and C. Xu. A hierarchal framework for finger-vein image classification. In *2nd IAPR Asian Conference on Pattern Recognition (ACPR)*, pages 833–837. IEEE, 2013.

[18] D. Tang, B. Huang, R. Li, and W. Li. A person retrieval solution using finger vein patterns. In *20th International Conference on Pattern Recognition (ICPR)*, pages 1306–1309. IEEE, 2010.

[19] Y. Wang, L. Wang, Y.-m. Cheung, and P. C. Yuen. Fingerprint geometric hashing based on binary minutiae cylinder codes. In *22nd International Conference on Pattern Recognition (ICPR)*, pages 690–695. IEEE, 2014.

[20] 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.

[21] L. Yang, G. Yang, L. Zhou, and Y. Yin. Superpixel based finger vein roi extraction with sensor interoperability. In *International Conference on Biometrics (ICB)*, pages 444–451. IEEE, 2015.

[22] Y. Yin, L. Liu, and X. Sun. Sdumla-hmt: a multimodal biometric database. In *Biometric Recognition*, pages 260–268. Springer, 2011.