Electronic Research Archive

http://www.aimspress.com/journal/era

ERA, 31(9): 5845–5865.
DOI: 10.3934/era.2023297
Received: 25 May 2023
Revised: 13 July 2023
Accepted: 14 August 2023
Published: 30 August 2023

Research article

Feature adaptive multi-view hash for image search

Li Sun* **and Bing Song**

Department of Internet Security, Henan Police College, Zhengzhou 450046, China

* **Correspondence:** Email: SL@hnp.edu.cn.

Abstract: With the rapid development of network technology and small handheld devices, the amount of data has significantly increased and various kinds of data can be supplied to us at the same time. Recently, hashing technology has become popular in executing large-scale similarity search and image matching tasks. However, most of the prior hashing methods are mainly focused on the choice of the high-dimensional feature descriptor for learning effective hashing functions. In practice, real world image data collected from multiple scenes cannot be descriptive enough by using a single type of feature. Recently, several unsupervised multi-view hashing learning methods have been proposed based on matrix factorization, anchor graph and metric learning. However, large quantization error will be introduced via a sign function and the robustness of multi-view hashing is ignored. In this paper we present a novel feature adaptive multi-view hashing (FAMVH) method based on a robust multi-view quantization framework. The proposed method is evaluated on three large-scale benchmarks CIFAR-10, CIFAR-20 and Caltech-256 for approximate nearest neighbor search task. The experimental results show that our approach can achieve the best accuracy and efficiency in the three large-scale datasets.

Keywords: feature adaptive; compact hashing code; multi-view; approximate nearest neighbor search

1. Introduction

The rise of mobile phone and internet users is also fueling the explosive growth of data. This brings two challenges, i.e., (1) how to save the storage cost [1, 2]; (2) how to achieve fast nearest neighbor retrieval [3, 4]. The storage cost of saving the original feature is prohibitively high for the first challenge. The classic linear scan solution to nearest neighbor search is not scalable in practical applications for the second challenge. For the second challenge, approximate nearest neighbor search gives faster search speed than nearest neighbor retrieval by balancing accuracy and speed of arithmetic. Among them, tree-based search schemes perform well for low-dimensional data by partitioning the data space via various tree structures [5, 6]. For high-dimensional data, these methods cannot achieve satisfactory performance and do not guarantee faster search compared to linear scan [7].

For the first challenge, hash methods [7, 8] have been proposed to encode large-scale data, such as images, videos, documents and so on, into a sequence of bits, called hash code. They are designed to embed data from high dimensional feature space into a similarity-preserving low dimensional Hamming space. Hash methods can be categorized into two main categories of hashing algorithms, namely data-independent and data-dependent methods. Data-independent methods include the famous locality sensitive hashing (LSH). The hash codes are generated by a random hash functions satisfying the local sensitive property for various distance measures [9–12] followed by rounding. Data-dependent methods can be categorized into unsupervised hashing methods and supervised hashing methods. In unsupervised hashing methods, graph-based [8, 13–17] and vector quantization hashing [18–20] methods have attracted considerable attention in hashing. In graph-based hashing methods, data-dependent compact binary codes can be obtained by graph based dimensionality reduction. In vector quantization based hashing methods, the feature space is divided into a set of subspaces and compact binary codes are obtained via quantizing each subspace separately into clusters. For example, Jégou et al. [18] introduced a product quantization based approach for approximate nearest neighbor search. The basic idea is to divide the feature space into a Cartesian product of low dimensional subspaces and then to quantize each subspace separately into clusters. Norouzi et al. [19] extended product quantization and introduced a rotation matrix into quantization loss. Zhang et al. [20] proposed a novel compact coding approach. The basic idea is to use the composition of several elements selected from the dictionaries to accurately approximate a vector. A constraint condition is introduced into the objective function that the summation of the inner products of all pairs of elements that are used to approximate the vector from different dictionaries is constant. In the supervised hashing method, the class labels information approximate nearest neighbor search. Shakhnarovich [21] designed a boosted similarity sensitive coding to learn a similarity from weighted Hamming distance for task-specific similarity search. Hinton et al. [22] proposed a binary encoding method with stacked restricted boltzmann machines (RBMs), which can be applied to learn binary codes. Kulis et al. [23] proposed a binary reconstructive embedding (BRE) hashing method based on explicitly minimizing the reconstruction error between the metric space and Hamming metric. Wang et al. [24] proposed a semi-supervised hashing (SSH) method, which minimized empirical error over the labeled set and maximized the variance of each hash function at the same time.

Most of the prior data-dependent hashing methods take only a single type of feature into consideration, referred to as single-view hashing. In practice, images captured from different scenes in the real world should be represented by different kinds of features to obtain a comprehensive and accurate description. These features describe different characteristics of the given image from different views. Incorporating the multiple features into hash learning, which is referred to as multi-view hashing, has been explored widely. The multi-view hashing methods can be divided into two categories: view-specific hashing methods and view-integrated hashing methods. View-specific hashing methods learn independent hash codes for each view of an instance, and then concatenate multiple view-specific binary codes into the final hash codes. Bronstein et al. [25] proposed a framework for supervised similarity learning based on embedding data from different feature spaces into the Hamming space. The hash codes are learned using AdaBoost framework. Kumar et al. [26] proposed a cross-view hashing method, which is based on a spectral hashing framework. The similarity of pairwise instances in the Hamming space is computed by summing the Hamming distance of inter views and intra views. Zhen and Yeung [27] proposed a Co-Regularized hashing method, which is based on the large-margin strategy while effectively preserving the inter-modality similarity. The view-integrated hashing methods learn

unified hash codes for each instance. Zhang et al. [28] proposed a composite hashing with multiple information sources (CHMIS) method, which is based on spectral hashing and the maximum margin framework. The affinity matrix is computed by the summation of the affinity matrix from each view. A consistency constraint that the output of hash functions should be close to final integrated hash codes is introduced. Kim et al. [29] proposed a multi-view anchor graph hashing (MVAGH) method, which extended single view spectral hashing to multi-view hashing. All single view anchor graphs are combined to approximate the averaged neighborhood graph. Then a random walk method on the multi-view anchor graph is utilized to compute the similarity of pairwise instances. Kim et al. [30] proposed a multi-view spectral hashing (MVSH) method, which integrated multi-view information into binary codes, and uses the product of codewords to avoid undesirable embedding. Wang et al. [31] proposed a partial multi-modal, i.e., some views of the instances are missing, hashing method. This type of hashing method combines the formulation of matrix decomposition that can find the latent shared semantics, and spectral clustering. Liu et al. [32] proposed a multi-view alignment hashing method based on regularized kernel nonnegative matrix factorization. Liu et al. [33] proposed a multiple feature kernel hashing method, which preserved certain similarities with linearly combined multiple kernels corresponding to different features. However, large quantization error will be introduced via a sign function and the robustness of multi-view hashing is ignored.

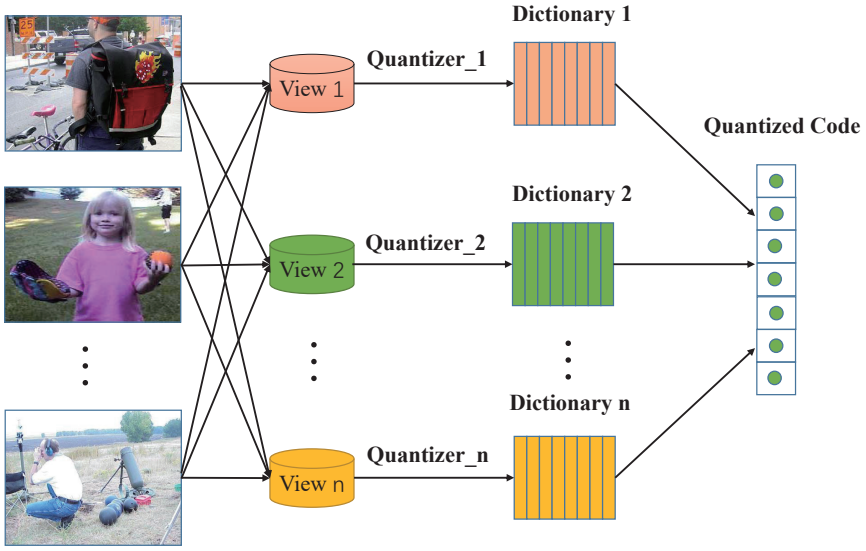

Figure 1. The motivation of the proposed multi-view quantization method.

In this paper, we propose a novel feature adaptive multi-view hashing (FAMVH) method based on a robust multi-view quantization framework as shown in Figure 1. Specifically, we utilize multiple feature representations for each image and learning adaptive weights in the vector quantization process. In addition, the $\ell_{2,1}$-norm loss is embedded in minimizing the reconstruction error and alleviating the impact of outliers. Finally, our approach is evaluated on three large-scale datasets for an approximate nearest neighbor (ANN) search task. All in all, our method is a view-integrated hashing method based on $\ell_{2,1}$-norm loss, which is robust to outlier data. Our contributions in this paper are summarized in the following:

- We propose a novel feature adaptive hashing method which can integrate multi-view to generate compact and integrated hash codes.
- An efficient algorithm is designed to optimize the proposed method.
- Our approach achieves superior performances in ANN experiments on three large scale datasets.

The remainder of this paper is organized as follows: In Section 2, we briefly introduce the related work. Section 3 presents the detail of our proposed approach. Discussions and analyses are given in Section 4. Section 5 provides extensive experimental validation on three datasets. The conclusions and future work are given in Section 6.

2. Related work

2.1. Product quantization

Product quantization (PQ) [18] splits each P dimensional feature x_i into M disjoint subvectors, where the dimension of each subvector is S_m and $\Sigma_{m=1}^{M} S_m = P$. The PQ method formulates the encoding problem as follows:

$$\begin{aligned} \min\ \mathscr{L}(\mathbf{C},\mathbf{B}) &= \sum_{i}^{N} \|\mathbf{x}_i - \begin{bmatrix} \mathbf{C}^1\mathbf{b}_i^1 \\ \vdots \\ \mathbf{C}^M\mathbf{b}_i^M \end{bmatrix}\|^2 \\ s.t.\ \mathbf{b}_i^m &\in \mathcal{H}_{1/K} \\ \|\mathbf{b}_i^m\|_1 &= 1, i \in \{1,\cdots,N\}, m \in \{1,\cdots,M\}, \end{aligned} \tag{2.1}$$

where N is the number of sample instances, $\mathbf{C}^i$ is m-th sub codebook and each column is a S_m dimensional sub codeword, $\mathcal{H}_{1/K} \equiv \{\mathbf{b}|\mathbf{b} \in \{0,1\}^K \ and\ \|\mathbf{b}\| = 1\}$, i.e., $\mathbf{b}$ is a binary vector comprising a 1-of-K encoding. The disadvantages of this method are as follows: 1) the ℓ_2-norm used in the objective function is not robust to the outlier data, 2) the reconstruction error is high and thus leads to limited search accuracy.

2.2. Cartesian k-means

The original p-dimensional feature space is splitted into m subspaces. m subcenters can be obtained by sub-quantizing the features in each subspace into h centers. In [19], each center of original feature space is modeled with a compositional parameterization of m subcenters, each with h elements. The learning objective for Cartesian k-means is to minimise the within-cluster squared distances:

$$\begin{aligned} \min\ \mathscr{L}(\mathbf{C},\mathbf{B}) &= \sum_{\mathbf{x}\in\mathbb{D}} \min_{\{\mathbf{b}^{(i)}\}_{i=1}^{m}} \|\mathbf{x} - \sum_{i=1}^{m} \mathbf{C}^{(i)}\mathbf{b}^{(i)}\|^2 \\ s.t.\ \mathbf{b}^{(i)} &\in \mathcal{H}_{1/h}, \end{aligned} \tag{2.2}$$

where $\mathbb{D} \equiv \{\mathbf{x}_j\}_{j=1}^{n}$ denotes a dataset comprising n p-dim data points. $\mathbf{C} \equiv [\mathbf{C}^{(1)},\cdots,\mathbf{C}^{(m)}] \in \mathbb{R}^{p\times mh}$ is block diagonal matrix comprising m subcenters set, where $\mathbf{C}^{(i)} \in \mathbb{R}^{p\times h}$ is a matrix whose columns comprise the elements of the i^{th} subcenter set and $\mathbf{C}^{(i)^T}\mathbf{C}^{(j)} = 0_{h\times h}, \forall i \neq j$. $\mathbf{C}$ can be expressed as a product $\mathbf{RD}$, where $\mathbf{R} \in \mathbb{R}^{p\times p}$ is an orthogonal matrix, i.e., $\mathbf{R}^T\mathbf{R} = \mathbf{I}_{p\times p}$, and $\mathbf{D} \in \mathbb{R}^{p\times mh}$ is a block diagonal matrix.

$$\mathbf{R} = [\mathbf{R}^{(1)}, \cdots, \mathbf{R}^{(m)}]$$

$$\mathbf{D} = \begin{bmatrix} \mathbf{D}^{(1)} & 0 & \dots & 0 \\ 0 & \mathbf{D}^{(2)} & & 0 \\ \vdots & & \ddots & \vdots \\ 0 & 0 & \dots & \mathbf{D}^{(m)} \end{bmatrix}$$

Then $\mathbf{C}^{(i)} = \mathbf{R}^{(i)}\mathbf{D}^{(i)}$. $\mathbf{D}^{(i)} \in \mathbb{R}^{s_i \times h}$ and $\mathbf{R}^{(i)} \in \mathbb{R}^{p \times s_i}$, where $s_i = rank(C^i)$, $\sum_{i=1}^{m} s_i \leq p$.

The objective in Eq (2.2) can be solved by an iterative coordinate descent method. Since the rotation matrix $\mathbf{R}$ is optimally learned, the CKM can achieve lower quantization error than PQ. The performance with ℓ_2-norm embedded in the objective function is susceptible to the outlier data. In addition, the CKM is a single view hashing method and can not integrate multiple features naturally.

2.3. Multi-view k-means clustering

Cai et al. [34] proposed a heterogeneous features clustering method by sharing the same clustering results in different views. $\mathbf{X}^{(v)} \in \mathbb{R}^{d_v \times n}$,$v = 1, \dots, M$, is used to represent the input data matrix with n instances and d_v-dim features in the v-th view. $\mathbf{F}^{(v)} \in \mathbb{R}^{d_v \times K}$ is the cluster centroid matrix in the v-th view. $\mathbf{G} \in \mathbb{R}^{K \times n}$ is the shared cluster assignment matrix and each row of $\mathbf{G}$ satisfies the 1-of-K encoding scheme, i.e., $\mathcal{H}_{1/K}$. The proposed multi-view K-means clustering method can be solved:

$$\begin{aligned} \min_{\mathbf{F}^v, \mathbf{G}, \alpha^{(v)}} \ & \sum_{v=1}^{M} (\alpha^{(v)})^{\gamma} \|\mathbf{X}^{(v)} - \mathbf{F}^{(v)}\mathbf{G}\|_{2,1} \\ & s.t.\ G_{ki} \in \{0, 1\},\ \sum_{k=1}^{K} G_{ki} = 1,\ \sum_{v=1}^{M} \alpha^{(v)} = 1, \end{aligned} \tag{2.3}$$

where $\|\mathbf{X}\|_{2,1} = \sum_{i=1}^{n} \|\mathbf{x}_i\|_2$, $\mathbf{x}_i$ is the i-th column of $\mathbf{X}$. $\ell_{2,1}$-norm has been proved robust to noise points [35–37].

The solution of the energy function in Eq (2.3) can be obtained by alternate optimizing over all the variables and the intermediate auxiliary variable. It can not generate compact hash codes as the cost of storing the centers grows linearly with K.

3. Proposed method

The framework of the proposed method is demonstrated in Figure 2. In this section, we first define some terms and notations that will be utilized throughout the paper. In the following subsection, we introduce our feature adaptive multi-view hashing approach, referred as FAMVH. The alternate optimization algorithm is presented in the next subsection. The convergence analysis of FAMVH is given in the end.

3.1. Terms and notations

Let V denote the number of views, N be the number of instances, $\mathbf{X}^v = [\mathbf{x}_1^v, \mathbf{x}_2^v, \dots, \mathbf{x}_N^v] \in \mathbb{R}^{d_v \times N}$ denote the feature matrix corresponding to the v-th view of all the instances. Each $\mathbf{x}_i^v$ is split into

M disjoint subvectors $\{[\mathbf{x}_i^v]^m\}$. Assume the m-th subvector $\{[\mathbf{x}_i^v]^m\}$ contains s_m^v dimensions and then $\sum_{m=1}^{M} s_m^v = d_v$.

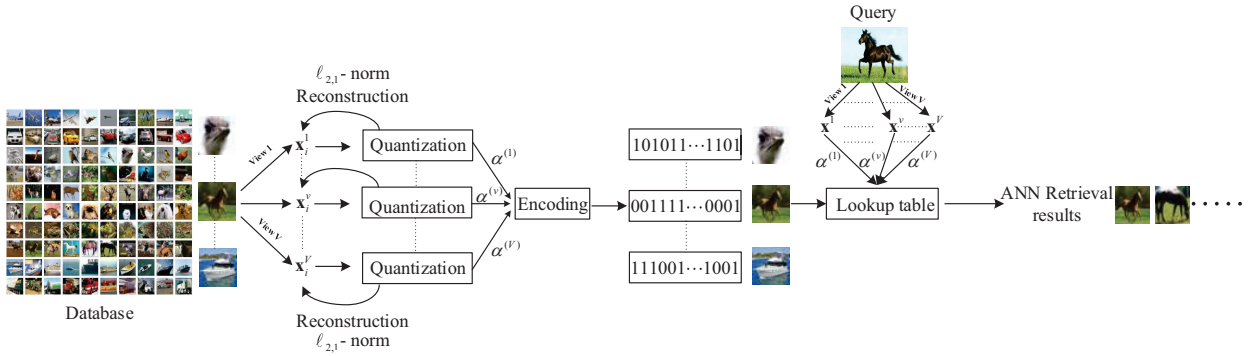

Figure 2. The framework of the proposed FAMVH method.

3.2. Formulation

The straight-forward way of combining all views of features is to concatenate all features together, which has been used in scene classification [38], object recognition [39] and etc. Feature concatenation preserves the raw information such that hashing methods can utilize the correlation among views. However, in such method, these features are often very high dimension, at the same time, the important view of features and the less important view of features are treated equally, which may not yield optimal hash codes since the structural information of each view has been lost. It is ideal to simultaneously perform the encoding using each view of features and unify their results based on their importance to the encoding task. To achieve the above purpose, three challenging problems have to be solved: 1) how to naturally assemble the multiple encoding losses, 2) how to learn the importance of views to the encoding task, 3) how to learn a compact hash code.

When performing a multi-view clustering algorithm, the clustering results in different views are required to be unique. Therefore, in multi-view hashing, we force the hash codes to be the same across different views. In the meantime, the data outliers can greatly affect the performance of hashing. To alleviate the effect of outlier data in the iterative optimization process, the sparsity-inducing norm, i.e., $\ell_{2,1}$-norm, is utilized in our quantization loss. The formulation of robust multi-view k-means hashing is as follows:

$$\begin{aligned} \min_{\mathbf{R}^v,\mathbf{D}_v^m,\mathbf{b}_i^m,\alpha^{(v)}} \quad & \sum_{i=1}^{N}\sum_{v=1}^{V}(\alpha^{(v)})^\gamma \|\mathbf{x}_i^v - \mathbf{R}^v \begin{bmatrix} \mathbf{D}_v^1\mathbf{b}_i^1 \\ \vdots \\ \mathbf{D}_v^M\mathbf{b}_i^M \end{bmatrix}\|_{2,1} \\ s.t.\ & (\mathbf{R}^v)^T\mathbf{R}^v = \mathbf{I}, \\ & \sum_{v=1}^{V}\alpha^{(v)} = 1, \\ & \|\mathbf{b}_i^m\|_1 = 1, \mathbf{b}_i^m \in \{0,1\}^K \\ & \forall\, i \in \{1,\cdots,N\}, m \in \{1,\cdots,M\}, \end{aligned} \tag{3.1}$$

where $\mathbf{D}_v^m \in \mathbb{R}^{s_m^v \times K}$ is the m-th sub codebook for v-th view and each column is a s_m^v-dimensional sub

codeword. $\mathbf{b}_i^m$ is the 1-of-K encoding on the m-th subvector. $\mathbf{R}^v \in \mathbb{R}^{d_v \times d_v}$ is a rotation matrix for the v-th view. $\alpha^{(v)}$ is the weight factor for the v-th view and γ is the parameter to control the weights distribution.

3.3. Optimization algorithm

The problem in Eq (3.1) is NP hard and difficult to solve with the discrete variable $\mathbf{b}_i^m$. There are four types of unknown variables, $\mathbf{R}^v, \mathbf{D}_v^m, \mathbf{b}_i^m$ and $\boldsymbol{\alpha}$. An alternating iteration process is performed where three of them are fixed, and the other one is optimized. Before performing optimization, we introduce some symbols for convenience as follows:

$$\mathbf{X}^v \triangleq [\mathbf{x}_1^v, \mathbf{x}_2^v, \cdots, \mathbf{x}_N^v], \tag{3.2}$$

$$\widehat{\mathbf{D}}_v \triangleq \begin{bmatrix} \mathbf{D}_v^1 & & \\ & \ddots & \\ & & \mathbf{D}_v^M \end{bmatrix}, \tag{3.3}$$

$$\widehat{\mathbf{b}}_i = [\mathbf{b}_i^{1^T}, \mathbf{b}_i^{2^T}, \cdots, \mathbf{b}_i^{M^T}]^T, \tag{3.4}$$

$$\mathbf{B} \triangleq [\widehat{\mathbf{b}}_1, \widehat{\mathbf{b}}_2, \cdots, \widehat{\mathbf{b}}_N]. \tag{3.5}$$

The object function is re-written in a matrix form as

$$\begin{aligned} \min_{\mathbf{R}^v, \widehat{\mathbf{D}}_v, \mathbf{B}, \boldsymbol{\alpha}} \quad & \sum_{v=1}^{V} (\alpha^{(v)})^\gamma \|\mathbf{X}^v - \mathbf{R}^v \widehat{\mathbf{D}}_v \mathbf{B}\|_{2,1} \\ s.t. \ & (\mathbf{R}^v)^T \mathbf{R}^v = \mathbf{I}, \\ & \sum_{v=1}^{V} \alpha^{(v)} = 1, \\ & \|\mathbf{b}_i^m\|_1 = 1, \mathbf{b}_i^m \in \{0,1\}^K \\ & \forall\ i \in \{1, \cdots, N\}, m \in \{1, \cdots, M\}, \end{aligned} \tag{3.6}$$

where $\mathbf{B} \in \{0,1\}^{MK \times N}$ has been defined in Eq (3.5).

To solve the problem in Eq (3.6), we add intermediate variables $\{\boldsymbol{\Lambda}^v\}$ and reformulate the object function as

$$\mathcal{L}_0 = \min_{\mathbf{R}^v, \widehat{\mathbf{D}}_v, \mathbf{B}, \alpha^{(v)}} \sum_{v=1}^{V} (\alpha^{(v)})^\gamma H^v, \tag{3.7}$$

where $H^v = Tr\{(\mathbf{X}^v - \mathbf{R}^v \widehat{\mathbf{D}}_v \mathbf{B}) \boldsymbol{\Lambda}^v (\mathbf{X}^v - \mathbf{R}^v \widehat{\mathbf{D}}_v \mathbf{B})^T\}$. $\boldsymbol{\Lambda}^v \in \mathbb{R}^{N \times N}$ is the diagonal matrix for v-th view. The i-th entry on the diagonal is defined as follows:

$$\Lambda_{ii}^v = \frac{1}{2\|\mathbf{E}_i^v\|_2 + \delta}, \ \forall\ i = 1, 2, \cdots, N, \tag{3.8}$$

where δ is a very small positive value to avoid the denominator being zero and $\mathbf{E}_i^v$ is the i-th column of $\mathbf{E}^v$, which is defined as follows:

$$\mathbf{E}^v = \mathbf{X}^v - \mathbf{R}^v\widehat{\mathbf{D}}_v\mathbf{B}. \tag{3.9}$$

- Solve $\mathbf{R}^v$ with $\widehat{\mathbf{D}}_v$, $\mathbf{\Lambda}^v$, $\mathbf{B}$ and $\boldsymbol{\alpha}$ fixed:
  The objective function can be written as

$$\begin{aligned} &\min_{\mathbf{R}^v} \ Tr\{(\mathbf{X}^v - \mathbf{R}^v\widehat{\mathbf{D}}_v\mathbf{B})\mathbf{\Lambda}^v(\mathbf{X}^v - \mathbf{R}^v\widehat{\mathbf{D}}_v\mathbf{B})^T\} \\ &s.t. \ (\mathbf{R}^v)^T\mathbf{R}^v = \mathbf{I}. \end{aligned} \tag{3.10}$$

  Equation (3.10) can be rewritten as

$$\begin{aligned} &\min_{\mathbf{R}^v} \ \|\mathbf{X}^v(\mathbf{\Lambda}^v)^{\frac{1}{2}} - \mathbf{R}^v\widehat{\mathbf{D}}_v\mathbf{B}(\mathbf{\Lambda}^v)^{\frac{1}{2}}\|_2^2 \\ &s.t. \ (\mathbf{R}^v)^T\mathbf{R}^v = \mathbf{I}, \end{aligned} \tag{3.11}$$

  where $(\mathbf{\Lambda}^v)^{\frac{1}{2}}$ denotes the square root of entries on the diagonal of matrix $\mathbf{\Lambda}^v$. Then optimizing $\mathbf{R}^v$ is the standard Procrustes Problem [40]. The solution of the problem can be obtained by single value decomposition (SVD): $\mathbf{X}^v\mathbf{\Lambda}^v(\widehat{\mathbf{D}}_v\mathbf{B})^T = \mathbf{U}\mathbf{\Sigma}\mathbf{V}^T$, the optimal $\mathbf{R}^v$ will be $\mathbf{U}\mathbf{V}^T$.
- Solve $\widehat{\mathbf{D}}_v$ with $\mathbf{R}^v$, $\mathbf{\Lambda}^v$, $\mathbf{B}$ and $\boldsymbol{\alpha}$ fixed:
  With straightforward algebraic manipulation, Eq (3.11) can be written as

$$\min_{\widehat{\mathbf{D}}_v} \ \|(\mathbf{R}^v)^T\mathbf{X}^v(\mathbf{\Lambda}^v)^{\frac{1}{2}} - \widehat{\mathbf{D}}_v\mathbf{B}(\mathbf{\Lambda}^v)^{\frac{1}{2}}\|_2^2 \tag{3.12}$$

  since

$$\begin{aligned} &\|\mathbf{X}^v(\mathbf{\Lambda}^v)^{\frac{1}{2}} - \mathbf{R}^v\widehat{\mathbf{D}}_v\mathbf{B}(\mathbf{\Lambda}^v)^{\frac{1}{2}}\|_2^2 \\ =&\|(\mathbf{R}^v)^T\mathbf{X}^v(\mathbf{\Lambda}^v)^{\frac{1}{2}} - \widehat{\mathbf{D}}_v\mathbf{B}(\mathbf{\Lambda}^v)^{\frac{1}{2}}\|_2^2 + \|(\mathbf{R}^v)^{\perp T}\mathbf{X}^v(\mathbf{\Lambda}^v)^{\frac{1}{2}}\|_2^2 \end{aligned} \tag{3.13}$$

  where columns of $(\mathbf{R}^v)^\perp$ span the orthogonal complement of the column-space of $\mathbf{R}^v$, i.e., the block matrix $[\mathbf{R}^v \ (\mathbf{R}^v)^\perp]$ is orthogonal. The second term in the formula described above is independent of $\widehat{\mathbf{D}}_v$.
  Taking the derivative of $\mathcal{L}_1 = \|(\mathbf{R}^v)^T\mathbf{X}^v(\mathbf{\Lambda}^v)^{\frac{1}{2}} - \hat{\mathbf{D}}_v\mathbf{B}(\mathbf{\Lambda}^v)^{\frac{1}{2}}\|_2^2$ with respect to $\widehat{\mathbf{D}}_v$, we can get

$$\frac{\partial\mathcal{L}_1}{\hat{\mathbf{D}}_v} = 2\hat{\mathbf{D}}_v\mathbf{B}\mathbf{\Lambda}^v\mathbf{B}^T - 2\mathbf{R}^v\mathbf{X}^v\mathbf{\Lambda}^v\mathbf{B}^T. \tag{3.14}$$

  Setting Eq (3.14) as 0, each $\widehat{\mathbf{D}}_v$ can be updated as $\mathbf{R}^v\mathbf{X}^v\mathbf{\Lambda}^v\mathbf{B}^T(\mathbf{B}\mathbf{\Lambda}^v\mathbf{B}^T)^\dagger$, where $(\cdot)^\dagger$ denotes the matrix pseudo inverse. As the pseudo inverse will cause numerical instability, we will adopt another optimization method which is more simple. Since $\hat{\mathbf{D}}_v$ is a diagonal block matrix and $\mathbf{\Lambda}^v$ is a diagonal matrix where the entries in the diagonal are positive, the Eq (3.13) can rewritten as

$$\min_{\{\mathbf{D}_v^m\}_{m=1}^M} \ \sum_{i=1}^{N}\sum_{m=1}^{M}\Lambda_{ii}^v\|[(\mathbf{R}^v)^T\mathbf{x}_i^v]^m - \mathbf{D}_v^m\mathbf{b}_i^m\|_2^2, \tag{3.15}$$

  where $[(\mathbf{R}^v)^T\mathbf{x}_i^v]^m$ is the m-th subvector of $(\mathbf{R}^v)^T\mathbf{x}_i^v$. $\{\mathbf{D}_v^m\}_{m=1}^M$ can be optimized independently by solving

$$\min_{\mathbf{D}_v^m} \sum_{i=1}^{N} \Lambda_{ii}^v \|[(\mathbf{R}^v)^T \mathbf{x}_i^v]^m - \mathbf{D}_v^m \mathbf{b}_i^m\|_2^2, \tag{3.16}$$

where $\mathbf{b}_i^m$ is the class label vector Eq (3.16) is similar to the weighted k-means clustering. Let $\mathcal{K}$ be the number of classes. Then $\mathbf{D}_v^m$ can be updated as

$$[\mathbf{D}_v^m]_p = \frac{\sum_{\mathbf{b}_i^m \in C_k} \Lambda_{ii}^v [(\mathbf{R}^v)^T \mathbf{x}_i^v]^m}{\sum_{\mathbf{b}_i^m \in C_k} \Lambda_{ii}^v}, \tag{3.17}$$

where C_k denotes class label k, p is the index corresponding to the p-th entry in vector $\mathbf{b}_i^m$ whose value is 1 and $[\mathbf{D}_v^m]_p$ denotes the p-th column of $\mathbf{D}_v^m$.

- Solve $\mathbf{B}$ with $\mathbf{R}^v$, $\mathbf{\Lambda}^v$, $\widehat{\mathbf{D}}_v$ and $\boldsymbol{\alpha}$ fixed:
From Eq (3.6), $\mathbf{B}$ can be decoupled into columns and optimized column by column.

$$\begin{aligned} \min_{\mathbf{b}_i^m} & \sum_{v=1}^{V} (\alpha^{(v)})^\gamma \|\mathbf{x}_i^v - \mathbf{R}^v \widehat{\mathbf{D}}_v \widehat{\mathbf{b}}_i\|_2 \\ s.t.\ & \|\mathbf{b}_i^m\|_1 = 1, \mathbf{b}_i^m \in \{0,1\}^K \\ & \forall\ m \in \{1, \cdots, M\}, \end{aligned} \tag{3.18}$$

where $\widehat{\mathbf{b}_i}$ has been given in Eq (3.4). With the intermediate variable $\mathbf{\Lambda}^v$, Eq (3.18) can be reformulated

$$\begin{aligned} \min_{\mathbf{b}_i^m} & \sum_{v=1}^{V} (\alpha^{(v)})^\gamma \Lambda_{ii}^v \|\mathbf{x}_i^v - \mathbf{R}^v \widehat{\mathbf{D}}_v \widehat{\mathbf{b}}_i\|_2^2 \\ s.t.\ & \|\mathbf{b}_i^m\|_1 = 1, \mathbf{b}_i^m \in \{0,1\}^K \\ & \forall\ m \in \{1, \cdots, M\}, \end{aligned} \tag{3.19}$$

Equation (3.18) can be expanded as in Eq (3.13). The formula is given as

$$\begin{aligned} \min_{\mathbf{b}_i^m} & \sum_{v=1}^{V} \sum_{m=1}^{M} (\alpha^{(v)})^\gamma \Lambda_{ii}^v \|[(\mathbf{R}^v)^T \mathbf{x}_i^v]^m - \mathbf{D}_v^m \mathbf{b}_i^m\|_2^2 + const \\ s.t.\ & \|\mathbf{b}_i^m\|_1 = 1, \mathbf{b}_i^m \in \{0,1\}^K \\ & \forall\ m \in \{1, \cdots, M\}, \end{aligned} \tag{3.20}$$

where $[(\mathbf{R}^v)^T \mathbf{x}_i^v]^m$ is the m-th subvector of $(\mathbf{R}^v)^T \mathbf{x}_i^v$, $const$ is independent of variable $\mathbf{b}_i^m$. $\mathbf{b}_i^m$ can be solved by optimizing

$$\begin{aligned} \min_{\mathbf{b}_i^m} & \|\widetilde{\mathbf{x}}_i^{(m)} - \widetilde{\mathbf{D}}^{(m)} \mathbf{b}_i^m\|_2^2 \\ s.t.\ & \|\mathbf{b}_i^m\|_1 = 1, \mathbf{b}_i^m \in \{0,1\}^K, \end{aligned} \tag{3.21}$$

where

$$\begin{aligned}\widetilde{\mathbf{x}}_i^{(m)} &= [(\alpha^{(1)})^\gamma \Lambda_{ii}^1([(\mathbf{R}^1)^T \mathbf{x}_i^1]^m)^T, \cdots, (\alpha^{(V)})^\gamma \Lambda_{ii}^V([(\mathbf{R}^V)^T \mathbf{x}_i^V]^m)^T]^T, \\ \widetilde{\mathbf{D}}^{(m)} &= [(\alpha^{(1)})^\gamma \Lambda_{ii}^1(\mathbf{D}_1^m)^T, \cdots, (\alpha^{(V)})^\gamma \Lambda_{ii}^V(\mathbf{D}_V^m)^T]^T,\end{aligned} \tag{3.22}$$

The $\mathbf{b}_i^m$ can be obtained by performing nearest neighbor (NN) search for $\widetilde{\mathbf{x}}_i^{(m)}$ to find the assignments coefficients. $\mathbf{\Lambda}^v$ can be updated by Eq (3.8).

$$\begin{aligned}&\min_{\alpha^{(v)}} \sum_{v=1}^{V} (\alpha^{(v)})^\gamma H^v \\ &s.t. \sum_{v=1}^{V} \alpha^{(v)} = 1,\ \alpha^{(v)} \geq 0.\end{aligned} \tag{3.23}$$

The Lagrange function associated with Eq (3.23) is defined as

$$\mathcal{L}_3(\alpha^{(v)}, \lambda) = \sum_{v=1}^{V} (\alpha^{(v)})^\gamma H^v + \lambda(\sum_{v=1}^{V} \alpha^{(v)} - 1). \tag{3.24}$$

Taking the derivative of $\mathcal{L}_3$ with respect to $\alpha^{(v)}$, we can find the optimum $\alpha^{(v)}$ by setting the derivative to zero.

$$\frac{\mathcal{L}_3}{\alpha^{(v)}} = \gamma(\alpha^{(v)})^{\gamma-1} H^v + \lambda = 0, \tag{3.25}$$

which yields the optimal

$$\alpha^{(v)} = (-\frac{\lambda}{\gamma H^v})^{\frac{1}{\gamma-1}}. \tag{3.26}$$

Substitute the optimal $\alpha^{(v)}$ in Eq (3.26) into the constraint $\sum_{v=1}^{V} \alpha^{(v)} = 1$, we get

$$\alpha^{(v)} = \frac{(\gamma H^v)^{\frac{1}{1-\gamma}}}{\sum_{v=1}^{V} (\gamma H^v)^{\frac{1}{1-\gamma}}}. \tag{3.27}$$

The optimization process is performed by alternatively updating $\mathbf{\Lambda}^v$, $\mathbf{R}^v$, $\widetilde{\mathbf{D}}^{(m)}$, $\mathbf{b}_i^m$ and $\alpha^{(v)}$ until convergence. The whole algorithm is summarized in Algorithm 1.

Algorithm 1 The Algorithm of FAMVH.

Input:

The training set for V views $\{\mathbf{X}^1, \cdots, \mathbf{X}^V\}$ and $\mathbf{X}^v \in \mathbb{R}^{d_v \times N}, v = 1, \cdots, V$;
The number of splitting feature subvectors M;
The number of sub codewords K;
The parameter γ;

Output:

The rotation matrix $\mathbf{R}^v \in \mathbb{R}^{d_v \times d_v}$ for each view;
The codebook on m-th subvector $\widehat{\mathbf{D}}_v \in \mathbb{R}^{s_m^v \times K}$ for v-th view;
The learned weight $\alpha^{(v)}$ for each view;
The learned hash codes $\mathbf{B} \in \{0, 1\}^{KM \times N}$;

Initialization:

Initialize $\alpha^{(v)} = \frac{1}{M}, v = 1, \cdots, V$;
$\mathbb{R}^v$ is initialized as the identity matrix $\mathbf{I}$;
$\mathbf{D}_v^m$ is initialized by selecting K data points from the m-th subvector of v-th view randomly;
$\mathbf{b}_i^m \in \{0, 1\}^K$ is initialized by optimizing Eq (3.21);

1: **repeat**
2: Compute the diagonal matrix $\mathbf{\Lambda}^v$ via Eq (3.8);
3: Update $\mathbf{R}^v$ through optimizing Eq (3.11);
4: Update $\mathbf{D}_v^m$ with Eq (3.17);
5: Update $\mathbf{b}_i^m$ via optimizing Eq (3.20);
6: Update $\alpha^{(v)}$ by Eq (3.26);
7: **until** Converges

3.4. Convergence analysis

The convergence proof of our proposed method is given as follows. Let the original objective function in Eq (3.6) be $\mathcal{L}(\mathbf{R}, \widehat{\mathbf{D}}, \mathbf{B}, \boldsymbol{\alpha})$, where $\mathbf{R} = \{\mathbf{R}^v\}_{v=1}^V, \widehat{\mathbf{D}} = \{\widehat{\mathbf{D}}_v\}_{v=1}^V$. For the t-step iteration, the alternate update procedure contains the following four subproblems:

$$\begin{aligned}
\mathbf{R}^{(t)} &\longleftarrow \underset{\mathbf{R}}{\operatorname{argmin}}\, \mathcal{L}(\mathbf{R}, \widehat{\mathbf{D}}^{(t-1)}, \mathbf{B}^{(t-1)}, \boldsymbol{\alpha}^{(t-1)}), \\
\widehat{\mathbf{D}}^{(t)} &\longleftarrow \underset{\widehat{\mathbf{D}}}{\operatorname{argmin}}\, \mathcal{L}(\mathbf{R}^{(t)}, \widehat{\mathbf{D}}, \mathbf{B}^{(t-1)}, \boldsymbol{\alpha}^{(t-1)}), \\
\mathbf{B}^{(t)} &\longleftarrow \underset{\mathbf{B}}{\operatorname{argmin}}\, \mathcal{L}(\mathbf{R}^{(t)}, \widehat{\mathbf{D}}^{(t)}, \mathbf{B}, \boldsymbol{\alpha}^{(t-1)}), \\
\boldsymbol{\alpha}^{(t)} &\longleftarrow \underset{\boldsymbol{\alpha}}{\operatorname{argmin}}\, \mathcal{L}(\mathbf{R}^{(t)}, \widehat{\mathbf{D}}^{(t)}, \mathbf{B}^{(t)}, \boldsymbol{\alpha}),
\end{aligned} \tag{3.28}$$

where the update procedure of auxiliary variables $\{\mathbf{\Lambda}^v\}_{v=1}^V$ is omitted for convenience. Thus we have the following inequality:

$$\begin{aligned}
&\mathcal{L}(\mathbf{R}^{(t-1)}, \widehat{\mathbf{D}}^{(t-1)}, \mathbf{B}^{(t-1)}, \boldsymbol{\alpha}^{(t-1)}),\\
&\geq \mathcal{L}(\mathbf{R}^{(t)}, \widehat{\mathbf{D}}^{(t-1)}, \mathbf{B}^{(t-1)}, \boldsymbol{\alpha}^{(t-1)}),\\
&\geq \mathcal{L}(\mathbf{R}^{(t)}, \widehat{\mathbf{D}}^{(t)}, \mathbf{B}^{(t-1)}, \boldsymbol{\alpha}^{(t-1)}),\\
&\geq \mathcal{L}(\mathbf{R}^{(t)}, \widehat{\mathbf{D}}^{(t)}, \mathbf{B}^{(t)}, \boldsymbol{\alpha}^{(t-1)}),\\
&\geq \mathcal{L}(\mathbf{R}^{(t)}, \widehat{\mathbf{D}}^{(t)}, \mathbf{B}^{(t)}, \boldsymbol{\alpha}^{(t)}) \geq \cdots,
\end{aligned} \tag{3.29}$$

It indicates that $\mathcal{L}(\mathbf{R}^{(t)}, \widehat{\mathbf{D}}^{(t)}, \mathbf{B}^{(t)}, \boldsymbol{\alpha}^{(t)})$ are monotonic nonincreasing as $t \to \infty$. Finally, the algorithm will converge to a local optimal solution.

4. Discussion and analysis

4.1. Computation complexity analysis

The update of the diagonal matrix $\mathbf{\Lambda}^v$ depends on the matrix $\mathbf{E}^v$ in Eq (3.8). The time complexity to compute $\mathbf{E}^v$ is $O(Nd_v^2)$. The complexity of updating the diagonal matrix $\mathbf{\Lambda}^v$ is $O(Nd_v)$. The complexity of updating $\mathbf{R}^v$ requires $O(Nd_v^2 + Nd_v + d_v^3)$ time. The complexity of updating the dictionary $\mathbf{D}_v^m$ is $O(N(K + s_m^v) + Ns_m^v d_v)$. The complexity of updating $\mathbf{b}_i^m$ is $O(K + \sum_{v=1}^V s_m^v d_v + K\sum_{v=1}^V s_m^v)$. The updating of $\widehat{\mathbf{D}}_v$ and $\mathbf{B}$ can be implemented in parallel computating. The complexity of updating $\boldsymbol{\alpha}$ is $O(V)$. The total time complexity of the proposed method is $O(TN\sum_{v=1}^V d_v^2)$, where T is the number of iterations.

4.2. Approximate nearest neighbor search

In this subsection, we discuss the ANN for comparing quantization techniques. Let $\mathbf{q}^v \in \mathbb{R}^{d_v}$ be the v-th view of a query instance and $\mathbf{x}_i^v \in \mathbb{R}^{d_v}$ be the v-th view of the i-th database point. The i-th instance is encoded as $\widehat{\mathbf{b}}_i \in \{0, 1\}^{MK}$ which has been defined in Eq (3.4). The asymmetric quantizer distance between a query q and the binary code $\widehat{\mathbf{b}}_i$ of i-th instance is

$$distAQD(q, \widehat{\mathbf{b}}_i) = \sum_{v=1}^{V} (\alpha^{(v)})^\gamma \|\mathbf{q}^v - \mathbf{R}^v \widehat{\mathbf{D}}_v \widehat{\mathbf{b}}_i\|_2. \tag{4.1}$$

Given a query, these distances for each v, and all $K \times M$ possible values of $\widehat{\mathbf{b}}_i$ can be pre-computed and stored in a query-specific $V \times K \times M$ lookup table. Finally, there are V addition operations to compute the distance $distAQD(q, \widehat{\mathbf{b}}_i)$. If the query point is also represented by the binary codes, the symmetric quantizer distance between the query binary codes $\widehat{\mathbf{b}}_q$ and $\widehat{\mathbf{b}}_i$ is given by

$$distSQD(\widehat{\mathbf{b}}_q, \widehat{\mathbf{b}}_i) = \sum_{v=1}^{V} (\alpha^{(v)})^\gamma \|\mathbf{R}^v \widehat{\mathbf{D}}_v \widehat{\mathbf{b}}_q - \mathbf{R}^v \widehat{\mathbf{D}}_v \widehat{\mathbf{b}}_i\|_2. \tag{4.2}$$

We can pre-compute $(\alpha^{(v)})^\gamma \|\mathbf{R}^v \widehat{\mathbf{D}}_v \widehat{\mathbf{b}}_q - \mathbf{R}^v \widehat{\mathbf{D}}_v \widehat{\mathbf{b}}_i\|_2$ and store them in an $V \times M \times K \times K$ lookup table. Finally, there are V addition operations to compute the distance $distSQD(\widehat{\mathbf{b}}_q, \widehat{\mathbf{b}}_i)$.

5. Experiments

5.1. Evaluation metric

For each dataset, we use each instance in the testing set as a query example to retrieve instances in the training set. The Hamming ranking by the approximate distance criteria is adopted to perform fair evaluation. In our following experiments, we use three different metrics [24]: the precision, recall and mean average precision (MAP) for the top N retrieved instances.

$$precision@top(N) = \frac{N_t}{N}, \tag{5.1}$$

where N_t and R_t denotes the true number of retrieved true nearest neighbor instances in top N retrieved instances.

$$recall@top(N) = \frac{N_t}{T}, \tag{5.2}$$

where T denotes the number of relevant true nearest neighbors.

Average precision (AP) of top N retrieved instances is defined as [41]:

$$AP = \frac{1}{N_t}\sum_{i=1}^{N} precision@top(i) \times \delta(i), \tag{5.3}$$

where $\delta(i)$ is a indicator function, i.e., if the item at rank i is a relevant true nearest neighbor of the given query, $\delta(i) = 1$, otherwise $\delta(i) = 0$. Then MAP can be computed as the mean of AP.

5.2. Experimental settings

In this section, we evaluate our proposed methods for the high dimensional nearest neighbour search problem with multiple features. Our experiments are carried on three different datasets: Caltech-256[*], CIFAR-10 and CIFAR-20 [†]. The images from Caltech-256 dataset spans 256 object categories ranging from grasshopper to tuning fork with rich color information and texture detail information. The CIFAR-10 and CIFAR-20 datasets share the same images, which are widely used in the community of image retrieval.

Caltech-256. It consists of 30,607 images, each of which is associated with 256 object categories with more than 80 images per category.

CIFAR-10. It consists of 60,000 32×32 color images collected from a subset of 80-million tiny images dataset[‡] [42]. It contains 10 class labels with 6,000 images in each class.

CIFAR-20. It consists of 60,000 32×32 color images, which belong to 20 superclasses. It is referred as the CIFAR-100 dataset with the 100 fine classes label removed.

For all datasets, we randomly select 1000 images as the query set and the rest of the dataset is used as the training set, following the experimental setting in [29, 43]. We use scene descriptor GIST [44], shape descriptor HOG [45], local binary descriptor LBP [46], color histogram descriptor, and the bag of visual words descriptor BOW. More concretely, the experimental settings are listed as follows:

-GIST: Gabor filters are applied with 8 different orientations and 4 scales. Each Gabor-filtered image is then averaged over 4×4 grid, which leads to a 512-dimensional vector ($8 \times 4 \times 16 = 512$).

[*] http://www.vision.caltech.edu/Image_Datasets/Caltech256/

[†] https://www.cs.toronto.edu/~kriz/cifar.html

[‡] http://horatio.cs.nyu.edu/mit/tiny/data/index.html

-HOG: The image gradients are computed in non-overlapping windows, where the orientation of gradients is quantized into 9 bins and normalized with 4 different normalization factors. Each image is resized into 64×64. We use the publicly available VLFeat toolbox[§] [47] to extract HOG features. The cell size is set to 8×8. The length of the ultimate feature vector is 1984-dimensions ($8 \times 8 \times 31 = 1984$).

-LBP: The uniform LBP labels the pixels of an image by thresholding a 4×4 neighborhood, and the response is mapped to a 243-dimensional vector.

-ColorHist: The intensities are quantized into 64 bins for each color channel which yield a 192-dimensional color histogram vector ($3 \times 64 = 192$).

-BOW: The dense SIFT features are quantized into 300-dimensional visual words.

In the test phase, a retrieved point is regarded as a true nearest neighbor if it lies in the top 100, 500 and 500 points closest to a query for Caltech-256, CIFAR-10 and CIFAR-20, respectively, similar to [32]. Since we use $\ell_{2,1}$-norm in our model, we also verify the existing methods based on $\ell_{2,1}$-norm distance for fair comparison. The distance between a query $\mathbf{q}$ and an instance $\mathbf{x}$ in the database is calculated as $dist(\mathbf{q}, \mathbf{x}) = \sum_{v=1}^{V} \|\mathbf{q}^v - \mathbf{x}^v\|_2$, where $\mathbf{q}^v, \mathbf{x}^v$ are the v-th view of $\mathbf{q}$ and $\mathbf{x}$, respectively. The experiments are performed using Matlab 2014a on a server configured with a 2-core processor and 32 GB of RAM running the Windows OS.

5.3. Evaluation on large scale datasets

In order to validate the efficiency of our approach, we design two experiments: approximate nearest neighbor experiments and comparison experiments on multiple views.

5.3.1. Approximate nearest neighbor experiments

In ANN experiments, we compare our FAMVH with CKM [19] hash and PQ [18]. We set $K = 256$ to make lookup tables small as the same in [19]. All the features are normalized. For CKM and PQ, the normalized multiple features are concatenated into a single representation. We use the publicly available source codes of CKM and PQ to perform the comparison experiments. In our approach, a parameter γ is used to control the weight factor distribution among all views. The optimal γ is selected from one of $\{0.001, 0.01, 0.1, 1.01, 10, 100\}$ which yields the best performance. We report the experimental results with M being set to 4, 8, 16 for code length 32, 64, 128. The maximal number of iterations for the whole algorithm was set to 10. In Figures 2, 3 and 4, the abbreviations AQ and SQ denote the asymmetric quantizer distance and symmetric quantizer distance metrics in approximate nearest neighbor search, respectively.

Figures 3, 4 and 5 illustrate the recall on Caltech-256, CIFAR-10 and CIFAR-20, respectively. As shown in Figure 3, our approach and CKM outperform PQ for the code length 32, 64 and 128 under the same type of approximate nearest neighbor distance metric. The performances of our approach and CKM are competitive. FAMVH does not seem to have a significant advantage over CKM. The main reason can be attributed to the larger category sizes and larger clutter categories. From the results on CIFAR-10 and CIFAR-20 as shown in Figures 4 and 5, we can see that our approach outperforms all the others under the same type of approximate nearest neighbor distance metric. The performance of methods with the AQ metric significantly outperforms the corresponding methods with the SQ metric. The reason is that the symmetric quantizer distance encodes both the query and the database instances, while the asymmetric quantizer distance only encodes the database instances. As the code length increases, the

[§]http://www.vlfeat.org/

performances of our approach and CKM increase too. The performance of PQ increases as the code length increases from 32 bit to 64 bit and decreases from 64 bit to 128 bit. The possible reason is that the lack of an optimal rotation matrix yields inconsistent results with the true nearest neighbors.

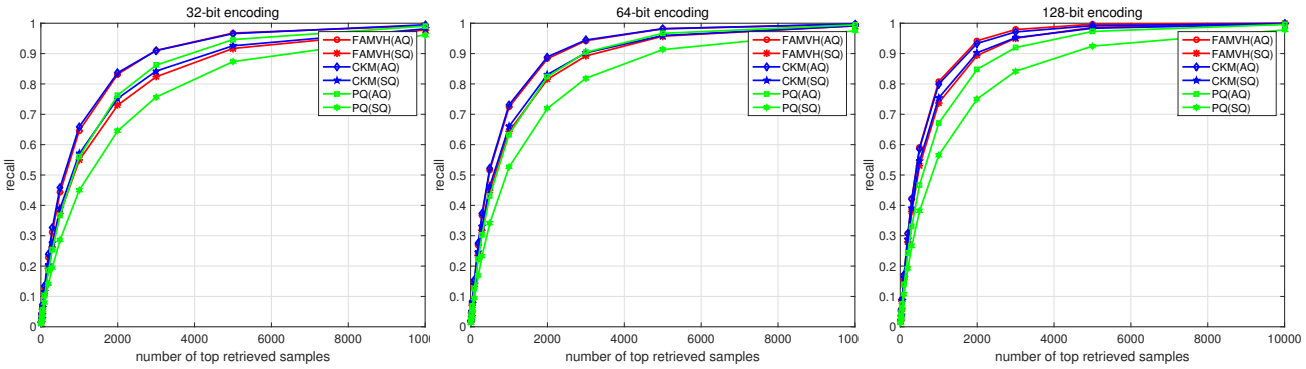

Figure 3. Recall for ANN search based on different quantizers and corresponding distance functions on Caltech256 dataset. The suffix 'AQ' or 'SQ' in the bracket refers to the asymmetric quantization distance or the symmetric quantization distance in ANN search.

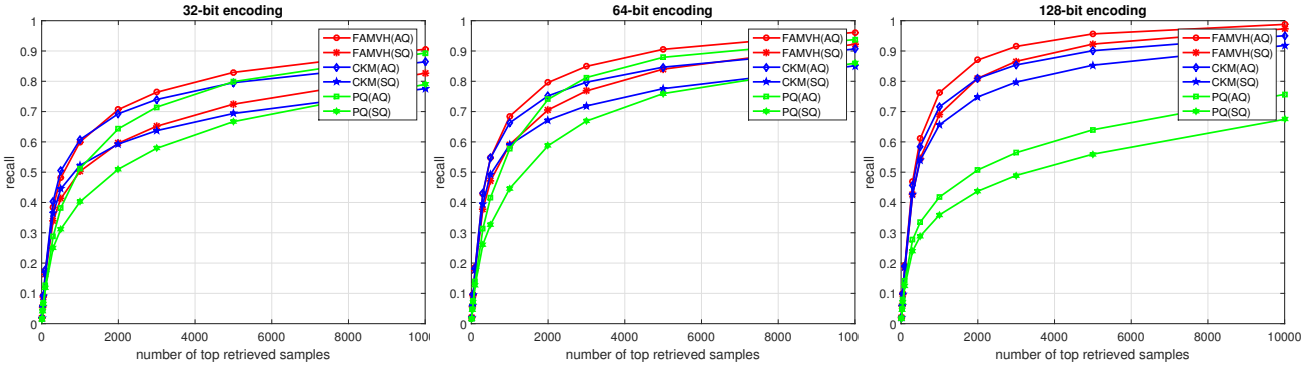

Figure 4. Recall for ANN search based on different quantizers and corresponding distance functions on CIFAR-10 dataset. The suffix 'AQ' or 'SQ' in the bracket refers to the asymmetric quantization distance or the symmetric quantization distance in ANN search.

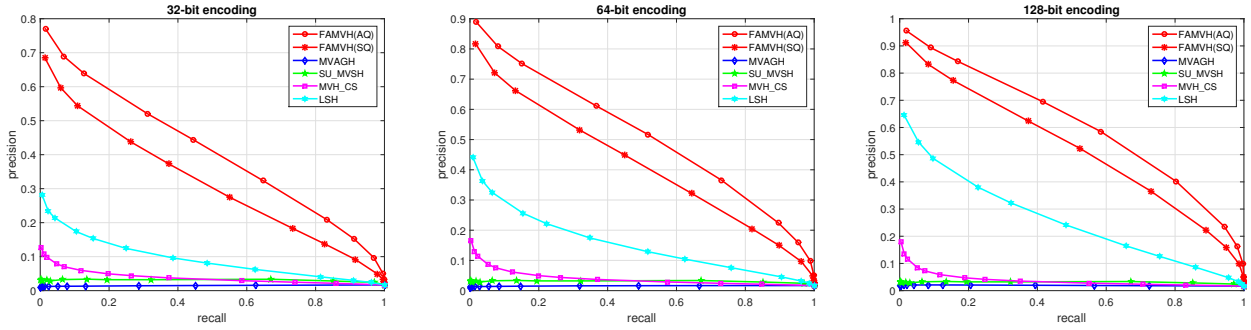

Figure 5. The precision-recall curves of all compared algorithms on Caltech-256 dataset. The suffix 'AQ' or 'SQ' in the bracket refers to the asymmetric quantization distance or the symmetric quantization distance in ANN search.

5.3.2. Comparison experiments on multiples views

In the multiple views experiment, we compare our approach (FAMVH) against four popular unsupervised multiview hashing algorithms, i.e., MVAGH [29], Sequential Update for Multi-View Spectral Hashing (SU_MVSH) [30], Multi-View Hashing (MVH_CS) [26] and LSH. Since LSH is a data independent hashing method, it can be extended to deal with the multiple views hashing problem with multiple features being concatenated into a single representation. The experimental settings are the same in ANN experiments. It is worth noting that these benchmark methods including MVAGH, SU_MVSH and MVH_CS share similar ideas of feature clustering with the proposed method. The source codes of [26, 29, 30] have not been published. We implement them ourselves. The results in terms of precision-recall curves are reported in Figures 6, 7 and 8 respectfully. The corresponding MAP values are listed in Table 1. As shown in Figures 6, 7 and 8, our approach achieves the most superior performance gain of all the compared algorithms on Caltech-256, CIFAR-10 and CIFAR-20 datasets. The performance of LSH is next to us on the three datasets. MVAGH performs the worst over the three datasets. The main possible reason is that the computation of similarities is based on the low-rank approximation of the averaged similarity, which is computed by random walk, while the true nearest neighbor is calculated by the summation of the square root of distances in each view.

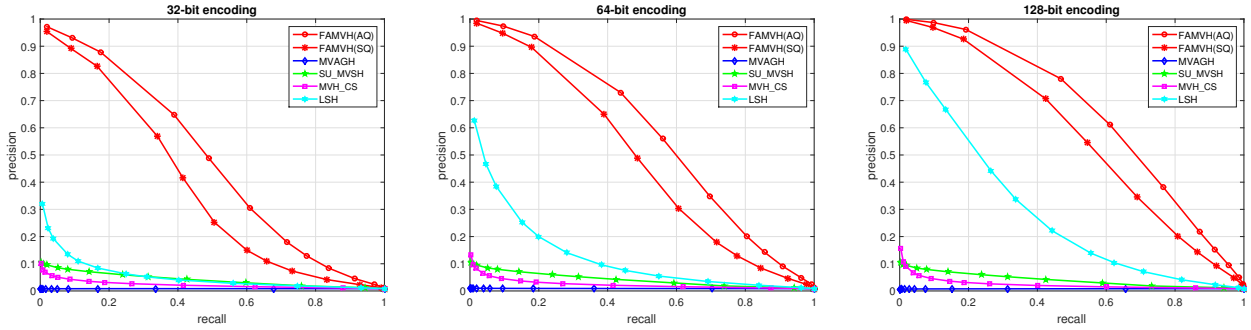

Figure 6. The precision-recall curves of all compared algorithms on CIFAR-10 dataset. The suffix 'AQ' or 'SQ' in the bracket refers to the asymmetric quantization distance or the symmetric quantization distance in ANN search.

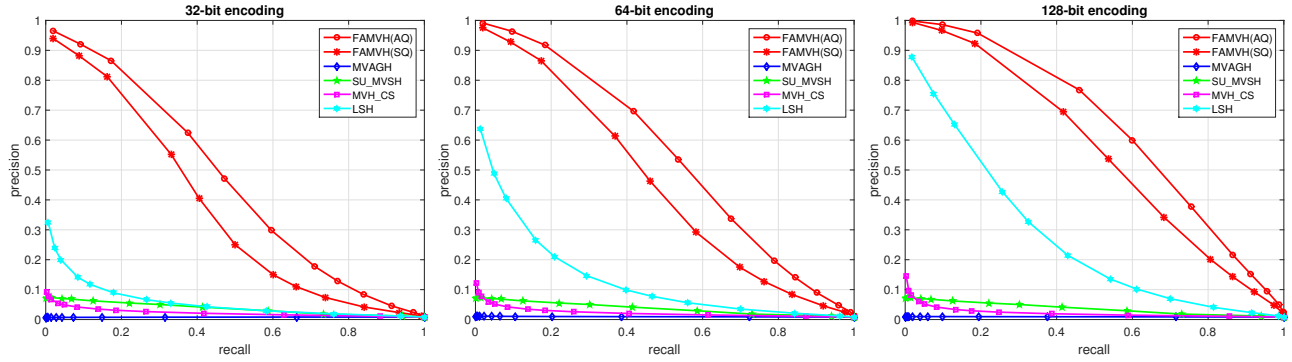

Figure 7. The precision-recall curves of all compared algorithms on CIFAR-20 dataset. The suffix 'AQ' or 'SQ' in the bracket refers to the asymmetric quantization distance or the symmetric quantization distance in ANN search.

Table 1. The comparison results of MAP on Caltech-256, CIFAR-10 and CIFAR-20 datasets.

Methods	Caltech-256			CIFAR-10			CIFAR-20		
	32 bits	64 bits	128 bits	32 bits	64 bits	128 bits	32 bits	64 bits	128 bits
MVAGH	0.0150	0.0160	0.0191	0.0084	0.0090	0.0080	0.0080	0.0095	0.0092
SU_MVSH	0.0312	0.0312	0.0312	0.0413	0.0408	0.0405	0.0370	0.0369	0368
MVH_CS	0.0381	0.0392	0.0374	0.0229	0.0242	0.0239	0.0226	0.0227	0.0220
LSH	0.0930	0.1542	0.2591	0.0529	0.1250	0.2825	0.0574	0.1338	0.2700
FAMVH(SQ)	0.3169	0.4188	5254	0.3808	0.4855	0.5675	0.3705	0.4532	0.5588
FAMVH(AQ)	**0.4114**	**0.5154**	**0.6125**	**0.4800**	**0.5809**	**0.6532**	**0.4619**	**0.5506**	**0.6408**

6. Conclusions

In this paper, we propose a feature adaptive multi-view hash method, which aims to generate a binary code to approximate the nearest neighbor (NN) search in multiple views. The method extends the classic cartesian k-means hashing method to multi-view. It not only integrates the multi-view naturally, but also generates compact hash codes. In experiments on large scale ANN search tasks, the experimental results are good. In our future work, the kernel extension of our method will be investigated in dealing with nonlinear feature mapping.

Use of AI tools declaration

The authors declare they have not used artificial intelligence (AI) tools in the creation of this article.

Acknowledgments

I would like to thank all anonymous reviewers for their constructive comments through each stage of the process.

Conflict of interest

The authors declare there is no conflicts of interest.

References

1. K. D. Doan, P. Yang, P. Li, One loss for quantization: Deep hashing with discrete wasserstein distributional matching, in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, (2022), 9447–9457.
2. J. M. Guo, A. W. H. Prayuda, H. Prasetyo, S. Seshathiri, Deep learning based image retrieval with unsupervised double bit hashing, *IEEE Trans. Circ. Syst. Vid. Technol.*, (2023), 1–15. https://doi.org/10.1109/TCSVT.2023.3268091
3. S. Li, X. Li, J. Lu, J. Zhou, Self-supervised video hashing via bidirectional transformers, in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, (2021), 13549–13558.
4. L. Wang, Y. Pan, C. Liu, H. Lai, J. Yin, Y. Liu, Deep hashing with minimal-distance-separated hash centers, in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, (2023), 23455–23464.
5. V. Gaede, O. Günther, Multidimensional access methods, *ACM Comput. Surv.*, **30** (1998), 170–231. https://doi.org/10.1145/280277.280279
6. J. H. Friedman, J. L. Bentley, R. A. Finkel, An algorithm for finding best matches in logarithmic expected time, *ACM Trans. Math. Softw.*, 1977. https://doi.org/10.1145/355744.355745
7. A. Gionis, P. Indyk, R. Motwani, Similarity search in high dimensions via hashing, in *International Conference on Very Large Data Bases*, (1999), 518–529.
8. Y. Weiss, A. Torralba, R. Fergus, Spectral hashing, in *Advances in Neural Information Processing Systems*, (2008), 1753–1760.
9. A.Z. Broder, On the resemblance and containment of documents, in *Compression and Complexity of Sequences*, (1997), 21–29.
10. M. S. Charikar, Similarity estimation techniques from rounding algorithms, *Appl. Comput. Harmon. Anal.*, (2002), 380–388.
11. M. Raginsky, S. Lazebnik, Locality-sensitive binary codes from shift-invariant kernels, in *Advances in Neural Information Processing Systems*, (2009), 1509–1517.
12. A. Torralba, R. Fergus, Y. Weiss, Small codes and large image databases for recognition, in *IEEE Conference on Computer Vision and Pattern Recognition*, (2008), 1–8. https://doi.org/10.1109/CVPR.2008.4587633

13. Y. Weiss, R. Fergus, A. Torralba, Multidimensional spectral hashing, in *European Conference on Computer Vision*, Springer, (2012), 340–353. https://doi.org/10.1007/978-3-642-33715-4_25

14. W. Liu, J. Wang, S. Kumar, S. F. Chang, Hashing with graphs, in *Proceedings of the 28th International Conference on Machine Learning*, (2011).

15. Q. Y. Jiang, W. J. Li, Scalable graph hashing with feature transformation, in *International Joint Conference on Artificial Intelligence*, (2015), 2248–2254.

16. L. Chen, D. Xu, I. W. H. Tsang, X. Li, Spectral embedded hashing for scalable image retrieval. *IEEE Trans. Cybern.*, **44** (2014), 1180–1190. https://doi.org/10.1109/TCYB.2013.2281366

17. Y. Gong, S. Lazebnik, A. Gordo, F. Perronnin, Iterative quantization: A procrustean approach to learning binary codes for large-scale image retrieval, in *IEEE Transactions on Pattern Analysis and Machine Intelligence*, **35** (2013), 2916–2929. https://doi.org/10.1109/TPAMI.2012.193

18. H. Jegou, M. Douze, C. Schmid, Product quantization for nearest neighbor search, *Trans. Pattern Anal. Mach. Intell.*, **33** (2011), 117–128. https://doi.org/10.1109/TPAMI.2010.57

19. M. Norouzi, D. J. Fleet, Cartesian k-means, in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, (2013), 3017–3024.

20. T. Zhang, C. Du, J. Wang, Composite quantization for approximate nearest neighbor search, in *Proceedings of the 31st International Conference on Machine Learning*, **32** (2014), 838–846.

21. G. Shakhnarovich, *Learning Task-Specific Similarity*, PhD thesis, Massachusetts Institute of Technology, 2005.

22. G. E. Hinton, R. R. Salakhutdinov, Reducing the dimensionality of data with neural networks, *Science*, **313** (2006),504–507. https://doi.org/10.1126/science.1127647

23. B. Kulis, T. Darrell, Learning to hash with binary reconstructive embeddings, in *Advances in Neural Information Processing Systems*, (2009), 22.

24. J. Wang, S. Kumar, S. F. Chang, Semi-supervised hashing for large-scale search, *IEEE Trans. Pattern Anal. Mach. Intell.*, **34** (2012), 2393–2406. https://doi.org/10.1109/TPAMI.2012.48

25. M. M. Bronstein, A. M. Bronstein, F. Michel, N. Paragios, Data fusion through cross-modality metric learning using similarity-sensitive hashing, in *2010 IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, (2010), 3594–3601. https://doi.org/10.1109/CVPR.2010.5539928

26. S. Kumar, R. Udupa, Learning hash functions for cross-view similarity search, in *Proceedings of the Twenty-Second International Joint Conference on Artificial Intelligence*, (2011), 1360–1365.

27. Y. Zhen, D. Y. Yeung, Co-regularized hashing for multimodal data, in *Advances in Neural Information Processing Systems*, (2012), 25.

28. D. Zhang, F. Wang, L. Si, Composite hashing with multiple information sources, in *Proceedings of the 34th international ACM SIGIR conference on Research and development in Information Retrieval*, (2011), 225–234. https://doi.org/10.1145/2009916.2009950

29. S. Kim, S. Choi, Multi-view anchor graph hashing, in *2013 IEEE International Conference on Acoustics, Speech and Signal Processing*, (2013), 3123–3127. https://doi.org/10.1109/ICASSP.2013.6638233

30. S. Kim, Y. Kang, S. Choi, Sequential spectral learning to hash with multiple representations, in *European Conference on Computer Vision*, Springer, (2012), 538–551. https://doi.org/10.1007/978-3-642-33715-4_39

31. Q. Wang, L. Si, B. Shen, Learning to hash on partial multi-modal data, in *Proceedings of the Twenty-Fourth International Joint Conference on Artificial Intelligence*, (2015), 3904–3910.

32. L. Liu, M. Yu, L. Shao, Multiview alignment hashing for efficient image search, *IEEE Trans. Image Process.*, **24** (2015), 956–966. https://doi.org/10.1109/TIP.2015.2390975

33. X. Liu, J. He, B. Lang, Multiple feature kernel hashing for large-scale visual search, *Pattern Recogn.*, **47** (2014), 748–757. https://doi.org/10.1016/j.patcog.2013.08.022

34. X. Cai, F. Nie, H. Huang, Multi-view k-means clustering on big data, in *Proceedings of the Twenty-Third International Joint Conference on Artificial Intelligence*, 2013.

35. J. Song, Y. Yang, X. Li, Z. Huang, Y. Yang, Robust hashing with local models for approximate similarity search, *IEEE Trans. Cybern.*, **44** (2014), 1225–1236. https://doi.org/10.1109/TCYB.2013.2289351

36. F. Nie, H. Huang, X. Cai, C. Ding, Efficient and robust feature selection via joint 2, 1-norms minimization, in *Advances in Neural Information Processing Systems*, (2010), 1813–1821.

37. Z. Ma, F. Nie, Y. Yang, J. R. R. Uijlings, N. Sebe, Web image annotation via subspace-sparsity collaborated feature selection, *IEEE Trans. Multimedia*, **14** (2012), 1021–1030. https://doi.org/10.1109/TMM.2012.2187179

38. X. Ren, L. Bo, D. Fox, RGB-(D) scene labeling: Features and algorithms, in *2012 IEEE Conference on Computer Vision and Pattern Recognition*, (2012), 2759–2766. https://doi.org/10.1109/CVPR.2012.6247999

39. K. Lai, L. Bo, X. Ren, D. Fox, A large-scale hierarchical multi-view RGB-D object dataset, in *2011 IEEE International Conference on Robotics and Automation*, (2011), 1817–1824. https://doi.org/10.1109/ICRA.2011.5980382

40. P. H. Schönemann, A generalized solution of the orthogonal procrustes problem, *Psychometrika*, **31** (1966), 1–10. https://doi.org/10.1007/BF02289451

41. G. Ding, Y. Guo, J. Zhou, Collective matrix factorization hashing for multimodal data, in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, (2014), 2083–2090.

42. A. Torralba, R. Fergus, W. T. Freeman, 80 million tiny images: A large data set for nonparametric object and scene recognition, *IEEE Trans. Pattern Anal. Mach. Intell.*, **30** (2008), 1958–1970. https://doi.org/10.1109/TPAMI.2008.128

43. A. Fitzgibbon, S. Lazebnik, P. Perona, Y. Sato, C. Schmid, *Computer Vision–ECCV 2012: 12th European Conference on Computer Vision, Florence, Italy, October 7–13, 2012, Proceedings, Part I*, Springer, Heidelberg, 2012.

44. A. Oliva, A. Torralba, Modeling the shape of the scene: A holistic representation of the spatial envelope, *Int. J. Comput. Vision*, **42** (2001), 145–175. https://doi.org/10.1023/A:1011139631724

45. N. Dalal, B. Triggs, Histograms of oriented gradients for human detection, in *2005 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'05)*, (2005), 886–893. https://doi.org/10.1109/CVPR.2005.177

46. T. Ojala, M. Pietikainen, T. Maenpaa, Multiresolution gray-scale and rotation invariant texture classification with local binary patterns, *IEEE Trans. Pattern Anal. Mach. Intell.*, **24** (2002), 971–987. https://doi.org/10.1109/TPAMI.2002.1017623

47. A. Vedaldi, B. Fulkerson, VLFeat: An open and portable library of computer vision algorithms, in *Proceedings of the 18th ACM international conference on Multimedia*, (2010), 1469–1472. https://doi.org/10.1145/1873951.1874249

AIMS Press