Learning Efficient Binary Codes From High-Level Feature Representations for Multilabel Image Retrieval

Lei Ma, Hongliang Li, *Senior Member, IEEE*, Fanman Meng, *Member, IEEE*, Qingbo Wu, *Member, IEEE*, and King Ngi Ngan, *Fellow, IEEE*

***Abstract*—Due to the efficiency and effectiveness of hashing technologies, they have become increasingly popular in large-scale image semantic retrieval. However, existing hash methods suppose that the data distributions satisfy the manifold assumption that semantic similar samples tend to lie on a low-dimensional manifold, which will be weakened due to the large intraclass variation. Moreover, these methods learn hash functions by relaxing the discrete constraints on binary codes to real value, which will introduce large quantization loss. To tackle the above problems, this paper proposes a novel unsupervised hashing algorithm to learn efficient binary codes from high-level feature representations. More specifically, we explore nonnegative matrix factorization for learning high-level visual features. Ultimately, binary codes are generated by performing binary quantization in the high-level feature representations space, which will map images with similar (visually or semantically) high-level feature representations to similar binary codes. To solve the corresponding optimization problem involving nonnegative and discrete variables, we develop an efficient optimization algorithm to reduce quantization loss with guaranteed convergence in theory. Extensive experiments show that our proposed method outperforms the state-of-the-art hashing methods on several multilabel real-world image datasets.**

***Index Terms*—Efficient binary codes, image semantic retrieval, nonnegative matrix factorization.**

I. Introduction

The exponential growth of big image data on the Web has posed a great challenge for the storage, analysis, and management of them. How to design an efficient and effective algorithm for large scale image semantic retrieval task has become a hot topic. The traditional real-valued descriptors like LBP [1]–[4], GIST [5], BOW [6], color histogram [7], color co-occurrence descriptor [8], etc. require huge computing and memory cost. In contrast, the binary descriptors have a great advantage in improving retrieving speed and reducing memory cost, which makes them easy for large-scale image retrieval [9]–[11].

For solving a large-scale image search problem based on real-valued descriptors, the classical tree-based methods such as kd-tree [12] and R-tree [13] methods are proposed. However, due to the existence of curse of dimensionality, the efficiency of the similarity search decreases with the increased dimension of data. In order to reduce the effects of the curse of dimensionality, some early data independent hash methods including Locality Sensitive Hashing (LSH) [14] and its variants [15] use random linear projections to map real-valued descriptors in a high-dimensional space into a low-dimensional Hamming space. The main limitation of these methods is that the retrieval performance is unsatisfactory with a small number of bits.

In order to learn more efficient and compact hash codes, some data dependent hashing methods [9]–[11], [16], [17] are designed to learn hash functions via machine learning. The main idea of data dependent methods is to learn a low-dimensional representation of data and then quantize [9] or threshold [11], [16] them into binary codes. Existing data dependent hashing methods can be roughly divided into three categories [18]: supervised hashing methods, semi-supervised hashing methods, and unsupervised hashing methods. The supervised hashing methods [19]–[30] make use of completely labeled data for training. However, with the rapidly growing data, it is expensive to label a large number of training samples. In order to reduce human labor intensity, semi-supervised hashing methods [16], [31] make use of both labeled and unlabeled data for training. Although the number of labeled data is reduced, semi-supervised hashing methods require experienced people to manually label some training samples.

Unsupervised hashing methods utilize the intrinsic data properties of the examples without using any label information to learn binary codes. To preserve the global variance structure of the input data, Principle Component Hashing (PCAH) [16], Iterative Quantization (ITQ) [9] and Isotropic Hashing (IsoH) [32] try to identify a set of hyperplanes with anisotropic [9], [16] or isotropic [32] PCA projection based on a linear manifold assumption i.e, the underling manifold is a linear subspace.

Manuscript received November 8, 2016; revised April 13, 2017; accepted April 28, 2017. Date of publication May 10, 2017; date of current version October 13, 2017. This work was supported in part by the National Natural Science Foundation of China under Grant 61525102, Grant 61502084, and Grant 61601102. The associate editor coordinating the review of this manuscript and approving it for publication was Prof. Enrico Magli. *(Corresponding author: Hongliang Li.)*

L. Ma, H. Li, F. Meng, and Q. Wu are with the School of Electronic Engineering, the Intelligent Visual Information Processing and Communication Laboratory (IVIPC), University of Electronic Science and Technology of China, Chengdu 611731, China (e-mail: leima_uestc@163.com; hlli@uestc.edu.cn; fmmeng@uestc.edu.cn; wqb.uestc@gmail.com).

K. N. Ngan is with the Department of Electronic Engineering, The Chinese University of Hong Kong, Hong Kong, China, and also with the School of Electronic Engineering, University of Electronic Science and Technology of China, Chengdu 611731, China (e-mail: knngan@ee.cuhk.edu.hk).

Color versions of one or more of the figures in this paper are available online at http://ieeexplore.ieee.org.

Digital Object Identifier 10.1109/TMM.2017.2703089

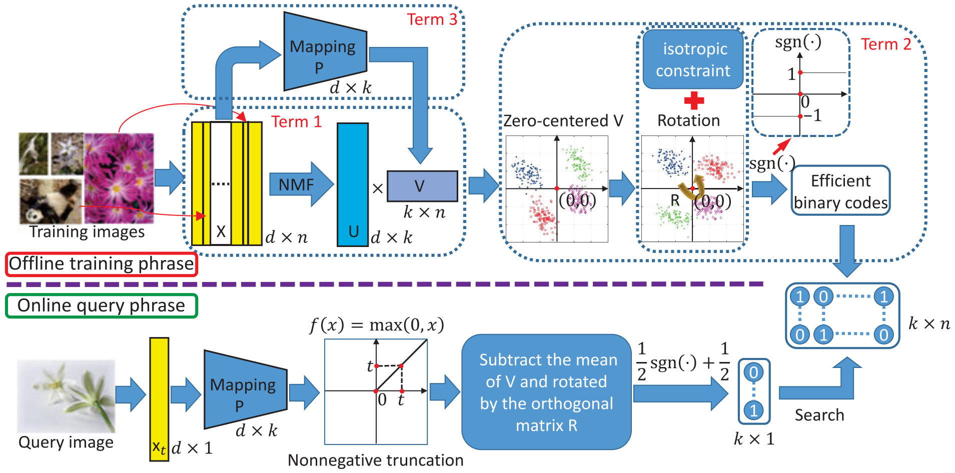

Fig. 1. Framework of the proposed method. The overall algorithm can be divided into two stages: offline training phrase and online query phrase. In the offline training phrase, three terms including Term 1, Term 2, and Term 3 are involved in generating efficient binary codes, where Term 1 refers to the highlevel feature representations learning term, Term 2, refers to the efficient binary codes learning term, and Term 3 refers to the out-of-sample extension term. In the online query phrase, the hash code of the query image can be obtained from the output of the process shown below the purple dotted line. Please see text for more details.

However, the linear manifold assumption is strict to a real-world application. Instead of assuming global linearity, Spectral Hashing (SH) [17], Anchor Graph Hashing (AGH) [10], Locally Linear Hashing (LLH) [33], and Induced Manifold Hashing (IMH) [11] make a weaker local neighborhood preserving assumption i.e., non-linear manifold assumption that nearby points in high-dimensional input space are close in the low-dimensional manifold subspace. The core idea of these methods is to learn hash functions based on manifold learning methods including: linear manifold learning methods (e.g. Principle Component Analysis (PCA)) or non-linear manifold learning methods (e.g. Spectral Clustering (SC) [34], Laplacian Eigenmaps (LE) [35], Isomap [36], Locally Linear Embedding (LLE) [37], Elastic Embedding (EE) [38], and t-Distributed Stochastic Neighbor Embedding (t-SNE) [39]). However, these methods widely adopt the manifold assumption that semantic similar samples tend to lie on a low-dimensional manifold. This assumption will be weakened when the intra-class similarities are low.

While the impact of intra-class variations can be alleviated by leveraging deep learning, vector quantization and nonnegative matrix factorization technologies, promising results are achieved in image retrieval [40]–[46]. Autoencoder (AE) [40], [47] and Deep Hashing (DH) [41] resort to the powerful fitting ability of deep neural networks to learn non-linear hash functions. Outputs of the last layer are signed to obtain the resulting binary codes, which will lead to the accumulated quantization error [31]. In addition, these methods usually have great computational burden and are prone to overfitting. To preserve high fitting ability and circumvent the risks of overfitting, Product Quantization (PQ) [42], Cartesian k-means (CKM) [43] and Composite Quantization (CQ) [44] turn to explore Vector Quantization (VQ) [48] technologies for learning binary codes. The key idea of VQ based methods is that the dimensions of the data can be separated into several disjoint subsets under the assumption of independence across dimensions, and each subset is quantized into a number of discrete state respectively. However, the assumption of independence across dimensions may destroy the latent semantic structure of data. To preserve the latent semantic structure of input data, Nonnegative Matrix Factorization Hashing (NMFH) [45] and Latent Structure Preserving Hashing (LSPH) [46] employ Nonnegative Matrix Factorization (NMF) [49] technology to decompose a input data matrix into a product of two nonnegative factors, which can be understood as the parts and the parts-based representations respectively since they only allow addition (not substraction) and its combination [50]. However, the NMF based hashing methods choose to solve a relaxed constraint problem and the quality of approximate solution is unsatisfactory due to the accumulated quantization error. For a more comprehensive introduction to the existing hashing methods, please refer to the review on hashing [51], [52].

The above unsupervised hashing methods either ignore the existing huge inter-class variations in real-world images or release discrete constraints on binary codes in the optimization, which may limit the effectiveness of learning efficient binary codes for complex scene/object images retrieval task. Therefore, it is a valuable research issue to simultaneously learn efficient binary codes and better data representation to solve image retrieval problem on real-world image databases.

In this paper, we propose a novel unsupervised hashing method by learning efficient binary codes from high-level feature representations which can describe the widely varying data distributions well. Due to the straightforward interpretability of NMF, it has attracted great attention recently and has been widely applied in several applications including face recognition [45], document clustering [53], recommender systems [54], etc. NMF can decompose the original real-value inputs into a basis matrix and a coefficient matrix which is the high-level representations of the original real-value inputs [55]. In addition, the flexibility of matrix factorization enables dealing with widely varying data distributions. Thus hash codes learned from high-level feature representations via NMF are expected to be more effective. As shown in Fig. 1, the (visually or semantically) similar images are expected to be represented as similar high-level feature representations via NMF and further mapped to similar binary codes. However, directly applying NMF to generate hash codes will result in two problems: 1) the solution of NMF problem is in general not unique [56], i.e., the basis matrix and the coefficient matrix are not unique, and as a result the resulting hashing codes are not unique; 2) there is no straightforward extension scheme for computing high-level feature representations of inputs in the test set, i.e., the problem of out-of-sample extension. From the point of view of information theory, all information bits in the resulting hash codes should be equally probable and independent of each other. In other words, efficient hash codes should simultaneously satisfy bits balance property, i.e., each bit has a 50% chance of being one or zero, and bits independence property, i.e., different bits are independent of each other [17]. Interestingly, the bits balance and bits independence constraints can be approximated by constraining the high-level feature representations with equal isotropic covariance, i.e., a diagonal matrix with all diagonal elements being the same. With this isotropic constraint, we can obtain unique solution of NMF problem like [57]. For the problem out-of-sample extension, we can introduce an explicit mapping between the original real-value inputs and the high-level feature representations. Therefore, we model our problem within a NMF framework, which includes the following three terms: a high-level feature representations learning term, an efficient binary code learning term, and an out-of-sample extension term. The first term aims to learn the high-level feature representations of input data. The second term aims to learn efficient binary codes. The last term aims to learn an explicit mapping from the input space to the high-level embedding space so that the high-level representations of the items in the test set can be computed directly. The introduction of the independence and balance constraints renders the optimization problem more complex and difficult. The problem can be simplified by relaxing the independence and balance constraints to one isotropic constraint on the high-level feature representations. However, the relaxed optimization problem is still a challenge optimization problem which contains nonnegative variables, continuous variables, a discrete variable and orthogonal constraints. Therefore, we develop an efficient coordinate descent algorithm to solve this challenge optimization problem. The convergent property of our proposed algorithm can be well guaranteed by a theoretical analysis. We evaluate our proposed algorithm on several multi-label real-world image datasets. The experimental results verify the effectiveness of our method.

The rest of this paper is organized as follows. In Section II, we present the details of the proposed method. Section III reports the experimental results by comparing with existing state-of-the-art methods on several real-world datasets. The conclusion is given in Section IV.

II. Learning Efficient Binary Codes From High-Level Feature Representations

Assume that we have a database $\mathbf{X} = \{\mathbf{x}_1, \ldots, \mathbf{x}_n\} \in \mathbb{R}^{d\times n}$, where $\mathbf{x}_i$ is a d-dimensional image descriptor of the i-th sample and n is the number of images. The purpose of image hashing is to learn hash functions $\mathcal{H} : \mathbb{R}^d \rightarrow \{-1, 1\}^k$ to map a training example $\mathbf{x}_i$ to a k-dimensional binary hash code $\mathbf{b}_i = \mathcal{H}(\mathbf{x}_i)$ in the Hamming space, while maintaining some notion of similarity evaluated in the real-value feature space.

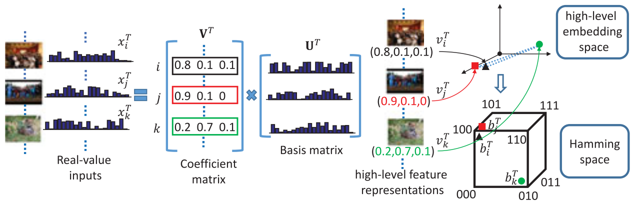

Fig. 2. Schematic illustration of the motivation. NMF can reconstruct the input nonegative real-value features with a basis matrix and a coefficient matrix, where each row in the coefficient matrix corresponds to a high-level feature representation of the original real-value inputs. The (visually or semantically) similar images are expected to have similar high-level feature representations, i.e., the distance between them in the high-level embedding space is small, while the distances are large for dissimilar images. The goal is to learn the hash codes which can preserve the neighborhood relationship from the high-level embedding space into the Hamming space.

In this section, we solve the hashing learning problem by learning efficient binary codes from high-level feature representations, of which the flowchart is shown in Fig. 2. As is illustrated in Fig. 2, the proposed method includes two stages: the offline training phase and the online query phrase. In the offline training phrase, a d-dimensional real-value visual feature is extracted from each image firstly. These features constitute an input data matrix $\mathbf{X} \in \mathbb{R}^{d\times n}$. We expect to learn a basis matrix $\mathbf{U} \in \mathbb{R}^{d\times k}$ and the high-level feature representations $\mathbf{V} \in \mathbb{R}^{k\times n}$ from $\mathbf{X} \in \mathbb{R}^{d\times n}$. $\mathbf{V}$ is zero-centered by subtracting the mean in each feature dimension, and further rotated by an orthogonal rotation matrix $\mathbf{R}$ to narrow down the gap between zero-centered $\mathbf{V}$ and binary codes $\mathbf{B} \in \{-1, 1\}^{k\times n}$. The isotropic constraint on $\mathbf{V}$ is introduced to approximate the bits balance and bits independence constraints. As a result, we can obtain efficient binary codes from the zero-centered and rotated $\mathbf{V}$. Meanwhile, we learn a projection matrix $\mathbf{P} \in \mathbb{R}^{d\times k}$ from $\mathbf{X}$ to $\mathbf{V}$ for facilitating encoding new image features into high-level feature representations. Finally, the hash function learning can be formulated as the following optimization objective function:

$$L(\mathbf{U}, \mathbf{V}, \mathbf{P}, \mathbf{B}, \mathbf{R}) = L_1(\mathbf{U}, \mathbf{V}) + \alpha L_2(\mathbf{B}, \mathbf{R}, \mathbf{V}) + \beta L_3(\mathbf{P}, \mathbf{V}). \tag{1}$$

Where α, β are two parameters to balance the effects of corresponding terms. The objective function includes three terms: the first term is the high-level feature representations learning term, which means that the (visually or semantically) similar examples can have similar high-level feature representations; the second term is the efficient binary codes learning term, which means that similar high-level feature representations should be mapped to similar binary codes and the learned hash codes should satisfy the properties of bits balance and independence; the third term is an out-of-sample extension term, which implies that the high-level feature representations of new input data can be computed by a straightforward projection from the input space to the high-level embedding space. In the online query phrase, a new query image is characterized as $\mathbf{x}_t$. Next, we project $\mathbf{x}_t$ into the high-level embedding space as $\mathbf{v}_t = \mathbf{P}^T \mathbf{x}_t$, followed by a truncation operator as $\hat{\mathbf{v}}_t = \max(\mathbf{0}, \mathbf{v}_t)$. We further subtract the mean of $\mathbf{V}$ from $\hat{\mathbf{v}}_t$ and rotate the result with $\mathbf{R}$. Finally, we encode the real-value output into a binary code $\mathbf{b}_t$ by a sgn$(\cdot)$ function, where sgn$(x) = 1$ if $x > 0$ and -1 otherwise.

In this paper, we use a bold uppercase fonts to denote matrix and a bold lowercase fonts to denote a vector. In addition, given a matrix $\mathbf{X} = \{x_{ij}\}$, we utilize $\mathbf{x}_j$ to denote j-th column respectively. The inverse of $\mathbf{X}$ is represented as $\mathbf{X}^{-1}$, and its transpose is denoted as $\mathbf{X}^T$. The Frobenius norm of $\mathbf{X}$ is denoted as $\|\mathbf{X}\|_F^2$. $\mathbf{I}_k$ is the identity matrix of size k. $\mathbf{1}_n$ is the all-ones vector of size n.

A. Learning High-Level Feature Representations

We utilize the nonnegative factorization method [49], [50], [55] to learn high-level feature representations. Specifically, NMF can decompose the original real-value inputs X into two matrices U and V which have only nonnegative elements. The nature of NMF is to minimize the reconstruction error as follows:

$$\begin{aligned} &\min L_1(\mathbf{U}, \mathbf{V}) = \ ||\mathbf{X} - \mathbf{UV}||_F^2 \\ &\text{s.t.} \qquad \mathbf{U} \geq 0, \mathbf{V} \geq 0 \end{aligned} \tag{2}$$

where the two nonnegative factors $\mathbf{U} \in \mathbb{R}^{d\times k}$ and $\mathbf{V} \in \mathbb{R}^{k\times n}$ denote the basis matrix and the the coefficient matrix respectively. $\mathbf{H} = \mathbf{I}_n - \frac{1}{n}\mathbf{1}_n\mathbf{1}_n^T$ is the centering projection matrix which is also a symmetric and idempotent matrix, i.e., $\mathbf{H} = \mathbf{H}^T = \mathbf{HH}^T$. Each column of $\mathbf{U}$ corresponds to a basis and each column of $\mathbf{V}$ is a high-level feature representation for each training example [55]. The number of hash bits corresponds to the number of bases, i.e., each hash bit corresponds to a latent basis in $\mathbf{U}$.

B. Learning Efficient Binary Codes

The solution of NMF problem is in general not unique [56], i.e., the basis matrix and the coefficient matrix are not unique. Therefore, the resulting hashing codes learned from the high-level feature representations are not unique. There may exist redundant and correlated information in the bits of the obtained various hashing codes. However, from the information-theoretic point of view, the information provided by each bit of the resulting hashing codes is expected to be maximal and different bits are independent of each other, i.e., the resulting hashing codes are efficient binary codes. To obtain efficient binary codes, we require that the hash bits satisfy two constraints, i.e. bits balance $\mathbf{B1}_n = 0$ and bits independence $\frac{1}{n}\mathbf{BB}^T = \mathbf{I}_k$, the former requires that each bit has a 50% chance of being one or zero and the latter requires that different bits are independent of each other [17]. In addition, we expect that similar (visually or semantically) high-level feature representations should be mapped to similar binary codes [9]. We model the binary codes learning problem by minimizing the following binary quantization error:

$$\begin{aligned} &\min L_2(\mathbf{B}, \mathbf{R}, \mathbf{V}) = \ ||\mathbf{B} - \mathbf{RVH}||_F^2 \\ &\text{s.t.} \qquad \mathbf{R}^T\mathbf{R} = \mathbf{I}_k, \mathbf{B1}_n = 0, \frac{1}{n}\mathbf{BB}^T = \mathbf{I}_k \\ &\qquad\qquad \mathbf{B} \in \{-1, 1\}^{k\times n} \end{aligned} \tag{3}$$

where $\mathbf{R} \in \mathbb{R}^{k\times k}$ is an orthogonal matrix that brings the zero-centered high-level feature representations $\mathbf{VH}$ closely to the nearest vertex of the binary hypercube $\{-1, 1\}^k$. The reason of using the orthogonality constraint is that the orthogonal $\mathbf{R}$ can balance the variance of different dimensions of the zero-centered $\mathbf{V}$ which will benefit the reduction of the quantization error [9]. Here, $\mathbf{V}$ is zero-centered to eliminate the difference between the center of $\mathbf{V}$ and the center of $\mathbf{B}$. According to (3), each data point in $\mathbf{V}$ is mapped to the nearest vertex of the binary hypercube $\{-1, 1\}^k$. In other words, the smaller the quantization loss in (3), the better the locality structure of the data points in $\mathbf{V}$ will preserved in the corresponding Hamming space $\mathbf{B}$.

C. Out-of-Sample Extension

There is no straightforward extension scheme for computing high-level feature representations of inputs in the test set via NMF. Therefore, we have to learn an explicit mapping between the inputs and the high-level feature representations in the training phrase. Now, assume that we are given the training examples $\mathbf{X}$ and their corresponding high-level feature representations $\mathbf{V}$. Then the mapping from $\mathbf{X}$ to $\mathbf{V}$ can be considered as a regression problem. We leverage a simple linear regression to model the relationship between $\mathbf{X}$ and $\mathbf{V}$ as $f(\mathbf{x}_i) = \mathbf{P}^T\mathbf{x}_i$, where $\mathbf{P} \in \mathbb{R}^{d\times k}$ is the parameter of the linear regression model f. In order to optimize $\mathbf{P}$, we choose to minimize the following function:

$$\min L_3(\mathbf{V}, \mathbf{P}) = \sum_{i=1}^{n} \|\mathbf{v}_i - \mathbf{P}^T\mathbf{x}_i\|_F^2 + \gamma_1\|\mathbf{P}\|_F^2. \tag{4}$$

The function include two terms: the first term is a linear regression term to reduce the regression error, and the second term is a regularization term to avoid overfitting. The parameter γ_1 is used to balance the importance of the two terms. By a simple algebraic calculation, the matrix form of the out-of-sample extension term is given as

$$\min L_3(\mathbf{V}, \mathbf{P}) = \|\mathbf{V} - \mathbf{P}^T\mathbf{X}\|_F^2 + \gamma_1\|\mathbf{P}\|_F^2. \tag{5}$$

D. Overall Objective Function

The overall objective function consists of three terms including learning high-level feature representations term, learning efficient binary codes term and the out-of-sample extension term, i.e.,

$$\begin{aligned} \min_{\mathbf{U},\mathbf{V},\mathbf{P},\mathbf{B},\mathbf{R}} \ & ||\mathbf{X} - \mathbf{UV}||_F^2 + \alpha\|\mathbf{B} - \mathbf{RVH}\|_F^2 \\ & + \beta\|\mathbf{V} - \mathbf{P}^T\mathbf{X}\|_F^2 + \gamma\|\mathbf{P}\|_F^2 \\ \text{s.t.} \qquad & \mathbf{R}^T\mathbf{R} = \mathbf{I}_k, \mathbf{B1}_n = 0, \frac{1}{n}\mathbf{BB}^T = \mathbf{I}_k \\ & \mathbf{U} \geq 0, \mathbf{V} \geq 0, \mathbf{B} \in \{-1, 1\}^{k\times n} \end{aligned} \tag{6}$$

where $\alpha, \beta, \gamma = \beta\gamma_1$ are the tradeoff parameters which are utilized to balance the importance of the three terms.

E. Optimization

The optimization problem in (6) is non-smooth and non-convex. The reasons are that: 1) the optimization problem involves one discrete variable and two discrete constraints; 2) the overall objective function and the real-value orthogonal constraint are non-convex. In addition, the bits balance and bits independence constraints make the problem NP hard [17]. Intuitively, we can relax the bits balance and bits independence constraints and use an alternative optimization algorithm to obtain an approximate solution. One way is to ignore the bits balance and bits independence constraints in (6). The hash codes will be computed with $\mathbf{B} = \mathrm{sgn}(\mathbf{RVH})$. However, the resulting hash codes may be unbalanced or dependent of each other. The other way is to approximate $\mathbf{B}$ with the magnitude of $\mathbf{RVH}$. Then, the bits balance and the bits independence constraints will be $\mathbf{RVH1}_n = 0$ and $\frac{1}{n}\mathbf{RVHH}^T\mathbf{V}^T\mathbf{R}^T = \mathbf{I}_k$. We can see that: 1) the bits balance constraint can be satisfied naturally according to the definition of the centering projection matrix $\mathbf{H}$; 2) the bits independence constraint can be further simplified as $\frac{1}{n}\mathbf{VHV}^T = \mathbf{I}_k$, where the invertible property of orthogonal matrix $\mathbf{R}$: $\mathbf{R}^{-1} = \mathbf{R}^T$, and a property of idempotent matrix $\mathbf{H}$: $\mathbf{HH}^T = \mathbf{H}$ are used. Therefore, the bits balance constraint $\mathbf{B1}_n = 0$ and bits independence constraint $\frac{1}{n}\mathbf{BB}^T = \mathbf{I}_k$ in (6) can be approximated by one orthogonal constraint $\frac{1}{n}\mathbf{VHV}^T = \mathbf{I}_k$. The overall objective function can be rewritten as

$$\begin{aligned}\min_{\mathbf{U},\mathbf{V},\mathbf{P},\mathbf{B},\mathbf{R}} \ & \|\mathbf{X} - \mathbf{UV}\|_F^2 + \alpha\|\mathbf{B} - \mathbf{RVH}\|_F^2 + \beta\|\mathbf{V} - \mathbf{P}^T\mathbf{X}\|_F^2 \\ & + \gamma\|\mathbf{P}\|_F^2 \\ \text{s.t.} \quad & \mathbf{R}^T\mathbf{R} = \mathbf{I}_k, \frac{1}{n}\mathbf{VHV}^T = \mathbf{I}_k \\ & \mathbf{U} \geq 0, \mathbf{V} \geq 0, \mathbf{B} \in \{-1, 1\}^{k\times n}. \end{aligned} \tag{7}$$

It is worth noting that the covariance of matrix $\mathbf{V}$ is $\frac{1}{n}\mathbf{VHV}^T$ which is required to be isotropic [32], i.e., isotropic constraint. According to [32], it is unreasonable to use the same length of binary codes for different embedding dimensions with unequal variances since larger-variance dimensions will carry more information. Hence, the high-level feature representations with isotropic variances will be better than those with anisotropic variances for hashing. Moreover, the orthogonal constraint on the covariance of matrix $\mathbf{V}$ can enhance the discriminative property of $\mathbf{V}$.

The introduction of the isotropic (orthogonal) constraint $\frac{1}{n}\mathbf{VHV}^T = \mathbf{I}_k$ further increases the difficulty of the optimization problem. There are two typical approximate solutions for solving the objective function in (7) with the isotropic constraint $\frac{1}{n}\mathbf{VHV}^T = \mathbf{I}_k$ by introducing different orthogonality penalty terms. One uses the Lagrangian multiplier method as presented in [58] by introducing a trace-penalty term. The other employs an orthogonal subspace method as done in [57] by introducing a ℓ_2-norm penalty term. The former requires huge computation cost in updating the Lagrange multiplier which is a symmetric matrix with many parameters. The latter decreases the computational complexity by introducing a single parameter. Here, we introduce a ℓ_2-norm penalty term in the objective function to replace the orthogonal constraint as done in [57]. The optimization problem in (7) can be rewritten as

$$\begin{aligned}\min_{\mathbf{U},\mathbf{V},\mathbf{P},\mathbf{B},\mathbf{R}} \ & O(\mathbf{U}, \mathbf{V}, \mathbf{P}, \mathbf{B}, \mathbf{R}) \\ & = \|\mathbf{X} - \mathbf{UV}\|_F^2 + \alpha\|\mathbf{B} - \mathbf{RVH}\|_F^2 + \beta\|\mathbf{V} - \mathbf{P}^T\mathbf{X}\|_F^2 \\ & \quad + \gamma\|\mathbf{P}\|_F^2 + \frac{\eta}{2}\left\|\frac{1}{n}\mathbf{VHV}^T - \mathbf{I}_k\right\|_F^2 \\ \text{s.t.} \quad & \mathbf{U} \geq 0, \mathbf{V} \geq 0, \mathbf{R}^T\mathbf{R} = \mathbf{I}_k, \mathbf{B} \in \{-1, 1\}^{k\times n} \end{aligned} \tag{8}$$

where η is the penalty parameter which penalizes for the deviations from the identity matrix.

We utilize the Lagrangian multiplier method to transform the above nonnegative constrained optimization problem in (8) into an unconstrained one with respect to $\mathbf{U}$ and $\mathbf{V}$ as follows:

$$\begin{aligned}\mathcal{L}(\Theta) &= \|\mathbf{X} - \mathbf{UV}\|_F^2 + \alpha\|\mathbf{B} - \mathbf{RVH}\|_F^2 + \beta\|\mathbf{V} - \mathbf{P}^T\mathbf{X}\|_F^2 \\ & \quad + \gamma\|\mathbf{P}\|_F^2 + \frac{\eta}{2}\left\|\frac{1}{n}\mathbf{VHV}^T - \mathbf{I}_k\right\|_F^2 + \mathrm{Tr}(\mathbf{\Phi U}^T) \\ & \quad + \mathrm{Tr}(\mathbf{\Psi V}^T) \\ &= \|\mathbf{X}\|_F^2 - 2\mathrm{Tr}(\mathbf{X}^T\mathbf{UV}) + \mathrm{Tr}(\mathbf{UVV}^T\mathbf{U}^T) + \alpha\|\mathbf{B}\|_F^2 \\ & \quad - 2\alpha\mathrm{Tr}(\mathbf{B}^T\mathbf{RVH}) + \beta\mathrm{Tr}(\mathbf{V}^T\mathbf{V}) - 2\beta\mathrm{Tr}(\mathbf{V}^T\mathbf{P}^T\mathbf{X}) \\ & \quad + \beta\mathrm{Tr}(\mathbf{P}^T\mathbf{XX}^T\mathbf{P}) + \gamma\mathrm{Tr}(\mathbf{P}^T\mathbf{P}) \\ & \quad + \frac{\eta}{2n^2}\mathrm{Tr}(\mathbf{VHV}^T\mathbf{VHV}^T) + \left(\alpha - \frac{\eta}{n}\right)\mathrm{Tr}(\mathbf{VHV}^T) \\ & \quad + \frac{\eta}{2}\|\mathbf{I}_k\|_F^2 + \mathrm{Tr}(\mathbf{\Phi U}^T) + \mathrm{Tr}(\mathbf{\Psi V}^T) \end{aligned} \tag{9}$$

where $\mathbf{\Phi}$ and $\mathbf{\Psi}$ are the dual variables, Θ involves all variables $\mathbf{U}$, $\mathbf{V}$, $\mathbf{P}$, $\mathbf{B}$, $\mathbf{R}$, $\mathbf{\Phi}$ and $\mathbf{\Psi}$ in (9). The derivation of (8) uses the orthogonal property of matrix $\mathbf{R}$: $\mathbf{R}^T\mathbf{R} = \mathbf{RR}^T = \mathbf{I}_k$, some properties of idempotent matrix $\mathbf{H}$: $\mathbf{H} = \mathbf{H}^T = \mathbf{HH}^T$, and the matrix-trace properties: $\mathrm{Tr}(\mathbf{A}) = \mathrm{Tr}(\mathbf{A}^T)$, $\mathrm{Tr}(\mathbf{AC}) = \mathrm{Tr}(\mathbf{CA})$, and $\|\mathbf{A}\|_F^2 = \mathrm{Tr}(\mathbf{A}^T\mathbf{A})$.

In order to solve the problem in (9), an efficient coordinate descent algorithm is developed to alteratively optimize unknown variables Θ until convergence. The detail optimization procedure works as follows.

1) Fix $\mathbf{V}, \mathbf{P}, \mathbf{B}, \mathbf{R}$ and set the derivative w.r.t. $\mathbf{U}$ to zero. We have

$$\nabla_{\mathbf{U}}\mathcal{L} = 2\mathbf{UVV}^T - 2\mathbf{XV}^T + \mathbf{\Phi} = 0. \tag{10}$$

By using the Karush-Kuhn-Tucker (KKT) complementarity condition $\mathbf{\Phi} \odot \mathbf{U} = 0$, where $\odot$ means the Hadamard product (entrywise product), we can multiply $\mathbf{U}$ on the entries of (10) and obtain

$$(\mathbf{UVV}^T - \mathbf{XV}^T) \odot \mathbf{U} = 0. \tag{11}$$

We get the following updating rule:

$$\mathbf{U} \leftarrow \mathbf{U} \odot \frac{\mathbf{XV}^T}{\mathbf{U}(\mathbf{VV}^T)}. \tag{12}$$

2) Fix $\mathbf{U}, \mathbf{P}, \mathbf{B}, \mathbf{R}$ and set the derivative w.r.t. $\mathbf{V}$ to zero. We have

$$\begin{aligned}\nabla_{\mathbf{V}}\mathcal{L} &= 2\mathbf{U}^T\mathbf{U}\mathbf{V} - 2\mathbf{U}^T\mathbf{X} + 2\left(\alpha - \frac{\eta}{n}\right)\mathbf{V}\mathbf{H}\\ &\quad - 2\alpha\mathbf{R}^T\mathbf{B}\mathbf{H} + 2\beta\mathbf{V} - 2\beta\mathbf{P}^T\mathbf{X}\\ &\quad + \frac{2\eta}{n^2}\mathbf{V}\mathbf{H}\mathbf{V}^T\mathbf{V}\mathbf{H} + \mathbf{\Psi} = 0. \end{aligned} \tag{13}$$

By using the Karush-Kuhn-Tucker (KKT) complementarity condition $\mathbf{\Psi} \odot \mathbf{V} = 0$, we can multiply $\mathbf{V}$ on the entries of (12) and obtain

$$\begin{aligned}&\bigg(\mathbf{U}^T\mathbf{U}\mathbf{V} - \mathbf{U}^T\mathbf{X} + \left(\alpha - \frac{\eta}{n}\right)\mathbf{V}\mathbf{H} - \alpha\mathbf{R}^T\mathbf{B}\mathbf{H} + \beta\mathbf{V}\\ &\quad - \beta\mathbf{P}^T\mathbf{X} + \frac{\eta}{n^2}\mathbf{V}\mathbf{H}\mathbf{V}^T\mathbf{V}\mathbf{H}\bigg) \odot \mathbf{V} = 0\end{aligned} \tag{14}$$

where $\mathbf{R}^T\mathbf{B}\mathbf{H}$ and $\mathbf{P}^T\mathbf{X}$ are mixed sign, which not only includes negative sign but also positive sign. We can use the following operator to decompose the mixed sign matrix $\mathbf{M}$ into a positive part and a negative part, i.e. $\mathbf{M} = [\mathbf{M}]_+ - [\mathbf{M}]_-$, where $[\mathbf{M}]_+ = (\mathbf{M} + |\mathbf{M}|)/2, [\mathbf{M}]_- = (|\mathbf{M}| - \mathbf{M})/2$. Then, we can get the following updating rule as in (15), as shown at the bottom of the page, where $\mathbf{Q}_1 = (\mathbf{V}\mathbf{1}_n)(\mathbf{V}\mathbf{1}_n)^T\mathbf{V}, \mathbf{Q}_2 = (\mathbf{V}\mathbf{V}^T)(\mathbf{V}\mathbf{1}_n)\mathbf{1}_n^T$, and $\mathbf{Q}_3 = (\mathbf{V}\mathbf{1}_n)((\mathbf{V}\mathbf{1}_n)^T\mathbf{V}\mathbf{1}_n)\mathbf{1}_n^T$.

3) Fix $\mathbf{U}, \mathbf{V}, \mathbf{B}, \mathbf{R}$ and set the derivative w.r.t. $\mathbf{P}$ to zero. We have

$$\nabla_{\mathbf{P}}\mathcal{L} = 2(\beta\mathbf{X}\mathbf{X}^T + \gamma\mathbf{I}_d)\mathbf{P} - 2\beta\mathbf{X}\mathbf{V}^T = 0. \tag{16}$$

Then we get the following updating rule:

$$\mathbf{P} = \beta(\beta\mathbf{X}\mathbf{X}^T + \gamma\mathbf{I}_d)^{-1}\mathbf{X}\mathbf{V}^T. \tag{17}$$

4) Fix $\mathbf{U}, \mathbf{V}, \mathbf{P}, \mathbf{R}$ and update $\mathbf{B}$ by

$$\mathbf{B} = \mathrm{sgn}(\mathbf{R}\mathbf{V}\mathbf{H}). \tag{18}$$

5) Fix $\mathbf{U}, \mathbf{V}, \mathbf{P}, \mathbf{B}$ and update $\mathbf{R}$ by solving an orthogonal procrustes problem presented as in [9]. We get the following updating rule:

$$\mathbf{R} = (\mathbf{M}\mathbf{N})^T \tag{19}$$

where $\mathbf{M}, \mathbf{N}$ are the left and right singular values of the matrix $\mathbf{V}\mathbf{H}\mathbf{B}^T$ with its SVD as $\mathbf{M}\mathbf{\Sigma}\mathbf{N}$.

After the convergence of the algorithm, we can obtain the average of the low-dimensional parts-based representations of training examples $\bar{\mathbf{v}} = \frac{1}{n}\mathbf{V}\mathbf{1}_n$. For a test sample $\mathbf{x}_t$, we can obtain the latent embedding representation $\hat{\mathbf{v}}_t$ as follows:

$$\hat{\mathbf{v}}_t = \max\left(0, (\mathbf{P}^*)^T\mathbf{x}_t\right) \tag{20}$$

where $\mathbf{P}^*$ is the optimal solution of $\mathbf{P}$. Here, the negative entries of $(\mathbf{P}^*)^T\mathbf{x}_t$ are considered as embedding noise. The binary code of $\mathbf{x}_t$ can be obtained by

Algorithm 1: Learning Efficient Binary Codes From High-Level Feature Representations

```
Input: The training data $\mathbf{X} \in \mathbb{R}^{d\times n}$, the parameters
    $\alpha, \beta, \gamma, \eta, \mu$.
Output: The optimal $\mathbf{P}^*, \mathbf{R}^*$, and $\bar{\mathbf{v}}$.
Initialization:
    Initialize $\mathbf{U}, \mathbf{V}, \mathbf{P}, \mathbf{B}, \mathbf{R}$;
repeat
    Update $\mathbf{U}$ according to (12);
    Update $\mathbf{V}$ according to (15);
    Update $\mathbf{P}$ according to (17);
    Update $\mathbf{B}$ according to (18);
    Update $\mathbf{R}$ according to (19);
until convergence or reaching the maximum number of
    iterations.
return $\mathbf{P}^*$, $\mathbf{R}^*$, and $\bar{\mathbf{v}}$.
```

$$\mathbf{b}_t = \mathrm{sgn}(\mathbf{R}^*(\hat{\mathbf{v}}_t - \bar{\mathbf{v}})) \tag{21}$$

where $\mathbf{R}^*$ is the optimal solution of $\mathbf{R}$. Then the hash function can be defined as

$$\mathcal{H}(\mathbf{x}_i) = \mathrm{sgn}(\mathbf{R}^*(\max(0, (\mathbf{P}^*)^T\mathbf{x}_i) - \bar{\mathbf{v}})). \tag{22}$$

The whole optimization procedure is summarized in Algorithm 1.

The objective function in (8) is non-increasing under the updating rules in (12), (15), (17), (18) and (19). In addtion, the objective is obviously bounded. Therefore, the successive iteration of Algorithm 1 will converge (see Appendix A).

The solution of NMF is nonunique in general. Assume that we have two matrices $\mathbf{Y} \in \mathbb{R}^{k\times k}$ and $\mathbf{Z} \in \mathbb{R}^{k\times k}$ which meet the following conditions:

$$\mathbf{Y}\mathbf{Z} = \mathbf{I}_k, \mathbf{U}\mathbf{Y} \geq 0, \mathbf{Z}\mathbf{V} \geq 0. \tag{23}$$

We can find that $(\mathbf{U}\mathbf{Y}, \mathbf{Z}\mathbf{V})$ is also the solution with the same residue $\|\mathbf{X} - \mathbf{U}\mathbf{V}\|_F^2$. With the isotropic (orthogonal) constraint, the uniqueness of NMF can be given as in the following proposition.

Proposition 1: With the isotropic constraint $\frac{1}{n}\mathbf{V}\mathbf{H}\mathbf{V}^T = \mathbf{I}_k$ in the NMF, there exist no matrices $\mathbf{Y}$ and $\mathbf{Z}$ that meet both (23) and the isotropic constraint $\frac{1}{n}\mathbf{Z}\mathbf{V}\mathbf{H}\mathbf{V}^T\mathbf{Z}^T = \mathbf{I}_k$, except when $\mathbf{Y}$ and $\mathbf{Z}$ are permutation matrices, i.e., $\mathbf{Y} = \mathbf{Q}, \mathbf{Z} = \mathbf{Q}^T$ where $\mathbf{Q}^T\mathbf{Q} = I, q_{i,j} = 0$ or 1.

Proof: Please refer to the proof of Proposition 1 in [58].

F. Complexity Analysis

The updating of $\mathbf{U}$ requires $O((d+k)kn)$. The main computational cost in updating $\mathbf{V}$ lies in computing $\mathbf{R}^T\mathbf{B}\mathbf{H}$, which takes $O(kn^2)$ to compute it directly. To reduce the

$$\mathbf{V} \leftarrow \mathbf{V} \odot \frac{\mathbf{U}^T\mathbf{X} + \frac{\alpha}{n}(\mathbf{V}\mathbf{1}_n)\mathbf{1}_n^T + \alpha[\mathbf{R}^T\mathbf{B}\mathbf{H}]_+ + \beta[\mathbf{P}^T\mathbf{X}]_+ + \frac{\eta}{n^3}\mathbf{Q}_1 + \frac{\eta}{n}\mathbf{V} + \frac{\eta}{n^3}\mathbf{Q}_2}{(\mathbf{U}^T\mathbf{U})\mathbf{V} + (\alpha+\beta)\mathbf{V} + \alpha[\mathbf{R}^T\mathbf{B}\mathbf{H}]_- + \beta[\mathbf{P}^T\mathbf{X}]_- + \frac{\eta}{n^2}(\mathbf{V}\mathbf{1}_n)\mathbf{1}_n^T + \frac{\eta}{n^2}(\mathbf{V}\mathbf{V}^T)\mathbf{V} + \frac{\eta}{n^4}\mathbf{Q}_3} \tag{15}$$

computational complexity, we turn to computing $\mathbf{R}^T\mathbf{B}\mathbf{H}$ as $\mathbf{R}^T\mathbf{B} - \frac{1}{n}(\mathbf{R}^T(\mathbf{B}\mathbf{1}_n))\mathbf{1}_n^T$. The updating of $\mathbf{V}$ requires $O((d+k)kn)$. The updating of $\mathbf{P}$ requires $O(nd(d+k))$. The updating of $\mathbf{B}$ takes $O((k^2+k)n)$, which takes the similar calculating trick as in updating $\mathbf{V}$ to reduce the computational complexity. The computational cost of updating of $\mathbf{R}$ can be reduced to $O((k^2+2k)n)$ by replacing $\mathbf{V}\mathbf{H}\mathbf{B}^T$ with $\mathbf{V}\mathbf{B}^T - \frac{1}{n}(\mathbf{V}\mathbf{1}_n)(\mathbf{B}\mathbf{1}_n)^T$. We assume that the number of iterations needed for convergence is T. The overall computational complexity of the proposed method is $O(T(d+k)kn)$. For space complexity, the proposed approach requires $O(d\times r)$ for storing $\mathbf{U}$, $O(d\times n)$ for $\mathbf{X}$, $O(r\times n)$ for $\mathbf{V}$, $O(k\times n)$ for $\mathbf{B}$, and $O(k\times k)$ for $\mathbf{R}$. We can easily store all the variables in memory for large-scale image retrieval problem. We can see that the computational and storage cost is linear to the number of training samples n.

G. Relationship Between the Proposed Method and Principal Component Analysis (PCA)

The Principal Component Analysis [59], [60] technology projects data points $\mathbf{X}\in\mathbb{R}^{d\times n}$ to the k-dimensional subspace $\mathbf{U}\in\mathbb{R}^{d\times k}$, i.e., the principal directions, and obtains k-dimensional data representations $\mathbf{V}\in\mathbb{R}^{k\times n}$ in the subspace. The objective function of the PCA method can be defined as

$$\min_{\mathbf{U},\mathbf{V}} \|\mathbf{X}-\mathbf{U}\mathbf{V}\|_F^2. \qquad \text{s.t. } \mathbf{U}^T\mathbf{U} = \mathbf{I}_k. \tag{24}$$

The objective function in (24) can be transformed into the following objective function:

$$\max_{\mathbf{U}} \operatorname{Tr}(\mathbf{U}^T\mathbf{X}\mathbf{X}^T\mathbf{U}). \qquad \text{s.t. } \mathbf{U}^T\mathbf{U} = \mathbf{I}_k. \tag{25}$$

The above function is the well-known PCA optimization objective function, which can be solved by eigenvalue decomposition of the correlation matrix $\mathbf{X}\mathbf{X}^T$ and output the eigenvectors for the k largest eigenvalues.

The goals of PCA and NMF are both to find new data representations by minimizing the reconstruction error. PCA chooses to maximize variance among all directions of the projected data. However, unlike PCA, an isotropic covariance constraint on the high-level feature representations is induced by the efficient coding hypothesis in the paper. In addition, PCA requires orthogonality of basis $\mathbf{U}$, while the proposed method has no such constraint.

Compared with the proposed method, there are three main shortcomings in the PCA based hashing method [16]. First, since PCA assumes the subspace $\mathbf{U}$ to be orthogonal, the number of desired PCA embeddings or hash bits cannot be greater than the dimension d of the original input data, i.e., $k\leq d$. Second, the Hamming Accumulated Errors [31] will be accumulated with the real-value principal components converted into binary codes as the code length increasing. Third, the variances of different PCA projected dimensions are different and, as a result, it is unreasonable to encode the different projected dimensions with the same number of bits. On the contrary, the proposed method can generate longer codes whose code lengths can be larger than the dimension d of the input data, i.e., $k>d$. In addition, the Hamming Accumulated Errors can be reduced by the introduction of the binary quantization error term in the proposed method.

III. Experiments

In this section, we evaluate the proposed method for semantic retrieval task on three multi-label benchmark real-world datasets including MIRFlickr,[1] ADE20K[2] and NUS-WIDE,[3] and two single-label benchmark real-world datasets including CIFAR-10[4] and Tiny-1M [16], [22]. We compare our method with seven state-of-the-art unsupervised hashing methods for image retrieval task, including Iterative Quantization (ITQ) [9], Isotropic Hashing (IsoH) [32], Inductive Hashing on Manifolds (IMH) [11], Anchor Graph Hashing (AGH) [10], Spectral Hashing (SH) [17], Principal Component Analysis Hashing (PCAH) [16], and Locality Sensitive Hashing (LSH) [14].

1) Iterative Quantization (ITQ) [9] learns hash codes by minimizing the quantization error of mapping PCA-projected data to vertices of the binary hypercube.
2) Isotropic Hashing (IsoH) [32] learns projection functions with isotropic variances for different dimensions of PCA-projected data, where lift and projection (LP) learning algorithm is adopted to obtain the resulting binary codes.
3) Inductive Hashing on Manifolds (IMH) [11] employs the intrinsic manifolds structure to learn compact binary embeddings, where the t-SNE [61] is utilized to capture the intrinsic manifolds structure.
4) Anchor Graph Hashing (AGH) [10] extends the traditional spectral hashing by utilizing anchor graphs to approximately compute pairwise similarity matrix.
5) Spectral Hashing (SH) [17] learns hash codes through solving the eigenvectors of the graph Laplacian.
6) Principal Component Analysis Hashing (PCAH) [16] learns hash codes by maximizing the variance of each bit.
7) Locality Sensitive Hashing (LSH) [14] utilizes random linear projections to map real-value input to binary codes.

In addition, we also compare the proposed method with Euclidean distance based retrieval method (L2-scan), which adopts the original real-value features and Euclidean distance based similarity metric for image retrieval.

A. Experimental Settings

1) Dataset Settings: The MIRFlickr dataset consists of 25,000 images from the Flickr website with 24 concepts including stuffs like sky, flower, beach, and discrete objects like people, dog and bird. The dataset is randomly split into a collection of 24,000 training images and 1,000 test images. The ADE20K dataset consists of 20,210 images for training, and 2,000 images for validation. There are a total of 150 semantic categories in the dataset, which include stuffs like sea, sky, tree, lake, and discrete objects like people, dog, car. We mix the training set with the validation set. The data are randomly split into a

[1] [Online]. Available: http://press.liacs.nl/mirflickr/

[2] [Online]. Available: http://groups.csail.mit.edu/vision/datasets/ADE20K/

[3] [Online]. Available: http://lms.comp.nus.edu.sg/research/NUS-WIDE.htm

[4] [Online]. Available: https://www.cs.toronto.edu/ kriz/cifar.html

collection of 21,210 training images and 1,000 test images. The NUS-WIDE dataset contains 269,648 images and the associated 81 concepts collected from the Flickr website. We choose the 21 most frequent concepts including a total of 195,834 images as the groundtruth labels, where each of these concepts contains at least 5,000 images. We randomly sample 100 images from each concept as the test queries and the remaining images for training. The CIFAR-10 dataset consists of 60,000 natural images grouped into 10 classes with 6,000 images for each class. Following [9], we randomly sample 1000 images to serve as test queries and and the remaining images for training. The Tiny-1M dataset is a subset of the 80M tiny image benchmark[5] [62], which does not include any semantic label. We sample a subset of one million images to construct the training set and a separate subset of 2K images as query test [16], [22]. The returned top 5% nearest neighbors in the training set closest to each query in terms of the Euclidean distance are considered groundtruth neighbors. So each query has 50K groundtruth neighbors.

For MIRFlickr, ADE20K and CIFAR-10 datasets, we extract 320-dimensional GIST features [5] from each image as the visual feature for image representation. For the NUS-WIDE dataset, we use the 128-dimensional wavelet texture features which have been extracted for each image. The global 384-dimensional GIST descriptors of the images in the Tiny-1M dataset are publicly available. For all the compared methods, the features are rescaled to zero mean and unit variance. For the proposed method, we rescale the range of the features to $[0, 1]$ along each feature dimension. In the test phase, the retrieval images are regarded as the true neighbours if they share at least two semantic labels with the query for the three multi-label datasets (MIRFlickr, ADE20K, and NUS-WIDE) and the same label with the query for single-label dataset CIFAR-10. The image retrieval performance is evaluated using the Mean Average Precision (MAP) metric [10], [11]. In addition, we employ Average Cumulative Gain (ACG) [27], Normalized Discounted Cumulative Gain (NDCG) [27] metrics to evaluate the Hamming ranking quality. As the computation of MAP is slow on NUS-WIDE and Tiny-1M datasets, the top-50,000 returned neighbours are used to evaluate the MAP performance of all methods.

MAP is defined as the mean of the average precision (AP) of all queries, where average precision (AP) of top R retrieved instances is defined as

$$AP = \frac{1}{N}\sum_{r=1}^{R} P(r)\delta(r). \tag{26}$$

N denotes the number of relevant instances in retrieved set. $P(r)$ denotes the precision of top r retrieved instances. $\delta(i)$ is a indicator function, i.e., if i-th retrieved instance is a relevant true neighbor of the given query, $\delta(i) = 1$, and otherwise $\delta(i) = 0$.

For a single query q, ACG is defined as the average of the similarity levels of the returned data points within top-p positions:

$$ACG@p = \frac{1}{p}\sum_{i=1}^{p} r_i \tag{27}$$

where r_i is the similarity level of the returned data point at the i-th position in the ranking list of the query q.

NDCG is a popular ranking measure in the information retrieval community, which is defined as

$$NDCG@p = \frac{1}{Z}\sum_{i=1}^{p} \frac{2^{r_i} - 1}{\log(i+1)} \tag{28}$$

where Z is normalization factor to ensure the NDCG score for the correct ranking is one. The final ranking performance is given by averaging the ACG and NDCG scores of all queries.

The first criterion MAP reflects the change of the percentage of retrieved relevant instances with respect to the number of retrieved instances. The second criterion ACG shows the precision weighted by the similarity level of each retrieved instance. The third criterion NDCG evaluates the ranking quality of retrieved instances by giving bigger penalty for the incorrect higher ranked items.

2) Implementation Details: We initialize $\mathbf{U}$ and $\mathbf{V}$ as two random matrices whose elements are uniformly distributed in the interval (0,1). $\mathbf{P}$ can be initialized by (16). $\mathbf{R}$ is initialized as a random orthogonal matrix, and $\mathbf{B}$ is initialized by sgn($\mathbf{RVH}$). For the proposed method, we empirically set $\alpha = 1, \beta = 10^{-2}, \gamma = 10, \eta = 10^7$. The maximum number of iterations in the following experiments is set to 100. For AGH, and IMH, 1,000 anchor points are generated by K-means clustering. All experiments are conducted using Matlab 2015a on a server with Intel(R) Xeon(R) CPU E5-2690@2.90 GHz and 128 G of RAM, running the 64-bit Linux system. We run the experiments on three multi-label datasets and CIFAR-10 dataset 30 times, while 10 times experiments are conducted for Tiny-1M by considering the tradeoff between the efficiency and the computation cost. The mean performances over all repetitions are reported. For each run, we randomly split the datasets into training and test sets with the datasets setting reported above.

B. Experimental Results and Analyses

In this section, we compare our proposed method with state-of-the-art hashing methods for multi-label image semantic retrieval task.

1) Results on **MIRFlickr***:* The results on **MIRFlickr** are shown in Table I. The proposed method performs better than the compared hashing methods for all cases. The improvement of the performances can be attributed to the efficient hash codes learned from the high-level feature representations extraction via NMF. In addition, the proposed method can surpass L2-scan, e.g., from 32 bits to 256 bits for MAP and ACG@ 100 metrics and from 64 bits to 256 bits for NDCG@ 100 metric. The reason is that the proposed method can reduce redundant information in the original real-value inputs. On the other hand, it also verifies the effectiveness of the proposed method. With the increment of the codes length, the performance of the pro-

[5] [Online]. Available: http://groups.csail.mit.edu/vision/TinyImages/

TABLE I
RETRIEVAL PERFORMANCE OF DIFFERENT METHODS ON THREE MULTILABEL DATASETS FROM 16 BITS TO 256 BITS

Metric	Method	MIRFlickr				ADE20K				NUS-WIDE			
		32 bits	64bits	128 bits	256 bits	32 bits	64bits	128 bits	256 bits	32 bits	64bits	128 bits	256 bits
MAP	**L2-scan**	0.3210±0.0047				0.4844±0.0056				0.1880±0.0024			
	ITQ	0.3208±0.0050	0.3229±0.0051	0.3244±0.0051	0.3253±0.0050	0.4620±0.0053	0.4642±0.0048	0.4666±0.0048	0.4679±0.0050	0.1916±0.0034	0.1952±0.0021	0.1975±0.0024	-
	IsoH	0.3198±0.0046	0.3219±0.0045	0.3235±0.0044	0.3247±0.0043	0.4602±0.0042	0.4640±0.0042	0.4664±0.0042	0.4680±0.0043	0.1926±0.0041	0.1966±0.0040	0.1763±0.0027	-
	IMH	0.3142±0.0062	0.3154±0.0070	0.3140±0.0060	0.3150±0.0059	0.4536±0.0098	0.4587±0.0061	0.4631±0.0062	0.4639±0.0060	0.1676±0.0042	0.1672±0.0063	0.1698±0.0062	0.1678±0.0054
	AGH	0.3083±0.0060	0.3069±0.0060	0.3054±0.0049	0.3035±0.0048	0.4822±0.0067	0.4592±0.0059	0.4722±0.0055	0.4666±0.0055	0.1864±0.0026	0.1828±0.0025	0.1770±0.0023	0.1718±0.0021
	LSH	0.3157±0.0065	0.3205±0.0055	0.3243±0.0055	0.3255±0.0051	0.4526±0.0066	0.4592±0.0063	0.4638±0.0060	0.4667±0.0060	0.1939±0.0060	0.2004±0.0064	0.2037±0.0036	0.2048±0.0043
	PCAH	0.3058±0.0045	0.3043±0.0045	0.3028±0.0045	0.3008±0.0045	0.4438±0.0051	0.4402±0.0051	0.4363±0.0050	0.4236±0.0049	0.1770±0.0022	0.1727±0.0021	0.1724±0.0021	-
	SH	0.3085±0.0046	0.3082±0.0050	0.3090±0.0045	0.3070±0.0044	0.4495±0.0055	0.4549±0.0050	0.4543±0.0050	0.4489±0.0049	0.1720±0.0022	0.1715±0.0024	0.1776±0.0023	-
	Ours	**0.3260±0.0051**	**0.3286±0.0050**	**0.3322±0.0060**	**0.3362±0.0061**	**0.4837±0.0107**	**0.4896±0.0076**	**0.4945±0.0073**	**0.4973±0.0072**	**0.1940±0.0107**	**0.2017±0.0049**	**0.2067±0.0078**	**0.2088±0.0047**
ACG@100	**L2-scan**	1.1824±0.0248				2.5675±0.0367				1.0215±0.0125			
	ITQ	1.1754±0.0218	1.1889±0.0246	1.1992±0.0254	1.2047±0.0251	2.2155±0.0392	2.2369±0.0345	2.2607±0.0336	2.2760±0.0418	0.8014±0.0166	0.8860±0.0186	0.9475±0.0147	-
	IsoH	1.1630±0.0238	1.1764±0.0262	1.1914±0.0249	1.1994±0.0254	2.1648±0.0364	2.2269±0.0338	2.2750±0.0400	2.3145±0.0389	0.8541±0.0177	0.9210±0.0158	0.8339±0.0109	-
	IMH	1.0396±0.0272	1.0450±0.0251	1.0416±0.0270	1.0420±0.0273	1.8022±0.0507	1.9113±0.0448	1.9832±0.0368	2.0183±0.0322	0.3106±0.0227	0.3050±0.0308	0.3068±0.0235	0.2998±0.0228
	AGH	1.0852±0.0237	1.0886±0.0248	1.0861±0.0261	1.0726±0.0257	2.0983±0.0339	2.1357±0.0307	2.1430±0.0303	2.1493±0.0306	0.8459±0.0119	0.9020±0.0117	0.9241±0.0116	0.9363±0.0113
	LSH	1.1334±0.0314	1.1731±0.0318	1.2004±0.0249	1.2143±0.0241	2.0706±0.0503	2.1714±0.0405	2.2470±0.0431	2.2876±0.0325	0.8500±0.0315	0.9292±0.0187	0.9778±0.0127	1.0073±0.0174
	PCAH	1.0776±0.0188	1.0818±0.0204	1.0794±0.0190	1.0651±0.0185	2.0555±0.0274	2.0584±0.0283	2.0304±0.0271	1.9799±0.0260	0.7993±0.0093	0.7850±0.0074	0.8251±0.0097	-
	SH	1.0839±0.0197	1.1019±0.0233	1.1189±0.0225	1.0981±0.0221	2.0656±0.0262	2.1542±0.0256	2.2334±0.0292	2.2061±0.0292	0.7134±0.0097	0.7531±0.0094	0.8326±0.0109	-
	Ours	**1.1880±0.0286**	**1.2237±0.0251**	**1.2493±0.0287**	**1.2635±0.0291**	**2.3258±0.0504**	**2.4393±0.0489**	**2.5225±0.0448**	**2.5740±0.0438**	**0.8697±0.0247**	**0.9451±0.0149**	**0.9922±0.0171**	**1.0216±0.0140**
NDCG@100	**L2-scan**	0.1971±0.0051				0.1422±0.0038				0.1816±0.0026			
	ITQ	0.1908±0.0049	0.1916±0.0047	0.1929±0.0051	0.1933±0.0052	**0.1212±0.0041**	0.1255±0.0039	0.1288±0.0037	0.1307±0.0041	0.1348±0.0030	0.1495±0.0031	0.1606±0.0028	-
	IsoH	0.1881±0.0039	0.1893±0.0042	0.1909±0.0037	0.1920±0.0039	0.1175±0.0028	0.1235±0.0024	0.1280±0.0030	0.1315±0.0031	0.1426±0.0027	0.1562±0.0022	0.1454±0.0017	-
	IMH	0.1681±0.0044	0.1688±0.0053	0.1680±0.0055	0.1683±0.0044	0.0907±0.0035	0.0935±0.0033	0.0974±0.0027	0.0944±0.0028	0.0729±0.0028	0.0722±0.0024	0.0723±0.0035	0.0719±0.0032
	AGH	0.1782±0.0046	0.1790±0.0041	0.1791±0.0040	0.1772±0.0039	0.1040±0.0031	0.1071±0.0030	0.1082±0.0030	0.1085±0.0030	0.1400±0.0024	0.1527±0.0025	0.1579±0.0023	0.1611±0.0023
	LSH	0.1840±0.0040	0.1906±0.0061	0.1947±0.0055	0.1966±0.0057	0.1105±0.0038	0.1197±0.0040	0.1262±0.0039	0.1305±0.0043	0.1436±0.0045	0.1595±0.0041	0.1696±0.0027	0.1757±0.0030
	PCAH	0.1798±0.0038	0.1804±0.0038	0.1803±0.0042	0.1782±0.0038	0.1067±0.0031	0.1080±0.0029	0.1065±0.0028	0.1034±0.0027	0.1363±0.0021	0.1350±0.0021	0.1423±0.0023	-
	SH	0.1811±0.0044	0.1829±0.0043	0.1855±0.0047	0.1826±0.0041	0.1068±0.0034	0.1127±0.0034	0.1173±0.0035	0.1163±0.0032	0.1189±0.0023	0.1266±0.0022	0.1442±0.0023	-
	Ours	**0.1939±0.0052**	**0.2023±0.0055**	**0.2071±0.0047**	**0.2089±0.0052**	0.1202±0.0040	**0.1291±0.0034**	**0.1363±0.0042**	**0.1417±0.0040**	**0.1463±0.0043**	**0.1632±0.0033**	**0.1736±0.0029**	**0.1798±0.0032**

posed method climbs up due to the increasing representative ability of the resulting binary codes. It is worth noting that the representative ability of the learned high-level feature representations will also increase with the increment of the codes length. However, the performances of PCAH and SH degrade as the number of bits increases. The main reason is that the eigendecomposition solution in PCAH will make the bit's discriminative power lower with longer codes, and the assumption that data are sampled from a multidimensional uniform distribution in SH is too restrictive in practice. AGH ultizes a low-rank method to approximate the adjacency matrix and performs better than PCAH and SH. The performances of LSH, ITQ, IsoH, and IMH increase as the increasement of hash bits since the larger hash bits improve the representative ability of binary codes. However, they are inferior to the proposed method: LSH is a data independent method which does not use any information from the data, ITQ and IsoH resort to a heuristic two-stage approach including a dimension reduction via PCA stage and a rotation stage to obtain the binary codes which will make the resulting binary codes suboptimal, and IMH relies initial anchors selection which will affect the performance of the resulting binary codes.

2) Results on **ADE20K***:* The results on **ADE20K** are shown in Table I. The retrieval performances of all the methods on **ADE20K** are better than **MIRFlickr**. The main reason is that the number of images is close while the number of images with the same category or co-occurrence categories on **MIRFlickr** is larger than **ADE20K** as shown in Fig. 3. The proposed method performs better than the baseline hashing methods for most cases. However, for the NDCG@100 metric at 32 bits, the proposed method performs slightly inferior to ITQ. This result can be attributed to the slightly weak representative ability of the learned 32-dimensional high-level feature representations (NMF tends to produce sparse factor matrices due to nonnegativity constraints, and as a result, PCA is more optimal than NMF in terms of the reconstructed error). ITQ and IsoH perform better than IMH, PCAH, AGH, SH and LSH in this dataset, but is still inferior to the proposed method. The experimental performances in this dataset are similar to those on **MIRFlickr** dataset.

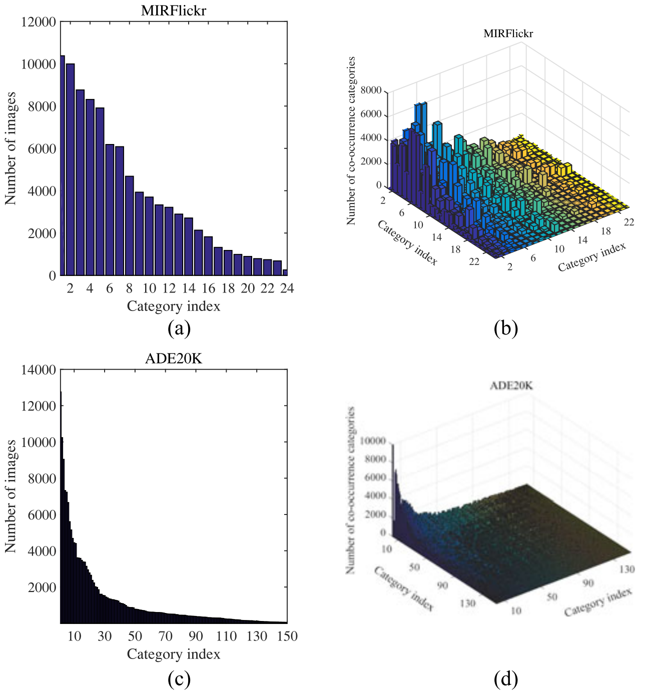

Fig. 3. Sematic categories histograms on (a) MIRFlickr and (c) ADE20K. The categories co-occurrence matrix on (b) MIRFlickr and (d) ADE20K.

3) Results on **NUS – WIDE***:* The results on **NUS – WIDE** are shown in Table I. Since the feature representation of **NUS-WIDE** is somewhat weak, the performances of all the methods on **NUS-WIDE** are inferior to

TABLE II
AVERAGE OFFLINE TRAINING AND THE ONLINE HASHING TIME OF ALL EVALUATED METHODS ON THE THREE MULTILABEL DATASETS

Datasets	Methods Times(s)	LSH	PCAH	SH	AGH	ITQ	IsoH	IMH	Proposed method
MIRFlickr	Training time	0.002	0.138	0.760	1.235	7.261	4.429	10.867	129.334
	Test time	0.004	0.004	0.137	0.031	0.006	0.006	0.018	0.007
ADE20K	Training time	0.004	0.140	0.823	0.668	7.077	4.451	13.585	120.409
	Test time	0.004	0.004	0.137	0.024	0.007	0.005	0.014	0.007
NUS-WIDE	Training time	0.001	0.069	0.849	5.774	20.532	1.451	14.485	402.644
	Test time	0.003	0.003	0.067	0.048	0.005	0.004	0.022	0.005

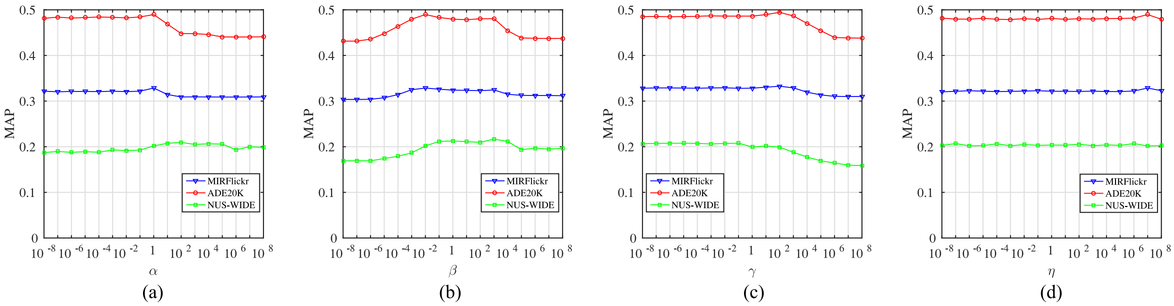

Fig. 4. MAP versus parameters α, β, γ, and η. (a) $\beta = 10^{-2}, \gamma = 10, \eta = 10^7$. (b) $\alpha = 1, \gamma = 10, \eta = 10^7$. (c) $\alpha = 1, \beta = 10^{-2}, \eta = 10^7$. (d) $\alpha = 1, \beta = 10^{-2}, \gamma = 10$.

MIRFlickr and **ADE20K** datasets. Similarity, the proposed method outperforms state-of-the-art hashing methods for all cases. The code length of ITQ, IsoH, PCAH, and SH on **NUS-WIDE** can't be greater than the dimension of the original input data, since they require PCA preprocessing of the input data at first. LSH performs better than ITQ and IsoH. This can be because the data distribution tends to the uniform distribution with large intra-class variation. Therefore, the manifold assumption will be weakened. The manifold based hashing methods such as IMH, AGH, PCAH and SH perform worse than LSH.

C. *Comparison of Training and Hashing Time*

In order to demonstrate the efficiency of the proposed method, we list the average offline training time and the online hashing (encoding) time of different methods on the three datasets in Table II, where all the comparison hash methods are executed 30 times with randomly generated training and test sets. We evaluate the average offline training time and the online time of all these methods with the code length of 256 bits on MIRFlickr and ADE20K and 128 bits on **NUS-WIDE**. Note that L2-scan is not included in Table II, due to the fact that L2-scan utilizes the original real-value features for image retrieval and doses not generate hash codes. Although, it needs more training time to make the proposed algorithm converge than the compared methods, the online hashing time of the proposed method is less than SH, AGH and IMH, which verifies the efficiency of the proposed method.

D. *Parameter Sensitivity*

There are four parameters in the proposed method, including α, β, γ and η in the objective function. We tune the parameters on MIRFlickr, ADE20K and NUS-WIDE by varying the value from 10^{-8} to 10^8. For the parameter sensitivity analysis, we fix the code length to 64 bits. In previous experiments, we utilize the fixed hyper-parameter configurations ($\alpha = 1, \beta = 10^{-2}, \gamma = 10, \eta = 10^7$). Here, we use the traditional approach of varying one parameter at a time while holding the others fixed to investigate the effects of different parameter settings on the algorithm performance. The experiments are conducted 30 times and the average experimental performances of MAP, ACG@ 100 and ADCG@ 100 are reported in Figs. 4–6 respectively. From Figs. 4–6, we can observe that the parameters α, β and γ are more sensitive compared to η. The reasons are that 1) α controls the quantization error loss in the offline training phrase which is essential for generating the resulting hash codes; 2) β and γ control the out-of-sample extension term which is a critical term for generating the hash codes of the test samples. More specifically, α controls the quantization error. It can be observed from Figs. 4(a), 5(a), and 6(a) that as α increases the experimental performances first increase and then decrease. The experimental results demonstrate that small α [e.g., $\alpha \in [10^{-8}, 10^{-2}]$ in Fig. 6(a)] will introduce large quantization error, while large α [e.g., $\alpha \in [10^5, 10^8]$ in Fig. 6(a)] will affect the balance of all components. β controls the correspondence between the original real-value features and the high-level feature representations, where small values of β (e.g., $\beta \in [10^3, 10^8]$ in Figs. 4–6) can weaken the correspondence relations between the original real-value features and the high-level feature representations, while large β (e.g., $\beta \in [10^{-8}, 10^{-2}]$ in Figs. 4–6) will affect the balance of the components in the model. In addition, γ controls the complexity of the proposed method, where large γ (e.g., $\beta \in [10^2, 10^8]$ in Figs. 4–6) will lead to under-fitting and small γ can lead to over-fitting. η can enforce the discrimination of the the high-level

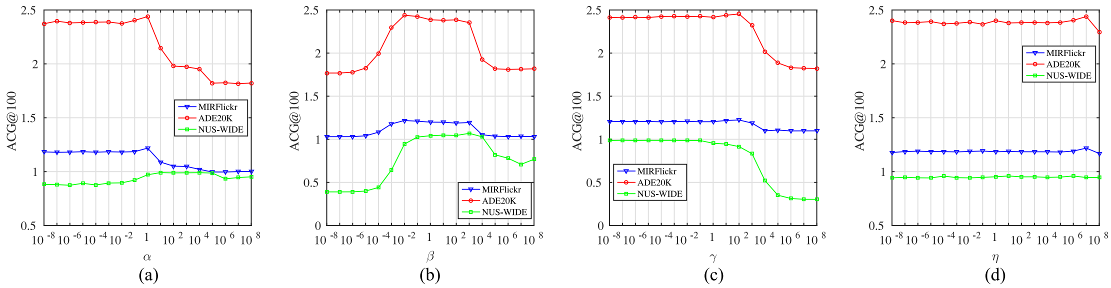

Fig. 5. ACG@100 versus parameters α, β, γ, and η. (a) $\beta = 10^{-2}, \gamma = 10, \eta = 10^7$. (b) $\alpha = 1, \gamma = 10, \eta = 10^7$. (c) $\alpha = 1, \beta = 10^{-2}, \eta = 10^7$. (d) $\alpha = 1, \beta = 10^{-2}, \gamma = 10$.

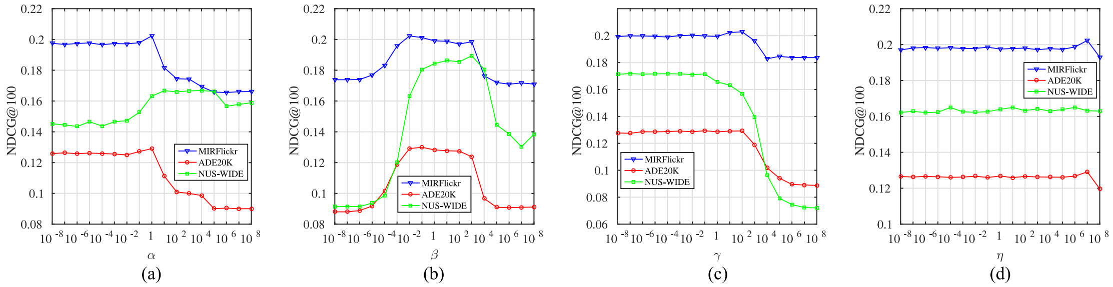

Fig. 6. NDCG@100 versus parameters α, β, γ, and η. (a) $\beta = 10^{-2}, \gamma = 10, \eta = 10^7$. (b) $\alpha = 1, \gamma = 10, \eta = 10^7$. (c) $\alpha = 1, \beta = 10^{-2}, \eta = 10^7$. (d) $\alpha = 1, \beta = 10^{-2}, \gamma = 10$.

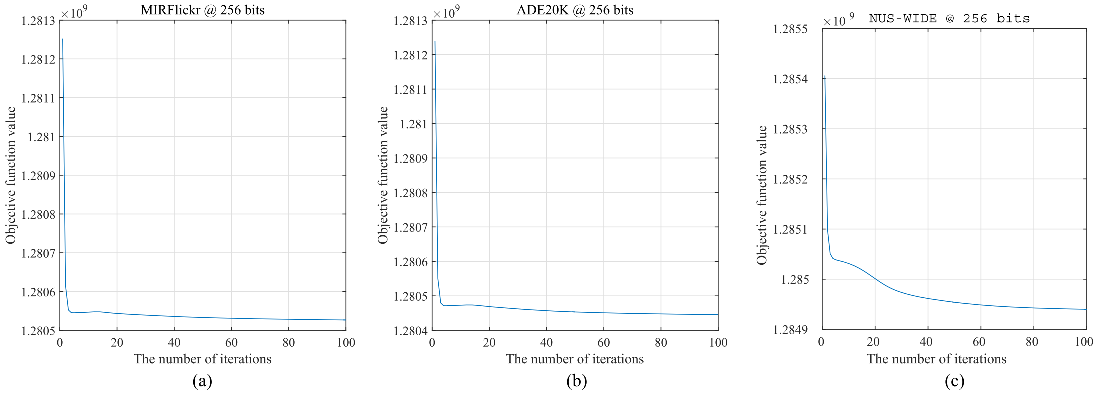

Fig. 7. Convergence curves on MIRFlickr, ADE20K, and NUS-WIDE datasets at 256 bits.

feature representations and generate efficient binary codes. η can be chosen from large values [e.g., $\gamma \in [10^6, 10^7]$ in Fig. 6(d)].

E. Convergence Study

The convergence of the iterative update rules in Algorithm 1 can be seen in Appendix A. Here, we also experimentally verify our analysis. Fig. 7 shows the objective value of the proposed method along the number of iterations. For each figure, the y-axis is the objective function value and x-axis is the number of iterations. It can be seen that the objective function decreases at each iteration and converges within 100 iterations. The objective function value is not decreasing smoothly after the third and the fifth iterations. The reason is that the objective function in (8) is non-smooth.

F. The Contributions of Different Terms

The overall objective function $O(\mathbf{U}, \mathbf{V}, \mathbf{P}, \mathbf{B}, \mathbf{R}) = O_1(\mathbf{U}, \mathbf{V}) + \alpha O_2(\mathbf{B}, \mathbf{R}, \mathbf{V}) + \beta O_3(\mathbf{V}, \mathbf{P}) + \gamma O_4(\mathbf{P}) + \eta O_5(\mathbf{V})$ includes five terms, where $\{O_i\}_{i=1}^5$ correspond successively to all the terms in (8). Since the high-level feature representations learning term $O_1(\mathbf{U}, \mathbf{V})$, out-of-sample extension term $O_3(\mathbf{V}, \mathbf{P}) + \gamma_1 O_4(\mathbf{P})$, and binary codes generation term $O_3(\mathbf{B}, \mathbf{R}, \mathbf{V})$ are indispensable part of the proposed model, we include $O_1(\mathbf{U}, \mathbf{V}), O_3(\mathbf{V}, \mathbf{P}), O_4(\mathbf{P})$ and $O_2(\mathbf{B}, \mathbf{R}, \mathbf{V})$ in the baselines defined as follows:

1) B_1: Learning binary codes from $O_1(\mathbf{U}, \mathbf{V}) + \alpha O_2(\mathbf{B}, \mathbf{R}, \mathbf{V}) + \beta O_3(\mathbf{V}, \mathbf{P}) + \gamma O_4(\mathbf{P})$, i.e.,

$$\min_{\Theta_1} O(\Theta_1) = ||\mathbf{X} - \mathbf{U}\mathbf{V}||_F^2 + \alpha\|\mathbf{B} - \mathbf{R}\mathbf{V}\mathbf{H}\|_F^2 + \beta\|\mathbf{V} - \mathbf{P}^T\mathbf{X}\|_F^2 + \gamma\|\mathbf{P}\|_F^2$$
$$\text{s.t.} \quad \mathbf{R}^T\mathbf{R} = \mathbf{I}_k, \mathbf{B} \in \{-1, 1\}^{k \times n}$$
$$\mathbf{U} \geq 0, \mathbf{V} \geq 0 \tag{29}$$

where $\Theta_1 = \{\mathbf{U}, \mathbf{V}, \mathbf{P}, \mathbf{B}, \mathbf{R}\}$.

2) T_1: Decomposing B_1 into two separate steps including $O_1(\mathbf{U}, \mathbf{V}) + \beta O_3(\mathbf{V}, \mathbf{P}) + \gamma O_4(\mathbf{P})$ and $O_2(\mathbf{B}, \mathbf{R}, \mathbf{V})$,

TABLE III
RETRIEVAL PERFORMANCE OF THE PROPOSED METHOD AND OTHER BASELINES ON THREE MULTILABEL DATASETS WITH THE MAP, ACG@ 100, AND NDCG@ 100 EVALUATION CRITERIONS AT 256 BITS

Dataset	Metric	B1	T1	T2	Proposed method
MIRFlickr	MAP	0.3320 ± 0.0054	0.3248 ± 0.0055	0.3284 ± 0.0069	**0.3362 ± 0.0061**
	ACG@100	1.2555 ± 0.0330	1.2095 ± 0.0286	1.2338 ± 0.0335	**1.2635 ± 0.0241**
	NDCG@100	0.2081 ± 0.0042	0.2011 ± 0.0040	0.2079 ± 0.0053	**0.2089 ± 0.0052**
ADE20K	MAP	0.4963 ± 0.0072	0.4875 ± 0.0064	0.4913 ± 0.0061	**0.4973 ± 0.0073**
	ACG@100	2.5232 ± 0.0451	2.5105 ± 0.0520	2.5325 ± 0.0410	**2.5740 ± 0.0438**
	NDCG@100	0.1372 ± 0.0046	0.1365 ± 0.0036	0.1392 ± 0.0040	**0.1417 ± 0.0040**
NUS-WIDE	MAP	0.2042 ± 0.0051	0.1929 ± 0.0053	0.1980 ± 0.0058	**0.2088 ± 0.0047**
	ACG@100	1.0071 ± 0.0159	0.9569 ± 0.0177	0.9856 ± 0.0152	**1.0216 ± 0.0140**
	NDCG@100	0.1783 ± 0.0029	0.1678 ± 0.0030	0.1741 ± 0.0026	**0.1798 ± 0.0032**

i.e.,

$$\min_{\Theta_2} O(\Theta_2) = ||\mathbf{X} - \mathbf{U}\mathbf{V}||_F^2 + \beta\|\mathbf{V} - \mathbf{P}^T\mathbf{X}\|_F^2 + \gamma\|\mathbf{P}\|_F^2 \quad \text{s.t. } \mathbf{U} \geq 0, \mathbf{V} \geq 0 \tag{30}$$

$$\min_{\Theta_3} O(\Theta_3) = \|\mathbf{B} - \mathbf{R}\mathbf{V}\mathbf{H}\|_F^2 \quad \text{s.t. } \mathbf{R}^T\mathbf{R} = \mathbf{I}_k, \mathbf{B} \in \{-1, 1\}^{k \times n} \tag{31}$$

where $\Theta_2 = \{\mathbf{U}, \mathbf{V}, \mathbf{P}\}$ and $\Theta_3 = \{\mathbf{B}, \mathbf{R}\}$.

3) T_2: Decomposing the whole learning procedure into two steps including $O_1(\mathbf{U}, \mathbf{V}) + \beta O_3(\mathbf{V}, \mathbf{P}) + \gamma O_4(\mathbf{P}) + \eta O_5(\mathbf{V})$ and $O_2(\mathbf{B}, \mathbf{R}, \mathbf{V})$, i.e.,

$$\min_{\Theta_4} O(\Theta_4) = ||\mathbf{X} - \mathbf{U}\mathbf{V}||_F^2 + \beta\|\mathbf{V} - \mathbf{P}^T\mathbf{X}\|_F^2 + \gamma\|\mathbf{P}\|_F^2 + \frac{\eta}{2}\left\|\frac{1}{n}\mathbf{V}\mathbf{H}\mathbf{V}^T - \mathbf{I}_k\right\|_F^2 \quad \text{s.t. } \mathbf{U} \geq 0, \mathbf{V} \geq 0 \tag{32}$$

$$\min_{\Theta_5} O(\Theta_5) = \|\mathbf{B} - \mathbf{R}\mathbf{V}\mathbf{H}\|_F^2 \quad \text{s.t. } \mathbf{R}^T\mathbf{R} = \mathbf{I}_k, \mathbf{B} \in \{-1, 1\}^{k \times n} \tag{33}$$

where $\Theta_4 = \{\mathbf{U}, \mathbf{V}, \mathbf{P}\}$ and $\Theta_5 = \{\mathbf{B}, \mathbf{R}\}$.

B_1 is the baseline without the isotropic constraint and T_1 is the corresponding two-step extension of B_1. The baseline B_1 can be solved by setting η in Algorithm 1 to zero. We can solve T_1 with two steps. The first step alternatively optimizes $\mathbf{U}, \mathbf{V}$, and $\mathbf{P}$ from the subproblem in (30) with the same updating rule as in (12), (15) and (17) by setting α and η to zero. The second step alternatively optimizes $\mathbf{B}$ and $\mathbf{R}$ from the subproblem in (31) with the same updating rule as in (18) and (19). A similar optimization procedure can be applied to the T_2. We compare the proposed method and other three baselines on the three multi-label datasets with the three evaluation criterions at 256 bits. The experimental results are shown in Table III. From this table we can see that the joint optimization methods including B_1 and the proposed method performs better than two-step methods T_1 and T_2. It demonstrates that the joint optimization methods can learn optimal binary codes. The performances of the proposed method and T_2 are consistently better than B_1 and T_1. It verifies that the isotropic variance constraint can improve the binary codes learning performance. In addition, the performance of the proposed method is better than the other baselines which demonstrates that incorporating them into a joint optimization framework can further improve the experimental performance for multi-label image retrieval.

G. Single Label Image Semantic Retrieval and Nearest-Neighbor Retrieval

Besides the experiments on three multi-label datasets, we also conduct comparison experiments on CIFAR-10 for single-label image retrieval and Tiny-1M for nearest neighbor retrieval. We employ the widely used MAP metric in single label image retrieval task and nearest neighbor retrieval to evaluate the performance of various methods. The experiments are conducted 30 times and 10 times for CIFAR-10 and Tiny-1M respectively. The average experimental performance of MAP is reported. The performances of different methods for single-label image semantic retrieval task are evaluated on CIFAR-10. The experimental results are shown in Table IV.

As we can see in Table IV, the proposed method achieves the best performance at 512 bits and surpasses the compared methods. The performances of different methods for nearest-neighbor retrieval task are evaluated on Tiny-1M. The experimental results are shown in Table V.

From Table V, we can observe that ITQ performs best at 32 bits and 64 bits, IsoH performs best at 128 bits and 256 bits, and LSH performs best at 320 bits and 512 bits. The proposed method achieves competitive performance with LSH from 320 bits to 512 bits. Let us make additional observations about the experiments on three multi-label datasets (MIRFlickr, ADE20K, and NUS-WIDE), CIFAR-10 and Tiny-1M. The proposed method performs better on the multi-label datasets with dispersed and complex classes in each image, while ITQ using PCA to extract low-dimensional feature representations performs well from 32 bits to 256 bits in the following two tasks: 1) the image semantic retrieval on single-label datasets with one compact class in each image, 2) nearest-neighbor retrieval. The reason can be that PCA is more optimal than NMF in terms of the reconstructed error (since NMF tends to produce sparse factor matrices due to nonnegativity constraints) which will benefit the single-label image retrieval and nearest-neighbor re-

TABLE IV
MAP PERFORMANCE OF DIFFERENT METHODS ON CIFAR10 FROM 32 BITS TO 512 BITS

	Ours	ITQ	IsoH	IMH	AGH	LSH	PCAH	SH	L2-scan
32 bits	0.1622 ± 0.0044	$\mathbf{0.1737 \pm 0.0019}$	0.1664 ± 0.0023	0.1538 ± 0.0068	0.1497 ± 0.0017	0.1441 ± 0.0046	0.1317 ± 0.0010	0.1342 ± 0.0010	0.1724 ± 0.0022
64 bits	0.1719 ± 0.0042	$\mathbf{0.1792 \pm 0.0020}$	0.1720 ± 0.0021	0.1575 ± 0.0052	0.1425 ± 0.0014	0.1564 ± 0.0031	0.1245 ± 0.0008	0.1329 ± 0.0009	
128 bits	0.1801 ± 0.0039	$\mathbf{0.1840 \pm 0.0023}$	0.1770 ± 0.0021	0.1554 ± 0.0063	0.1372 ± 0.0012	0.1666 ± 0.0036	0.1182 ± 0.0005	0.1304 ± 0.0008	
256 bits	0.1866 ± 0.0032	$\mathbf{0.1883 \pm 0.0023}$	0.1817 ± 0.0023	0.1556 ± 0.0061	0.1318 ± 0.0009	0.1766 ± 0.0025	0.1132 ± 0.0004	0.1273 ± 0.0008	
320 bits	0.1876 ± 0.0031	$\mathbf{0.1896 \pm 0.0024}$	0.1832 ± 0.0023	0.1542 ± 0.0065	0.1300 ± 0.0009	0.1793 ± 0.0023	0.1118 ± 0.0004	0.1258 ± 0.0008	
512 bits	$\mathbf{0.1899 \pm 0.0030}$	–	–	0.1538 ± 0.0068	0.1270 ± 0.0008	0.1827 ± 0.0025	–	–	

TABLE V
MAP PERFORMANCE OF DIFFERENT METHODS ON TINY1M FROM 32 BITS TO 512 BITS

	Ours	ITQ	IsoH	IMH	AGH	LSH	PCAH	SH
32 bits	0.1802 ± 0.0059	$\mathbf{0.3184 \pm 0.0044}$	0.3008 ± 0.0058	0.1049 ± 0.0019	0.2877 ± 0.0039	0.2029 ± 0.0085	0.1367 ± 0.0005	0.1606 ± 0.0017
64 bits	0.2223 ± 0.0040	$\mathbf{0.3477 \pm 0.0044}$	0.3383 ± 0.0044	0.1049 ± 0.0042	0.2951 ± 0.0026	0.2633 ± 0.0042	0.1199 ± 0.0005	0.1799 ± 0.0020
128 bits	0.3055 ± 0.0061	0.3635 ± 0.0051	$\mathbf{0.3689 \pm 0.0044}$	0.1054 ± 0.0026	0.2857 ± 0.0021	0.3244 ± 0.0101	0.1030 ± 0.0003	0.2124 ± 0.0013
256 bits	0.3676 ± 0.0008	0.3721 ± 0.0053	$\mathbf{0.3932 \pm 0.0035}$	0.1059 ± 0.0015	0.2616 ± 0.0014	0.3738 ± 0.0075	0.0872 ± 0.0001	0.2280 ± 0.0016
320 bits	0.3940 ± 0.0014	0.3779 ± 0.0052	0.1151 ± 0.0004	0.1052 ± 0.0012	0.2432 ± 0.0009	$\mathbf{0.3942 \pm 0.0098}$	0.0791 ± 0.0001	0.2242 ± 0.0016
512 bits	0.4069 ± 0.0037	–	–	0.1058 ± 0.0038	0.2290 ± 0.0008	$\mathbf{0.4073 \pm 0.0038}$	–	–

trieval, whereas NMF can provide a better characterization of the intra-latent information (data-driven latent attributes, e.g., clusterings) which will benefit uncovering the common latent structure shared by multi-label images.

H. Discussion

This paper has considered the problem of learning efficient hash codes from high-level feature representations. The basic idea behind the proposed method is that the extracted high-level feature representations should capture the intra-latent structure of data and the resulting hash codes learned from the high-level feature representations are expected to be efficient (bits balance and bits independence constraints are satisfied). In designing the proposed method, we explore a joint NMF optimization framework to learn the high-level feature representations and efficient hash codes.

Compared to existing methods, the proposed method has several advantages. The data independent hashing methods (e.g. LSH) require relatively long hash codes to produce good performance, which will lead to longer query time as well as a larger storage cost than would otherwise be necessary. Different from the data independent hashing methods, the proposed method can generate compact binary codes which will lead to faster query time with fewer memory cost. The manifold-based hashing methods (e.g. IMH, AGH, PCAH, SH, IsoH and ITQ) widely adopt non-linear or linear manifold assumption that semantic similar samples tend to lie on a low-dimensional manifold. For muti-label with dispersed and complex classes in each image, the manifold assumption will be weakened due to the large intra-class variation. Different from nonlinear manifold-based hashing methods, we do not need to make any assumption about the data distributions except the nonnegativity of the input data and data-driven latent attributes (e.g., inherent clustering property) of the data can be mined via NMF which make our method more suitable to perform complex (multi-label) image retrieval. In addition, It is worth noting that ITQ and IsoH utilize two separate steps to learn hash functions, which will lead to suboptimal binary codes. On the contrary, the proposed method adopts a joint optimization framework for learning hashing functions, which can generate optimal binary codes. Therefore, the proposed method outperforms previous work for multi-label image semantic retrieval task.

In consideration of computational and memory cost, the geometrical structure of the data space is not exploited in the proposed method. As a result, it would be helpful to explicitly model such information by constructing a nearest neighbor graph to further improve the performance. In addition, considering reducing model complexity, we utilize single-layer NMF (factorizing the inputs into two factors) in the paper. However, the deep multi-layer NMF can generate more informational high-level representations as in [55]. It is promising to incorporate the deep multi-layer NMF structure into the model to learn higher-level representations. In order to reduce human labor, we introduce a novel method for completely unsupervised hashing learning. However, the performance is not satisfactory for some practical applications. Therefore, label information can be utilized to further improve the image retrieval performance.

IV. CONCLUSION

In this paper, we propose an effective unsupervised hashing to learn efficient binary codes using NMF. The inspirations for this work lie in that NMF can produce high-level feature representations, and the higher-level features are expected to generate effective and efficient hash codes. However, directly applying NMF to generate hash codes will result in two problems: 1) the solution of NMF problem is in general not unique; 2) there is no straightforward extension scheme for computing high-level feature representations of inputs in the test set. Therefore, we introduce three terms into our model, i.e., a high-level feature representations learning term to learn high-level feature representations, an efficient binary code learning term to learn efficient binary codes while ensuring that similar (visually or semantically) high-level feature representations should be mapped to similar binary codes, and an out-of-sample extension term to facilitate the encoding of new samples. To solve the non-convex and non-smooth problem, an efficient optimization algorithm is developed which can decrease quantization loss and guarantee the convergence of the proposed algorithm in theory. Extensive experimental results on several benchmark real-world image datasets demonstrate the effectiveness of the proposed method.

Appendix A Proofs of Convergence

In order to prove the convergence of our proposed algorithm, we only need to prove that the objective value decreases in the alternate iteration procedure, i.e.,

$$\begin{aligned} &O(\mathbf{U}^{t-1}, \mathbf{V}^{t-1}, \mathbf{P}^{t-1}, \mathbf{B}^{t-1}, \mathbf{R}^{t-1}) \geq \\ &O(\mathbf{U}^{t}, \mathbf{V}^{t-1}, \mathbf{P}^{t-1}, \mathbf{B}^{t-1}, \mathbf{R}^{t-1}) \geq \\ &O(\mathbf{U}^{t}, \mathbf{V}^{t}, \mathbf{P}^{t-1}, \mathbf{B}^{t-1}, \mathbf{R}^{t-1}) \geq \cdots O(\mathbf{U}^{t}, \mathbf{V}^{t}, \mathbf{P}^{t}, \mathbf{B}^{t}, \mathbf{R}^{t}). \end{aligned} \tag{34}$$

The main difficulty in proving the convergence lies in the proofs of the first two inequalities. We choose to prove the second inequality, since the proofs of the first two inequalities are similar. The proofs of the other inequalities are omitted because they are well-known. We use an auxiliary function similar to that in [49], [57]. The definition of auxiliary function is given as follows:

Definition 1: [49] $G(h, \hat{h})$ is an auxiliary function for $F(h)$ if the conditions

$$G(h, \hat{h}) \geq F(h), G(h, h) = F(h) \tag{35}$$

are satisfied.

Lemma 1 ([49]): If G is an auxiliary function, then F is nonincreasing under the update

$$h^{(t+1)} = \arg\max_{h} G(h, h^{(t)}). \tag{36}$$

Proof: $F(h^{(t+1)}) \leq G(h^{(t+1)}, h^{(t)}) \leq G(h^{(t)}, h^{(t)}) = F(h^{(t)})$. ■

Considering any element v_{ij} in $\mathbf{V}$, F_{ij} is used to denote the part of O, which is only relevant to v_{ij}. By defining the appropriate auxiliary function $G(v, v_{ij}^{t})$ for F_{ij}, the updating rule in (15) easily follows (36). The first-order and second-order derivatives of F_{ij} with respect to v_{ij} are shown as follows:

$$\begin{aligned} F'_{ij} = \left(\frac{\partial O}{\partial \mathbf{V}}\right)_{ij} = \bigg(&-2\mathbf{U}^T\mathbf{X} + 2\mathbf{U}^T\mathbf{U}\mathbf{V} + 2\alpha\mathbf{V}\mathbf{H} \\ &- 2\alpha\mathbf{R}^T\mathbf{B}\mathbf{H} + 2\beta\mathbf{V} - 2\beta\mathbf{P}^T\mathbf{X} \\ &+ \frac{2\eta}{n^2}\mathbf{V}\mathbf{H}\mathbf{V}^T\mathbf{V}\mathbf{H} - \frac{2\eta}{n}\mathbf{V}\mathbf{H}\bigg)_{ij} \end{aligned} \tag{37}$$

$$\begin{aligned} F''_{ij} = {} & 2(\mathbf{U}^T\mathbf{U})_{ii} + 2\alpha h_{jj} + 2\beta \\ & + \frac{2\eta}{n^2}\Big((\mathbf{V}\mathbf{H}\mathbf{V}^T)_{ii} h_{jj} + (\mathbf{V}\mathbf{H})_{ii}^2 - n h_{jj}\Big). \end{aligned} \tag{38}$$

Lemma 2: Function

$$\begin{aligned} G\big(v, v_{ij}^{(t)}\big) = {} & F_{ij}\big(v_{ij}^{(t)}\big) + F'_{ij}\big(v_{ij}^{(t)}\big)\big(v - v_{ij}^{t}\big) + \big(v - v_{ij}^{t}\big)^2 \cdot \\ & ((\mathbf{U}^T\mathbf{U})\mathbf{V} + (\alpha + \beta)\mathbf{V} + \alpha[\mathbf{R}^T\mathbf{B}\mathbf{H}]_- \\ & + \beta[\mathbf{P}^T\mathbf{X}]_- + \frac{\eta}{n^2}\mathbf{V}\mathbf{V}^T\mathbf{V} + \frac{\eta}{n^2}(\mathbf{V}\mathbf{1}_n)\mathbf{1}_n^T \\ & + \frac{\eta}{n^4}(\mathbf{V}\mathbf{1}_n)\big((\mathbf{V}\mathbf{1}_n)^T\mathbf{V}\mathbf{1}_n\big)\mathbf{1}_n^T)_{ij} / v_{ij}^{t}. \end{aligned} \tag{39}$$

is an auxiliary function for $F_{ij}(v)$.

Proof: Since $G(v, v) = F_{ij}(v)$ is obvious, we need only verify that $G(v, v_{ij}^{(t)}) \geq F_{ij}(v)$. To verify this, we compare the Taylor series expansion of $F_{ij}(v)$

$$\begin{aligned} F_{ij}(v) = {} & F_{ij}\big(v_{ij}^{(t)}\big) + F'_{ij}\big(v_{ij}^{(t)}\big)\big(v - v_{ij}^{(t)}\big) + \big(v - v_{ij}^{(t)}\big)^2 \cdot \\ & \bigg((\mathbf{U}^T\mathbf{U})_{ii} + \alpha h_{jj} + \beta \\ & + \frac{\eta}{n^2}((\mathbf{V}\mathbf{H}\mathbf{V}^T)_{ii} h_{jj} + (\mathbf{V}\mathbf{H})_{ii}^2 - n h_{jj})\bigg) \end{aligned} \tag{40}$$

with (39). We can have the following inequalities:

$$(\mathbf{U}^T\mathbf{U}\mathbf{V})_{ij} = \sum_{l=1}^{k} (\mathbf{U}^T\mathbf{U})_{il} v_{lj}^{(t)} \geq (\mathbf{U}^T\mathbf{U})_{ii} v_{ij}^{(t)} \tag{41}$$

$$\alpha v_{ij} \geq \alpha h_{jj} v_{ij}^{(t)} \tag{42}$$

$$\alpha([\mathbf{R}^T\mathbf{B}\mathbf{H}]_-)_{ij} \geq 0, \beta([\mathbf{P}^T\mathbf{X}]_-)_{ij} \geq 0 \tag{43}$$

$$\begin{aligned} & (\mathbf{V}\mathbf{V}^T\mathbf{V})_{ij} \\ & + \frac{1}{n^2}\Big((\mathbf{V}\mathbf{1}_n)\big((\mathbf{V}\mathbf{1}_n)^T\mathbf{V}\mathbf{1}_n\big)\mathbf{1}_n^T\Big)_{ij} + \big((\mathbf{V}\mathbf{1}_n)\mathbf{1}_n^T\big)_{ij} \\ & = \sum_{l=1}^{n} (\mathbf{V}\mathbf{V}^T)_{il} v_{lj} + \frac{1}{n^2}\sum_{l=1}^{n} v_{il}\big(\mathbf{1}_n\mathbf{1}_n^T\mathbf{V}^T\mathbf{V}\mathbf{1}_n\mathbf{1}_n^T\big)_{lj} + \sum_{l=1}^{n} v_{il} \\ & \geq \bigg((\mathbf{V}\mathbf{V}^T)_{ii} + \frac{1}{n^2}\big(\mathbf{1}_n\mathbf{1}_n^T\mathbf{V}^T\mathbf{V}\mathbf{1}_n\mathbf{1}_n^T\big)_{jj} + 1\bigg) v_{ij} \\ & = (\mathbf{V}\mathbf{V}^T)_{ii} v_{ij}^{(t)} + \frac{1}{n^2} v_{ij}^{(t)} \mathbf{1}_n^T\mathbf{V}^T\mathbf{V}\mathbf{1}_n + v_{ij}^{(t)} \\ & \geq \big((\mathbf{V}\mathbf{H}\mathbf{V}^T)_{ii} h_{jj} + (\mathbf{V}\mathbf{H})_{ii}^2 - n h_{jj}\big) v_{ij}^{(t)} \end{aligned} \tag{44}$$

where the second inequality in (44) holds because of the condition $\frac{1}{n}(\mathbf{V}\mathbf{H}\mathbf{V}^T)_{ii} = 1$, and

$$\begin{aligned} (\mathbf{V}\mathbf{H})_{ii}^2 &= \left(\mathbf{V} - \frac{1}{n}\mathbf{V}\mathbf{1}_n\mathbf{1}_n^T\right)_{ii}^2 = v_{ii}^2 - \frac{2}{n}\sum_{l=1}^{n} v_{il} + \frac{1}{n^2}\left(\sum_{l=1}^{n} v_{il}\right)^2 \\ &\leq v_{ii}^2 + \frac{1}{n^2}\left(\sum_{l=1}^{n} v_{il}\right)^2 \leq (\mathbf{V}\mathbf{V}^T)_{ii} + \frac{1}{n^2}\mathbf{1}_n^T\mathbf{V}^T\mathbf{V}\mathbf{1}_n. \end{aligned} \tag{45}$$

Thus the inequality $G(v, v_{ij}^{(t)}) \geq F_{ij}(v)$ holds. ■

Replacing $G(v, v_{ij}^{(t)})$ in (36) by (39) and setting the derivation of $G(v, v_{ij}^{(t)})$ with respect to v to zero result in the following update rule:

$$\begin{aligned} &v_{ij}^{(t+1)} \\ &= v_{ij}^{(t)} - v_{ij}^{(t)} F'_{ij}\big(v_{ij}^{(t)}\big) \Big/ \bigg((\mathbf{U}^T\mathbf{U})\mathbf{V} + (\alpha+\beta)\mathbf{V} \\ &\qquad + \alpha[\mathbf{R}^T\mathbf{B}\mathbf{H}]_- + \beta[\mathbf{P}^T\mathbf{X}]_- \\ &\qquad + \frac{\eta}{n^2}(\mathbf{V}\mathbf{V}^T)\mathbf{V} + \frac{\eta}{n^2}(\mathbf{V}\mathbf{1}_n)\mathbf{1}_n^T \\ &\qquad + \frac{\eta}{n^4}(\mathbf{V}\mathbf{1}_n)((\mathbf{V}\mathbf{1}_n)^T\mathbf{V}\mathbf{1}_n)\mathbf{1}_n^T \bigg)_{ij}. \qquad (46) \end{aligned}$$

By simple algebra manipulations, we can have the same updating rule as in (15). Since $G(v, v_{ij}^{(t)})$ is an auxiliary function, $F_{ij}(v)$ is nonincreasing under the updating rule in (15). The other inequalities in (34) hold under the corresponding updating rules. The convergence of the proposed method is proved.

REFERENCES

[1] T. Song and H. Li, "Wavelbp based hierarchical features for image classification," *Pattern Recog. Lett.*, vol. 34, no. 12, pp. 1323–1328, 2013.

[2] T. Song, H. Li, B. Zeng, and M. Gabbouj, "Texture classification using joint statistical representation in space-frequency domain with local quantized patterns," in *Proc. IEEE Int. Symp. Circuits Syst.*, Jun. 2014, pp. 886–889.

[3] T. Song and H. Li, "Local polar DCT features for image description," *IEEE Signal Process. Lett.*, vol. 20, no. 1, pp. 59–62, Jan. 2013.

[4] T. Song *et al.*, "Noise-robust texture description using local contrast patterns via global measures," *IEEE Signal Process. Lett.*, vol. 21, no. 1, pp. 93–96, Jan. 2014.

[5] A. Oliva and A. Torralba, "Modeling the shape of the scene: A holistic representation of the spatial envelope," *Int. J. Comput. Vis.*, vol. 42, no. 3, pp. 145–175, 2001.

[6] F. Li and P. Perona, "A Bayesian hierarchical model for learning natural scene categories," in *Proc. IEEE Comput. Soc. Conf. Comput. Vis. Pattern Recog.*, Jun. 2005, vol. 2, pp. 524–531.

[7] F. Meng, H. Li, G. Liu, and K. N. Ngan, "Object co-segmentation based on shortest path algorithm and saliency model," *IEEE Trans. Multimedia*, vol. 14, no. 5, pp. 1429–1441, Oct. 2012.

[8] H. Li, F. Meng, and K. N. Ngan, "Co-salient object detection from multiple images," *IEEE Trans. Multimedia*, vol. 15, no. 8, pp. 1896–1909, Dec. 2013.

[9] Y. Gong, S. Lazebnik, A. Gordo, and F. Perronnin, "Iterative quantization: A procrustean approach to learning binary codes for large-scale image retrieval," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 35, no. 12, pp. 2916–2929, Dec. 2013.

[10] W. Liu, J. Wang, S. Kumar, and S. Chang, "Hashing with graphs," in *Proc. Int. Conf. Mach. Learn.*, 2011, pp. 1–8.

[11] F. Shen, C. Shen, Q. Shi, A. van den Hengel, and Z. Tang, "Inductive hashing on manifolds," in *Proc. IEEE Conf. Comput. Pattern Recog.*, Jun. 2013, pp. 1562–1569.

[12] J. H. Friedman, J. L. Bentley, and R. A. Finkel, "An algorithm for finding best matches in logarithmic expected time," *ACM Trans. Math. Softw.*, vol. 3, no. 3, pp. 209–226, 1977.

[13] A. Guttman, "R-trees: A dynamic index structure for spatial searching," in *Proc. Annu. Meet.*, 1984, pp. 47–57.

[14] A. Gionis, P. Indyk, and R. Motwani, "Similarity search in high dimensions via hashing," in *Proc. Int. Conf. Very Large Data Bases*, 1999, pp. 518–529.

[15] M. Datar, N. Immorlica, P. Indyk, and V. S. Mirrokni, "Locality-sensitive hashing scheme based on p-stable distributions," in *Proc. ACM Symp. Comput. Geometry*, 2004, pp. 253–262.

[16] J. Wang, S. Kumar, and S. F. Chang, "Semi-supervised hashing for large-scale search," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 34, no. 12, pp. 2393–2406, Dec. 2012.

[17] Y. Weiss, A. Torralba, and R. Fergus, "Spectral hashing," in *Proc. Adv. Neural Inf. Process. Syst.*, 2008, pp. 1753–1760.

[18] Y. Sun, M. K. Ng, and Z. Zhou, "Multi-instance dimensionality reduction," in *Proc. 24th AAAI Conf. Artif. Intell.*, 2010, pp. 587–592.

[19] B. Kulis and T. Darrell, "Learning to hash with binary reconstructive embeddings," in *Proc. Adv. Neural Inf. Process. Syst.*, 2009, pp. 1042–1050.

[20] C. Strecha, A. M. Bronstein, M. M. Bronstein, and P. Fua, "Ldahash: Improved matching with smaller descriptors," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 34, no. 1, pp. 66–78, Jan. 2012.

[21] M. Norouzi and D. J. Fleet, "Minimal loss hashing for compact binary codes," in *Proc. Int. Conf. Mach. Learn.*, 2011, pp. 353–360.

[22] W. Liu, J. Wang, R. Ji, Y. Jiang, and S. Chang, "Supervised hashing with kernels," in *Proc. IEEE Conf. Comput. Vis. Pattern Recog.*, Jun. 2012, pp. 2074–2081.

[23] T. Ge, K. He, and J. Sun, "Graph cuts for supervised binary coding," in *Proc. Eur. Conf. Comput. Vis.*, 2014, pp. 250–264.

[24] Y. Yang, W. Chen, Y. Luo, F. Shen, J. Shao, and H. T. Shen, "Zero-shot hashing via transferring supervised knowledge," in *Proc. ACM Multimedia Conf.*, 2016, pp. 1286–1295.

[25] F. Shen, C. Shen, W. Liu, and H. T. Shen, "Supervised discrete hashing," in *Proc. IEEE Conf. Comput. Vis. Pattern Recog.*, Jun. 2015, pp. 37–45.

[26] R. Xia, Y. Pan, H. Lai, C. Liu, and S. Yan, "Supervised hashing for image retrieval via image representation learning," in *Proc. Conf. Artif. Intell.*, 2014, pp. 2156–2162.

[27] F. Zhao, Y. Huang, L. Wang, and T. Tan, "Deep semantic ranking based hashing for multi-label image retrieval," in *Proc. IEEE Conf. Comput. Vis. Pattern Recog.*, Jun. 2015, pp. 1556–1564.

[28] K. Lin, H. Yang, J. Hsiao, and C. Chen, "Deep learning of binary hash codes for fast image retrieval," in *Proc. Conf. Comput. Vis. Pattern Recog. Workshops*, Jun. 2015, pp. 27–35.

[29] H. Lai *et al.*, "Instance-aware hashing for multi-label image retrieval," *IEEE Trans. Image Process.*, vol. 25, no. 6, pp. 2469–2479, Jan. 2016.

[30] Z. Chen, J. Lu, J. Feng, and J. Zhou, "Nonlinear discrete hashing," *IEEE Trans. Multimedia*, vol. 19, no. 1, pp. 123–135, Jan. 2017.

[31] C. Wu, J. Zhu, D. Cai, C. Chen, and J. Bu, "Semi-supervised nonlinear hashing using bootstrap sequential projection learning," *IEEE Trans. Knowl. Data Eng.*, vol. 25, no. 6, pp. 1380–1393, Jun. 2013.

[32] W. Kong and W. Li, "Isotropic hashing," in *Proc. Conf. Neural Inf. Process. Syst.*, 2012, pp. 1655–1663.

[33] G. Irie, Z. Li, X. Wu, and S. Chang, "Locally linear hashing for extracting non-linear manifolds," in *Proc IEEE Conf. Comput. Vis. Pattern Recog.*, Jun. 2014, pp. 2123–2130.

[34] Y. Weiss, "Segmentation using eigenvectors: A unifying view," in *Proc. 7th IEEE Int. Conf. Comput. Vis.*, Sep. 1999, pp. 975–982.

[35] M. Belkin and P. Niyogi, "Laplacian eigenmaps for dimensionality reduction and data representation," *Neural Comput.*, vol. 15, no. 6, pp. 1373–1396, 2003.

[36] J. B. Tenenbaum, V. de Silva, and J. C. Langford, "A global geometric framework for nonlinear dimensionality reduction," *Science*, vol. 290, no. 5500, pp. 2319–2323, 2000.

[37] S. T. Roweis and L. K. Saul, "Nonlinear dimensionality reduction by locally linear embedding," *Science*, vol. 290, pp. 2323–2326, 2000.

[38] M. Á. Carreira-Perpiñán, "The elastic embedding algorithm for dimensionality reduction," in *Proc. Int. Conf. Mach. Learn.*, 2010, pp. 167–174.

[39] L. van der Maaten and G. Hinton, "Visualizing high-dimensional data using t-sne," *J. Mach. Learn. Res.*, vol. 25, no. 9, pp. 2579–2605, 2008.

[40] G. Hinton and R. Salakhutdinov, "Reducing the dimensionality of data with neural networks," *Science*, vol. 313, no. 5786, pp. 504–507, 2006.

[41] V. E. Liong, J. Lu, G. Wang, P. Moulin, and J. Zhou, "Deep hashing for compact binary codes learning," in *Proc. IEEE Conf. Comput. Vis. Pattern Recog.*, Jun. 2015, pp. 2475–2483.

[42] H. Jégou, M. Douze, and C. Schmid, "Product quantization for nearest neighbor search," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 33, no. 1, pp. 117–128, Jan. 2011.

[43] M. Norouzi and D. J. Fleet, "Cartesian k-means," in *Proc. IEEE Conf. Comput. Vis. Pattern Recog.*, Jun. 2013, pp. 3017–3024.

[44] T. Zhang, C. Du, and J. Wang, "Composite quantization for approximate nearest neighbor search," in *Proc. Int. Conf. Mach. Learn.*, 2014, pp. 838–846.

[45] L. Mukherjee, S. N. Ravi, V. K. Ithapu, T. Holmes, and V. Singh, "An NMF perspective on binary hashing," in *Proc. IEEE Int. Conf. Comput. Vis.*, Dec. 2015, pp. 4184–4192.

[46] Z. Cai, L. Liu, M. Yu, and L. Shao, "Latent structure preserving hashing," in *Proc. Brit. Mach. Vis. Conf.*, 2015, pp. 172-1–172-11.

[47] D. Wang, P. Cui, M. Ou, and W. Zhu, "Learning compact hash codes for multimodal representations using orthogonal deep structure," *IEEE Trans. Multimedia*, vol. 17, no. 9, pp. 1404–1416, Sep. 2015.

[48] R. Gray and D. Neuhoff, “Quantization,” *IEEE Trans. Inf. Theory*, vol. 44, no. 6, pp. 2325–2384, Oct. 1998.

[49] D. D. Lee and H. S. Seung, “Algorithms for non-negative matrix factorization,” in *Proc. Conf. Neural Inf. Process. Syst.*, 2000, pp. 556–562.

[50] D. D. Lee and H. S. Seung, “Learning the parts of objects by nonnegative matrix factorization,” *Nature*, vol. 401, pp. 788–791, 1999.

[51] J. Wang, W. Liu, S. Kumar, and S. Chang, “Learning to hash for indexing big data— survey,” *Proc. IEEE*, vol. 104, no. 1, pp. 34–57, Jan. 2016.

[52] J. Wang, T. Zhang, J. Song, N. Sebe, and H. T. Shen, “A survey on learning to hash,” *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. PP, no. 99, p. 1, 2017.

[53] M. W. Berry and M. Browne, “Email surveillance using non-negative matrix factorization,” *Comput. Math. Org. Theory*, vol. 11, no. 3, pp. 249–264, 2005.

[54] Y. Bao, H. Fang, and J. Zhang, “Topicmf: Simultaneously exploiting ratings and reviews for recommendation,” in *Proc. 28th AAAI Conf. Artif. Intell.*, 2014, pp. 2–8.

[55] G. Trigeorgis, K. Bousmalis, S. Zafeiriou, and B. W. Schuller, “A deep semi-nmf model for learning hidden representations,” in *Proc. Int. Conf. Mach. Learn.*, 2014, pp. 1692–1700.

[56] K. Huang, N. D. Sidiropoulos, and A. Swami, “Non-negative matrix factorization revisited: Uniqueness and algorithm for symmetric decomposition,” *IEEE Trans. Signal Process.*, vol. 62, no. 1, pp. 211–224, Jan. 2014.

[57] Z. Li, X. Wu, and H. Peng, “Nonnegative matrix factorization on orthogonal subspace,” *Pattern Recog. Lett.*, vol. 31, no. 9, pp. 905–911, 2010.

[58] C. Ding, T. Li, W. Peng, and H. Park, “Orthogonal nonnegative matrix t-factorizations for clustering,” in *Proc. Int. Conf. Knowl. Discovery Data Mining*, 2006, pp. 126–135.

[59] M. Collins, S. Dasgupta, and R. E. Schapire, “A generalization of principal components analysis to the exponential family,” in *Proc. Conf. Neural Inf. Process. Syst.*, 2001, pp. 617–624.

[60] I. Jolliffe, *Principal Component Analysis*, 2nd ed. New York, NY, USA: Springer, 2002.

[61] L. van der Maaten and G. E. Hinton, “Visualizing high-dimensional data using t-sne,” *J. Mach. Learn. Res.*, vol. 9, pp. 2579–2605, 2008.

[62] A. Torralba, R. Fergus, and W. T. Freeman, “80 million tiny images: A large data set for nonparametric object and scene recognition,” *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 30, no. 11, pp. 1958–1970, Nov. 2008.

Lei Ma received the B.Sc. degree in communication engineering from Hubei University, Wuhan, China, in 2012, and is currently working toward the Ph.D degree in signal and information processing at the Intelligent Visual Information Processing and Communication Laboratory, University of Electronic Science and Technology of China, Chengdu, China.

His research interests include large-scale multimedia indexing and retrieval, computer vision, pattern recognition, and machine learning.

Hongliang Li (M’06–SM’11) received the Ph.D. degree in electronics and information engineering from Xian Jiaotong University, Xian, China, in 2005.

From 2005 to 2006, he was with the Visual Signal Processing and Communication Laboratory, Chinese University of Hong Kong (CUHK), Hong Kong, China, as a Research Associate. From 2006 to 2008, he was a Postdoctoral Fellow with the same laboratory at CUHK. He is currently a Professor with the School of Electronic Engineering, University of Electronic Science and Technology of China, Chengdu, China. He has authored or coauthored numerous technical articles in well-known international journals and conferences. He is a co-editor of the book *Video Segmentation and its Applications* (Springer, 2011). His research interests include image segmentation, object detection, image and video coding, visual attention, and multimedia communication systems.

Prof. Li was involved in many professional activities. He is a Member of the Editorial Board of the *Journal on Visual Communications and Image Representation*, and the Area Editor of *Signal Processing: Image Communication, Elsevier Science*. He served as a Technical Program Co-Chair for VCIP2016 and ISPACS 2009, General Co-Chair of the ISPACS 2010, Publicity Co-Chair of IEEE VCIP 2013, Local Chair of the IEEE ICME 2014, and a TPC member in a number of international conferences, e.g., ICME 2013, ICME 2012, ISCAS 2013, PCM 2007, PCM 2009, and VCIP 2010.

Fanman Meng (S’12–M’13) received the Ph.D. degree in signal and information processing from the University of Electronic Science and Technology of China, Chengdu, China, in 2014.

From July 2013 to July 2014, he was with the Division of Visual and Interactive Computing, Nanyang Technological University, Singapore, Singapore, as a Research Assistant. He is currently an Associate Professor with the School of Electronic Engineering, University of Electronic Science and Technology of China. He has authored or coauthored numerous technical articles in international journals and conferences. His research interests include image segmentation and object detection.

Prof. Meng is a member of the IEEE CAS society. He was the recipient of the “Best Student Paper Honorable Mention Award” for the 12th Asian Conference on Computer Vision (ACCV 2014) in Singapore and the “Top 10% Paper Award” in the IEEE International Conference on Image Processing (ICIP 2014) at Paris, France.

Qingbo Wu (S’12–M’15) received the B.E. degree in education of applied electronic technology from Hebei Normal University, Shijiazhuang, China, in 2009, and the Ph.D. degree in signal and information processing from the University of Electronic Science and Technology of China, Chengdu, China, in 2015.

From February 2014 to May 2014, he was a Research Assistant with the Image and Video Processing (IVP) Laboratory, Chinese University of Hong Kong, Hong Kong, China. Then, from October 2014 to October 2015, he served as a Visiting Scholar with the Image & Vision Computing (IVC) Laboratory, University of Waterloo, Waterloo, ON, Canada. He is currently a Lecturer with the School of Electronic Engineering, University of Electronic Science and Technology of China. His research interests include image/video coding, quality evaluation, and perceptual modeling and processing.

King Ngi Ngan (M’79–S’79–M’82–SM’91–F’00) received the Ph.D. degree in electrical engineering from Loughborough University, Loughborough, U.K., in 1982.

He is currently a Chair Professor with the Department of Electronic Engineering, Chinese University of Hong Kong, Hong Kong, China. He was previously a Full Professor with the Nanyang Technological University, Singapore, and the University of Western Australia, Crawley, WA, Australia. He has been appointed a Chair Professor at the University of Electronic Science and Technology, Chengdu, China, under the National Thousand Talents Program since 2012. He holds honorary and visiting professorships of numerous universities in China, Australia, and South East Asia. He has published extensively, including 3 authored books, 7 edited volumes, more than 400 refereed technical papers, and edited 9 special issues in journals. In addition, he holds 15 patents in the areas of image/video coding and communications.

Prof. Ngan is a Fellow of IET (U.K.) and IEAust (Australia), and was an IEEE Distinguished Lecturer from 2006–2007. He served as an Associate Editor of IEEE TRANSACTIONS ON CIRCUITS AND SYSTEMS FOR VIDEO TECHNOLOGY, *Journal on Visual Communications and Image Representation*, *EURASIP Journal of Signal Processing: Image Communication*, and *Journal of Applied Signal Processing*. He chaired and co-chaired a number of prestigious international conferences on image and video processing including the 2010 IEEE International Conference on Image Processing, and served on the advisory and technical committees of numerous professional organizations.