IEICE TRANS. INF. & SYST., VOL.E100–D, NO.4 APRIL 2017

INVITED PAPER *Special Section on Award-winning Papers*

Multimodal Learning of Geometry-Preserving Binary Codes for Semantic Image Retrieval

Go IRIE[†a)], **Hiroyuki ARAI**[†], ***Members***, ***and*** **Yukinobu TANIGUCHI**[††], ***Senior Member***

SUMMARY This paper presents an unsupervised approach to feature binary coding for efficient semantic image retrieval. Although the majority of the existing methods aim to preserve neighborhood structures of the feature space, semantically similar images are not always in such neighbors but are rather distributed in non-linear low-dimensional manifolds. Moreover, images are rarely alone on the Internet and are often surrounded by text data such as tags, attributes, and captions, which tend to carry rich semantic information about the images. On the basis of these observations, the approach presented in this paper aims at learning binary codes for semantic image retrieval using multimodal information sources while preserving the essential low-dimensional structures of the data distributions in the Hamming space. Specifically, after finding the low-dimensional structures of the data by using an unsupervised sparse coding technique, our approach learns a set of linear projections for binary coding by solving an optimization problem which is designed to jointly preserve the extracted data structures and multimodal data correlations between images and texts in the Hamming space as much as possible. We show that the joint optimization problem can readily be transformed into a generalized eigenproblem that can be efficiently solved. Extensive experiments demonstrate that our method yields significant performance gains over several existing methods.

key words: *image retrieval, multimodal learning, binary coding*

1. Introduction

Image retrieval is a central topic in many research fields like image processing, multimedia, and computer vision. Suppose we have an image database of n images (image features) $\mathcal{X} := \{\mathbf{x}_i \in \mathcal{F}_x\}_{i=1}^n$, where $\mathcal{F}_x \subset \mathbb{R}^{d_x}$ is a d_x-dimensional image feature space. Given a query image $\mathbf{q} \in \mathcal{F}_x$, the task is to find a subset of database images $\mathcal{X}^* \subset \mathcal{X}$ that is relevant to the query $\mathbf{q}$. Here, the retrieved subset $\mathcal{X}^*$ is expected to be semantically similar to the query, e.g., images depicting the same object as the query. A straightforward solution would be to use an exhaustive L_2 scan. Unfortunately, such a scan is often prohibitive due to its linear time complexity with respect to the database size n and feature dimensionality d_x. Classical tree-based indexing techniques like kd-tree [1] are not satisfactory either, because they are inefficient for the high dimensional features that are often used in semantic image retrieval problems.

One promising approach is to use feature binary coding, a.k.a. binary hashing. In binary coding, the original database entries $\forall \mathbf{x} \in \mathcal{X}$ (resp. the query $\mathbf{q} \in \mathcal{F}_x$) are encoded into c-bit binary vectors as $\mathbf{z} := \phi(\mathbf{x}) \in \mathcal{H}$ (resp. $\mathbf{z}_q := \phi(\mathbf{q}) \in \mathcal{H}$), where $\mathcal{H} := \{\pm 1\}^c$ is a c-dimensional Hamming space such that $c \leq d_x$ and ϕ is a mapping from $\mathcal{F}_x$ to $\mathcal{H}$. It has been proven that the search time with binary codes in a Hamming space can be reduced to sub-linear time [2], so the retrieval process is significantly accelerated. Binary coding has received much attention recently because of this property.

Our focus in this paper is on unsupervised feature binary coding for efficient semantic image retrieval. The challenge is how to obtain effective binary codes, or the function ϕ, for semantic image retrieval without any explicit supervision (e.g., class labels). One idea would be to design learning algorithms that can preserve neighborhood structures of data in the Hamming space $\mathcal{H}$ as much as possible. A variety of methods have been proposed for this, including PCA-based [3], [4], graph-based [5]–[7], and clustering-based [8], [9], just to name a few. The majority of these methods aim to preserve the neighborhood structures of the feature space $\mathcal{F}_x$. However, such neighbors do not always capture semantically similar images in many real-world problems. Moreover, in emerging real-world scenarios like web or social media retrieval, an image is rarely alone. Rather, an image is often surrounded by various pieces of text information, such as tags, attributes, and captions. These text data may carry rich semantic information about images and thus would be useful for learning semantic binary codes.

In this paper, we present an approach to unsupervised feature binary coding for semantic image retrieval. Based on an observation that semantically similar images tend to form non-linear manifolds in $\mathcal{F}_x$ [10]–[12], we first extract such inherent data structures relevant to image semantics from $\mathcal{X}$ by using an unsupervised sparse coding technique [13]. In order to learn a set of linear projections used as ϕ, we design an optimization problem that jointly preserves the discovered inherent data structures and the multimodal data correlations between images and their associated texts in the Hamming space $\mathcal{H}$. Furthermore, we show that this optimization problem can be readily transformed into a generalized eigenproblem which can be readily solved. Extensive experiments indicate that our method significantly improves semantic image retrieval accuracy compared with several existing methods.

The remainder of this paper is organized as follows.

Manuscript received October 28, 2016.
Manuscript revised December 21, 2016.
Manuscript publicized January 6, 2017.

[†]The authors are with Nippon Telegraph & Telephone Corporation, Yokosuka-shi, 239–0847 Japan.
[††]The author is with Tokyo University of Science, Tokyo, 125–8585 Japan.
a) E-mail: irie.go@lab.ntt.co.jp

DOI: 10.1587/transinf.2016AWI0003

We briefly review the previous studies in the next section. Then we present an overview and the details of our approach in Sect. 3. Section 4 describes experimental results demonstrating the effectiveness of our method, and Sect. 5 gives concluding remarks.

2. Related Work

As mentioned in Sect. 1, many unsupervised binary coding approaches have been proposed. We here briefly review representative methods. One popular approach is to use principal component analysis (PCA) to learn efficient linear projections for binary coding [3]. A state-of-the-art PCA-based method is Iterative Quantization (ITQ) [4]. ITQ refines the initial PCA projections by finding an orthonormal projection matrix that can minimize the quantization errors between the projected real-valued vectors and binary codes. Some methods learns non-linear mapping functions to preserve the local proximity of the data in binary codes. Spectral hashing (SH) [5] and anchor graph hashing [6] learn binary codes based on the graph Laplacian [14]; they reduce the feature dimensionality while preserving pairwise data proximities. Inductive hashing on manifolds [7] is based on a similar idea but uses t-distributed stochastic neighbor embedding [15] instead of a graph Laplacian. There are other studies [8], [9] that rely on clustering assumptions.

A majority of the existing approaches aim to preserve the neighborhood structures of the feature space $\mathcal{F}_x$, such as dominant dimensions of the data variances [3], [4] or local data proximities [5]–[9]. However, even data proximity is not enough to capture the intrinsic structures (i.e., essential low-dimensional structures) of multiple manifolds, especially if they are close to each other in the space. This is because no matter how accurately the data proximities in the feature space are preserved, it is not possible to distinguish one manifold from the others. One method [13] considers the case and uses a locally linear sparse coding technique to extract the intrinsic structures of separate manifolds. Following this idea, our approach also utilizes locally linear sparse coding to capture the data structures. However, unlike [13], which uses non-linear mapping for the binary coding, our formulation is designed to learn linear projections for more efficient coding.

Furthermore, our approach uses multimodal sources to learn binary codes, which allows us to obtain much more semantic code by leveraging the rich information carried by text data. Certain recent studies with a similar motivation use multimodal binary codes obtained from multiple features for retrieval [16], [17]. By contrast, we do not assume that multiple features are available in the retrieval stage, and instead we aim at improving a single-modal binary code by leveraging multimodal information sources. Some studies consider cross-modal hashing [18]–[21], which also uses multimodal information sources to learn binary codes. However, they mainly focus on the cross-modal retrieval problem in which the task is to retrieve data of one modality type with a query of another type (e.g., retrieve text data from an image query). Rather, our focus is on uni-modal retrieval, and our motivation is in improving uni-modal retrieval performance by multimodal learning. In addition, our formulation is different from existing cross-modal hashing methods and preserves both multimodal data correlations and intrinsic data structures discovered by the sparse coding technique.

3. Method

We first define our problem. Let us denote by $\mathcal{F}_x \subset \mathbb{R}^{d_x}$ and $\mathcal{F}_y \subset \mathbb{R}^{d_y}$ an image and a text feature spaces. Suppose we have data matrices of images and text, $X := [\mathbf{x}_1, \ldots, \mathbf{x}_n]$ and $Y := [\mathbf{y}_1, \ldots, \mathbf{y}_n]$ where $\mathbf{x}_i$ and $\mathbf{y}_i$, $i = 1, \ldots, n$, are i.i.d. distributed in each of $\mathcal{F}_x$ and $\mathcal{F}_y$, respectively. Furthermore, we assume that $\mathbf{x}_i$ and $\mathbf{y}_i$, $\forall i$, are semantically relevant to each other, e.g., an image and a text description of the image. Given such a dataset, our goal is to obtain a function $\phi : \mathcal{F}_x \longrightarrow \mathcal{H}$ for binary coding of image features. Without loss of generality, we will hereafter assume that the empirical means over X and Y are $\mathbf{0}$. Following [3], [4], we will consistently consider the following signed linear function for ϕ.

$$\phi(\mathbf{x}) = \mathrm{sign}(A^\top \mathbf{x}), \tag{1}$$

where $A \in \mathbb{R}^{d_x \times c}$ is a linear projection matrix, and sign() is an element-wise sign function. Compared with non-linear functions, linear projections have fewer parameters and thus are more efficient for computing binary codes in terms of both space and time. The main problem now is how to determine A.

Our approach determines A so as to preserve both the intrinsic manifold structures of X and Y, as well as multimodal data correlations between X and Y. To this end, we cast this problem as a joint optimization problem of A. The basic structure of our loss function (to be minimized) can be written as follows.

$$\lambda \mathcal{G}(X) + (1 - \lambda)\mathcal{G}(Y) + \eta \mathcal{C}(X, Y), \tag{2}$$

where $\mathcal{G}$ and $\mathcal{C}$ are called the *geometric loss function* and *multimodal loss function*, respectively. λ and η are balancing parameters. We give the details of $\mathcal{G}$ and $\mathcal{C}$ in Sects. 3.1 and 3.2, respectively; then we describe the total problem and the algorithm to solve it in Sect. 3.3.

3.1 Geometric Loss Function

This section presents our geometric loss function $\mathcal{G}$ which is designed to preserve the intrinsic data structures of X (resp. Y). The key idea is based on [13] which aims to extract the structures of multiple manifolds sampled in $\mathcal{X}$ (resp. $\mathcal{Y}$) by using an unsupervised sparse coding technique similar to [22].

First, we introduce a key observation that supports the above idea. Suppose we have a $d_{\mathcal{M}_\mathbf{x}}$-dimensional manifold $\mathcal{M}_\mathbf{x}$ in which $\mathbf{x}$ lies (typically $d_{\mathcal{M}_\mathbf{x}} \ll d_x$).

Definition 1. $\mathcal{M}_{\mathbf{x}}$ is a $d_{\mathcal{M}_{\mathbf{x}}}$-dimensional manifold if it is a topological space where each point has a neighborhood that is homeomorphic to $\mathbb{R}^{d_{\mathcal{M}_{\mathbf{x}}}}$.

Observation 1. If $\mathcal{M}_{\mathbf{x}}$ is a $d_{\mathcal{M}_{\mathbf{x}}}$-dimensional manifold, $\forall \mathbf{x}$ in $\mathcal{M}_{\mathbf{x}}$ can be linearly spanned by $d_{\mathcal{M}_{\mathbf{x}}}+1$ points in the tangent space at $\mathbf{x}$. If $\mathcal{M}_{\mathbf{x}}$ is densely enough sampled by $\mathcal{X}$, it can be approximately spanned by $d_{\mathcal{M}_{\mathbf{x}}}+1$ neighbor points in $\mathcal{M}_{\mathbf{x}}$ sampled from $\mathcal{X}$.

If $\mathcal{M}_{\mathbf{x}}$ is a linear manifold, then the observation stands because $\mathcal{M}_{\mathbf{x}}$ can be globally spanned by arbitrary $d_{\mathcal{M}_{\mathbf{x}}}+1$ points on $\mathcal{M}_{\mathbf{x}}$ sampled from $\mathcal{X}$[†]. If $\mathcal{M}_{\mathbf{x}}$ is non-linear, $\mathcal{M}_{\mathbf{x}}$ is locally homeomorphic to $\mathbb{R}^{d_{\mathcal{M}_{\mathbf{x}}}}$ in its tangent space.

The above implies that the local structure of a manifold around $\forall \mathbf{x} \in \mathcal{F}$ can be effectively captured through a linear combination of its $d_{\mathcal{M}_{\mathbf{x}}}+1$ neighbor points on $\mathcal{M}_{\mathbf{x}}$ sampled from $\mathcal{X}$. Accordingly, the function $\mathcal{G}$ is constructed in the following two steps: (1) extracting locally linear structures at each point of X (resp. Y) in order to capture the structure of a manifold around the point. This can be done by finding a linear reconstruction of $\mathbf{x}$ using its (at least) $d_{\mathcal{M}_{\mathbf{x}}}+1$ neighbor points; (2) designing a function to preserve the extracted linear structures in $\mathcal{H}$. In this paper, the same form of $\mathcal{G}$ is used for both X and Y; we hereafter explain the case of X only, for simplicity.

Extracting locally linear structures [13]. What we want to do here is to find linear reconstruction weights for $\forall \mathbf{x} \in \mathcal{X}$. If we can assume that one manifold $\mathcal{M}_{\mathbf{x}}$ is far enough from the others in $\mathcal{F}_x$ and that $d_{\mathcal{M}_{\mathbf{x}}}$ is known, then it is enough to just retrieve $d_{\mathcal{M}_{\mathbf{x}}}+1$ Euclidean neighbor points and compute linear reconstruction weights. However, neither can be assumed in practice. Fortunately, it is possible to take a set of Euclidean neighbor points $\mathcal{N}(\mathbf{x})$ which is large enough to contain the desired $d_{\mathcal{M}_{\mathbf{x}}}+1$ neighbor points in $\mathcal{M}_{\mathbf{x}}$ ($|\mathcal{N}(\mathbf{x})| \geq d_{\mathcal{M}_{\mathbf{x}}}+1$). Therefore, after obtaining such a set $\mathcal{N}(\mathbf{x})$, we try to select only those in $\mathcal{M}_{\mathbf{x}}$ from $\mathcal{N}(\mathbf{x})$. The most efficient, i.e., sparsest, linear reconstruction of $\mathbf{x}$ is achieved when we use only the neighbor points in the same tangent space of $\mathcal{M}_{\mathbf{x}}$ at $\mathbf{x}$. This leads to the following locally linear sparse coding problem [13].

$$\min_{\mathbf{w}_i} \frac{1}{2}\|\mathbf{x}_i - \sum_{j\in\mathcal{N}(\mathbf{x}_i)} w_{ij}\mathbf{x}_j\|^2 + \tau\|\mathbf{s}_i\mathbf{w}_i\|_1 \tag{3}$$

$$\text{s.t.: } \mathbf{w}_i^\top \mathbf{1} = 1, \tag{4}$$

where $\mathbf{w}_i \in \mathbb{R}^n$ is a vector of linear reconstruction weights to be optimized ($w_{ij}=0$ if $j \notin \mathcal{N}(\mathbf{x}_i)$) and $\mathbf{1}$ is the vector of all ones with the same size as $\mathbf{w}_i$. $\mathbf{s}_i\mathbf{w}_i$ means element-wise multiplication between $\mathbf{s}_i$ and $\mathbf{w}_i$. The first term is the locally linear reconstruction error, and the second term is a sparsity inducing term that penalizes distant points with $\mathbf{s}_i := (s_{i1},\ldots,s_{in})^\top$. s_{ij} is determined so that it takes a larger value as the distance between $\mathbf{x}_i$ and $\mathbf{x}_j$ is larger; we use $s_{ij} := \frac{\|\mathbf{x}_i-\mathbf{x}_j\|}{\sum_j\|\mathbf{x}_i-\mathbf{x}_j\|}$ if $j \in \mathcal{N}(\mathbf{x}_i)$ and 0 otherwise. τ is a parameter which balances two terms. The solution $\mathbf{w}_i$ ($w_{ij}=0$ if $j \notin \mathcal{N}(\mathbf{x}_i)$) obtained by solving this problem is the sparse linear reconstruction of $\mathbf{x}_i$; thus, the non-zero dimensions are expected to correspond to $d_{\mathcal{M}_{\mathbf{x}_i}}+1$ neighbor points in $\mathcal{M}_{\mathbf{x}_i}$. Hence, the intrinsic structures of the manifolds can be captured with $W := [\mathbf{w}_1,\ldots,\mathbf{w}_n]$. Note that this is a simple small weighted sparse coding problem with $|\mathcal{N}(\mathbf{x})|$ unknown variables so it can be efficiently solved with a typical sparse coding solver. In this study, we use a homotopy algorithm [23] since we expect $\mathbf{w}_i$ to be rather sparse.

Designing $\mathcal{G}$. $\mathcal{G}$ is designed to preserve W in $\mathcal{H}$ as faithfully as possible. Here, we denote the binary code for $\mathbf{x}_i$ ($i=1,\ldots,n$) by $\mathbf{z}_i \in \mathcal{H}$. $\mathcal{G}$ is naturally defined as follows.

$$\mathcal{G}(X) := \frac{1}{n}\sum_{i=1}^{n}\|\mathbf{z}_i - \sum_{j=1}^{n} w_{ij}\mathbf{z}_j\|^2. \tag{5}$$

This objective is the linear reconstruction error of $\{\mathbf{z}_1,\ldots,\mathbf{z}_n\}$ with fixed W, hence, minimizing it with respect to $\{\mathbf{z}_1,\ldots,\mathbf{z}_n\}$ will optimally preserve W in $\mathcal{H}$. Note that it can be rewritten in matrix form as

$$\mathcal{G}(X) = \frac{1}{n}\sum_{i=1}^{n}\|\mathbf{z}_i - \sum_{j=1}^{n} w_{ij}\mathbf{z}_j\|^2 = \mathrm{tr}(Z^\top M_x Z), \tag{6}$$

where $M_x := (I_n - W)^\top(I_n - W)/n$ and $Z := [\mathbf{z}_1,\ldots,\mathbf{z}_n]^\top$. Since $Z = \mathrm{sign}(X^\top A)$ by Eq. (1),

$$\mathcal{G}(X) = \mathrm{tr}(\mathrm{sign}(A^\top X)\, M_x\, \mathrm{sign}(X^\top A)), \tag{7}$$

which gives the final definition of $\mathcal{G}$.

It is worth noting that minimizing the above objective $\mathcal{G}$ immediately gives a uni-modal learning formulation of geometry-preserving binary codes.

$$\min_A \ \mathcal{G}(X) = \mathrm{tr}(\mathrm{sign}(A^\top X)\, M_x\, \mathrm{sign}(X^\top A)) \tag{8}$$

$$\text{s.t.: } \mathrm{sign}(A^\top X)\mathrm{sign}(X^\top A) = nI_c, \tag{9}$$

where the constraint is imposed to ensure that the bits are independent of each other [5]. Unfortunately, this is an NP-hard problem due to the existence of the sign functions. By replacing each sign function by its signed magnitude [3], [4], [18], the problem can be relaxed into

$$\min_A \ \mathrm{tr}(A^\top X M_x X^\top A) \tag{10}$$

$$\text{s.t.: } A^\top X X^\top A = nI_c. \tag{11}$$

Since both $XM_xX^\top$ and $XX^\top$ are symmetric positive semi-definite, the above is a standard generalized eigenproblem and its solution consists of c eigenvectors corresponding to c minimum eigenvalues of $(XX^\top)^{-1}(XM_xX^\top)$. After obtaining the optimal A, one can generate binary codes for $\forall \mathbf{x} \in \mathcal{F}_x$ as $\mathbf{z} = \mathrm{sign}(A^\top \mathbf{x})$. Note that this problem is similar to a linear dimensionality reduction technique called neighborhood preserving embedding (NPE) [24]. NPE and our approach differ in the way W is constructed. NPE uses standard linear

[†]For instance, a 1-dimensional linear manifold (a straight line) can be uniquely specified by arbitrary two points, and a 2-dimensional linear manifold (a plane) can be specified by three points.

reconstruction weights obtained by solving a least-squares problem [24], while we optimize W by using sparse coding.

3.2 Multimodal Loss Function

Next, we introduce our multimodal loss function C. Given n data pairs $\{(\mathbf{x}_i, \mathbf{y}_i)\}_{i=1}^n$, we want that the distance between two binary codes is small in every data pair in $\mathcal{H}$ as much as possible. Here, we denote by $Z_x := [\mathbf{z}_1^{(x)}, \ldots, \mathbf{z}_n^{(x)}]^\top$ and $Z_y := [\mathbf{z}_1^{(y)}, \ldots, \mathbf{z}_n^{(y)}]^\top$ the binary codes for $X = [\mathbf{x}_1, \ldots, \mathbf{x}_n]$ and $Y = [\mathbf{y}_1, \ldots, \mathbf{y}_n]$, respectively. We define the function C as the average distance between Z_x and Z_y which can be represented as

$$C(X, Y) := \frac{1}{n} \sum_{i=1}^{n} \|\mathbf{z}_i^{(x)} - \mathbf{z}_i^{(y)}\|^2. \tag{12}$$

Note that $\mathbf{z}_i^{(x)} = \mathrm{sign}(A^\top \mathbf{x}_i)$ and $\mathbf{z}_i^{(y)} = \mathrm{sign}(B^\top \mathbf{y}_i)$ where $B \in \mathbb{R}^{d_y \times c}$ is the projection matrix for Y; therefore, C can be defined as

$$C(X, Y) = \frac{1}{n} \sum_{i=1}^{n} \|\mathrm{sign}(A^\top \mathbf{x}_i) - \mathrm{sign}(B^\top \mathbf{y}_i)\|^2. \tag{13}$$

By minimizing this expression with respect to A and B, we can find A (and B) that minimize the average Hamming distance over the all pairs $\{(\mathbf{z}_i^{(x)}, \mathbf{z}_i^{(y)})\}_{i=1}^n$.

3.3 Multimodal Learning of Geometry-Preserving Binary Codes

By combining the two geometric loss functions, $\mathcal{G}(X)$ and $\mathcal{G}(Y)$, one for each modality, and the multimodal loss function $C(X, Y)$ together, the total problem of learning A (and B) can be defined as follows.

$$\min_{A,B} \lambda \mathcal{G}(X) + (1 - \lambda)\mathcal{G}(Y) + \eta C(X, Y) \tag{14}$$

$$\text{s.t.: } \mathrm{sign}(A^\top X)\mathrm{sign}(X^\top A) = nI_c, \tag{15}$$

$$\mathrm{sign}(B^\top Y)\mathrm{sign}(Y^\top B) = nI_c. \tag{16}$$

The constraints Eq. (15) and Eq. (16) are imposed to ensure that the learned bits are independent of each other. λ ($0 \leq \lambda \leq 1$) and $\eta \geq 0$ are balancing parameters. Although this problem optimizes both A and B (i.e., projections for images and texts, respectively), we are interested in binary coding of image features in this paper; thus, we will use only A and discard B.

When we set $\lambda = 1$ (resp. $\lambda = 0$) and $\eta = 0$, the problem becomes exactly the same as the uni-modal case Eqs. (8), (9), where A (resp. B) is trained so as to optimally preserve the locally linear geometries of X (resp. Y) in $\mathcal{H}$. On the other hand, when we take $\eta \to \infty$ and ignore sign(), the problem turns into

$$\min_{A,B} \frac{1}{n} \sum_{i=1}^{n} \|A^\top \mathbf{x}_i - B^\top \mathbf{y}_i\|^2 \tag{17}$$

$$\text{s.t.: } A^\top X X^\top A = nI_c, \tag{18}$$

$$B^\top Y Y^\top B = nI_c. \tag{19}$$

This is exactly canonical correlation analysis (CCA); namely, in this case A and B are trained to maximize the multimodal correlations between X and Y. These facts indicate that our formulation finds A (and B) such that the intrinsic data geometries and multimodal correlations between the two modality spaces are jointly preserved in the Hamming space.

Solution. Interestingly, we can show that the total problem is equivalent to the uni-modal case presented in Eqs. (8), (9). Let us define the following two block matrices.

$$Q := \begin{bmatrix} X & 0 \\ 0 & Y \end{bmatrix}, \quad P := \begin{bmatrix} A \\ B \end{bmatrix}. \tag{20}$$

Then the first two terms of the total objective Eq. (14) can be transformed into

$$\begin{aligned} &\lambda \mathcal{G}(X) + (1 - \lambda)\mathcal{G}(Y) \\ &= \lambda \mathrm{tr}(\mathrm{sign}(A^\top X)\, M_x\, \mathrm{sign}(X^\top A)) \\ &\quad + (1 - \lambda)\mathrm{tr}(\mathrm{sign}(B^\top Y)\, M_y\, \mathrm{sign}(Y^\top B)) \\ &= \mathrm{tr}(\mathrm{sign}(P^\top Q)\, M_{\mathcal{G}}\, \mathrm{sign}(Q^\top P)), \end{aligned} \tag{21}$$

where

$$M_{\mathcal{G}} := \begin{bmatrix} \lambda M_x & 0 \\ 0 & (1 - \lambda)M_y \end{bmatrix}. \tag{22}$$

The last term in Eq. (14) can be rewritten as

$$\begin{aligned} \eta C(X, Y) =& \frac{\eta}{n} \sum_{i=1}^{n} \|\mathrm{sign}(A^\top \mathbf{x}_i) - \mathrm{sign}(B^\top \mathbf{y}_i)\|^2 \\ =& \mathrm{tr}(\mathrm{sign}(P^\top Q)\, M_C\, \mathrm{sign}(Q^\top P)), \end{aligned} \tag{23}$$

where

$$M_C := \frac{\eta}{n}(I_{2n} - W_C), \quad W_C := \begin{bmatrix} 0 & I_n \\ I_n & 0 \end{bmatrix}. \tag{24}$$

By using Eq. (21) and Eq. (23), we can rewrite the total objective Eq. (14) as

$$\begin{aligned} &\lambda \mathcal{G}(X) + (1 - \lambda)\mathcal{G}(Y) + \eta C(X, Y) \\ &= \mathrm{tr}(\mathrm{sign}(P^\top Q)\, M_{\mathcal{G}}\, \mathrm{sign}(Q^\top P)) \\ &\quad + \mathrm{tr}(\mathrm{sign}(P^\top Q)\, M_C\, \mathrm{sign}(Q^\top P)) \\ &= \mathrm{tr}(\mathrm{sign}(P^\top Q)(M_{\mathcal{G}} + M_C)\mathrm{sign}(Q^\top P)). \end{aligned} \tag{25}$$

Similarly, the constraints Eqs. (15), (16) can be reformulated as:

$$\mathrm{sign}(P^\top Q)\mathrm{sign}(Q^\top P) = 2nI_c. \tag{26}$$

Here, we re-define $M := (M_{\mathcal{G}} + M_C)$, $A := P$, $X := Q$, and $n := 2n$, respectively. Finally, the total problem can be transformed into

$$\min_{A} \mathrm{tr}(\mathrm{sign}(A^\top X)\, M\, \mathrm{sign}(X^\top A)) \tag{27}$$

$$\text{s.t.: } \text{sign}(A^\top X)\text{sign}(X^\top A) = nI_c. \tag{28}$$

This problem is indeed equivalent to the uni-modal problem Eqs. (8), (9). Because of this property, the total problem can be optimally and efficiently solved in exactly the same way as in the uni-modal case discussed in Sect. 3.1. Specifically, we obtain the initial solution A of Eqs. (27), (28) by solving the relaxed problem Eqs. (10), (11) which is a generalized eigenproblem. The solution A corresponds to $P = [A^\top\ B^\top]^\top$ as in Eq. (20). Then, after obtaining the final A, we can generate binary codes for $\forall \mathbf{x} \in \mathcal{F}_x$ as $\mathbf{z}^{(x)} = \text{sign}(A^\top \mathbf{x})$.

Minimizing the quantization error. Because of the signed magnitude relaxation, the projection A is likely to incur significant binary quantization errors, leading to unsatisfactory retrieval performance. Following [4], we refine the initial projection A by solving the following orthogonal Procrustes problem to minimize the quantization error incurred by the relaxation.

$$\min_{Z,R} \|Z - X^\top AR\|_F^2 \tag{29}$$

$$\text{s.t. } R^\top R = I_c. \tag{30}$$

where $Z \in \{\pm 1\}^{n\times c}$ is the matrix of n binary codes and R is an orthonormal matrix of size $c \times c$. A local optimum can readily be obtained by alternating minimization between Z and R [4], [25], after which the projection can be refined as $A \leftarrow AR$. Note that this new projection does not change the optimality of the relaxed problem Eqs. (10), (11).

Observation 2. The objective value of Eq. (10) does not change as a result of the transformation into A with an arbitrary orthonormal matrix R.

Since $R^\top R = RR^\top = I_c$, by substituting $A = AR$ for Eq. (10), we get

$$\text{tr}((R^\top A^\top)XMX^\top(AR)) = \text{tr}(A^\top XMX^\top A). \tag{31}$$

Therefore, we can minimize the binary quantization error by using this post-processing refinement without losing the optimality with respect to geometric loss and multimodal loss. After obtaining the final solution $A \leftarrow AR$, we can generate c-bit codes for $\forall \mathbf{x} \in \mathcal{F}_x$ as $\mathbf{z} = \text{sign}(A^\top \mathbf{x})$. Hereafter, we refer to our method as Multimodal learning of Geometry-preserving Linear Projections (mGLP). Moreover, we call its uni-modal learning version (Eqs. (8), (9)) Uni-modal learning of Geometry-preserving Linear Projections (uGLP). We can say that mGLP is a multimodal learning extension of uGLP which has been investigated in our previous work [26].

3.4 Computational Complexity

Now let us analyze the computational complexity of our method. Overall, as the following discussion shows, its binary coding time is linear with respect to d_x; and its training time scales linearly in n and quadratically in d, where we define $d := d_x + d_y$ and $m := 2n$.

Binary coding. Our method is based on linear projections for binary coding in which an image feature $\mathbf{x}$ is encoded as $\mathbf{z} = \text{sign}(A^\top \mathbf{x})$. Hence, the time and space complexity for encoding is $O(cd_x)$ which is constant with respect to n and linear in d_x.

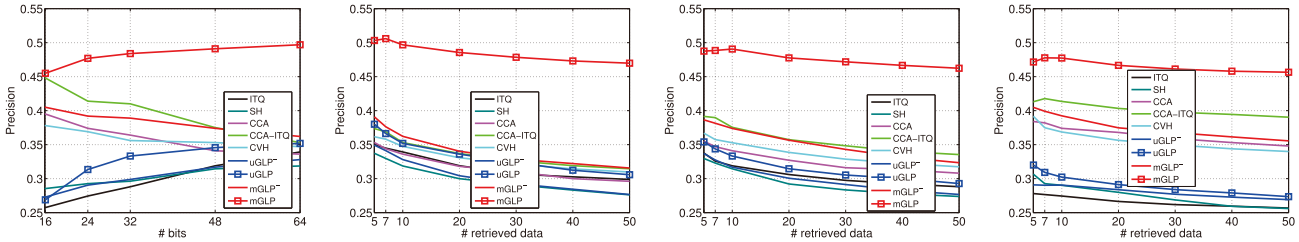
Training. The training stage consists of three steps: extracting linear reconstruction weights W for the two feature spaces by solving the locally linear sparse coding problem Eqs. (3), (4), optimizing the projection by solving Eqs. (27), (28), and refining the projections through the post-processing to minimize the quantization errors by solving Eqs. (29), (30). Solving the locally linear sparse coding takes on average $O(t^3 + |\mathcal{N}(\mathbf{x})|)$ by using a homotopy algorithm [23], where t is the number of non-zeros of $\mathbf{w}_i$ in Eq. (3) ($t \ll d$) and $|\mathcal{N}(\mathbf{x})|$ is the number of the candidate Euclidean neighbor points. An exhaustive Euclidean search typically takes $O(n)$ to collect the candidate $\mathcal{N}(\mathbf{x})$. However, we do not need the exact Euclidean neighbors, because the sparse coding itself has the ability to select only desirable points from $\mathcal{N}(\mathbf{x})$ (see Sect. 3.1); hence, $\mathcal{N}(\mathbf{x})$ can be collected using arbitrary approximate nearest neighbor search methods such as locality sensitive hashing (LSH) [27] in sub-linear time in n[†]. Moreover, this step can readily be parallelized using multiple CPUs or cores. Second, we solve Eqs. (10), (11) and Eqs. (29), (30). Note that M is very sparse and there are only $t \times t$ non-zero entries on average ($t \ll d$). Therefore, $XMX^\top$ and $XX^\top$ can be computed in $O((d^2+t^2)m)$. Moreover, the sparse eigenproblem can be efficiently solved by, for example, using the Lanczos method in $O(cdt)$ time [28]. Lastly, refining the initial projections A by Eqs. (29), (30) takes $O(mcd + c^2)$ time which is also linear in m. To this end, the total time required for training is $O((d^2 + cd + t^2)m + cdt + c^2) \sim O((d^2 + cd + t^2)n + cdt + c^2)$ which is linear in n and quadratic in d. The space complexity is $O((d + c + t)n + d^2 + c^2)$ which is also linear in n and quadratic in d.

4. Experiments

We experimentally analyze the retrieval performance of mGLP and uGLP in semantic image retrieval tasks using two popular benchmark datasets, a-Pascal [29] and COCO [30]. mGLP and uGLP obtain different binary coding results depending on whether or not each is coupled with post-processing for minimizing the quantization error. Thus, we evaluate both versions in our experiments. Below, we refer to the versions with the quantization error minimization as mGLP and uGLP and those without it as mGLP$^-$ and uGLP$^-$.

We compare our methods with several of the previous methods for unsupervised binary coding including uni-modal (ITQ [4] and SH [5]) and cross-modal ones (CCA [4],

[†]Typically, a few hundred Euclidean neighbor points collected with short LSH codes (say 16 bits) were enough in our experiments. There was no significant difference in retrieval quality from that of an exhaustive linear search.



(a) P vs #bits (b) 64 bits (c) 48 bits (d) 24 bits

Fig. 1 Retrieval performance on a-Pascal. Comparison with existing methods. (a) Precision of top 10 retrieved images vs. number of bits; Precision vs. number of retrieved images for (b) 64-, (c) 48-, and (d) 24-bit codes.

CCA-ITQ [4] and CVH [18]). To run these methods, we use the Matlab code provided by the authors of each method. Their parameters (if they have any) are tuned using a grid-search. Specifically, k for the k-nearest neighbor graphs used in CVH is set as the best one from $\{2, 3, 5, 10, 20\}$. For mGLP, λ is chosen from $[0, 1]$ and η is the best from $[0, 3]$.

We follow the common evaluation protocol used in the previous studies [4], [5]. We split each dataset into database and query sets. The database set is used to train binary codes for up to 64-bit codes and construct the database against which the queries are performed. We evaluate Hamming ranking performance; i.e., the retrieved images are sorted according to their Hamming distances to the query. This procedure is exhaustive but fast enough in practice [4]. Following [4], we measure the performance in terms of the precision of the top ranked images (averaged over the query set). The precision is measured using the semantic labels; i.e., we judge the retrieval to be successful if and only if the label of the retrieved item is the same as that of the query.

4.1 Datasets

We use the following benchmark datasets.

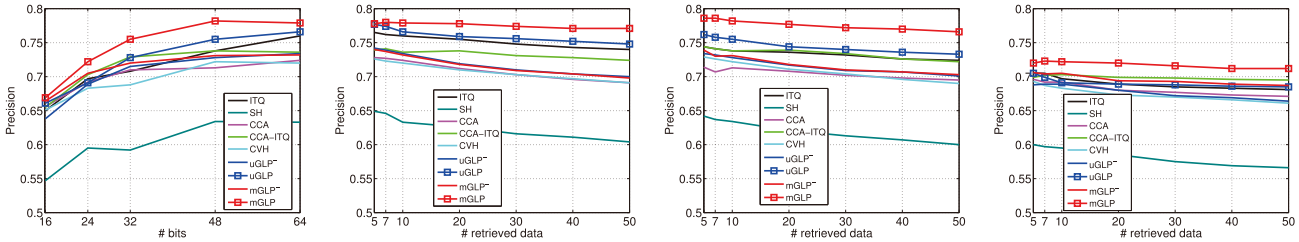
a-Pascal[†] [29]: a-Pascal contains 12,695 images in the 20 object categories defined in the Pascal VOC 2008 challenge (e.g., *people*, *dog*, and *car*). Following [29], we use multiple low-level features as our image feature; we first extract texture, HOG, edge, and color descriptors, which result in a 9751-dimensional vector, and reduce the dimensionality of the vector to 512 by using PCA. Each image is also associated with 64-dimensional binary attributes, each of which indicates a part or some semantic property of an object (e.g., *leg*, *wing*, and *2D-boxy*). We use these binary attributes as our text features. We randomly sample 500 images for the query set and use the rest for the database set.

COCO[††] [30]. Microsoft COCO v2014.1 is a large-scale image dataset that contains 123,558 images in 80 categories of objects. Each image is associated with five short sentences describing its content. In our experiments, we keep only the first sentence for each image. Our image feature is extracted by using the Caffe implementation[†††] of a convolutional neural network (CNN) called AlexNet [31]. Specifically, we extract 4,096-dimensional activation features from its *fc6* layer, then reduce their dimension to 128 by PCA, as is done in [32]. For the text feature, we use 300-dimensional skip-gram word vectors [33] learned by word2vec[††††] and compute the mean vector of the word vectors of the words appearing in each text description. We randomly sample 1,000 images for the query and use the rest to construct the training and database sets. In the dataset, each image is allowed to have multiple labels, therefore, we judge the retrieval is successful if a query and a retrieved image share at least one common label.

4.2 Results

Results on a-Pascal. Figure 1 compares the results of our approaches (mGLP, mGLP⁻, uGLP, and uGLP⁻) with the other methods (ITQ, SH, CCA, CCA-ITQ, and CVH). Figure 1 (a) gives the precision of the top 10 retrieved images for various code lengths, and Fig. 1 (b-d) plot the precisions for different numbers of retrieved images. We can make several observations about the results. First, mGLP is always superior to the other methods on this dataset. The gain of mGLP relative to the second best method, CCA-ITQ, is 40.7% for 64-bit codes. Second, some methods that learn binary codes using multimodal information sources, such as CCA, CCA-ITQ, CVH, mGLP and mGLP⁻, tend to yield better retrieval performance compared with the uni-modal learning approaches like ITQ, SH, uGLP and uGLP⁻. This clearly indicates that multimodal learning is important for improving binary codes for semantic image retrieval. Third, mGLP⁻ is consistently better than CCA and CVH. This highlights how effective our formulation, Eqs. (14)–(16), is compared with these approaches. In contrast, mGLP⁻ is slightly worse than CCA-ITQ. However, this difference mainly comes from the quantization error minimization in CCA-ITQ, which is vital to improving binary codes, especially for learning relatively longer codes [4]. As mentioned above, once it is coupled with the quantization error minimization, i.e., mGLP, its retrieval performance becomes su-

[†]http://vision.cs.uiuc.edu/attributes/
[††]http://mscoco.org/
[†††]http://caffe.berkeleyvision.org/
[††††]https://code.google.com/p/word2vec/



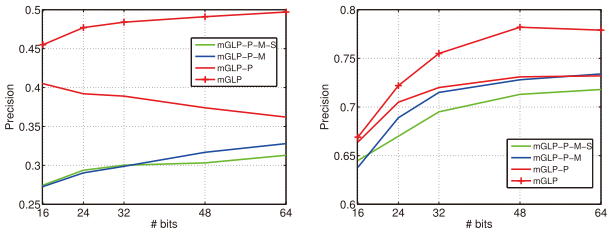
(a) P vs #bits (b) 64 bits (c) 48 bits (d) 24 bits

Fig. 2 Retrieval performance on COCO. Comparison with existing methods. (a) Precision of top 10 retrieved images vs. number of bits; Precision vs. number of retrieved images for (b) 64-, (c) 48-, and (d) 24-bit codes.

perior to CCA-ITQ. Lastly, uGLP$^-$ is always competitive to or better than the existing uni-modal learning methods such as ITQ and SH. This suggests that capturing the intrinsic structures of the data is essential for improving semantic image retrieval accuracies, and our geometric loss function is designed to achieve this goal.

Results on COCO. Figure 2 shows the results for COCO. Figure 2 (a) shows the precision of top 10 retrieved images for different code lengths, and Fig. 2 (b-d) show the precision for different numbers of retrieved images. Again, mGLP consistently performs better than the other methods. The gain of mGLP to the second best method, CCA-ITQ, is around 5.9% for 48-bit codes. Some of the other tendencies are similar to the a-Pascal case; mGLP is consistently better than CCA and CVH, and its performance is further boosted when it is coupled with the quantization error minimization. These results highlight the effectiveness of our formulation and the quantization error minimization. Comparing the results on a-Pascal, we can see that the differences between the uni-modal learning methods (ITQ, SH, uGLP, and uGLP$^-$) and the other methods which use multimodal information sources (CCA, CCA-ITQ, CVH, mGLP, and mGLP$^-$) are not so large. This may be because the image feature used in this dataset, the off-the-shelf CNN activation, is much more consistent with the semantic labels of the images [34], [35] than the hand-crafted features used in a-Pascal; hence, the binary codes learned by the uni-modal learning methods are already correlated to the semantic information of the images. Furthermore, our mGLP is always competitive with or better than the existing methods. This result emphasizes the consistency of our geometric loss function with the image semantics; our formulation preserves the semantics in the binary codes.

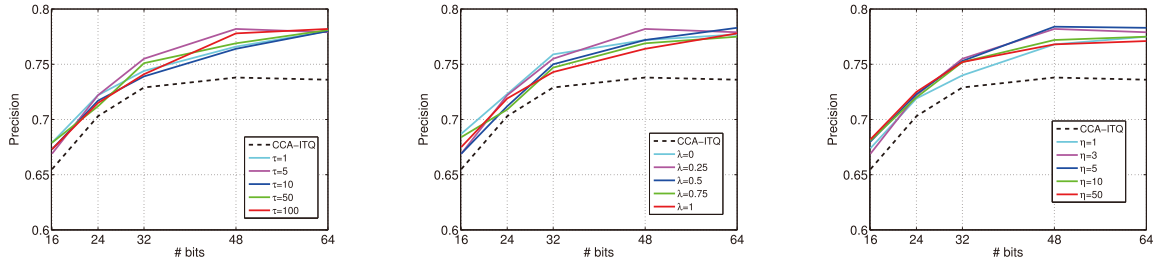
Impact of individual components. Now let us examine the impact of each component of mGLP on the retrieval performance. The main components of mGLP can be summarized as follows: (i) **S**parse coding: our method learns the intrinsic data structures W used in the geometric loss function by using locally linear sparse coding (Eqs. (3), (4)), instead of a least-squares approach as in NPE [24]; (ii) **M**ultimodal learning: our model considers not only the geometric loss but also multimodal loss to learn the projection A by solving a joint minimization problem (Eqs. (14)–(16); (iii) **P**rojection refinement: our method refines the initial A by solving Eqs. (29), (30) to minimize the quantization errors. We remove each of these components one-by-one from the full mGLP configuration and evaluate the resulting retrieval performance. The experiment compare the following four variants of mGLP: (1) **mGLP**: the full configuration; (2) **mGLP-P**: mGLP without the post-processing projection refinement, which is equivalent to mGLP$^-$; (3) **mGLP-P-M**: mGLP-P without multimodal learning, which reduces to uGLP$^-$; and (4) **mGLP-P-M-S**: mGLP-P-M without sparse coding. This can be done by taking the signs of real-valued vectors obtained by NPE. The results, shown in Fig. 3, indicate that mGLP clearly outperforms the other methods on both datasets; and that performance gradually decreases as components are removed. These results illustrate the importance of all of the components for learning semantic binary codes.



(a) a-Pascal (b) COCO

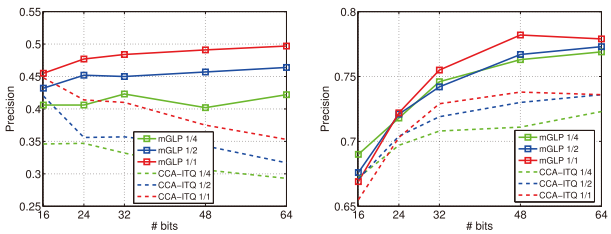
Fig. 3 Evaluation of main components. Precision of top 10 retrieved images vs. number of bits on (a) a-Pascal and (b) COCO.

Parameter studies. Our mGLP formulation has three parameters, τ, λ, and η (see Eq. (3) and Eq. (14). In Fig. 4, we analyze their impact on the retrieval performance. For comparison, we also show the results of CCA-ITQ which yields the best performance among the existing methods. One can see that the results are somewhat sensitive to these parameters, and fine tuning may improve the performance. However, these are better than CCA-ITQ for wide range of the parameter values. As for η, we examine the wide range of its value $\{1, 3, 5, 10, 50\}$. The precision value increases up to $\eta = 5$ and then slowly decreases as it becomes much larger. This indicates that multimodal learning is effective for improving semantic retrieval performance as long as it is well-balanced with the other two geometric loss terms, i.e.,



(a) τ (b) λ (c) η

Fig. 4 Parameter sensitivities of mGLP. Results on COCO. Precision of top 10 retrieved images vs. number of bits.



(a) a-Pascal (b) COCO

Fig. 5 Impact of the number of training data. Precision of top 10 retrieved images vs. number of bits on (a) a-Pascal and (b) COCO.

setting too large values for η may corrupt the meaningful geometric distributions of features, resulting in undesirable performance. Although the optimal value is around $\eta = 5$ in this setup, generally it may depend on how much the original image and text features are correlated with their semantic labels; hence, fine-tuning may improve performance.

Impact of the number of training data. The impact of the number of training data on the retrieval performance is analyzed by reducing the data (1/1) to 1/2 and 1/4 of its original size. CCA-ITQ is evaluated in the same setting for comparison. The results, shown in Fig. 5, indicate that the performance improved as the training data increases, which is natural behavior. mGLP performs better than CCA-ITQ, even with fewer number of the training data. For example, the results of mGLP with only 1/4 the training data are comparable to or better than those of CCA-ITQ with the full datasets.

Processing time. We recorded the empirical time taken for training and binary coding of the methods. All the results are obtained using MATLAB codes on a workstation equipped with a 2.6 GHz Intel Xeon CPU. The results for COCO are reported in Table 1. mGLP and uGLP take slightly longer to train than the other methods. However, the training takes only a few minutes, not much of a big problem in practice. Comparing mGLP with uGLP, one sees that mGLP is slower than uGLP because of the sparse coding used to train W. This process can be accelerated by using parallel computing on multiple CPUs or cores. As for the binary coding time, except for SH, which uses non-linear functions for coding, all of the compared methods are fast, as they are based on linear projections.

Table 1 Processing time on COCO dataset. Training time (sec) and binary coding time (msec).

	Training (sec)	Binary coding (msec)
ITQ	0.41	0.01
SH	0.20	0.05
CCA	0.53	0.01
CCA-ITQ	1.02	0.01
CVH	21.22	0.01
uGLP	33.66	0.01
mGLP	76.49	0.01

5. Conclusions

We presented an unsupervised learning method for feature binary coding which we call Multimodal learning of Geometry-preserving Projections (mGLP). We introduced a geometric loss function that can preserve the intrinsic structures of the data captured by sparse coding in binary codes. We considered a multimodal loss function that measures the distances between multimodal data pairs in the Hamming space and formulated a unified joint minimization problem of the geometric loss and multimodal loss in order to learn the binary codes. We showed that this problem can readily be transformed into a simple generalized eigenproblem and solved efficiently. By conducting an extensive set of experiments on two public datasets, we experimentally clarified the key properties of mGLP and demonstrated its superiority over several existing methods.

References

[1] T.K. Sellis, N. Roussopoulos, and C. Faloutsos, "The r+-tree: A dynamic index for multi-dimensional objects," VLDB endowments, pp.507–518, 1987.

[2] P. Indyk and R. Motwani, "Approximate nearest neighbors: Towards removing the curse of dimensionality," ACM Sympo. Theory of Computing (STOC), pp.604–613, 1998.

[3] 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, 2012.

[4] 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, 2013.

[5] Y. Weiss, A. Torralba, and R. Fergus, "Spectral hashing," Conf. Neural Information Processing Systems (NIPS), pp.1753–1760, 2008.

[6] W. Liu, J. Wang, S. Kumar, and S. Chang, "Hashing with graphs," Int. Conf. Machine Learning (ICML), pp.1–8, 2011.
[7] F. Shen, C. Shen, Q. Shi, A. van den Hengel, and Z. Tang, "Inductive hashing on manifolds," Conf. Computer Vision and Pattern Recognition (CVPR), pp.1562–1569, 2013.
[8] J.-P. Heo, Y. Lee, J. He, S.-F. Chang, and S.-E. Yoon, "Spherical hashing: Binary code embedding with hyperspheres," IEEE Trans. Pattern Anal. Mach. Intell., vol.37, no.11, pp.2304–2316, 2015.
[9] K. He, F. Wen, and J. Sun, "K-means hashing: An affinity-preserving quantization method for learning binary compact codes," Conf. Computer Vision and Pattern Recognition (CVPR), pp.2938–2945, 2013.
[10] H. Murase and S.K. Nayar, "Visual learning and recognition of 3-d objects from appearance," Int. J. Comput. Vision., vol.14, no.1, pp.5–24, 1995.
[11] D. Zhou, J. Weston, A. Gretton, O. Bousquet, and B. Schölkopf, "Ranking on data manifolds," Conf. Neural Information Processing Systems (NIPS), pp.169–176, 2003.
[12] K.Q. Weinberger and L.K. Saul, "Unsupervised learning of image manifolds by semidefinite programming," Int. J. Comput. Vision., vol.70, no.1, pp.77–90, 2006.
[13] G. Irie, Z. Li, X.-M. Wu, and S.-F. Chang, "Locally linear hashing for extracting non-linear manifolds," Conf. Computer Vision and Pattern Recognition (CVPR), pp.2123–2130, 2014.
[14] M. Belkin and P. Niyogi, "Laplacian eigenmaps for dimensionality reduction and data representation," Neural Comput., vol.15, no.6, pp.1373–1396, 2003.
[15] L. van der Maaten and G. Hinton, "Visualizing data using t-sne," J. Machine Learning Research (JMLR), vol.9, pp.2579–2605, 2008.
[16] J. Song, Y. Yang, Z. Huang, H.T. Shen, and R. Hong, "Multiple feature hashing for real-time large scale near-duplicate video retrieval," Proc. 19th ACM international conference on Multimedia - MM '11, pp.423–432, 2011.
[17] X. Liu, J. He, D. Liu, and B. Lang, "Compact kernel hashing with multiple features," Proc. 20th ACM international conference on Multimedia - MM '12, pp.881–884, 2012.
[18] S. Kumar and R. Udupa, "Learning hash functions for cross-view similarity search," Int. Joint Conf. Artificial Intelligence (IJCAI), pp.1360–1365, 2011.
[19] M.M. Bronstein, A.M. Bronstein, F. Michel, and N. Paragios, "Data fusion through cross-modality metric learning using similarity-sensitive hashing," Conf. Computer Vision and Pattern Recognition (CVPR), pp.3594–3601, 2010.
[20] J. Song, Y. Yang, Y. Yang, Z. Huang, and H.T. Shen, "Inter-media hashing for large-scale retrieval from heterogenous data sources," Proc. 2013 international conference on Management of data - SIGMOD '13, pp.785–796, 2013.
[21] G. Irie, H. Arai, and Y. Taniguchi, "Alternating co-quantization for cross-modal hashing," Int. Conf. Computer Vision (ICCV), pp.1886–1894, 2015.
[22] E. Elhamifar and R. Vidal, "Sparse manifold clustering and embedding," Conf. Neural Information Processing Systems (NIPS), pp.55–63, 2011.
[23] D.L. Donoho and Y. Tsaig, "Fast solution of l_1-norm minimization problems when the solution may be sparse," IEEE Trans. Inf. Theory, vol.54, no.11, pp.4789–4812, 2008.
[24] X. He, D. Cai, S. Yan, and H.-J. Zhang, "Neighborhood preserving embedding," Int. Conf. Computer Vision (ICCV), vol.1, pp.1208–1213, 2005.
[25] S.X. Yu and J. Shi, "Multiclass spectral clustering," Int. Conf. Computer Vision (ICCV), vol.1, pp.313–319, 2003.
[26] G. Irie, H. Arai, and Y. Taniguchi, "Hashing with locally linear projections," IEICE Trans. Inf. Syst. (Japanese Edition), vol.J97-D, no.12, pp.1785–1796, Dec. 2014.
[27] M.S. Charikar, "Similarity estimation techniques from rounding algorithms," ACM Sympo. Theory of Computing (STOC), pp.380–388, 2002.
[28] G.W. Stewart, Matrix Algorithms Volume II, SIAM, 2001.
[29] A. Farhadi, I. Endres, D. Hoiem, and D.A. Forsyth, "Describing objects by their attributes," Conf. Computer Vision and Pattern Recognition (CVPR), pp.1778–1785, 2009.
[30] T.-Y. Lin, M. Maire, S. Belongie, J. Hays, P. Perona, D. Ramanan, P. Dollár, and C.L. Zitnick, "Microsoft COCO: Common objects in context," Europ. Conf. Computer Vision (ECCV), vol.8693, pp.740–755, 2014.
[31] A. Krizhevsky, I. Sutskever, and G.E. Hinton, "Imagenet classification with deep convolutional neural networks," Conf. Neural Information Processing Systems (NIPS), pp.1106–1114, 2012.
[32] A. Babenko, A. Slesarev, A. Chigorin, and V. Lempitsky, "Neural codes for image retrieval," Europ. Conf. Computer Vision (ECCV), pp.584–599, 2014.
[33] T. Mikolov, I. Sutskever, K. Chen, G.S. Corrado, and J. Dean, "Distributed representations of words and phrases and their compositionality," Conf. Neural Information Processing Systems (NIPS), pp.3111–3119, 2013.
[34] J. Donahue, Y. Jia, O. Vinyals, J. Hoffman, N. Zhang, E. Tzeng, and T. Darrell, "Decaf: A deep convolutional activation feature for generic visual recognition," Int. Conf. Machine Learning (ICML), pp.647–655, 2014.
[35] A.S. Razavian, H. Azizpour, J. Sullivan, and S. Carlsson, "CNN features off-the-shelf: An astounding baseline for recognition," Workshop on Computer Vision and Pattern Recognition (CVPR Workshops), pp.512–519, 2014.

Go Irie received the BS and MS degrees in system engineering from Keio University, Japan, in 2004 and 2006, respectively, and the PhD degree in information science and technology from the University of Tokyo, Japan, in 2011. He is currently a Research Engineer at NTT Corporation, Japan. He was a visiting research scholar at Columbia University from 2012 to 2013. His current research interests include multimedia search and multimodal analysis. He is a member of IEICE.

Hiroyuki Arai received the M.S. degree in physics and the Ph.D. degree in information science from Hokkaido University, Sapporo, Japan, in 1991 and 2009, respectively. He jointed NTT Corporation, Kanagawa, Japan, in 1991, where he was involved in research on map recognition systems. He moved to NTT Data Corporation in 2001, where he was involved in developing image processing technologies from 2001 to 2005. He is currently a Senior Research Engineer with the NTT Media Intelligence Laboratories, NTT Corporation. He is a member of ITE and IEICE.

Yukinobu Taniguchi received the B.E., M.E., and Dr.Eng. degrees in mathematical engineering from the University of Tokyo in 1990, 1992, and 2002, respectively. He was with NTT Corporation from 1992 to 2015. He is currently a professor of Tokyo University of Science, Japan. His research interests include image/video processing and multimedia applications. He is a member of ACM, IPSJ, ITE, and a senior member of IEICE.