Multi-view Cross-media Hashing with Semantic Consistency

Ruoyu Liu*, Shikui Wei*, Yao Zhao*, Zhenfeng Zhu*, and Jingdong Wang†
*Beijing Jiaotong University †Microsoft Research Asia

***Abstract*—We employ the cross-media hashing to handle both the cross-media representation and indexing simultaneously. Most existing methods attempt to bridge the semantic gap by maximizing the correlation of the heterogeneous instances describing the same information object. Although these methods guarantee that the heterogeneous instances of the same object are close in the commonly shared space, those belonging to different objects but the same category may be scattered. We propose a new cross-media hashing scheme named Multi-view Cross-Media Hashing with Semantic Consistency (MCMHSC) to address this problem. By fully exploiting the semantic correlation and complementary information among objects, the proposed scheme builds discriminative hashing codes. Experiments on two public benchmarks demonstrate the good performance in terms of search accuracy and time complexity.**

***Index Terms*—Multi-view, Cross-media, Hashing**

I. Introduction

Searching [1] is a basic manner of people to find out the needed information from a huge amount of data, which is widely applied to many applications. In the last two decades, many works have been reported towards improving search accuracy and reducing search time. However, most of them are essentially the single-media retrieval, *i.e.*, performing a search on the information objects carried by the same media type (*e.g.*, image). Recently, web users are becoming the main body of generating information contents. Since no unique rules are followed by them, the structure of the contents is informal and heterogeneous. For example, when a user publish an information object (*e.g.*, a record of a daily story) on Facebook, he may represent it by combining text, image, and video. Clearly, this object crosses multiple media types that share the same content. The goal of the cross-media research is to bridge the heterogeneous gap between different media types. As a hot point, cross-media retrieval has attracted much attention in recent years. It helps users to directly measure the similarity among heterogeneous data.

Due to the massive scale of web data, it is important for cross-media retrieval to perform searches efficiently. An effective way to speed up retrieve is hashing, which solves an approximate nearest neighbor (ANN) search problem. However, ANN cannot be directly obtained for cross-media retrieval since information objects cross multiple media types. Besides, most of the existing hashing methods are not applicable to cross-media retrieval. Therefore, cross-media hashing should be specially conducted.

To facilitate the discussions, we clarify some terms used in this paper. An information object is an entity that describes a certain semantic meaning (*e.g.*, an event), which can be carried by a single media type (*e.g.*, image) or several media types (*e.g.*, text and image). Fig. 1(a) and Fig. 1(b) illustrate a single-media and a multi-media cases respectively. For the instances of the same media type, they can be represented in a unique homogeneous feature space (single modality) or several heterogeneous feature spaces (multiple modalities). For example, an image can be represented by both SIFT and SURF. For the instances of different media types, they are heterogeneous whenever they are represented by single-modal or multi-modal features. For example, although image and video can be both represented by SURF, they are heterogeneous. In this paper, we claim that instances of the same media type are represented in a homogeneous feature space. To distinguish cross-media hashing from traditional schemes, we classify the hashing approaches into three categories.

Fig. 1. Two examples of information objects: (a) The information object is carried by an image, which describes 'a tiger'. (b) The information object describes an event about 'the Chinese food in APEC', which is carried by both an image and a text document.

- **uni-modal hashing**: Information objects are carried by the instances of a single media type, and all the instances are represented in a homogeneous feature space. The goal of uni-modal hashing is to learn hash functions to project homogeneous features into compact hash codes. (For example, LSH [2])
- **multi-modal hashing**: Information objects are carried by the instances of a single media type, but each instance is represented in several heterogeneous feature spaces. The goal of multi-modal hashing is to learn hash functions to individually project heterogeneous features into a shared and compact binary space. (For example, MFH [3])
- **cross-media hashing**: Information objects are carried by the instances of multiple media types, but the instances of

the same media type are represented in a homogeneous feature space. The goal of cross-media hashing is to learn hash functions to preserve the inter-media similarities. The similarities between the instances of different media types are measurable. (For example, CAMH [4])

In this paper, we focus on cross-media hashing and propose a new method named Multi-view Cross-Media Hashing with Semantic Consistency (MCMHSC), which is an extension of our previous work [4]. The core idea lies in that we treat the category as an independent view and introduce it into maximizing the correlation of heterogeneous instances. This constraint enforces that the instances of different objects but the same category locate near in the shared space (intra-category correlation). The key novelties of the proposed scheme are summarized as follows:

(1) A novel cross-media hashing based on three-view CCA [5] is proposed. By introducing categories as a third media type, the performance of hash codes is improved, since the heterogeneous instances of the same category are closer in the learned space. Another merit of the proposed scheme lies in that its time complexity is free from code length since the optimization is based on a generalized eigenvalue problem.
(2) MCMHSC is easily extended to more views. The extension of the method is simple and straightforward, and its formulation is also given explicitly in this paper. However, most of the previous works mainly focus on two media types, which are not easy to extend.
(3) A simple but effective fusion algorithm is proposed to generate a unique binary code for an object with multiple views (or modalities). It fully exploits the complementary information from multiple views to encode the semantic content.

The rest of the paper is organized as follows: In Section II, we review the previous works of cross-media hashing. In Section III, we present the details of the proposed scheme. Experimental results on two public benchmarks are illustrated and analyzed in Section IV. Finally, the conclusions are given.

II. Related Works of Cross-media Hashing

The target of cross-media hashing is to perform fast retrieval with limited loss of accuracy. When datasets are large, the search speed decreases rapidly since the brute-force search is usually performed. To address this issue, cross-media hashing projects heterogeneous instances into a shared binary space and uses the fast Hamming distance to measure the similarities. It not only improves search speed but also saves storage.

The problem of cross-media hashing was firstly studied by Bronstein *et al.* in cross-modal similarity sensitive hashing (CMSSH) [6], which is a Boosting algorithm. Cross-view hashing (CVH) [7] extends spectral hashing to preserve the intra-media and inter-media similarities simultaneously. Multimodal latent binary embedding (MLBE) [8] employs a probabilistic generative model to encode the homogeneous and heterogeneous similarities. Linear cross-modal hashing (LCMH) [9] obtains hash codes via thresholding the distance between data points and cluster centroids. Collective matrix factorization hashing (CMFH) [10] learns hash codes with latent factor model from different media types, and latent semantic sparse hashing (LSSH) [11] captures high-level semantic information, *e.g.*, sparse coding and matrix factorization, to improve search performance. In both of them, canonical correlation analysis (CCA) [12] is employed to preserve the inter-media similarity.

However, all the above schemes only take pairwise correlation into consideration, but category information is not considered. In many real-world applications, however, category information is available. For example, Flickr allows users to label their photos with several words. The photos labeled as the same category (word) are semantically correlative. To involve the categories into the training procedure, some cross-media hashing methods are proposed to preserve the category-level similarity. Semantic correlation maximization (SCM) [13] aims to make the distance of hash codes equal to the similarity of label vectors. Centroid approaching cross-media hashing (CAMH) [4] proposes a quadrangle model which introduces category information by calculating category centroids. Semantics-preserving hashing (SePH) [14] minimizes the KL-divergence between the probability distribution of hash codes and the one learned from the semantic affinities of training data.

Most of the recent works also employ deep learning to learn hash functions. Masci *et al.* proposed a multi-modal similarity-preserving hashing based on the coupled Siamese neural network [15]. Deep multimodal hashing with orthogonal regularization (DMHOR) [16] takes multi-modal and cross-modality encoders to preserve intra- and inter-modality correlations. Deep cross-modal hashing (DCMH) [17] is an end-to-end learning framework of deep neural networks.

III. Cross-Media Hashing with Semantic Consistency

In this section, we describe the details of the proposed method. We use boldface uppercase, boldface lowercase and letter to denote the matrices, vectors, and scales respectively. In addition, the transpose of $\mathbf{X}$ is denoted as $\mathbf{X}^T$ and the inverse of $\mathbf{X}$ is denoted as $\mathbf{X}^{-1}$.

A. Problem Description

Assume we have N information objects, each object is carried by a pair of heterogeneous instances from different media types: $\{x_i^{(1)}, x_i^{(2)}\}_{i=1}^N$, where $x_i^{(1)} \in R^{D(1)}$ and $x_i^{(2)} \in R^{D(2)}$. For example, $x_i^{(1)}$ can be the SIFT feature extracted from an image, and $x_i^{(2)}$ can be the latent Dirichlet allocation (LDA) feature extracted from a text document.

Our goal is to project heterogeneous instances into a shared binary space, in which both the intra-media and inter-media similarities can be directly measured. For the case of two media types (or modalities), the key is to learn two hash functions:

$$\begin{aligned} h^{(1)} &: R^{D(1)} \longmapsto \{-1,1\}^L \\ h^{(2)} &: R^{D(2)} \longmapsto \{-1,1\}^L \end{aligned} \tag{1}$$

where $\{-1,1\}^L$ is a shared binary space of L dimensions. In the space, the heterogeneous instances (*e.g.*, images and texts)

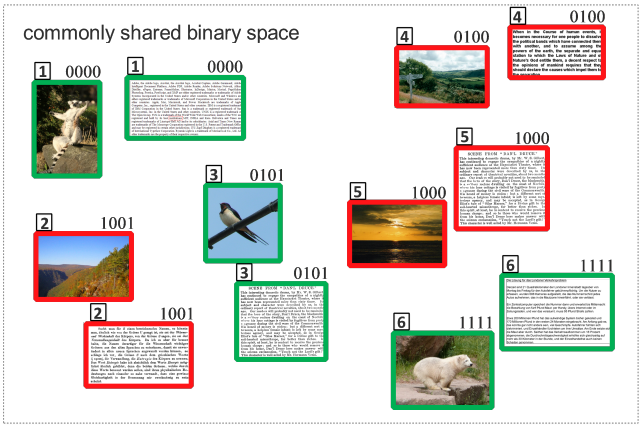

Fig. 2. Illustration of maximizing the correlation of the visual instance $x_i^{(1)}$ and the textual instance $x_i^{(2)}$ of the same information object.

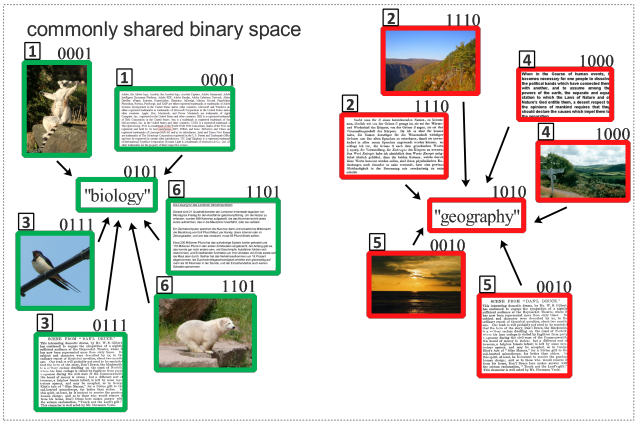

Fig. 3. Illustration of minimizing the distance between the visual instance $x_i^{(1)}$ and textual instance $x_i^{(2)}$ of the same information object, and the distance between an instance and its associated category.

are directly measured by the Hamming distance. The binary problem is often relaxed into a real-valued case, then the goal is changed to firstly learn two mapping functions:

$$f^{(1)} : R^{D(1)} \longmapsto R^L, \quad f^{(2)} : R^{D(2)} \longmapsto R^L \tag{2}$$

and then binarize the real-valued vectors.

Generally, most of the existing methods are to maximize the correlation between $x_i^{(1)}$ and $x_i^{(2)}$ as shown in Fig. 2. In this way, the heterogeneous instances of the same object are close in the shared binary space. However, the heterogeneous instances of different objects but the same category may be scattered. We attempt to address the issue by introducing categories into the learning procedure.

B. Formulation

Assume that each object is labeled by M categories. The key idea of the MCMHSC method is to treat the categories as the third view and introduce it into the learning procedure. It can be formulated as the following optimization problem:

$$\begin{aligned} \min_{h^{(1)},h^{(2)},h^{(3)}} & \|\mathbf{B}^{(1)}-\mathbf{B}^{(2)}\|_F^2 + \|\mathbf{B}^{(1)} - \mathbf{B}^{(3)}\|_F^2 \\ & + \|\mathbf{B}^{(2)} - \mathbf{B}^{(3)}\|_F^2 \\ s.t., \quad & \mathbf{B}^{(i)^T} e = 0, \quad b(i) \in \{-1, 1\}, \\ & \frac{1}{N}\mathbf{B}^{(i)^T}\mathbf{B}^{(i)} = \mathbf{I}_L, i = 1, 2, 3 \end{aligned} \tag{3}$$

where $\|.\|_F$ is the Frobenius norm, e is a $N \times 1$ vector whose entries are all 1 and $\mathbf{I}_L$ is an $L \times L$ identity matrix. $\mathbf{B}^{(1)}, \mathbf{B}^{(2)}, \mathbf{B}^{(3)} \in R^{N \times L}$, whose rows represent the hash codes of heterogeneous instances $x^{(1)}, x^{(2)}$ and categories $x^{(3)}$ respectively. The constraint $\mathbf{B}^{(i)^T} e = 0$ requires each bit has equal chance to be -1 or 1. $\frac{1}{N}\mathbf{B}^{(i)^T}\mathbf{B}^{(i)} = \mathbf{I}_L$ requires that each bit is obtained independently.

The first term of Eq. 3 minimizes the distance between heterogeneous instance pair ($x_i^{(1)}$ and $x_i^{(2)}$) of the same information object. The second and third terms minimize the distance between heterogeneous instances and categories. If the second and third terms are removed, MCMHSC degenerates to CCA that only preserves pairwise correlation between heterogeneous instances.

To introduce the categories, the second and the third items are added into the objective function. Here, MCMHSC treats categories as an independent view and learns a hash function for it as well. In this way, the instances of the same category will approach to the same target. Fig. 3 illustrates the optimizing process. To the best of our knowledge, this idea is firstly introduced to cross-media hashing, and the similar idea has been applied in the three-view CCA [5] for consistent representation.

C. Optimization

The optimization problem in Eq. 3 is equivalent to the balanced graph partition issue, which is NP hard. We relax it to a real-valued case, which is changed to learn three linear functions:

$$\begin{aligned} f^{(1)}(z_i^{(1)}) &= \mathbf{W}^{(1)^T} z_i^{(1)} \\ f^{(2)}(z_i^{(2)}) &= \mathbf{W}^{(2)^T} z_i^{(2)} \\ f^{(3)}(z_i^{(3)}) &= \mathbf{W}^{(3)^T} z_i^{(3)} \end{aligned} \tag{4}$$

where $\mathbf{W}^{(1)}, \mathbf{W}^{(2)} \in R^{K \times L}$, $\mathbf{W}^{(3)} \in R^{M \times L}$ are three linear projection matrices. $z_i^{(1)}, z_i^{(2)} \in R^K$ are the feature representation of $x_i^{(1)}, x_i^{(2)}$, which are individually obtained by concatenating their distances to K cluster centroids. $z_i^{(3)} \in \{0, 1\}^M$ is the binary vector representation of categories, in which the entries of the labeled categories are 1 and others are 0.

Then we can rewrite Eq. 3 to:

$$\begin{aligned} \min_{\mathbf{W}^{(1)},\mathbf{W}^{(2)},\mathbf{W}^{(3)}} & \|\mathbf{Z}^{(1)}\mathbf{W}^{(1)} - \mathbf{Z}^{(2)}\mathbf{W}^{(2)}\|_F^2 \\ & + \|\mathbf{Z}^{(1)}\mathbf{W}^{(1)} - \mathbf{Z}^{(3)}\mathbf{W}^{(3)}\|_F^2 \\ & + \|\mathbf{Z}^{(2)}\mathbf{W}^{(2)} - \mathbf{Z}^{(3)}\mathbf{W}^{(3)}\|_F^2 \\ s.t., \quad & \frac{1}{N}\mathbf{W}^{(i)^T}\mathbf{Z}^{(i)^T}\mathbf{Z}^{(i)}\mathbf{W}^{(i)} = \mathbf{I}_L; i = 1, 2, 3 \end{aligned} \tag{5}$$

where $\mathbf{Z}^{(1)}, \mathbf{Z}^{(2)}, \mathbf{Z}^{(3)}$ are three feature matrices and each row of them is a sample of $z_i^{(1)}, z_i^{(2)}, z_i^{(3)}$.

Eq. 5 has the same form as the three-view CCA, which can be reduced to the following generalized eigenvalue problem:

$$\begin{pmatrix} \Sigma_{11} & \Sigma_{12} & \Sigma_{13} \\ \Sigma_{21} & \Sigma_{22} & \Sigma_{23} \\ \Sigma_{31} & \Sigma_{32} & \Sigma_{33} \end{pmatrix} \begin{pmatrix} w_1 \\ w_2 \\ w_3 \end{pmatrix} = \lambda \begin{pmatrix} \Sigma_{11} & 0 & 0 \\ 0 & \Sigma_{22} & 0 \\ 0 & 0 & \Sigma_{33} \end{pmatrix} \begin{pmatrix} w_1 \\ w_2 \\ w_3 \end{pmatrix} \tag{6}$$

where Σ_{ij} is the covariance matrix between the i^{th} and j^{th} media types and w_i is a column of $\mathbf{W}^{(i)}$. Then $\mathbf{W}^{(1)}, \mathbf{W}^{(2)}, \mathbf{W}^{(3)}$ are calculated as follows:

$$\begin{aligned} \mathbf{W}^{(1)} = \mathbf{W}(1:K,:), \mathbf{W}^{(2)} = \mathbf{W}(K+1:2K,:), \\ \mathbf{W}^{(3)} = \mathbf{W}(2K+1:end,:) \end{aligned} \tag{7}$$

where $\mathbf{W}$ is constructed by the eigenvectors of the L largest eigenvalues in Eq. 6.

D. Binarization

After obtaining the three functions in Eq. 4, we can easily project heterogeneous features into a shared continue space. Then, the next step is binarization to hash codes. We employ a similar strategy as in [9]. Firstly, we relax $\mathbf{B}^{(i)}$ into its real-valued form $\mathbf{Y}^{(i)}$, which is calculated as follows:

$$\mathbf{Y}^{(i)} = \mathbf{Z}^{(i)}\mathbf{W}^{(i)} \tag{8}$$

where $i = 1, 2, 3$. Then we calculate the binarization threshold using the $mean$ function:

$$u^{(i)} = mean(\mathbf{Y}^{(i)}) \tag{9}$$

where $u^{(i)} \in R^L$.

Finally, we binarize $\mathbf{Y}^{(i)}$ as follows:

$$\begin{cases} b_{jk}^{(i)} = 1 & if \quad y_{jk}^{(i)} \geq u_k^{(i)} \\ b_{jk}^{(i)} = -1 & if \quad y_{jk}^{(i)} < u_k^{(i)} \end{cases} \tag{10}$$

where $\mathbf{Y}^{(i)} = [y_1^{(i)}, ..., y_N^{(i)}]^T$, $i = 1, 2, 3$, $j = 1, ..., N$ and $k = 1, ..., L$. (j is the index of the instances, k is the index of the elements of $y^{(i)}$ and $b^{(i)}$, and $y^{(i)}$ is the real-valued relaxation of $b^{(i)}$.)

E. Fusion

The complementary information from multiple views can be employed to further improve the discriminative capability of hash codes. Inspired by [14], we propose a simple but effective fusion algorithm to generate a unique binary code for an object with multiple views (or modalities). After $\mathbf{Y}^{(i)}$ and $u^{(i)}$ are calculated, we can use the sigmoid function to calculate approximate probability that $b_{jk}^{(i)}$ equals to 1 or -1:

$$\begin{aligned} p(b_{jk}^{(i)} = 1) &= \frac{1}{1 + e^{-(y_{jk}^{(i)} - u_k^{(i)})}} \\ p(b_{jk}^{(i)} = -1) &= 1 - p(b_{jk}^{(i)} = 1) \end{aligned} \tag{11}$$

Then Eq. 10 has the following equivalent form:

$$\begin{cases} b_{jk}^{(i)} = 1 & if \quad p(b_{jk}^{(i)} = 1) \geq p(b_{jk}^{(i)} = -1) \\ b_{jk}^{(i)} = -1 & if \quad p(b_{jk}^{(i)} = 1) < p(b_{jk}^{(i)} = -1) \end{cases} \tag{12}$$

We fuse the two probabilities calculated from $\mathbf{Y}^{(1)}$ and $\mathbf{Y}^{(2)}$ to generate a uniform hash code matrix $\mathbf{B}$ for both modalities. The probability that b_{jk} equals to 1 or -1 is set to the maximal value of the two heterogeneous views:

$$\begin{aligned} p(b_{jk} = 1) &= \max(p(b_{jk}^{(1)} = 1), \quad p(b_{jk}^{(2)} = 1)) \\ p(b_{jk} = -1) &= \max(p(b_{jk}^{(1)} = -1), \ p(b_{jk}^{(2)} = -1)) \end{aligned} \tag{13}$$

Then $\mathbf{B}$ is calculated in a similar way of Eq. 12:

$$\begin{cases} b_{jk} = 1 & if \quad p(b_{jk} = 1) \geq p(b_{jk} = -1) \\ b_{jk} = -1 & if \quad p(b_{jk} = 1) < p(b_{jk} = -1) \end{cases} \tag{14}$$

where $j = 1, ..., N$ and $k = 1, ..., L$.

F. Extension

MCMHSC can be easily extended to more than two media types. In that case, the objective function in Eq. 3 is changed to:

$$\begin{aligned} \min_{\mathbf{B}^{(i)}, i=1,...,P+1} & \sum_{i=1}^{P+1} \sum_{i<j}^{P+1} \|\mathbf{B}^{(i)} - \mathbf{B}^{(j)}\|_F^2 \\ s.t., \quad & \mathbf{B}^{(i)^T} e = 0, \quad b(i) \in \{-1, 1\}, \\ & \frac{1}{N} \mathbf{B}^{(i)^T} \mathbf{B}^{(i)} = \mathbf{I}_L, i = 1, ..., P+1 \end{aligned} \tag{15}$$

where P is the number of media types and the categories are treated as the $(P+1)^{\text{th}}$ media type.

Eq. 15 can be relaxed to:

$$\begin{aligned} \min_{\mathbf{W}^{(i)}, i=1,...,P+1} & \sum_{i=1}^{P+1} \sum_{i<j}^{P+1} \|\mathbf{Z}^{(i)}\mathbf{W}^{(i)} - \mathbf{Z}^{(j)}\mathbf{W}^{(j)}\|_F^2 \\ s.t., \quad & \frac{1}{N} \mathbf{W}^{(i)^T} \mathbf{Z}^{(i)^T} \mathbf{Z}^{(i)} \mathbf{W}^{(i)} = \mathbf{I}_L, \quad i = 1, ..., P+1 \end{aligned} \tag{16}$$

which can be reduced to a generalized eigenvalue problem:

$$\begin{pmatrix} \Sigma_{11} & \Sigma_{12} & ... & \Sigma_{1,P+1} \\ \Sigma_{21} & \Sigma_{22} & ... & \Sigma_{2,P+1} \\ ... & ... & ... & ... \\ \Sigma_{P+1,1} & \Sigma_{P+1,2} & ... & \Sigma_{P+1,P+1} \end{pmatrix} \begin{pmatrix} w_1 \\ w_2 \\ ... \\ w_{P+1} \end{pmatrix} = \lambda \begin{pmatrix} \Sigma_{11} & 0 & ... & 0 \\ 0 & \Sigma_{22} & ... & 0 \\ ... & ... & ... & ... \\ 0 & 0 & ... & \Sigma_{P+1,P+1} \end{pmatrix} \begin{pmatrix} w_1 \\ w_2 \\ ... \\ w_{P+1} \end{pmatrix} \tag{17}$$

where Σ_{ij} is the covariance matrix between the i^{th} and j^{th} media types.

IV. Experimental Analysis

In this section, we conduct empirical studies on cross-media hashing. Two popular benchmark datasets are employed, and each dataset is divided into two sets, *i.e.*, a database set (also used for training) and a query set.

A. Experimental Setup

In our experiments, Wiki and NUS-WIDE are employed as the benchmarks and Mean Average Precision (mAP) is used for measuring the accuracy of the hash codes. Time cost in the off-line and on-line phases is used for measuring the time complexity.

Wiki is generated from a group of 2,866 Wikipedia documents (or information objects). Each document is an image-text pair and is uniquely tagged with one of 10 labels. The images are represented by 128-dimensional SIFT histograms, and the text articles are represented by a probability distribution over 10 topics (10-dimensional LDA feature). It contains 2,173 data points as the database set and the other 693 data points as the query set. The features are provided with the dataset.

NUS-WIDE is downloaded from Flickr, which includes 269,648 images with associated tags. Here, each image and its tags together are treated as an information object. In addition to tags, each image is also assigned several categories. We retain a part of the samples of the top 20 most frequent categories, which contains 17,600 image-tags pairs as the database set and 7,040 pairs as the query set. Each image is represented by a 500-dimensional bag of visual words. The tags associated with an image together are represented as a 1000-dimensional tag vector. The features are provided with the dataset as well.

Mean Average Precision (mAP) is employed as the accuracy measure. Given a query and a set of R retrieved results, the value of its Average Precision (AP) is defined as:

$$\text{AP} = \frac{1}{l}\sum_{r=1}^{R} P(r)\delta(r) \tag{18}$$

where l is the number of true positives in the retrieved set. $P(r)$ denotes the precision of the top r retrieved documents, and $\delta(r) = 1$ if the r^{th} retrieved document is a true positive and $\delta(r) = 0$ otherwise. We then average the AP values over all the queries to obtain the mAP score. A larger mAP score indicates a better accuracy. In the experiments, we set $R = 50$, which is consistent with the setting in [8].

Time cost in the off-line and on-line phases is employed to measure the time complexity. The time cost in the off-line phase contains the time used for training and computing the hash codes of the database set. The time cost in the on-line phase contains the time spent on computing the hash codes of the query set and calculating the Hamming distances for cross-media retrieval.

B. Compared Methods

Seven previous methods are fully tested and compared with the proposed scheme, which include CMSSH [6], CVH [7], LCMH [9], CMFH [10], SCM [13] (SCM-Seq and SCM-Orth), LSSH [11] and SePH [14] (SePH$_{rnd}$ and SePH$_{km}$). The source codes of these methods except for LCMH are published by their authors. We implement LCMH according to the description in [9]. We also consider a baseline that generates hash codes from classifiers [18], which is named trivial solution hashing (TSH) in this paper.

Similar to the previous works, we evaluate all the methods on two retrieval tasks. One is to use a text query to search the relevant images (shorted for 'Text query *vs.* Image data'). The other is to use an image query to search the relevant texts (shorted for 'Image query *vs.* Text data').

TABLE I
ACCURACY EVALUATION ON WIKI.
THE BEST RESULTS ARE MARKED IN BOLDFACE AND THE SECOND BEST RESULTS ARE MARKED BY UNDERLINES (TSH IS NOT TAKEN INTO COMPARISON).

Task	Method	Code Length		
		L = 8	L = 16	L = 32
Image query *vs.* Text data	TSH	0.2381		
	CMSSH	0.1092	0.1412	0.1212
	CVH	0.2080	0.1960	0.1629
	LCMH	0.1435	0.1552	0.1682
	CMFH	0.2175	0.2156	0.2383
	SCM-Seq	0.2117	0.2282	0.2194
	SCM-Orth	0.1989	0.1871	0.1716
	LSSH	0.1945	0.2195	0.2146
	SePH$_{rnd}$	0.2094	**0.2380**	0.2419
	SePH$_{km}$	0.2212	0.2234	**0.2520**
	MCMHSC	**0.2361**	0.2228	0.2118
Text query *vs.* Image data	TSH	0.3093		
	CMSSH	0.1020	0.1122	0.1463
	CVH	0.3548	0.2741	0.2493
	LCMH	0.1445	0.1793	0.1996
	CMFH	0.3288	0.3379	0.4075
	SCM-Seq	0.3096	0.3576	0.4278
	SCM-Orth	0.3052	0.2425	0.2265
	LSSH	0.4818	0.5521	0.5658
	SePH$_{rnd}$	**0.6598**	**0.6725**	**0.6849**
	SePH$_{km}$	0.6368	0.6716	0.6822
	MCMHSC	0.6038	0.5601	0.5513

C. Parameters' Setting

For MCMHSC, we set K to the size of the training set. In that case, $\mathbf{Z}^{(1)}$ and $\mathbf{Z}^{(2)}$ are calculated by the distances to all the training samples. For LCMH, we set its parameters based on the settings in [9]. For the other schemes, since the source codes are provided by their authors, we use the default parameters of these programs.

The length of hash codes L in our experiments is set 8, 16, and 32, which is the same as in [9]. We set these values because the hash codes of these lengths can be easily stored in bytes and measured by fast bitwise operations.

D. Accuracy Evaluation

The experimental results on Wiki and NUS-WIDE are shown in Tables I and II respectively. Clearly, MCMHSC is comparable with other schemes on Wiki, and it outperforms other schemes on NUS-WIDE.

To give an in-depth discussion on the performance of various schemes, we divide the hashing methods into several groups. From the point of model optimization, we can separate these ten methods (one is ours) into three groups: (1) the ones based on iterative optimization (CMFH and LSSH), (2) the ones based on bitwise optimization (CMSSH, SCM-Seq, SePH$_{rnd}$ and SePH$_{km}$) and (3) the ones based on eigenvalue decomposition (CVH, SCM-Orth, LCMH and MCMHSC). By optimizing the model iteratively, the schemes belonging to the first group get the optimal value of one variable by fixing the others in

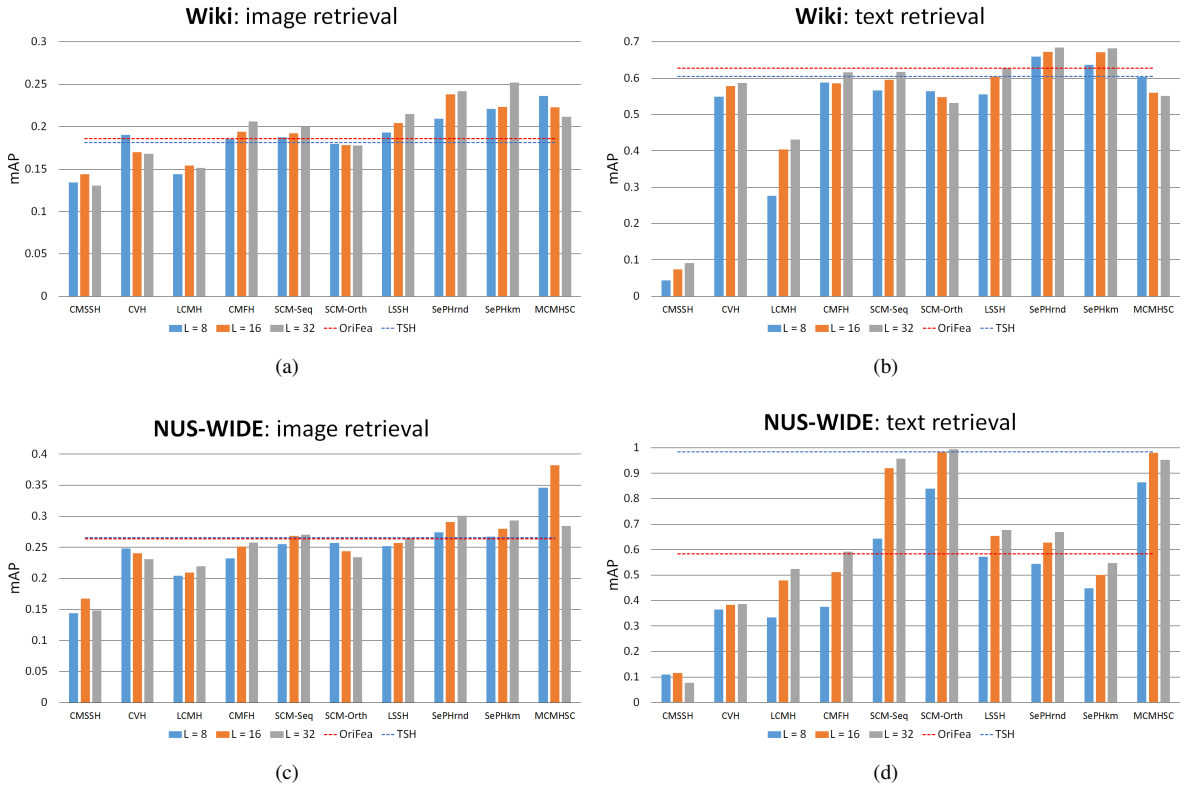

Fig. 4. Illustrations of the results of the single-media retrievals with the hash codes: (a) image retrieval on Wiki, (b) text retrieval on Wiki, (c) image retrieval on NUS-WIDE, (d) text retrieval on NUS-WIDE. 'OriFea' stands for the original feature.

each iteration. In this way, this kind of methods can reach a locally optimal solution and get a reliable result. Both CMFH and LSSH employ some high-level descriptors to improve the retrieval performance: CMFH uses latent semantic features learned by matrix factorization to represent both image and text, while LSSH changes the image representation to a more suitable feature, *i.e.*, sparse coding. Since the features of these two schemes are high-level, their accuracy is higher than some methods that use the original features, *e.g.*, CVH. Besides, the mAP scores of these schemes are usually increased with the code length. This is reasonable since a longer code encodes more information and thus improves the performance.

The schemes belonging to the second group optimize the hash function bit by bit in a loop. Specifically, CMSSH extends the AdaBoost method, whose accuracy is heavily related to the number of pairwise correlations observed. However, since most of the correlations are not observed, it fails to provide enough samples for training and its performance is not stable. SCM-Seq minimizes the differences between the distances calculated by hash codes and the ones calculated by label vectors, and it takes a non-orthogonal projection to learn the hash function bit by bit. The two schemes of SePH minimize the KL-divergence between the probability distribution of hash codes and the one learned from the semantic affinities of training data. Since these three schemes introduce categories, their accuracy is relatively high. The bit by bit optimization improves their performance with the code length in most cases. This is because the current bit helps to improve the discrimination of the previous bits.

TABLE II
ACCURACY EVALUATION ON NUS-WIDE.
THE BEST RESULTS ARE MARKED IN BOLDFACE AND THE SECOND BEST RESULTS ARE MARKED BY UNDERLINES (TSH IS NOT TAKEN INTO COMPARISON)

Task	Method	Code Length		
		$L = 8$	$L = 16$	$L = 32$
Image query *vs.* Text data	TSH	0.3304		
	CMSSH	0.1612	0.1200	0.1188
	CVH	0.2730	0.2639	0.2518
	LCMH	0.1969	0.1962	0.2005
	CMFH	0.2670	0.2755	0.3000
	SCM-Seq	0.3619	0.3817	**0.3985**
	SCM-Orth	**0.3625**	0.3552	0.2909
	LSSH	0.2960	0.3069	0.3092
	SePH_{rnd}	0.2739	0.2906	0.3009
	SePH_{km}	0.2669	0.2800	0.2931
	MCMHSC	0.3460	**0.3820**	0.2840
Text query *vs.* Image data	TSH	0.5137		
	CMSSH	0.1662	0.1630	0.1580
	CVH	0.2968	0.2875	0.2783
	LCMH	0.1987	0.1941	0.2025
	CMFH	0.2774	0.3126	0.3375
	SCM-Seq	0.3514	0.4786	0.5211
	SCM-Orth	0.4451	0.4586	0.3720
	LSSH	0.3487	0.3669	0.3870
	SePH_{rnd}	0.5450	0.6284	0.6690
	SePH_{km}	0.4483	0.5014	0.5467
	MCMHSC	**0.8642**	**0.9797**	**0.9523**

The experimental results on both Wiki and NUS-WIDE verify these conclusions.

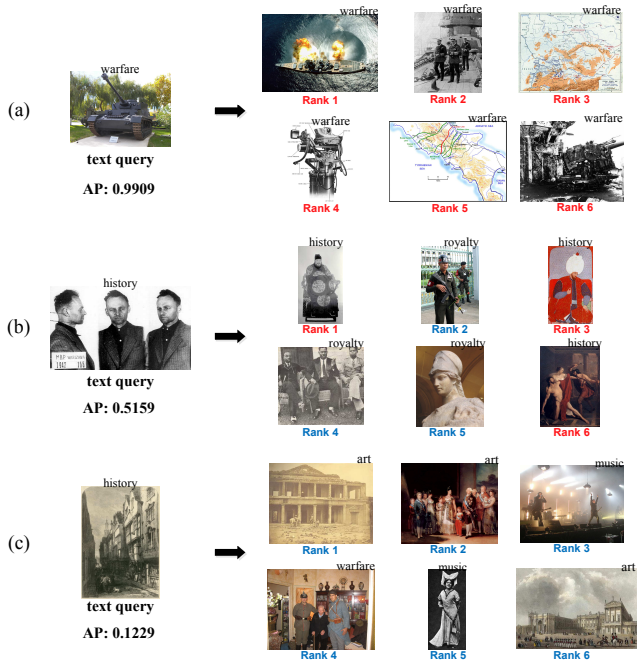

Fig. 5. Three examples of the 'Text query *vs.* Image data' task of MCMHSC on Wiki when the code length is 8, which include: (a) a good, (b) a medium and (c) a bad results. Text documents are represented with their associated images. The positive and negative retrieval results are marked in red and blue rank numbers. All the documents are labeled with their categories.

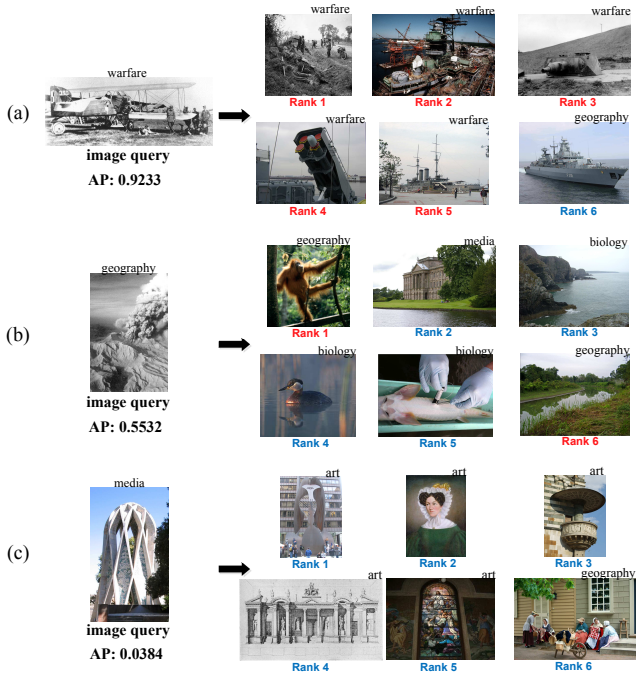

Fig. 6. Three examples of the 'Image query *vs.* Text data' task of MCMHSC on Wiki when the code length is 8, which include: (a) a good, (b) a medium and (c) a bad results. Text documents are represented with their associated images. The positive and negative retrieval results are marked in red and blue rank numbers. All the documents are labeled with their categories.

For the schemes in the last group, *i.e.*, CVH, SCM-Orth, LCMH and MCMHSC, they formulate the optimization into an eigenvalue decomposition problem. The performance of these schemes mainly depends on the optimal objective function, and they are all related with CCA. In particular, CVH preserves both intra-media and inter-media similarities, and it degenerates to CCA when the affinity matrix is not available. LCMH gets the representation of instances by calculating the distances to cluster centroids and fitting the Gaussian distribution. Since the representation is not effective, its performance is worse than CVH. SCM-Orth utilizes all the supervised information for training, which has a better performance than CVH. MCMHSC also introduces the category information by treating it as an independent view, which remarkably improves the performance. Besides, it takes a simple but effective fusion algorithm to further improve the search accuracy. In this way, it outperforms the previous works on NUS-WIDE, but only achieves a comparable performance on Wiki. The possible reason may lie in the datasets themselves. For Wiki, there are a lot of noise, and it is difficult to correctly bridge the heterogeneous gap between the instances under heavy noise. In contrast, NUS-WIDE is a more reliable benchmark. However, the mAP scores of these schemes are not consistently improved with the code length. This is because the high discriminative bits are from the first few projection directions that have high variances. In MCMHSC, we also learn a projection matrix for the label vectors in Eq. 5, which best retains the supervision when the code length L equals to the category number M. When $L < M$, it may introduce supervision loss, and when $L > M$, it possibly introduces noisy information. As a consequence, our scheme tends to have the best performance when the code length is close to the category number.

Besides, it can be observed from Tables I and II that TSH is a simple but effective baseline. It has a better performance than most of the hashing methods, especially in the task of 'Image query *vs.* Text data'. We also illustrate the results of image and text retrievals in Fig. 4. Compared to the original features, most of the schemes have similar and even better performance, especially for the methods that introduce categories (*i.e.*, SCM, SePH, and MCMHSC). The reason lies in that these schemes make the data of the same category more centralized in the shared space, which helps to improve the performance of the single-media retrievals.

Furthermore, TSH has a close performance to the best performing method for the image-to-text retrieval, but only has an average performance for the text-to-image retrieval as shown in Tables I and II. The reason lies in that the text features are better than the images features because they can train a better classifier for TSH. Fig. 4 also validates the conclusion, since the text-to-text retrieval of TSH is more accurate than the image-to-image retrieval on both two datasets. Considering the database set is also used for training in our experiments, the hash codes of the text gallery are more accurate than the image gallery. As a result, the performance of the image-to-text retrieval is better than the image-to-image retrieval for TSH. Meanwhile, the text-to-image retrieval has much lower accuracy than the text-to-text retrieval, and TSH only has an average performance for this task. Our method, however, uses a

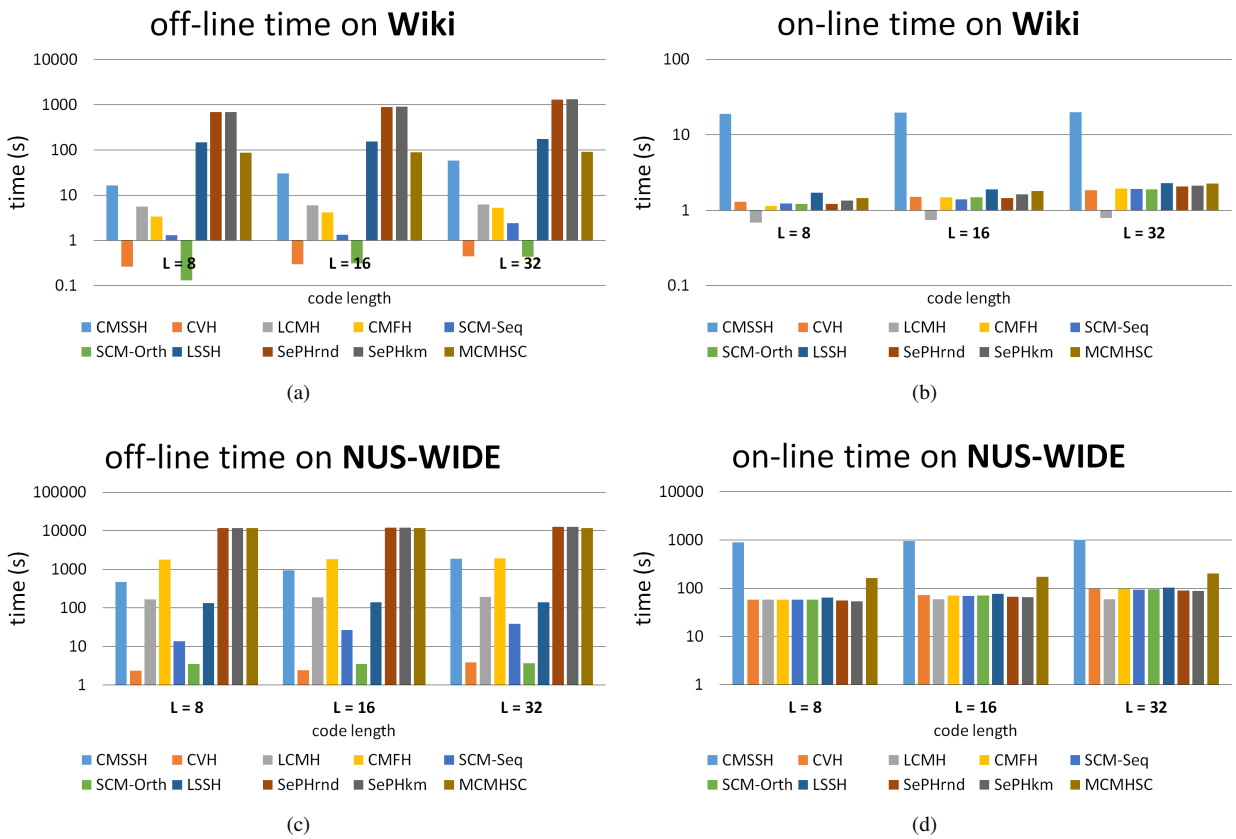

Fig. 7. Illustrations of the time comparisons between the schemes with different code lengths in: (a) the off-line phase on Wiki, (b) the on-line phase on Wiki, (c) the off-line phase on NUS-WIDE, (d) the on-line phase on NUS-WIDE. All the records are the real running time.

fusion algorithm to generate a unique hash code for each object. It observably improves the accuracy of the hash codes of the gallery images. As a consequence, the text-to-image retrieval of MCMHSC has a close performance to the text-to-text retrieval of TSH.

We present several examples of MCMHSC's retrieval results on Wiki, when the code length is 8, in Figures 6 and 5, which contain a good, a medium and a bad results of each retrieval task. We represent each text document with its corresponding image, and all the documents are labeled with their categories. It can be observed that MCMHSC works well. Even in the bad examples, the results still make sense. For example, in Fig. 6(c), even though the query is labeled as the category 'media', it looks like an artistic building.

MCMHSC is easy to scale to a larger number of categories. In that case, suppose the category number is M', then the size of $Z^{(3)}$ in Eq. 5 changes to $N \times M'$. The optimization steps are the same as in Section III.

E. Time Complexity Evaluation

To quantitatively evaluate the time complexity of the schemes, we make records of their time costs in both the off-line and on-line phases. Fig. 7 shows the results obtained on Wiki and NUS-WIDE. We test the schemes on a platform with two Intel Xeon 3.33GHz (6 cores) CPUs, 48GB RAM, and Linux x64 operating system.

Clearly, the two schemes in the iterative optimization group (CMFH and LSSH) have medium off-line time. This is because the training of these schemes is an iterative process which does not stop until it reaches convergence. It usually takes a long time to train the models.

The schemes in the bitwise optimization group learn their hash functions bit by bit in loops. The training time of these schemes increases with the code length and is mainly depended on the calculation complexity in each loop. CMSSH learns a scalar and updates the weights in each loop, which leads to short off-line time. SCM learns the current bit of the hash functions based on the previous bit, but it uses the multi-thread technique to speed up the training. SePH has the longest off-line time because it takes gradient descent to get the optimal value of each bit.

The benefit of the schemes in the eigenvalue decomposition group is that their training time is not increased with the code length. (But the off-line time includes calculating the hash codes of the database set, so it still increases with the code length.) Since their optimization is based on eigenvalue decomposition, the time complexity of these models is mainly related to the size of the matrix to be decomposed. However, since MCMHSC represents each instance with the distances to all the training samples, it has a long off-line time.

In the on-line phase, since their codes are mostly calculated by linear projection, their on-lime time is similar. CMSSH

has the longest on-line time. The main reason lies in that its similarity metric is weighted Hamming distance.

In brief, MCMHSC achieves comparable or better performance in terms of search accuracy and time complexity.

V. Conclusion

This paper proposes a new cross-media hashing method named Multi-view Cross-Media Hashing with Semantic Consistency (MCMHSC). The core idea is to treat the categories as the third view and preserve the correlation between heterogeneous instances and categories as well. In this way, both intra-category and pairwise correlations are considered simultaneously in learning the hash functions. In addition, a simple but effective fusion algorithm that fully exploits the complementary information from multiple views is proposed to further improve the search accuracy. Experiments on two public benchmarks show that the proposed scheme achieves comparable or better performance compared to the state-of-the-arts in terms of accuracy and time complexity. Since the scenarios with noisy categories and on large-scale datasets are more practical in the real-world applications, we will study it in the future work.

Acknowledgment

This work was supported in part by National Natural Science Foundation of China (No.61572065, No.61532005), Joint Fund of Ministry of Education of China and China Mobile (No.MCM20160102), and Fundamental Research Funds for the Central Universities (No.2015JBM028, No.2015JBZ002).

References

[1] L. Zheng, Y. Yang, and Q. Tian, "Sift meets cnn: a decade survey of instance retrieval," *TMAPI*, preprint.
[2] M. Datar, N. Immorlica, P. Indyk, and V. Mirrokni, "Locality-sensitive hashing scheme based on p-stable distributions," in *SoCG*, 2004, pp. 253–262.
[3] J. Song, Y. Yang, Z. Huang, H. Shen, and R. Hong, "Multiple feature hashing for real-time large scale near-duplicate video retrieval," in *ACMMM*, 2011, pp. 423–432.
[4] R. Liu, Y. Zhao, S. Wei, and Z. Zhu, "Cross-media hashing with centroid approaching," in *ICME*, 2015, pp. 1–6.
[5] Y. Gong, Q. Ke, M. Isard, and S. Lazebnik, "A multi-view embedding space for modeling internet images, tags, and their semantics," *IJCV*, vol. 106, no. 2, pp. 210–233, 2014.
[6] M. Bronsten, A. Bronstein, F. Michel, and N. Paragios, "Data fusion through cross-modality metric learning using similarity-sensitive hashing," in *CVPR*, 2010, pp. 3594–3601.
[7] S. Kumar and R. Udupa, "Learning hash functions for cross-view similarity search," in *IJCAI*, vol. 22, no. 1, 2011, pp. 1360–1365.
[8] Y. Zhen and D. Yeung, "A probabilistic model for multimodal hash function learning," in *SIGKDD*, 2012, pp. 940–948.
[9] X. Zhu, Z. Huang, H. Shen, and X. Zhao, "Linear cross-modal hashing for efficient multimedia search," in *ACMMM*, 2013, pp. 143–152.
[10] G. Ding, Y. Guo, and J. Zhou, "Collective matrix factorization hashing for multimodal data," in *CVPR*, 2014, pp. 2083–2090.
[11] J. Zhou, G. Ding, and Y. Guo, "Latent semantic sparse hashing for cross-modal similarity search," in *SIGIR*, 2014, pp. 415–424.
[12] N. Rasiwasia, J. Pereira, E. Coviello, G. Doyle, G. Lanckriet, R. Levy, and N. Vasconcelos, "A new approach to cross-modal multimedia retrieval," in *ACMMM*, 2010, pp. 251–260.
[13] D. Zhang and W. Li, "Large-scale supervised multimodal hashing with semantic correlation maximization," in *AAAI*, 2014, pp. 2177–2183.
[14] Z. Lin, G. Ding, M. Hu, and J. Wang, "Semantics-preserving hashing for cross-view retrieval," in *CVPR*, 2015, pp. 3864–3872.
[15] J. Masci, M. Bronstein, A. Bronstein, and J. Schmidhuber, "Multimodal similarity-preserving hashing," *TPAMI*, vol. 36, no. 4, pp. 824–830, 2014.
[16] D. Wang, P. Cui, M. Ou, and W. Zhu, "Deep multimodal hashing with orthogonal regularization," in *IJCAI*, 2015.
[17] Q.-Y. Jiang and W.-J. Li, "Deep cross-modal hashing," *arXiv preprint arXiv:1602.02255*, 2016.
[18] A. Sablayrolles, M. Douze, H. Jégou, and N. Usunier, "How should we evaluate supervised hashing?" *arXiv preprint arXiv:1609.06753*, 2016.

Ruoyu Liu is a PhD student of the Institute of Information Science, Beijing Jiaotong University. His research interests include multimedia retrieval, computer vision and deep learning. Contact him at 12112062@bjtu.edu.cn.

Shikui Wei is currently a Professor with the Institute of Information Science, Beijing Jiaotong University. His research interests include computer vision, image/video analysis and retrieval, and copy detection. Contact him at shkwei@bjtu.edu.cn.

Yao Zhao is now the Director of the Institute of Information Science, BJTU. His research interests include image/video coding, digital watermarking and forensics, and video analysis and understanding. Zhao is the corresponding author for this article. Contact him at yzhao@bjtu.edu.cn.

Zhenfeng Zhu is currently a Professor with the Institute of Information Sciences, Beijing Jiaotong University. His current research interests include image and video understanding, computer vision, and machine learning. Contact him at zhfzhu@bjtu.edu.cn.

Jingdong Wang is a Lead Researcher at the Visual Computing Group, Microsoft Research Asia. His areas of interest include computer vision, machine learning, pattern recognition, and multimedia computing. Contact him at jingdw@microsoft.com.