Locality Preserving Hashing based on Random Rotation and Offsets of PCA in Image Retrieval

Shan Zhao and Yongsi Li∗

School of Computer Science and Technology, Henan Polytechnic University, Jiaozuo, 454000, China

Abstract

Manifold-based subspace feature extraction methods have recently been deeply studied in data dimensionality reduction. Inspired by PCA Hashing (PCAH), if the Locality Preserving Projection (LPP) is directly used in the hash image retrieval, it is prone to shortcomings such as being inefficient and time-consuming. In order to address these deficiencies, this paper mainly combines Principal Component Analysis (PCA) and manifold subspace feature extraction method LPP, and we present a RLPH framework using random rotation. Among them, PCA processing solves the eigenvalue problem encountered in the calculation of LPP, thereby improving the recognition effect of the algorithm. The PCA projection needs to ensure that the variance of the sample points after projection is as large as possible. However, projections of small variance may produce unnecessary redundancy and noise. Therefore, in the subspace after the PCA projection, we only extract the eigenvectors that contain most of the information at the top of the PCA projections. Then, we utilize a random orthogonal matrix to randomly rotate and shifts the eigenvectors and the reduced-dimension sample obtained after the top eigenvectors of the PCA projection is subjected to LPP mapping. Random rotation produces many thin projection matrices blocks that are then concatenated into one final projection matrix. Random rotation is a key step in this paper that minimizes the quantization error for codes. The proposed method greatly improves the retrieval efficiency, and extensive experiments demonstrate its effectiveness.

Keywords: manifold; data reduction; hashing; PCA; LPP; random rotation

(Submitted on August 20, 2018; Revised on September 15, 2018; Accepted on October 17, 2018)

© 2018 Totem Publisher, Inc. All rights reserved.

1. Introduction

Approximate Nearest Neighbor (ANN) search is an important research in many large-scale machine learning and computer vision research topics, such as mode recognition [1], object detection [2], and 3D reconstruction [3]. These tasks mainly find the nearest neighbor of a sample in a huge database. When faced with a large amount of data and high-dimensional data information in the database, an ideal retrieval effect and an acceptable retrieval time cannot be obtained in a high-dimensional space based on Nearest Neighbor (NN) query. NN search is infeasible in these scenarios because it has high time complexity. Therefore, ANN search plays a crucial role in the fast similarity retrieval on large scale datasets.

In recent years, hash-based retrieval technology [3-12] has attracted much attention in large-scale applications. In past literature, the tree-based similarity search method [13-17] has been a very active field in machine learning and computer vision. Although the tree-based retrieval technique greatly reduces the time response of a single retrieval, it requires a large storage space and is sensitive to the data distribution so that the tree-based method cannot guarantee faster search compared to the linear scan method in the large-scale image datasets. Thus, hashing methods based on ANN search are becoming increasingly popular. The main purpose of this method is to generate corresponding binary code for each sample so that similar samples have close codes. The ANN-based hashing method is very fast on large-scale datasets because the Hamming distance between binary codes in modern CPU is effectively computed by XOR instructions. In addition, binary code is very compact for memory storage.

The current hash function can be divided into data-independent and data-dependent [5-6,11-12,18-22]. Data-independent hashing methods do not rely on original data to learn hash functions, such as the most classic hashing method...
Locality Sensitive Hashing (LSH) [4, 11, 23]. They usually adopt random projections as hash functions without considering any data structure. In order to obtain satisfactory results, LSH typically increases the code length and the number of hash tables, but this generally increases the CPU’s storage space and query response time, which reduces the search efficiency. On the contrary, data dependence depends on the internal structure of the original data to project high-dimensional data into low-dimensional space, such as spectral graph analysis and semi-supervised learning, that are typically demonstrated to be more effective than data-independent LSH.

In the research of image retrieval, there are many hashing methods based on dimensionality reduction, among which the reduced-dimensional methods are various. The data-dependent PCA Hashing (PACH) [11] utilizes Principal Component Analysis (PCA) [24] of many reduced-dimensional methods to achieve fast similarity search in huge databases. The main purpose of PACH is to find a set of optimal unit-orthogonal bases through linear transformation and project the original data onto the set of orthogonal bases for dimensionality reduction so as to achieve higher retrieval efficiency. Because manifold-based Locality Preserving Projection (LPP) [25] and PCA are both unsupervised linear subspace feature extraction methods, LPP has many advantages over PCA: it has strong robustness and discriminate ability and can maintain the topological structure of the data. In many cases, the PCA projection was replaced with LPP, so LPP has a good research value in hash image retrieval.

If we apply LPP directly to hash image retrieval, it has disadvantages such as being inefficient and time-consuming. In order to address these deficiencies, we present a novel hashing method named RLPH, where R and b represent a random rotation and random offsets on the top principal projections of PCA data. This method integrates the two methods of PCA and manifold subspace feature extraction LPP, and PCA processing solves the eigenvalue problem encountered in the calculation of LPP, thereby improving the recognition effect of the algorithm. On this basis, we propose a unified framework utilizing random rotation.

More specifically, the specific framework is divided into three stages. In the first stage of the framework, we first perform PCA on the training samples, and then we extract only the eigenvectors with the most identifying information at the top of PCA so that unnecessary noise and information are reduced in the projection process. In the second stage of the framework, the training sample is projected onto the eigenvectors to obtain a sample of reduced dimension by PCA, and then the reduced-dimensional sample is subjected to LPP projection so that a thin projection matrix block is finally obtained. In the third stage of the framework, a random orthogonal matrix is exploited to perform multiple random rotations on the top eigenvectors of the above-mentioned PCA. The first two steps above are repeated for each rotation, from which we can obtain multiple thin projection matrix blocks. We only need to splice these thin matrix blocks together to get the projection matrix of the final number of coded bits. Finally, the training sample is projected onto the projection matrix to reduce the overall dimension of the data, and the obtained reduced-dimensional sample of LPP is then hash-coded to obtain the binary long code we need. Random rotation is a key step in this paper that minimizes the quantization error for codes and diversifies the projection matrix. Our main contributions are summarized as follows:

1. LPP is essentially a linear reduced-dimensional method that simultaneously has the ability to learn popularly. Furthermore, it can not only maintain the local structure of high-dimensional data effectively in data projection, but also easily obtain the embedded low-dimensional coordinate representation and extract new sample features. However, LPP itself is easily affected by the small sample problem, and the global structure of the data cannot be maintained during the projection process. Therefore, we combine PCA and LPP to retain the local and global similarity structure of the original data.

2. In this paper, the whole process integrates the binary coded segmentation idea of the bagging strategy in Bagging PCA (BPCA) [26] to matrix splicing of multiple projection matrix blocks. More importantly, inspired by the two-fold randomness PCAH (R^2PCA) [27], a single random rotation is applied to the top major projection of PCA data, reducing the quantization error between codes. In particular, the subtle integration of manifold subspace feature extraction method with the bagging strategy and random rotation greatly improves the retrieval efficiency of the original LPP hashing technique.

3. Our proposed approach outperforms several current state-of-the-art hashing methods. As can be seen in later chapters, we mainly conduct experiments in three publicly available face databases and analyse the comparison results precisely. Extensive experiments prove the feasibility and efficiency of this method.

2. Related Work

In the field of computer vision, almost all ANN searches are performed by partitioning the entire space, dividing it into many small subspaces, and performing recursive operations in the locked subspace. In this section, we will briefly introduce
several ANN-based hashing methods.

To the best of our knowledge, LSH is considered one of the important breakthroughs in fast ANN search in high-dimensional space. It uses the hash function family as a random projection to embed training data into the Hamming space. Depending on the similarity measure method and the data space used by the hash function family, the LSH methods can be divided into several classes, such as LSH based on random projection and LSH based on P-stable distribution. In addition, for use in a wide class of useful similarity searches, the Kernelized LSH (KLSH) [7] describes how to adapt the LSH to arbitrary kernel functions so that the sub-linear time similarity search guarantees of the algorithm are preserved as much as possible. Because the LSH method is data-independent, it does not use the data itself in constructing a hash function. In order to obtain high search accuracy, it is often necessary to use a very long code bit, but as the number of coded bits increases, the collision probability of the similarity training sample in the hashing process will decrease, which will reduce the recall rate in the search. Because of this, LSH often requires multiple hash tables to achieve satisfactory search results. However, this not only increases the storage space, but also increases the query time.

Recently, many data-dependent hashing methods have developed rapidly. It often needs to learn data-aware hash functions, and many hashing methods are based on the characteristic decomposition of the Laplacian matrix. However, the eigenvectors for different variances are typically unbalanced. In [28], instead of learning all the eigenvectors at once, Wang et al. proposed a successive learning framework in the early stage to learn hash function, which tends to minimize the errors made by the previous one. Moreover, Gong et al. proposed an alternate minimization method to reduce the dimension of the training data PCA and used an optimal way to rotate data of zero-centered to minimize the quantization error. This method is called Iterative Quantization (ITQ) [14]. Furthermore, a similar proposal was used in [29], in which PCA is often replaced by LPP. On the other hand, the method based on manifold subspace feature extraction has been widely used in fast hash-based retrieval. For example, Locally Linear Hashing (LLH) [30] prevailing utilizes sparse matrix decomposition to learn arbitrary low-dimensional manifolds to minimize the reconstruction error and quantization error.

From the above introduction, PCA is widely used in many data-dependent hash methods for eigendecomposition, and manifold-based hashing also occurs. Different from most of the existing hashing methods, we effectively combine PCA based on random rotation and LPP of manifold dimensionality reduction methods in our approach. To this end, we only extract the top eigenvectors of the PCA. Two-fold random rotation is applied to the top principal components of PCA, and the reduced-dimensional sample obtained after the top eigenvectors of the PCA projection is subjected to LPP mapping so as to generate multiple different and thin projection matrix blocks. Random rotation effectively reduces the quantization error between codes and leverages a number of different matrix blocks. Finally, we stitched together multiple projection matrix blocks. Owing to the intricate theory established in the ensemble study, our method enjoys several preponderances that are lacking in previous works.

3. Related Theoretical Background

3.1. Principal Component Background

PCA is a classical unsupervised linear subspace feature extraction method. The main purpose is to find an optimal set of orthonormal bases through linear changes and to utilize this linear combination of orthogonal bases to reconstruct the original data so that the mean square error of the reconstructed data and the original data is minimized. The larger the variance, the greater the amount of information it carries. If we want to separate the projections of all sample points as much as possible, we should maximize the variance of the sample points after the projection. Therefore, the optimization objective function can be written as:

\[
\arg\max_W \quad \text{tr}(W^TXX^TW) \quad \text{s.t.} \quad W^TW = I
\]  

(1)

Among them, \(X = \{x_1, x_2, \ldots, x_n\} \in \mathbb{R}^{d \times n}\) is a sample matrix. \(n\) and \(d\) respectively represent the number of sample points and the dimension of sample points. Under the constraint condition \(W^TW = I\), the matrix \(W\) satisfying the maximization objective function of Equation (1) is the projection matrix obtained after PCA. Utilizing the Lagrange multiplier method, the following generalized eigenvalue problem can be solved:

\[
XX^T w_i = \lambda_i w_i
\]  

(2)
Assuming that \( w_1, w_2, \ldots, w_t \) are the eigenvectors corresponding to the largest \( t \) eigenvalues of Equation (2), the PCA projection matrix is expressed as \( W=[w_1, w_2, \ldots, w_t] \in \mathbb{R}^{d \times t} \). The low-dimensional features of the sample set \( X \) are represented as \( \{y_i=W^T x_i, i=1, 2, \ldots, n\} \). In PCA, the most discriminative information is captured by the top eigenvectors. In addition to the principal predictions, the eigenvectors corresponding to the reduced eigenvalues are typically noise.

3.2. Locality Preserving Projection

LPP as a subspace extraction method based on manifold not only can effectively find the low-dimensional manifold embedded in the high-dimensional data space, but also can obtain its low-dimensional data representation for new test samples. As an unsupervised linear subspace feature extraction method, it can maintain the local structure between high-dimensional data.

In the identification problem, the closer the two samples, the greater the degree of similarity and the greater the probability that the samples belong to the same category. The basic idea of the LPP method is to find a projection matrix \( V \) and map the sample set \( X=[x_1, x_2, \ldots, x_n] \) in the high dimensional space \( \mathbb{R}^d \) to the sample set \( Y=[y_1, y_2, \ldots, y_n] \) in the low dimensional space \( \mathbb{R}^t, (t<d) \), which is \( \{y_i=V^T x_i, i=1, 2, \ldots, n\} \). The two neighbors in the \( \mathbb{R}^t \) space are still close neighbors in the \( \mathbb{R}^t \) space after \( V \) mapping. The objective function is:

\[
\sum_y (y_i - y_j)^T M_y
\]

(3)

With simple algebraic operations, the objective function Equation (3) can be simplified to:

\[
\frac{1}{2} \sum_y (y_i - y_j)^T M_y = \frac{1}{2} \sum_y (V^T x_i - V^T x_j)^T M_y = V^T X (D-M) X^T = V^T X D X^T V
\]

(4)

Among them, \( M_y \) is the weight matrix of the nearest neighbor graph, which is usually set by the Gaussian kernel method. If there is an edge between \( x_i \) and \( x_j \), \( M_{ij}=\exp(-\|x_i-x_j\|^2/2\delta^2) \). Otherwise, \( M_{ij}=0 \). The matrix \( D \) is a diagonal matrix whose value is the sum of the data elements of each row or column of the weight matrix \( M \). That is, \( D=\sum M_y \) and \( L=D-M \) is a Laplacian matrix of the nearest neighbor graph. In the case of constraint \( V^T X D X^T V=1 \), the final objective function is obtained by minimizing the objective function Equation (4) as follows:

\[
\arg \min_V V^T X D X^T V \text{ s.t. } V^T X D X^T V = 1
\]

(5)

The matrix of the LPP projection transformation is \( V=\{v_1, v_2, \ldots, v_t\} \), formed by the feature vector corresponding to the smallest \( t \) eigenvalues after Equation (5).

It can be seen that LPP generally replaces linear PCA dimensionality reduction methods. On the other hand, LPP itself is easily affected by the small sample problem, and it cannot maintain the global structure of the data during the projection process. If LPP is directly used for hash image retrieval, its retrieval efficiency is very inefficient. Therefore, we make full use of the random rotation to combine PCA and LPP together and retain the local and global similarity structure of the original data. The details are provided in the next section of this article.

4. Locality Preserving Hashing based on Random Rotations of PCA

In this section, we will introduce the proposed method in detail. Assume that given a dataset \( X=\{x_1, x_2, \ldots, x_n\} \in \mathbb{R}^{d \times n} \), the data requires t-bit hash encoding. The whole process has two key steps. On the one hand, we only take the top eigenvectors generated by the PCA. On the other hand, we utilize a random orthogonal matrix to randomly rotate the eigenvectors. The reduced dimension sample is hash-coded, and the resulting binary matrix is \( H=\{h_1, h_2, \ldots, h_n\} \in \{-1,1\}^{t \times n} \). The specific steps mainly include three parts:

1. The PCA is performed on covariance matrix \( XX^T \) of the training samples, and we only retain the first \( r (r<t) \) eigenvectors of discriminative information in the top of PCA to obtain an orthogonal matrix \( W^d \). Since most discriminative information is captured by the top eigenvectors of PCA, \( W^d \) contains the main information of the training sample and reduces the noise in the PCA projection process. We project the data onto the orthogonal matrix \( W^d \) to obtain the reduced-
dimension matrix $Y$. The function is as follows:

$$ Y = W^T X $$

(6)

In this formula, $W^i = \{ w^i_1, w^i_2, \ldots, w^i_r \}, W^i \in \mathbb{R}^{d \times r}$. Then, we utilize the random orthogonal matrix $\{ R^{(i)} \in \mathbb{R}^{d \times r}, i = 1, 2, \ldots, K \}$ to rotate $W^i$ and random offsets vector $\{ b^{(i)} \in \mathbb{R}^{d \times r}, i = 1, 2, \ldots, K \}$. One of the meanings of the random rotation minimizes the quantization error between the hash codes. On the other hand, top multiple principal components of PCA are generated, and the diversity of eigenvectors is increased. The matrix is marked as $W^i$:

$$ W^i = W^i R^{(i)} + b^{(i)}, \ (i = 1, 2, \ldots, K) $$

(7)

Moreover, $K = t/r$ represents $K$ rotations.

(2) We take the Equation (6) projection matrix $Y \in \mathbb{R}^{s \times n}$ for LPP, where $Y$ is the reduced-dimensional sample of the training data obtained after PCA projection. Performing PCA processing before LPP can effectively solve the eigenvalue problem encountered in the LPP calculation process, thereby further improving the recognition ability of the algorithm. We substitute $Y$ into Equation (5) to get the new objective function as follows:

$$ \arg \min_{V^i} \ V^T Y L Y^T V^i \ s.t. \ V^T Y D Y^T V^i = 1 $$

(8)

After solving for Equation (8), $V^i$ represents the feature vector $V^i = \{ v^i_1, v^i_2, \ldots, v^i_r \}$ corresponding to the smallest $r$ eigenvalues. Utilizing the Lagrange multiplier method, we can convert Equation (8) to:

$$ L(v, \lambda) = V^T Y L Y^T V^i - \lambda (V^T Y D Y^T V^i - 1) $$

Then, we take the derivative of $v$ and $\lambda$ in $L(v, \lambda)$:

$$ \frac{\partial L}{\partial V} = Y L Y^T V^i - \lambda Y D Y^T V^i = 0 $$

$$ \frac{\partial L}{\partial \lambda} = V^T Y L Y^T V^i - 1 = 0 $$

The final result is:

$$ Y L Y^T V^i = \lambda Y D Y^T V^i $$

(9)

The $Y L Y^T$ and $Y D Y^T$ in Equation (9) are non-singular matrices of $\mathbb{R}^{s \times s}$, and we only need to solve a generalized eigenvalue problem. The new projection matrix $V^{i1} \in \mathbb{R}^{s \times r}$ can be obtained by combining Equation (6) and Equation (9):

$$ V^{i1} = W^i V^i $$

(10)

Because a random orthogonal matrix is utilized in Equation (7) to rotate $W^i$, $W^i$ in Equation (10) is replaced by $W^i$. At this point, Equation (10) will become Equation (11):

$$ V^{i1(\cdot)} = W^i R^{(\cdot)} V^i, \ i = 1, 2, \ldots, K $$

(11)

Finally, we get $K$ projection matrix blocks containing $r$ feature vectors.
(3) We stitch together the $K$ projection matrix blocks $V^{(1)}, V^{(2)}, \ldots, V^{(K)}$ obtained above to get the final projection matrix $V = \{v^{(1)}, v^{(2)}, \ldots, v^{(K)}\}$. We project the training samples $X$ onto the matrix $V$ and utilize the sign($\cdot$) functions for performing binary encoding of 0 or 1 to get the binary code we ultimately need. The hash function $H$ expression is as follows:

$$H = \text{sign}(V^T X)$$ (12)

In this way, we use Equation (12) to finalize the training sample. Furthermore, the specific framework of our method RLPH learning binary coding is summarized in Algorithm 1.

---

**Algorithm 1 Binary Code Learning with RLPH**

**Input:** Training samples $X = \{x_1, x_2, \ldots, x_n\}$, code length $r$, the number of feature vectors $r$ in each small part of the projection matrix

**Output:** Hash encoding matrix $H$

1. Calculate the original data covariance matrix $XX^T$;
2. Extract the top $r$ largest eigenvectors $W^1$ of $XX^T$, where $W^1 = [w^1_1, w^1_2, \ldots, w^1_d] \in \mathbb{R}^{d \times r}$, and utilize Equation (6) to find the reduced-dimensional matrix $Y$;
3. Utilize Equation (9) to obtain the projection matrix $V^{(1)} = W^1 V^1$;
4. Calculate the number of projection matrices containing $r$ eigenvectors $K = \lceil tr \rceil$;
5. for $i := 1$ to $K$ do begin
6. Generate a random orthogonal matrix $K^i \in \mathbb{R}^{d \times r}$, $i = 1, 2, \ldots, K$ and random offsets vector $b(i) \in \mathbb{R}^{d}$, $i = 1, 2, \ldots, K$;
7. Rotate $W^1$ using random orthogonal matrix $R^i$, then Equation (7) is the rotated and shifted matrix $W^i$;
8. Utilize Equations (6) (7) (9) to obtain the projection matrix, which is transformed into $V^{(i)} = (W^i R^i + b(i))V^1$, $i = 1, 2, \ldots, K$;
9. end
10. The obtained K thin projection matrix blocks are stitched together to obtain the final projection matrix $V$. The training samples $X$ is projected onto a matrix $V$ and then hashed utilizing Equation (12) to obtain a binary code $H = \{h_1, h_2, \ldots, h_n\} \in \{-1, 1\}^{rn}$.

---

5. Experiments

All the experiments in this paper are tested under the MATLAB R2016a software and the Win8 system. In this section, we conducted extensive experiments with different datasets. This method is compared with the most advanced hashing method, and the performance of this method is evaluated.

5.1. Relevant Datasets

Three different face image databases were used in this experiment. Table 1 summarizes the size and dimensions of these three databases:

- **AR:** Consisting of 1680 face images from 120 volunteers, each image is represented by a 2000-dimensional feature vector. Each of them consists of 14 different face images.

- **ORL:** Consisting of 400 face images from 40 volunteers, each image is represented by a 10304-dimensional feature vector. Each person consists of 10 different images.

- **Yale:** Consisting of 165 face images from 15 volunteers, each image is represented by a 10000-dimensional feature vector. Each person consists of 11 different face images.

In order to guarantee the fairness of the experiment, for each of the above datasets, we randomly selected 30 data as the test sample for this experiment, and all the rest were used as training samples.

<table>
<thead>
<tr>
<th>Table 1. Description of datasets</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Datasets</strong></td>
</tr>
<tr>
<td><strong>Sizes</strong></td>
</tr>
<tr>
<td><strong>Dimensions</strong></td>
</tr>
</tbody>
</table>

5.2. Compared Methods and Parameter Setting

Our method is compared to six existing hashing methods:

- LSH [4]: Locality Sensitive Hashing. It is a similarity search method that mainly uses random projection as a hash function family.
CNNH [5]: Convolutional Neural Network (CNN) select relu activation function of hidden layer and output layer select Sigmoid activation function, and use respectively Stochastic Gradient Descent (SGD), Root Mean Square Prop (RMSProp) and the Adaptive Gradient (AdaGrad), the three kinds of optimization algorithm of face data set for training. And hash code it.

PCAH [11]: PCA Hashing. It utilizes the covariance matrix of the original data to find a group of optimal orthogonal bases through feature decomposition. This approach preserves the primary information of the original data as much as possible and utilizes a set of orthogonal bases to reconstruct the original data.

SH [12]: Spectral Hashing. This method mainly performs spectral analysis on the original high-dimensional dataset and then relaxes the constraints to turn the problem into a reduced-dimensional problem for the Laplacian feature map.

SpH [6]: Spherical Hashing. It is a binary coding embedding technique that uses a hypersphere to segment data points and improves retrieval performance.

In order to ensure the reliability of the experiment, the above six kinds of hash algorithms are all based on the original data without any processing, and its parameter settings are consistent with the original literature.

5.3. Evaluation Criteria

To perform fair evaluation, we utilize the Hamming ranking search, which is commonly used in literature. All sample points in the database are ranked according to their Hamming distance, and the expected neighbors are returned from the top $K$ samples. The retrieval performance is measured by two widely used metrics: mean average precision (MAP) and precision-recall curves. Among them, the MAP score is calculated by Equation (13):

$$\text{MAP} = \frac{1}{|Q|} \sum_{i=1}^{|Q|} \frac{1}{n_i} \sum_{k=1}^{n_i} \text{precision}(R_k)$$

Where $q \in Q$ is a query and $n_i$ is the number of points associated with $q_i$ in the dataset. Then, suppose that the relevant points are ordered as $[r_1, r_2, \cdots, r_n]$, and $R_a$ is the set of ranked retrieval results from the top result until the point $r_a$ is reached.

5.4. Experimental Results and Analysis

MAP is one of the most comprehensive and effective standards for evaluating image retrieval. In this paper, the whole process integrates the binarization segmentation idea of the bagging strategy to matrix splice multiple dimensionality reduction matrix blocks. In the PCA processing, we only take the top eigenvectors of the PCA projection, which contains most of the discriminating information so that unnecessary noise and information in the projection process are reduced. More importantly, we apply random rotation to the top eigenvectors to reduce the quantization error between encodings, which greatly improves the retrieval efficiency of the original LPP hashing technique.

In order to better verify the performance of the method in this paper, we chose the most advanced hashing method for comparison with RLPH. As illustrated in Table 2, the MAP scores for 96-bit, 128-bit, and 256-bit encodings are shown for our method and other six methods under the three face datasets. It is easy to see from the table that RLPH has the highest MAP scores. Specifically, we find that data-dependent hashing methods such as SpH can achieve generally superior performance to some eigendecomposition-based methods, such as PCAH, with an increase in hash code bits. However, some hash methods based on eigendecomposition such as SH and PCAH are inferior to the data-independent hash method LSH because some noise eigenvectors are introduced when long binary code is learned. For PCAH, its MAP scores gradually decrease as the number of coded bits increases because of increased noise information in the three face databases. RLPH succeeded in avoiding this phenomenon by simply extracting useful information from the PCA projection process. On the other hand, the retrieval performance of PCAH increases rapidly. It can be seen that the key of the random orthogonal matrix is the PCAH hashing approach. Our method makes full use of the random orthogonal matrix to rotate the top eigenvectors of PCA randomly, which makes it possible to obtain a variety of projection matrices and reduce the quantization error between codes.

In addition, it is easy to find that our method gradually increases MAP scores as the encoding length increases. This phenomenon is a normal trend. As the size of the code increases, much of the useful information is entirely in RLPH. From
this perspective, our approach is very similar to LSH. More importantly, our approach consistently outperforms other methods. This shows that its advantages are obvious. This shows that its advantages are obvious.

Table 2. Comparison scores of Mean Average Precision (MAP) for different methods with various code lengths in three different datasets of face images

<table>
<thead>
<tr>
<th>Methods</th>
<th>AR</th>
<th>ORL</th>
<th>Yale</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>96-bits</td>
<td>128-bits</td>
<td>256-bits</td>
</tr>
<tr>
<td>SpH</td>
<td>0.4173</td>
<td>0.4625</td>
<td>0.4625</td>
</tr>
<tr>
<td>SH</td>
<td>0.4284</td>
<td>0.3823</td>
<td>0.3129</td>
</tr>
<tr>
<td>PCAH</td>
<td>0.1965</td>
<td>0.1761</td>
<td>0.1409</td>
</tr>
<tr>
<td>CNNH-AdaGrad</td>
<td>0.3683</td>
<td>0.4678</td>
<td>0.3986</td>
</tr>
<tr>
<td>CNNH-RMSProp</td>
<td>0.4261</td>
<td>0.3749</td>
<td>0.2964</td>
</tr>
<tr>
<td>CNNH-SGD</td>
<td>0.4489</td>
<td>0.2300</td>
<td>0.4932</td>
</tr>
<tr>
<td>RLPH</td>
<td><strong>0.5665</strong></td>
<td><strong>0.6162</strong></td>
<td><strong>0.6779</strong></td>
</tr>
</tbody>
</table>

At the same time, the performance of RLPH with deep learning algorithms CNN-SGD, CNN-RMSProp and CNN-AdaGrad are compared. Three different kinds of optimization algorithm of CNN's consistent parameter is set, the choice of neural network hidden layer Relu activation function, the output layer select Sigmoid activation function, each iteration step length is 0.001, the momentum coefficient and the attenuation coefficient is 0.9. Figure 2 shows the MAP comparison curve for each method. It can be seen that, due to the small data set used in the experiment, the performance of the deep learning algorithm is not well reflected. The MAP curve of RLPH is higher than that of the latter several methods, and the MAP value increases gradually with the increase of hash coding number.

As illustrated in the figure, our approach RLPH consistently outperforms all other competitors after encoding from 64-bits on three datasets. From the figure, it can be clearly seen that the PCAH method rapidly decreases its MAP curve as the encoding length increases. This is because the eigenvectors enter some noise when the PCA performs eigendecomposition. The appearance of these noises makes the projection doped with unnecessary eigenvectors. In this paper, RLPH fully utilizes PCA processing to solve the eigenvalue problem encountered in the calculation of LPP, so as to improve the recognition effect of the algorithm. The results imply that the strategy proposed in our method is more effective than those in other approaches to handle the imbalance problem of eigendecomposition.

The Precision & Recall curve is another important evaluation criterion that reflects the overall performance of different hashing methods. We conducted comparative experiments under three datasets. As shown in Figure 2, our approach is better than other hashing performances. As we can see, the search rate of our method in image retrieval occupies a great advantage in many methods.

Since the number at the top eigenvector of the PCA projection is an important parameter in our RLPH method, we performed additional experiments to test its effect on our model by changing the number r of eigenvectors. We changed the number of r from 2 to 32 to calculate the MAP value under different bits of the three datasets. From the results of the MAP curve in Figure 4, we can see that the retrieval performance of our method varies with the number of selected feature vectors in the PCA subspace. The performance of the RLPH method is better when the number of eigenvectors is neither large nor small. When the value of r is larger, more principal component projections will be applied in each reduced-dimensional matrix block, which may perform many noise eigenvectors and degrade performance. When the value of r is small, only a few eigenvectors are extracted in each reduced-dimensional matrix block. The few major projections like this are far from enough to capture the structural information of the PCA projection data. It can be seen that the best experimental results are achieved when the number of eigenvectors is 8.
6. Conclusion

In this paper, we propose a new hashing method named RLPH and only retain the top discriminative eigenvectors after PCA data, and we use a random orthogonal matrix to rotate eigenvectors to minimize the quantization error between codes. Finally, combining the obtained multiple thin projection matrix blocks together greatly improves the efficiency of the RLPH method in image retrieval. In addition, this method can take into account the non-adjacent information of the data during the training process so as to maintain the global structure of the data in the projection process. Nevertheless, this method also
has some problems that need to be improved. Its limitation is that it is suitable mainly for small sample sets, and further improvement is needed for large training samples. Therefore, solving these existing deficiencies will be the focus of future works.

Acknowledgement

This work was supported in part by Key Research Project of Henan Province Higher School (18B520017) and Doctor Fund of Henan Polytechnic University (B2014-043)

References

27. P. Li and P. Ren, “RPCH: Hashing with Two-Fold Randomness on Principal Projections,” Neurocomputing, pp. 236-244,
2017

Shan Zhao graduated from Xidian University with a Ph.D. She entered the School of Computer Science and Technology of Henan Polytechnic University in 2007 and became an associate professor in 2010. She is currently a visiting scholar at the University of Limerick in Ireland. She is also a member of the China Computer Federation. Her current research interests include image processing and pattern recognition.

Yongsi Li is a Master’s student in the School of Computer Science and Technology at Henan Polytechnic University. Her research interests include image retrieval based on hash and deep learning.