Image Retrieval with Query-Adaptive Hashing

DONG LIU, Harbin Institute of Technology
SHUICHENG YAN, National University of Singapore
RONG-RONG JI, Harbin Institute of Technology
XIAN-SHENG HUA, Microsoft Research Asia
HONG-JIANG ZHANG, Microsoft Advanced Technology Center

Hashing-based approximate nearest-neighbor search may well realize scalable content-based image retrieval. The existing semantic-preserving hashing methods leverage the labeled data to learn a fixed set of semantic-aware hash functions. However, a fixed hash function set is unable to well encode all semantic information simultaneously, and ignores the specific user's search intention conveyed by the query. In this article, we propose a query-adaptive hashing method which is able to generate the most appropriate binary codes for different queries. Specifically, a set of semantic-biased discriminant projection matrices are first learnt for each of the semantic concepts, through which a semantic-adaptable hash function set is learnt via a joint sparsity variable selection model. At query time, we further use the sparsity representation procedure to select the most appropriate hash function subset that is informative to the semantic information conveyed by the query. Extensive experiments over three benchmark image datasets well demonstrate the superiority of our proposed query-adaptive hashing method over the state-of-the-art ones in terms of retrieval accuracy.

Categories and Subject Descriptors: H.3.1 [**Content Analysis and Indexing**]: Indexing Methods—*Evaluation/methodology*; H.3.3 [**Information Search and Retrieval**]: Relevance feedback—*Human Information Processing*

General Terms: Algorithms

Additional Key Words and Phrases: Image retrieval, query-adaptive, hashing, joint sparsity

ACM Reference Format:

Liu, D., Yan, S., Ji, R.-R., Hua, X.-S., and Zhang, H.-J. 2013. Image retrieval with query-adaptive hashing. ACM Trans. Multimedia Comput. Commun. Appl. 9, 1, Article 2 (February 2013), 16 pages.
DOI = 10.1145/2422956.2422958 http://doi.acm.org/10.1145/2422956.2422958

1. INTRODUCTION

Content-Based Image Retrieval (CBIR) [Rui et al. 1999] becomes infeasible when handling large-scale, for example, gigantic, image databases since sequential image comparison is not scalable. To realize real-time CBIR, hashing-based *Approximate Nearest-Neighbor* (ANN) search techniques [Wang et al. 2010; Kulis and Grauman 2009; Kulis and Darrell 2009; Kulis et al. 2009; Shakhnarovich et al. 2003]

Authors' addresses: D. Liu (corresponding author), Harbin Institute of Technology, 92 Xidazhi Street, Harbin, Heilongjiang, China, 150001; email: dongliu.hit@gmail.com; S. Yan, National University of Singapore, Singapore; R.-R. Ji, Harbin Institute of Technology, 92 Xidazhi Street, Harbin, Heilongjiang, China, 150001; X.-S. Hua, Microsoft Research Asia; H.-J. Zhang, Microsoft Advanced Teachnology Center.

DOI 10.1145/2422956.2422958 http://doi.acm.org/10.1145/2422956.2422958

have been proposed to explore the trade-off between search accuracy and efficiency. The existing hashing methods can be divided into two paradigms. The first paradigm is based on unsupervised hashing [Lee et al. 2010; Indyk and Motwani 1998; Weiss et al. 2008; Rainsky and Lazebnik 2010], which relies on the unlabeled data to generate the binary codes via a predefined similarity metric. The drawback of this paradigm is that the visual similarity measure may not necessarily reflect the semantic information, which hampers its effectiveness in vision applications. To relieve this deficiency, the second paradigm, namely for the supervised/semi-supervised hashing methods [Salakhutdinov and Hinton 2009; Shakhnarovich et al. 2003; Wang et al. 2010], pursues machine learning techniques to preserve the semantic information in Hamming space. The basic idea is to leverage a set of labeled data to learn the appropriate hash functions through optimizing certain hashing objective, enforcing the data points with the same label to be encoded with similar binary codes. The key issue of this paradigm is the label contradiction in the multilabel setting, where two images with the same label may be labeled differently for other labels. Such contradictory labeling information brings difficulties to the learning procedure, often resulting in a biased hash function set.

Despite the success of the supervised/semi-supervised hashing methods, all of them try to encode the semantic information of all concepts with a fixed set of hash functions. However, due to the gap between semantic concepts and visual features, a fixed hash function set is unable to well reveal all concepts simultaneously, especially for a huge image dataset with a large number of concepts. For CBIR applications, an ideal semantic-preserving hashing method needs to consider the semantic concept(s) conveyed by the query image, and maps all images with the same concepts into the identical binary codes accordingly. This calls for a *Query-Adaptive Hashing* (QAH) algorithm which is able to generate the most appropriate binary codes for different queries. By doing so, we may not only overcome the deficiency of encoding all concepts simultaneously, but also well capture the specific search intention of the user.

There are two challenges in creating a query-adaptive hashing algorithm. The first is the construction of semantic-adaptable hash function set, namely, ensuring that few hash functions from the whole set are sufficient to characterize a semantic concept. Intuitively, a straightforward approach to tackle the task is to construct different hash function sets for individual concepts, but obviously this is infeasible since mapping the images into multiple binary codes will lead to extremely heavy storage requirement. Instead, we learn a uniform hash function set in which different hash function subsets can well capture the specific semantic information of different concepts. To achieve this goal, we learn a set of semantic-biased discriminant projection matrices, each of which characterizes the most discriminative projection directions of one semantic concept. Then we further learn a hash projection matrix which is able to reconstruct each of the semantic-biased projection matrices with only a few of its constitutive hash vectors. By employing the learnt hash projection matrix as hash function set, the individual semantic information is naturally incorporated into different hash function subsets, making the uniform hash function set *adaptable* for the individual concepts. Note that each hash function can be employed to reconstruct more than one semantic-biased projection matrices, since the highly correlated semantic concepts may share some common patterns.

The second challenge is that we need a mechanism to select online from the pool of hash functions the most appropriate ones with respect to the semantic information conveyed in the query. To address this task, we select from the retrieved results of the query image a small set of feedback images, where each image is labeled by the user as favored or not. Based on these feedback images, we can learn a query-biased discriminant projection matrix, which characterizes the discriminative projection directions for the semantic information in the query. Meanwhile, the query image itself can be considered as a one-column discriminant projection matrix if the user does not want to provide feedback. Then the appropriate hash function subset is determined through a sparse representation procedure which aims

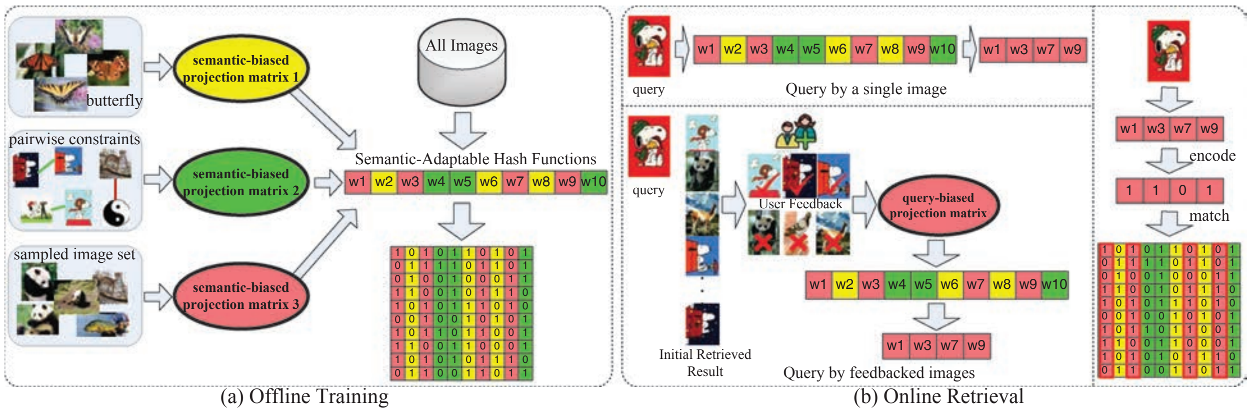


Fig. 1. (a) Offline training: a set of semantic-adaptable hash functions are learnt from the semantic-biased discriminant projection matrices and all images are encoded into binary codes accordingly. (b) Online retrieval: a sparse representation procedure is implemented to select the most appropriate hash functions to reconstruct the semantic information conveyed by the single query image or the feedback images. Finally, the hashing-based image retrieval can be implemented based on the selected query-aware hash function subset. For better viewing, please see the original color pdf file.

at utilizing the least number of hash functions to reconstruct the learnt discriminant projection matrix. At query time, only the bits generated from these selected hash functions are employed as the binary codes. Therefore, our proposed algorithm is able to achieve query-adaptive hashing only through hash function selection without binary codes updating, which is quite efficient with good scalability.

The proposed QAH algorithm is a general two-step framework as illustrated in Figure 1. In the training stage, a set of semantic-adaptable hash functions are learnt from the semantic-biased discriminant projection matrices. Then all images in the database are encoded as binary codes, which implicitly encode all semantic information with different bit combinations. In the query stage, a sparse reconstruction procedure is implemented to select the least number of hash functions to reconstruct the semantic information conveyed in the query image or in the feedback images. The main contributions of this article can be summarized as follows: (1) we propose a query-adaptive hashing algorithm to facilitate image retrieval. The hash functions can be automatically adapted to reflect the semantic information conveyed by the query image, and thus improve the accuracy of image retrieval; (2) to construct semantic-adaptable hash functions, we propose a joint sparsity variable selection model to learn a uniform set of semantic-aware hash functions that are adaptable to the individual concepts; (3) at query time, we propose to use the sparse representation to select the most appropriate hash function subset with respect to the semantic information conveyed by the query or the corresponding feedback images.

2. RELATED WORK

The most popular hashing-based ANN search technique is Locality-Sensitive Hashing (LSH) [Indyk and Motwani 1998]. The basic idea of LSH is to project each data point to a b-bit hash key by applying b binary-valued LSH hash functions $h_1, \ldots, h_b$. Although there exists an asymptotic theoretical guarantee for random projection-based LSH, it may lead to quite inefficient codes in order to achieve satisfactory search accuracy.

To overcome the limitation of LSH, many recently proposed hashing methods attempt to learn data-aware hash functions through machine learning techniques. Spectral Hashing (SpH) [Weiss et al. 2008] is an effective unsupervised hashing method based on the assumption that data points are spread in an Euclidean space with either uniform or Gaussian distribution. However, this assumption is very restrictive and rarely holds for real-world data. The work in Shakhnarovich et al. [2007] used stacked Restricted Boltzmann Machine (RBM) to learn the hash functions in a supervised manner, and showed to generate compact binary codes with semantic information preserved. Although these methods generate semantic-preserving binary codes, their performances heavily rely on the available training data. Either noisy or insufficient training data may degrade their effectiveness. To address this issue, Wang et al. [2010] proposed a semi-supervised hashing formulation where a supervised term minimizes the empirical error on the labeled data while an unsupervised term provides effective regularization from the unlabeled data. Despite the preceding progresses in semantic-preserving hashing, none of them has considered the underlying semantic information conveyed in the query. In contrast, our proposed QAH algorithm can generate adaptive hash functions (and binary codes) according to individual query.

The most related work to this article is Jégou et al. [2008], which proposed a query-adaptive hashing method based on LSH. It defines a criterion to measure the expected accuracy of a given hash function with respect to the query. At query time, the criterion is adopted to online select from a pool of predefined hash functions the most appropriate ones. This is shown to improve the search accuracy. However, its adaptation is solely based on a *heuristic* hash function relevance criterion, and does not consider the semantic information. The other related work is Jiang et al. [2011], which performs query-adaptive ranking for hashing-based image search. The basic concern of this work lies in the fact that a large number of images sharing equal Hamming distances to a query, and an effective method to adapt the Hamming distances between the images and the query is highly desired. From this aspect, the target of this work is to better rank images with respect to the query based on binary codes, which is essentially different from our work here which learns the most discriminative hash bits for the specific query.

3. QUERY-ADAPTIVE HASHING

In this section, we describe the query-adaptive hashing algorithm, starting with the pursuit of semantic-biased discriminant projections. Then we discuss the objective to minimize and develop a coordinate descend procedure for the optimization. Finally, we present how to realize query-adaptive hashing via a sparse representation procedure.

3.1 Semantic-Biased Discriminant Projection

Semantic-biased discriminant projection is a low-dimensional transformation matrix biased by a given semantic concept under consideration or other labeled information. By applying such a transformation matrix, the positive samples will be clustered together while the negative samples will be kept away. From this point of view, the projection matrix essentially carries important discriminative information with respect to the given concept. We employ the *Biased Discriminant Analysis* (BDA) algorithm [Weiss et al. 2001] to obtain the projection matrix for each concept.

For a given concept c, let $\mathcal{X}_c^p = \{\mathbf{x}_1^p, \mathbf{x}_2^p, \ldots, \mathbf{x}_{N_c^p}^p\}$ denote the positive images, and $\mathcal{X}_c^n = \{\mathbf{x}_1^n, \mathbf{x}_2^n, \ldots, \mathbf{x}_{N_c^n}^n\}$ denote the negative images. Each image in these two sets is represented as a feature vector in $\mathbb{R}^d$. N_c^p and N_c^n are the numbers of positive and negative images, respectively. With the asymmetric treatment biased towards the positive images, the objective function of BDA can be written as

$$\mathbf{\Psi}^* = \arg\max_{\mathbf{\Psi}} \frac{|\mathbf{\Psi}^\top \mathbf{S}_p \mathbf{\Psi}|}{|\mathbf{\Psi}^\top \mathbf{S}_n \mathbf{\Psi}|}, \tag{1}$$

where $\mathbf{\Psi}$ denotes the discriminant transformation matrix, $|\cdot|$ denotes the determinant of a matrix. The scatter matrix estimates $\mathbf{S}_n$ and $\mathbf{S}_p$ are obtained by

$$\mathbf{S}_n = \sum_{i=1}^{N_c^n} \big(\mathbf{x}_i^n - \mathbf{m}_p\big)\big(\mathbf{x}_i^n - \mathbf{m}_p\big)^\top,$$
$$\mathbf{S}_p = \sum_{i=1}^{N_c^p} \big(\mathbf{x}_i^p - \mathbf{m}_p\big)\big(\mathbf{x}_i^p - \mathbf{m}_p\big)^\top, \tag{2}$$

where $\mathbf{m}_p$ denotes the mean vector of the positive image subset $\mathcal{X}_c^p$, that is, $\mathbf{m}_p = \frac{1}{N_c^p}\sum_{i=1}^{N_c^p}\mathbf{x}_i^p$.

The solution to Eq. (1) can be obtained by generalized eigenanalysis with the eigenvector matrix $\mathbf{V}$ associated with the (nonzero) eigenvalue matrix $\mathbf{\Lambda}$ satisfying

$$\mathbf{S}_n\mathbf{V} = \mathbf{S}_p\mathbf{V}\mathbf{\Lambda}. \tag{3}$$

Then the discriminant transformation matrix $\mathbf{\Psi}^*$ for the given concept c can be calculated as $\mathbf{\Psi}^* = \mathbf{V}\mathbf{\Lambda}^{1/2}$. Denote by k the number of column vectors in $\mathbf{\Psi}^*$, we set $k = 100$ such that the column vectors of $\mathbf{V}$ corresponding to the 100 largest eigenvalues in $\mathbf{\Lambda}$ are retained in $\mathbf{\Psi}^*$. In the implementation of the BDA algorithm, instead of utilizing all the images in the large-scale database, we only sample from each category a small number of images as training images for learning the optimal discriminant matrix $\mathbf{\Psi}^*$. This makes the subsequent hash function learning procedure scalable for large-scale corpus, since $\mathbf{\Psi}^*$ compresses the semantic information of all samples into a compact matrix, and avoids the direct learning on all training samples, which is infeasible for large-scale applications.

3.2 Semantic-Adaptable Hash Function

The motivation of learning semantic-adaptable hash functions is to construct a hash function set in which different hash function subsets are sufficient to characterize different semantic concepts, which naturally achieves the query-adaptive hashing. Suppose we have an image collection containing N_c concepts, we can learn N_c projection matrices $\{\mathbf{\Psi}_1, \mathbf{\Psi}_2, \ldots, \mathbf{\Psi}_{N_c}\}$ with the aforementioned BDA algorithm, and each matrix $\mathbf{\Psi}_i \in \mathbb{R}^{d\times k}$ reflects the most discriminative projection directions of one concept. Intuitively, we can directly utilize the N_c learnt semantic-biased projection matrices as hash functions to generate semantic-aware binary codes. However, each projection matrix can only contribute for one concept. To fully describe all semantic concepts in the Hamming space, we need to map the images with all the N_c projection matrices. This will lead to extremely long binary codes with heavy storage requirement. Meanwhile, a more desirable technique is to learn a uniform hash function set that is adaptable to different concepts.

In this article, we propose a joint covariate selection model to learn the semantic-adaptable hash functions. The basic idea is to learn a hash projection matrix that can reconstruct all the N_c semantic-biased projection matrices. The proposed optimization objective is formulated as

$$\min_{\mathbf{W},\{\mathbf{U}_c\}} \sum_{c=1}^{N_c} \big[\|\mathbf{\Psi}_c - \mathbf{W}\mathbf{U}_c\|_F^2 + \lambda\|\mathbf{U}_c\|_{2,1}\big], \tag{4}$$

where $\mathbf{W} \in \mathbb{R}^{d\times b}$ is the desired hash projection matrix and $\mathbf{U}_c \in \mathbb{R}^{b\times k}$ is the coefficient matrix for the c-th concept. $\|\cdot\|_F$ denotes the Frobenius norm of a matrix. $\|\mathbf{U}_c\|_{2,1}$ denotes the $\ell_{2,1}$-norm of matrix $\mathbf{U}_c$, and can be calculated as $\|\mathbf{U}_c\|_{2,1} = \sum_{j=1}^{b}\|\mathbf{u}_c^j\|_2$, where $\mathbf{u}_c^j \in \mathbb{R}^{1\times k}$ denotes the j-th row of matrix $\mathbf{U}_c$. λ is a regularization parameter. In our implementation, we set the value of λ such that the scales of the two terms are at similar levels.

Intuitively, one may argue that the standard Linear Discriminant Analysis (LDA) algorithm can also be directly applied to learn a projection matrix that is able to separate the semantic concepts from each other and is used to replace $\mathbf{W}$. However, each projection vector from LDA is discriminative to all the semantic concepts simultaneously, and thus it is generally difficult, if not impossible, to obtain a more compact set of hashing functions for a specific concept from the derived projection matrix by LDA. A closer analysis on the optimization objective in Eq. (4) reveals why the learnt hash projection matrix $\mathbf{W}$ is adaptable to different semantic concepts. For a given semantic-biased projection matrix $\mathbf{\Psi}_c = [\boldsymbol{\psi}_1^c, \ldots, \boldsymbol{\psi}_k^c]$, the minimization of $\|\mathbf{\Psi}_c - \mathbf{W}\mathbf{U}_c\|_F^2$ actually enforces each column vector $\boldsymbol{\psi}_j^c$ in $\mathbf{\Psi}_c$ to be reconstructed by the linear combination of the column vectors in $\mathbf{W}$. Meanwhile, the minimization of $\|\mathbf{U}_c\|_{2,1}$ tends to select basis projection vectors in $\mathbf{W}$ based on the strength of all $\boldsymbol{\psi}_j^c$'s jointly rather than on the strength of individual $\boldsymbol{\psi}_j^c$, making the reconstructions of all $\boldsymbol{\psi}_j^c$'s share similar sparsity patterns. Moreover, the minimization over all semantic-biased projection matrices $\{\mathbf{\Psi}_c\}$ will result in a uniform projection matrix $\mathbf{W}$, in which different subsets of column vectors convey semantic information for different concepts, and thus achieves semantic-adaptable hash functions. Denote by $\mathbf{w}_i$ the i-th column vector of $\mathbf{W}$, we notice that each $\mathbf{w}_i$ can be used for reconstructing more than one semantic-biased projection matrices, which reflects the shared common patterns among the highly correlated semantic concepts.

To prevent $\mathbf{W}$ from being arbitrarily large (which leads to arbitrarily small values of $\mathbf{U}_c$), we compensate the norms of the column vectors in $\mathbf{W}$ into the coefficient matrix $\mathbf{U}_c$ and rewrite the objective function as

$$\min_{\mathbf{W},\{\mathbf{U}_c\}} \sum_{c=1}^{N_c} \left[\|\mathbf{\Psi}_c - \mathbf{W}\mathbf{U}_c\|_F^2 + \lambda\|\mathbf{Q}\mathbf{U}_c\|_{2,1}\right], \tag{5}$$

where the matrix $\mathbf{Q} = diag\{\|\mathbf{w}_1\|_2, \ldots, \|\mathbf{w}_b\|_2\}$.

This objective function is biquadratic, and there is no closed-form solution. Here we present an iterative procedure which applies the block coordinate descent approach to optimize $\mathbf{W}$ and $\{\mathbf{U}_c\}$ alternately. It transforms the original intractable problem into a set of tractable subproblems, and finally converges to the local optimum.

3.2.1 *Optimizing $\boldsymbol{W}$ for Given $\{\boldsymbol{U}_c\}$.* For fixed $\{\mathbf{U}_c\}$, the objective function in Eq. (5) with respect to $\mathbf{W}$ can be written as

$$F(\mathbf{W}) = \sum_{c=1}^{N_c} \left[\|\mathbf{\Psi}_{\boldsymbol{c}} - \mathbf{W}\mathbf{U}_c\|_F^2 + \lambda \sum_{j=1}^{b} s_c^j \|\mathbf{w}_j\|_2 \right], \tag{6}$$

where $s_c^j = \|\mathbf{u}_c^j\|_2$ denotes the ℓ_2-norm of the j-th row vector of matrix $\mathbf{U}_c$.

We choose the $\ell_{2,1}$-mixed norm Accelerated Proximal Gradient (APG) method [Tseng 2008] for the optimization. The APG method comprises alternative updating a weighting matrix sequence $\{\hat{\mathbf{W}}^t\}_{t\geq 1}$ and an aggregation matrix sequence $\{\hat{\mathbf{V}}^t\}_{t\geq 1}$. Each iteration consists of two steps: (1) a generative gradient mapping step to update matrix $\hat{\mathbf{W}}^{t+1}$ with current aggregation matrix $\hat{\mathbf{V}}^t$, and (2) an aggregation forward step to update $\hat{\mathbf{V}}^{t+1}$ by combining $\hat{\mathbf{W}}^{t+1}$ and $\hat{\mathbf{W}}^t$.

Generative gradient mapping step. Given the current matrix $\hat{\mathbf{V}}^t$, we update $\hat{\mathbf{W}}^{t+1}$ according to Tseng [2008] as

$$\begin{aligned} \hat{\mathbf{W}}^{t+1} &= \hat{\mathbf{V}}^t - \eta\nabla^t, \\ \hat{\mathbf{w}}_j^{t+1} &= \left[1 - \frac{\lambda p_j \eta}{\|\hat{\mathbf{w}}_j^{t+1}\|_2}\right]_+ \hat{\mathbf{w}}_j^{t+1}, \ j = 1, \ldots, b, \end{aligned} \tag{7}$$

where $\nabla^t = 2\sum_{c=1}^{N_c}(\mathbf{W}\mathbf{U}_c - \boldsymbol{\Psi}_c)\mathbf{U}_c^\top$, $p_j = \sum_{c=1}^{N_c} s_c^j$ is a constant, η is the step size parameter which is determined by line search according to Armijo-Goldstein condition [Boyd and Vandenberghe 2004], and $[\cdot]_+ = \max(\cdot, 0)$.

Aggregation step. We then construct a linear combination of $\hat{\mathbf{W}}^t$ and $\hat{\mathbf{W}}^{t+1}$ to update $\hat{\mathbf{V}}^{t+1}$ as

$$\hat{\mathbf{V}}^{t+1} = \hat{\mathbf{W}}^{t+1} + \frac{\alpha_{t+1}(1-\alpha_t)}{\alpha_t}(\hat{\mathbf{W}}^{t+1} - \hat{\mathbf{W}}^t). \tag{8}$$

Here the sequence $\{\alpha_t\}_{t\geq 1}$ can be conventionally set to be $\alpha_t = 2/(t+2)$ [Tseng 2008]. Furthermore, since the APG method has the convergence rate of $O(1/t^2)$ [Tseng 2008], the optimization is thus very efficient. After the optimization of $\mathbf{W}$, we normalize the column vector of $\mathbf{W}$ and consequently convey the norm to the coefficient matrix, namely,

$$\mathbf{u}_c^i \Leftarrow \mathbf{u}_c^i \times \|\mathbf{w}_i\|_2, \quad \forall i, c, \tag{9}$$

$$\mathbf{w}_i \Leftarrow \mathbf{w}_i / \|\mathbf{w}_i\|_2, \quad \forall i, \tag{10}$$

and the preceding updatings will not change the value of the objective function in Eq. (5).

3.2.2 *Optimizing $\{\boldsymbol{U}_c\}$ for Given Normalized **W**.* Based on the normalized $\mathbf{W}$, the objective function with respect to $\{\mathbf{U}_c\}$ for given $\mathbf{W}$ can be written as

$$F(\{\mathbf{U}_c\}) = \sum_{c=1}^{N_c}\left[\|\boldsymbol{\Psi}_c - \mathbf{W}\mathbf{U}_c\|_F^2 + \lambda\|\mathbf{Q}\mathbf{U}_c\|_{2,1}\right]. \tag{11}$$

From the objective function, we notice that different $\mathbf{U}_c$'s are independent to each other for optimization, and hence the objective function can be further simplified into matrix-wise form as

$$F(\mathbf{U}_c) = \|\boldsymbol{\Psi}_c - \mathbf{W}\mathbf{U}_c\|_F^2 + \lambda\|\mathbf{Q}\mathbf{U}_c\|_{2,1}. \tag{12}$$

To ease the representation, we omit the subscript in the previous formulation, and rewrite it as

$$F(\mathbf{U}) = \sum_{i=1}^{k}\|\boldsymbol{\psi}_i - \mathbf{W}\mathbf{u}_i\|_2^2 + \lambda\sum_{j=1}^{b} q_j\|\mathbf{u}^j\|_2, \tag{13}$$

where $\boldsymbol{\psi}_i$ denotes the i-th column of $\boldsymbol{\Psi}$, $\mathbf{u}_i$ is the i-th column of coefficient matrix $\mathbf{U}$, $\mathbf{u}^j$ denotes the j-th row of $\mathbf{U}$, and q_j is a constant which equals to $\|\mathbf{w}_j\|_2$.

The minimization of the this objective function can also be accomplished by APG method which alternatively updates a coefficient matrix sequence $\{\hat{\mathbf{U}}^t\}_{t\geq 1}$ and an aggregation matrix sequence $\{\hat{\mathbf{O}}^t\}_{t\geq 1}$. At each iteration, we implement the following two updating steps: (1) a generative gradient mapping step to update matrix $\hat{\mathbf{U}}^{t+1}$ with current aggregation matrix $\hat{\mathbf{O}}^t$, and (2) an aggregation step to update $\hat{\mathbf{O}}^{t+1}$ by combining $\hat{\mathbf{U}}^{t+1}$ and $\hat{\mathbf{U}}^t$.

Generative gradient mapping step. Given the current matrix $\hat{\mathbf{O}}^t$, we update $\hat{\mathbf{U}}^{t+1}$ as

$$\begin{aligned} \hat{\mathbf{U}}^{t+1} &= \hat{\mathbf{O}}^t - \eta\nabla^t, \\ \hat{\mathbf{u}}^{j,t+1} &= \left[1 - \frac{\lambda q_j \eta}{\|\hat{\mathbf{u}}^{j,t+1}\|_2}\right]_+ \hat{\mathbf{u}}^{j,t+1}, \; j = 1, \ldots, b, \end{aligned} \tag{14}$$

where $\nabla^t = 2(\mathbf{W}\mathbf{U}_c - \boldsymbol{\Psi}_c)\mathbf{W}^\top$, η is the step size parameter, and $[\cdot]_+ = \max(\cdot, 0)$.

Aggregation step. We then construct a linear combination of $\hat{\mathbf{U}}^t$ and $\hat{\mathbf{U}}^{t+1}$ to update $\hat{\mathbf{O}}^{t+1}$ as

$$\hat{\mathbf{O}}^{t+1} = \hat{\mathbf{U}}^{t+1} + \frac{\alpha_{t+1}(1-\alpha_t)}{\alpha_t}(\hat{\mathbf{U}}^{t+1} - \hat{\mathbf{U}}^t), \tag{15}$$

where, similar to the updating step for matrix $\mathbf{W}$, we set $\alpha_t = 2/(t+2)$.

3.3 Retrieval with Query-Adaptive Hashing

After we obtain the hash function matrix $\mathbf{W}$ incorporating the semantic information of different concepts, we further apply $\mathbf{W}$ to generate the binary codes for all the images in the database. Specifically, for a given image $\mathbf{x} \in \mathbb{R}^d$, its corresponding l-th hash bit is calculated as

$$y_l = \frac{1}{2}\left(1 + \operatorname{sgn}\left(\mathbf{w}_l^\top \mathbf{x}\right)\right), \tag{16}$$

where $\operatorname{sgn}(\cdot) \in \{-1, +1\}$. This is actually the standard method for generating the binary codes in most projection-based hash algorithms, giving that the mean of all the data is zero. Moreover, since we have b hash functions, the given image $\mathbf{x}$ is encoded as a b-bit binary code in which different bit subsets carry different semantic information for different concept.

Given a query image $\mathbf{x}_q \in \mathbb{R}^d$, there are two scenarios for determining the most appropriate hash function subset for $\mathbf{x}_q$, which correspond to the common actions taken by the general users in CBIR.

Query by feedback images. This scenario is actually the relevance feedback procedure for CBIR. For a small set of images returned from hashing-based image retrieval, the user may identify which images are favored and which are not. These feedback images can be used as additional training samples to learn a query-biased discriminant projection matrix via the BDA algorithm introduced in Section 3.1, which can be further used for determining the appropriate hash function subset. Denote by $\boldsymbol{\Psi}$ the projection matrix obtained from user feedback, the hash function subset selection procedure can be formulated as

$$\min_{\mathbf{U}} \|\boldsymbol{\Psi} - \mathbf{W}\mathbf{U}\|_F^2 + \rho\|\mathbf{U}\|_{2,1}. \tag{17}$$

Due to the $\ell_{2,1}$-norm regularization, the optimal $\mathbf{U}^*$ naturally contains many all-zero rows. We collect the indices for the nonzero rows into Ω. Therefore, the selected hash function subset informative for the query can be written as $\mathbf{W}(\boldsymbol{\Omega})$, which denotes the selected column vectors in $\mathbf{W}$ based on the indices in $\boldsymbol{\Omega}$. The lower part of Figure 1(b) shows an example of the query by feedback images. Suppose we have a set of semantic adaptive hashing functions $\{w_1, \ldots, w_{10}\}$ that are learned via the offline training procedure as shown in Figure 1(a). Then we learn from the feedback images a query-biased projection matrix. Based on the hash function subset selection criteria in Eq. (17), we can select a subset $\{w_1, w_3, w_7, w_9\}$ as the desired hash functions for the given query. Finally, only the binary bits corresponding to these selected hash functions will be utilized for hashing-based image retrieval.

Query by a single image. More generally, the users may only provide a single query image $\mathbf{x}_q$ to perform CBIR. In this case, we do not have any training samples to train a query-biased discriminant projection matrix as in the scenarios of query by feedback images. Instead, the query image itself can be seen as a discriminant vector conveying the category information, and thus the determination of query-adaptive hash function subset is formulated as a sparse coding problem

$$\min_{\boldsymbol{\alpha}} \|\mathbf{x}_q - \mathbf{W}\boldsymbol{\alpha}\|_2^2 + \rho\|\boldsymbol{\alpha}\|_1, \tag{18}$$

where $\boldsymbol{\alpha}$ denotes the reconstruction coefficient vector enforced to be sparse by ℓ_1-norm minimization.

We solve the optimal $\boldsymbol{\alpha}^*$ using ℓ_1-magic package [Candès and Romberg 2007]. Specifically, $\boldsymbol{\alpha}_j^* = 0$ means that the j-th hashing vector $\mathbf{w}_j$ in $\mathbf{W}$ does not contribute to the reconstruction, and thus is not informative for the query image. Therefore, we collect all nonzero entries in $\boldsymbol{\alpha}^*$ and employ the corresponding hash functions as the appropriate hash function subset for the query.

At query time, we can simply use the selected hash functions to realize query-adaptive hashing. More specifically, we only project the query image onto the selected hash functions, and obtain a condensed query binary code that reflects the concept information. For the images which have been encoded as binary codes by the uniform hash function set, only the bits from these selected hash functions (i.e., some fixed positions in the b-bit binary codes generated by the selected hash functions) are needed to be inspected for the given search task. By doing so, we naturally generate appropriate binary codes for the query concept, and thus achieve the query-adaptive hashing in CBIR.

3.4 Time Complexity Analysis

Since our method consists of an offline hash function learning step and online hash function subset selection step, we detail the time complexity for each respectively.

Offline hash function learning. As aforementioned, the optimization is solved by an iterative procedure. For each iteration, we need solve the minimization problem in Eq. (6) and Eq. (11). We first analyze the time complexity of the optimization problem in Eq. (6). Specifically, it costs $O(N_c dkb)$ floating point operations for evaluating the function value and the gradient of the objective function in Eq. (6). Moreover, it costs $O(bd)$ floating point operations to compute $\|\mathbf{w}_j\|_2$, for $j = 1, \ldots, b$. Therefore, the total time complexity for solving the optimization problem in Eq. (6) is $O(N_c dkb)$. On the other hand, the optimization problem in Eq. (11) is tackled by solving N_c subproblems as in Eq. (12), each of which is solved in the time complexity of $O(b(d + k))$. Therefore, the total time complexity for solving N_c subproblems in Eq. (11) is $O(N_c b(d + k))$. Putting everything together, the time complexity for the optimization problem in Eq. (5) is $O(N_c dkb)$. Moreover, the proposed hash function learning algorithm is suitable for incremental learning. When a new category comes in, we can start from the current $\mathbf{W}$ and then update it to incorporate the new category based on the discriminant project matrix of the new category. Therefore, the updating can be implemented very efficiently.

Online query-adaptive hash function subset selection. The time complexity of query-adaptive hash function subset selection can be summarized as follows. For the scenario with feedback images, the optimization problem in Eq. (17) is actually the same as the optimization problem in Eq. (12), and can thus be optimized in the time complexity of $O(b(d + k))$. In particular, when the number of hash functions is fixed as 20, the selection process can be finished within 0.1 seconds on the MATLAB platform of an Intel XeonX5450 workstation with 3.0 GHz CPU and 16GB memory. For the scenario with a single query image, the ℓ_1-magic package is able to solve the task with time complexity of $O(db)$, which can be typically processed within 0.02 seconds. Furthermore, after selecting the query-adaptive hash function subset, the hash keys for all buckets should also be adapted into the corresponding binary bit subset. Each bit subset selection can be processed with $O(1)$ time, and the overall time complexity is $O(m)$, where m is the bucket number. In our practice, each bit subset selection process can be finished with less than 10^{-5} seconds, and therefore, the adaptation procedure is very fast.

4. EXPERIMENTS

To empirically evaluate the effectiveness of our proposed QAH algorithm, we perform experiments on three image datasets: Caltech-101 ($\sim$10K images) [Fei-Fei et al. 2007], Photo Tourism Patch ($\sim$100K images) [Winder and Brown 2007], and ImageNet ($\sim$1.3 million images) [Jia et al. 2009]. We implement our proposed QAH in the scenario of *query by a **S**ingle image* (QAH-S) and *query by **F**eedback images* (QAH-F) respectively, and also compare the proposed algorithm with several state-of-the-art hashing methods in literature, including Locality-Sensitive Hashing (LSH) [Indyk and Motwani 1998], Spectral Hashing (SpH) [Weiss et al. 2008], Semi-Supervised Hashing (SSH) [Wang et al. 2010] and

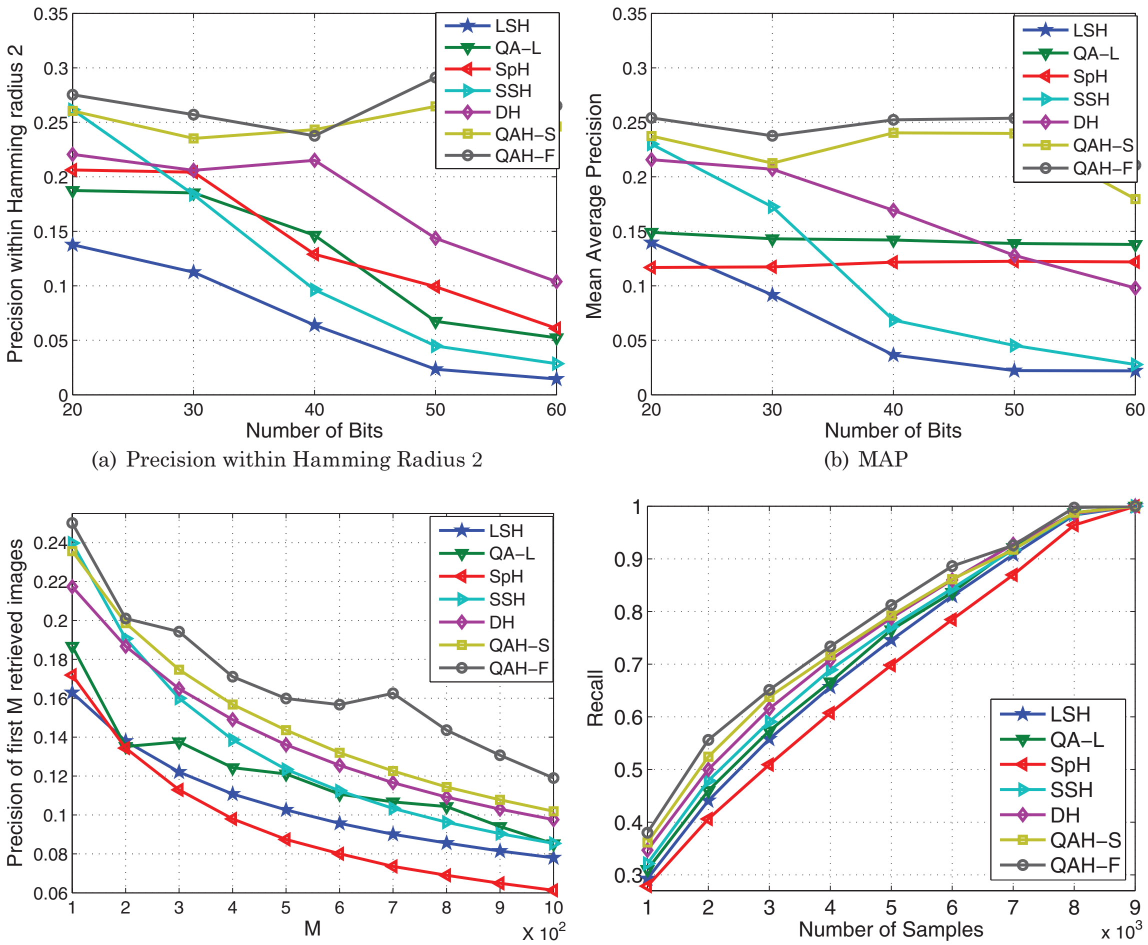
Query-Adaptive LSH (QA-L) [Jégou et al. 2008]. Moreover, we also employ the learnt semantic-adaptable hash functions introduced in Section 3.2 as a comparison baseline, which only performs Discriminating Hashing (DH) without query-adaptive hash function selection and is essentially a supervised hashing method. Therefore, we have five comparison baselines in total, varying from unsupervised hashing method (LSH, SpH, QA-L), supervised hashing (DH) to semi-supervised hashing (SSH), covering the existing hashing methods under different settings. For the random projection-based hashing methods including LSH and QA-L, we run the experiment 10 times and report the averaged results. It is worth noting that our algorithm can work in case where the query image does not belong to the categories for the training data, and thus we do not compare with the image classification algorithms.

To realize CBIR, there exist two commonly used methods in the literature. (1) Hash table lookup. Both the query image and all database images are mapped into binary codes with a predefined hash function set, and all images in the buckets that fall within a small Hamming radius to the query are returned as retrieval results. The complexity of such a similarity search method is of constant time. (2) Ranking with Hamming distance. All the images in the database are ranked according to their Hamming distances with respect to the query image and the top ranked images are returned as visually nearest neighbors of the query image.

As for the quantitative performance measurement, we inspect the search results based on whether the returned images and the query image share the same semantic label. Based on the preceding two visual similarity search scenarios, we use two different evaluation metrics to measure the results of different methods. For the hash-table-lookup-based method, the precision of the returned images which fall in buckets with Hamming radius not larger than 2 to the query image is employed to evaluate the performance. This evaluation metric is widely applied in hashing-based similarity search [Weiss et al. 2008; Wang et al. 2010]. Specifically, if a query returns no neighbors within Hamming radius of 2, it is treated as a failed query with zero precision. On the other hand, for the Hamming distance ranking-based evaluation method, we use the Mean Average Precision (MAP), which approximates the area under the precision-recall curve, as the numerical evaluation metric. Moreover, we also compute the precision within the top M retrieved images returned from Hamming ranking. Finally, recall is calculated for different methods via Hamming ranking by progressively scanning the entire database.

4.1 Image Retrieval on Caltech-101 Dataset (**10**K)

The first experiment is conducted on the Caltech-101 dataset, a standard benchmark for object recognition, which contains about 10K images from 101 object categories and 1 background category. Appearance information of each image is modeled using HSV-SIFT descriptors ($3 \times 128 = 384$ dims) [Bosch et al. 2006] which are computed on a regular grid on the image with a spacing of 10 pixels and a radius of 8. The descriptors are subsequently quantized into a vocabulary of 1000 visual words generated by k-means method. Then each HSV-SIFT feature is mapped into an integer (visual word index) between 1 and 1000, leading to the Bag-Of-Words (BOW) image representation. To perform CBIR, we randomly select from each category 10 images as queries, resulting in 1010 query images in total. The remaining images are used for hash table construction. In addition, the QAH (including QAH-S and QAH-F), DH and SSH need a set of labeled training data for learning semantic-aware hash functions. Therefore, we randomly sample half of the images from each category of the remaining images, and obtain a labeled image set with 4068 images. More specifically, QAH and DH learn semantic-biased discriminative projection matrices using the labeled images while SSH uses them to construct the pairwise label constraints. For QAH-F, we randomly select 20 positive images and 20 negative images as user feedback images.



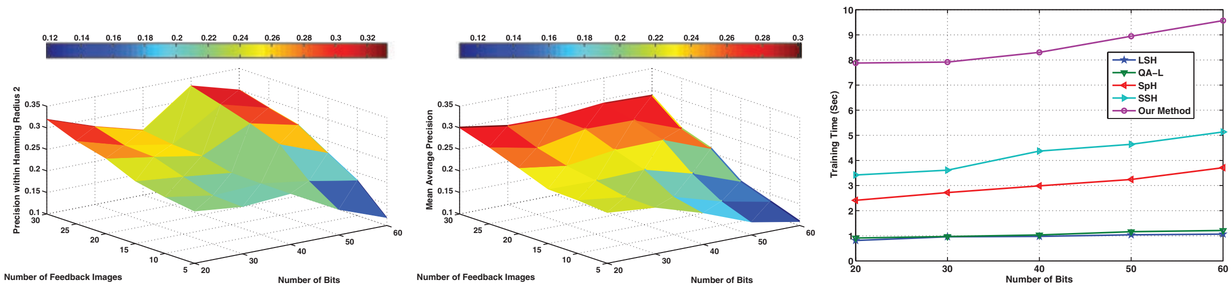
(a) Precision within Hamming Radius 2 (b) MAP

(c) Precision (d) Recall

Fig. 2. (a) Precision within Hamming radius of 2 using hash table lookup. (b) MAP curves using Hamming distance ranking. (c) Precision of the first M retrieved images using Hamming distance ranking with 60 bits. (d) Recall curves using Hamming distance ranking with 60 bits.

Figure 2(a) shows the precision curves of different methods using hash lookup table within Hamming radius of 2 with the number of bits varying from 20 to 60[1]. In Figure 2(b), we further plot the MAP curves for various methods with various number of bits. From the plots, we have the following observations: (1) Our proposed QAH algorithms including QAH-S and QAH-F achieve much better performance than the other five baselines. This clearly demonstrates the effectiveness of adapting hash functions to the individual query images, which is ignored by the state-of-the-art hashing algorithms in literature. (2) The QAH, despite depending on the DH, clearly beats the performance of DH, which is substantially due to the fact that the selected hash functions well capture the semantic information

[1]Here the bit number refers to the number of bits generated by the uniform hash function matrix, based on which, the bit subset used at the query time shall be adaptively selected. For example, in the query by a single image experiment, when the initial bit number is 40, the average number of bits selected for each query is about 13. Therefore, it is possible that the accuracy is even lower when the bit number is larger as shown in Figure 2(a)–2(b).



(a) (b) (c)

Fig. 3. (a) Precision within Hamming radius of 2 as a function of the combination of the hashing bit number and per-category feedback image number. (b) MAP as a function of the combination of the hashing bit number and per-category feedback image number. (c) Training time comparison of different hashing algorithms.

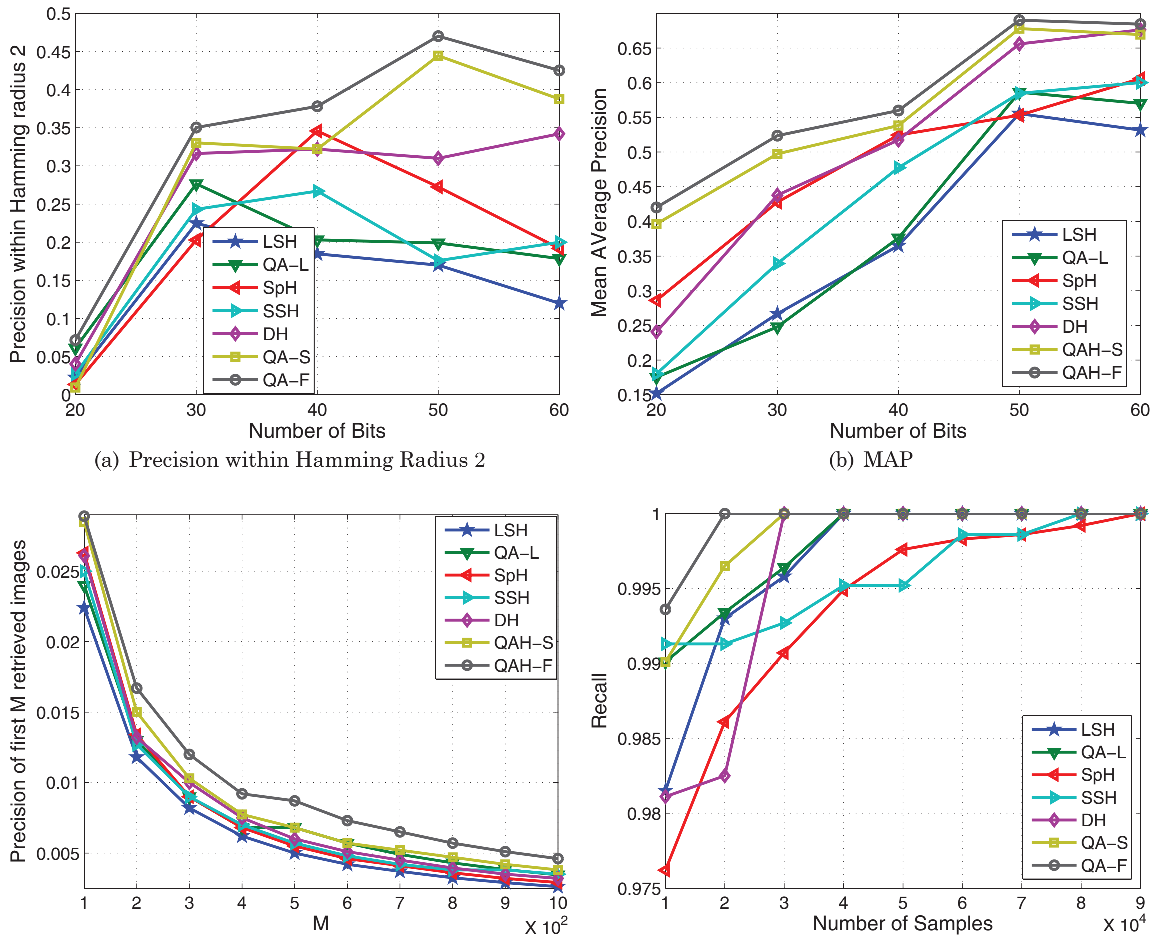
with respect to the query image, while the DH simply applies all of the learnt discriminant hash functions to different queries. (3) QAH outperforms QA-L, since QA-L only relies on a heuristic criterion while QAH explores the underlying semantic information via discriminative learning. (4) QAH-F achieves better performance than QAH-S, since the feedback images provide more discriminant information than a single image. We further plot the precision curves of the first M retrieved images and the recall curves for different methods in Figure 2(c) and Figure 2(d), where the number of bits is fixed as 60. From these results we can observe consistent performance improvements as in the previous two figures. Again, both QAH-F and QAH-S outperform all the baseline methods, and QAH-F clearly beats QAH-S. These results further confirm the superiority of our proposed algorithm. Figure 3(a) and Figure 3(b) show the performance of our proposed algorithm under various combinations of the per-category feedback image number [2] and hash bit number. Figure 3(c) further shows the training time comparison of different algorithms.

4.2 Patch Retrieval on Photo Tourism Data (**100**K)

Photo Tourism image patch dataset is a widely applied benchmark for evaluating the hashing-based retrieval methods. In our experiment, we use 100K 32×32 patches extracted from Notre Dame pictures in this dataset. For each patch, there is a label provided to describe its 3D position. Patches with the same label can be defined as in the same category, which are near-duplicate image patches of the same place of Notre Dame, with variations in lighting, camera viewpoints, and so on. Specifically, 10K patches are randomly selected as the query images and the rest 90K patches are used for hash table construction, where half of them are randomly selected for hash function learning. For each patch, we extract the 320-dimensional gist feature to represent its appearance information. To realize QAH-F, we randomly select 30 positive images and 30 negative images as feedback images for each category (if the total number of positive image for one category is less than 30, we select half of the positive images and the equal number of negative images as feedback images).

The experiment results of our proposed QAH method compared with the other baselines are illustrated as in Figure 4. Figure 4(a) shows the precision within Hamming radius of 2 for various hashing methods with varied number of bits. As can be seen, QAH including QAH-S and QAH-F yield significantly higher precision for all bit lengths, which are often several times higher than that of LSH and QA-L. Once again, our proposed query-adaptive hashing algorithm shows its effectiveness in the image retrieval task. Figure 4(b) further shows the MAP curves for different methods. We can see that our

[2] User provides {5, 10, 15, 20, 25, 30} positive and negative images per-category respectively as the feedback images.



(a) Precision within Hamming Radius 2 (b) MAP

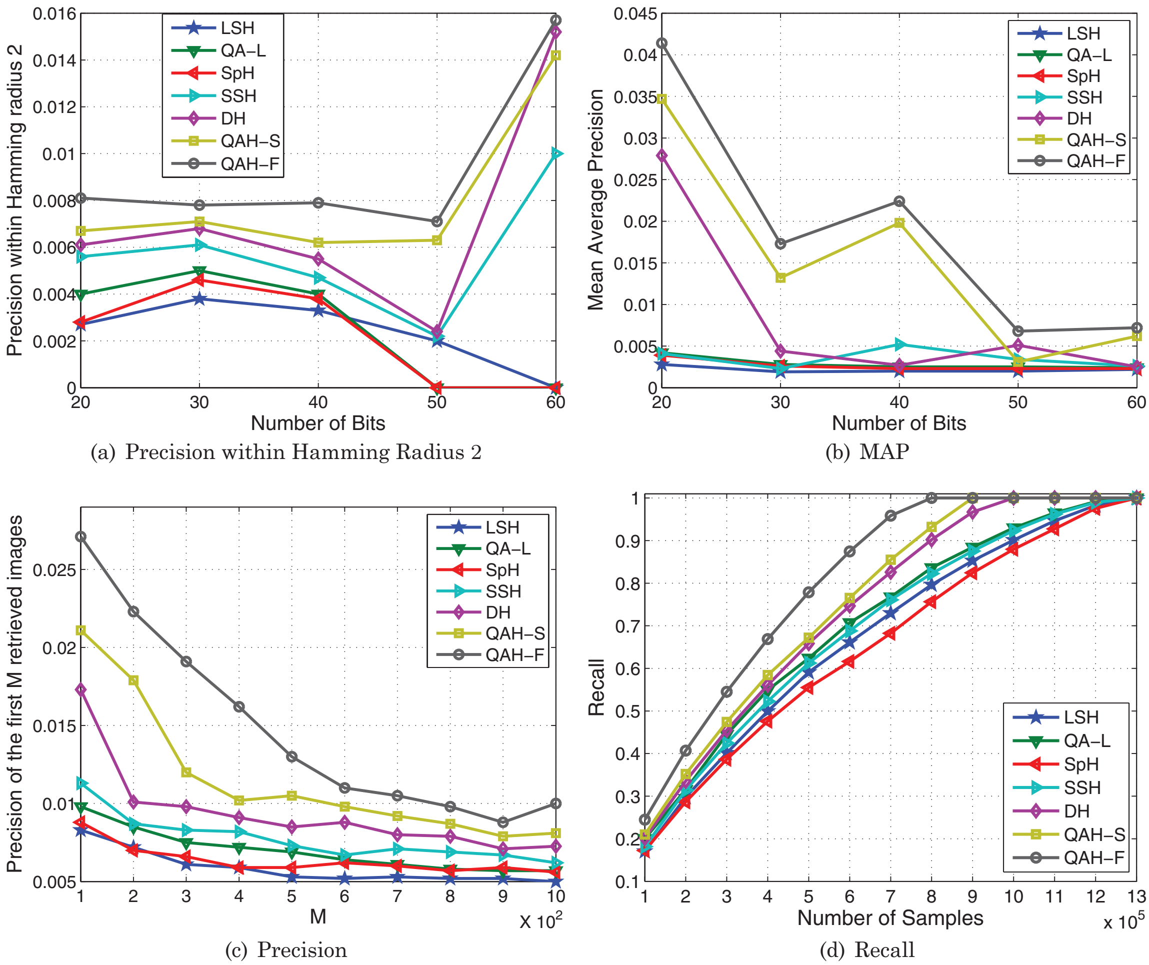
(c) Precision (d) Recall

Fig. 4. (a) Precision within Hamming radius of 2 using hash table lookup. (b) MAP curves using Hamming distance ranking. (c) Precision of the first M retrieved images using Hamming distance ranking with 60 bits. (d) Recall curves using Hamming distance ranking with 60 bits.

method is clearly better than the other hashing methods, and the improvements are consistent for different bit numbers, which indicates that our algorithm is robust under different setting. The precision curves of the top M retrieved images and the recall curves can be seen from Figure 4(c) and Figure 4(d) respectively, where the length of hash bits is fixed as 60. We can see that the QAH-based algorithms including QAH-F and QAH-S outperform the other baseline methods, and QAH-F consistently gives the best results among all the methods.

4.3 Image Retrieval on ImageNet Dataset (**1.3**M)

Finally, we test our proposed method over 1.3 million images, which are sampled from 1000 categories of the ImageNet dataset. For each image, we compute the SIFT descriptors on 20×20 overlapping patches with a spacing of 10 pixels, and obtain a set of dense SIFT descriptors. Then k-means method is performed on a random subset of 10 million SIFT descriptors to form a visual vocabulary of 1000 visual



(a) Precision within Hamming Radius 2 (b) MAP

(c) Precision (d) Recall

Fig. 5. (a) Precision within Hamming radius of 2 using hash table lookup. (b) MAP curves using Hamming distance ranking. (c) Precision of the first M retrieved images using Hamming distance ranking with 60 bits. (d) Recall curves using Hamming distance ranking with 60 bits.

words. Each SIFT descriptor is quantized into a visual word using the nearest cluster center. We select from each category 10 images and obtain a total number of 10K query images. The rest images are used for hash table construction. For the learning-based hashing methods, we further randomly select 50K images from all the categories (50 images from each category) to learn the semantic-preserving hash functions. For QAH-F, we randomly select 50 positive images and 50 negative images as feedback images.

We also compute the quantitative measurement under the hash table lookup and Hamming distance ranking to compare the performances of different methods. A returned image is considered as a matched image if it possesses the same semantic label as the query image. Figure 5(a) shows the precision curves for different methods using the hash lookup table. Once again, our proposed QAH methods including QAH-S and QAH-F achieve better results compared to other baselines in most cases. Figure 5(b) further shows the MAP for different methods with varied number of hash bits. As can be seen, QAH performs generally better than the other hash methods, which confirms the effectiveness of the proposed method. Figure 5(c) and Figure 5(d) display the precision and recall curves for various hash methods with the fixed hash bit number 60. Higher precision and recall values for QAH clearly indicate the advantages of query-adaptive hashing. These results, together with the experimental results obtained on Caltech-101 and Photo Tourism datasets, clearly confirm the effectiveness and robustness of the proposed query-adaptive hashing algorithm.

4.4 Discussion

Now we want to explain why the proposed query-adaptive hashing algorithm produces better results than the state-of-the-art hashing algorithms. The success of the proposed algorithm relies on two critical components: semantic adaptable hash function learning and query-adaptive hash function selection. In the hash function learning stage, we try to learn a uniform projection matrix in which different subsets of column vectors are sufficient to reconstruct each of the discriminant projection matrices corresponding to different categories. This actually compresses the semantic information of different categories into a condensed matrix, and provides a potential for query-adaptable hash functions. The effectiveness of the learnt discriminant projection matrix can be verified by the performance of the DH algorithm over the three image datasets, which generally outperforms the other unsupervised or semi-supervised hash algorithms. On the other hand, at query time, we use the learnt hash function matrix as the dictionary matrix to reconstruct the query image in a sparse manner. This facilitates the selection of a small number of hash functions for revealing the category information residing in the query. From the experimental evaluation on the three datasets, we can see that our proposed algorithm outperforms the other hash algorithms. More specifically, our algorithm clearly beats the DH algorithm, which employs the same hash function learning procedure without the query-adaptive hash function selection. This indicates the effectiveness of the query-adaptive hash function selection for image similarity search. Besides, we can observe the performance improvements of our proposed algorithm over all of the datasets, which shows that the effectiveness of the proposed algorithm is robust.

5. CONCLUSIONS AND FUTURE WORK

We have introduced a query-adaptive hashing algorithm, which is able to generate the most appropriate semantic-preserving binary codes with respect to the query images. An offline training stage is first applied to learn a semantic-adaptable hash function set. Then an online hash function subset selection stage is implemented to select the least number of hash functions for the reconstruction of the semantic information conveyed by the query image. Extensive experiments over three widely used benchmark image datasets have demonstrated the effectiveness of the proposed algorithm. For future work, we will leverage user click-through data generated within the image retrieval process to better learn query-adaptable hash functions.

REFERENCES

BOSCH, A., ZISSERMAN, A., AND MUÑOZ, X. 2006. Scene classification via plsa. In *Proceedings of the European Conference on Computer Vision*. 517–530.

BOYD, S. AND VANDENBERGHE, L. 2004. *Convex Optimization*. Cambridge University Press.

CANDÈS, E. AND ROMBERG, J. 2007. L1-magic. http://users.ece.gatech.edu/~justin/l1magic/.

FEI-FEI, L., ROBERT, F., AND PIETRO, P. 2007. Learning generative visual models from few training examples: An incremental bayesian approach tested on 101 object categories. *Comput. Vision Image Understand. 106,* 1, 59–70.

INDYK, P. AND MOTWANI, R. 1998. Approximate nearest neighbors: Towards removing the curse of dimensionality. In *Proceedings of the Annual ACM Symposium on Theory of Computing*. 604–613.

JÉGOU, H., AMSALEG, L., SCHMID, C., AND GROS, P. 2008. Query-Adaptive locality sensitive hashing. In *Proceedings of the International Conference on Acoustics, Speech, and Signal Processing*. 12–15.

JIA, D., WEI, D., RICHARD, S., LI-JIA, L., KAI, L., AND FEI-FEI, L. 2009. Imagenet: A large-scale hierarchical image database. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. 248–255.

JIANG, Y.-G., WANG, J., AND CHANG, S.-F. 2011. Lost in binarization: Query-adaptive ranking for similar image search with compact codes. In *Proceedings of the ACM International Conference on Multimedia Retrieval*.

KULIS, B. AND DARRELL, T. 2009. Learning to hash with binary reconstructive embeddings. In *Proceedings of the 23rd Annual Conference on Neural Information Processing Systems*. 1042–1050.

KULIS, B. AND GRAUMAN, K. 2009. Kernelized locality-sensitive hashing for scalable image search. In *Proceedings of the International Conference on Computer Vision*. 2130–2137.

KULIS, B., JAIN, B., AND GRAUMAN, K. 2009. Fast similarity search for learned metrics. *IEEE Trans. Pattern Anal. Mach. Intell. 31,* 12, 2143–2157.

LEE, D., KE, Q., AND ISARD, M. 2010. Partition min-hash for partial duplicate image discovery. In *Proceedings of the European Conference on Computer Vision*. 648–662.

RAGINSKY, M. AND LAZEBNIK, S. 2009. Locality-sensitive binary codes from shift-invariant kernels. In *Proceedings of 23rd Annual Conference on Neural Information Processing Systems*. 1509–1517.

RUI, Y., HUANG, T., AND CHANG, S.-F. 1999. Image retrieval: Current techniques, promising directions, and open issues. *J. Vis. Comm. Image Represent. 10,* 1, 39–62.

SALAKHUTDINOV, R. AND HINTON, G. 2009. Semantic hashing. *Int. J. Approx. Reasoning 50,* 7, 969–978.

SHAKHNAROVICH, G., VIOLA, P., AND DARRELL, T. 2003. Fast pose estimation with parameter-sensitive hashing. In *Proceedings of the International Conference on Computer Vision*. 750–757.

SHAKHNAROVICH, R. AND HINTON, G. 2007. Learning a nonlinear embedding by preserving class neighbourhood structure. In *Proceedings of the International Conference on Artificial Intelligence and Statistics*. 1–8.

TSENG, P. 2008. On accelerated proximal gradient methods for convex-concave optimization. *SIAM J. Optim.* (to appear).

WANG, J., KUMAR, S., AND CHANG, S.-F. 2010. Semi-Supervised hashing for scalable image retrieval. In *Proceedings of the International Conference on Computer Vision*. 3424–3431.

WEISS, Y., TORRALBA, A., AND FERGUS, R. 2001. Small sample learning during multimedia retrieval using biasmap. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. 11–17.

WEISS, Y., TORRALBA, A., AND FERGUS, R. 2008. Spectral hashing. In *Advances in Neural Information Processing*. 1753–1760.

WINDER, S. AND BROWN, M. 2007. Learning local image descriptors. In *Proceedings of the International Conference on Computer Vision*. 1–8.

Received May 2011; revised December 2011; accepted March 2012