Unsupervised Visual Hashing with Semantic Assistant for Content-Based Image Retrieval

Zhu, L., Shen, J., Xie, L., & Cheng, Z. (2017). Unsupervised Visual Hashing with Semantic Assistant for Content-Based Image Retrieval. *IEEE Transactions on Knowledge and Data Engineering*, *29*(2), 472-486. https://doi.org/10.1109/TKDE.2016.2562624

Published in:
IEEE Transactions on Knowledge and Data Engineering

Document Version:
Peer reviewed version

Published in IEEE Transactions on Knowledge and Data Engineering, 2017 February, Volume 29, Issue 2, Pages 472-486
http://doi.org/10.1109/TKDE.2016.2562624

Unsupervised Visual Hashing with Semantic Assistant for Content-Based Image Retrieval

Lei Zhu, Jialie Shen, Liang Xie, and Zhiyong Cheng

Abstract—As an emerging technology to support scalable content-based image retrieval (CBIR), hashing has recently received great attention and became a very active research domain. In this study, we propose a novel unsupervised visual hashing approach called semantic-assisted visual hashing (SAVH). Distinguished from semi-supervised and supervised visual hashing, its core idea is to effectively extract the rich semantics latently embedded in auxiliary texts of images to boost the effectiveness of visual hashing without any explicit semantic labels. To achieve the target, a unified unsupervised framework is developed to learn hash codes by simultaneously preserving visual similarities of images, integrating the semantic assistance from auxiliary texts on modeling high-order relationships of inter-images, and characterizing the correlations between images and shared topics. Our performance study on three publicly available image collections: *Wiki*, *MIR Flickr*, and *NUS-WIDE* indicates that SAVH can achieve superior performance over several state-of-the-art techniques.

Index Terms—Content-based image retrieval, semantic-assisted visual hashing, auxiliary texts, unsupervised learning

1 Introduction

With the continued advances in social media and mobile computing technology, the past decade has witnessed a tremendous growth in the availability of Web images. Consequently, there has been an increasing interest in the information retrieval and multimedia computing communities to study smart image retrieval techniques. In particular, techniques for content-based image retrieval (CBIR) [1], [2], where only visual image is used as query, are gaining in importance due to a wide range of promising applications.

To provide high quality content-based search services over huge volume of image collections, both efficiency and effectiveness are important issues. Advanced indexing structure is essential to scale the big data space and facilitate accurate search. The most naive approach for CBIR is to sequentially compare query image with each sample stored in the database. Its linear complexity leads to the poor efficiency and low scalability in real environment. Also, visual features usually have high dimensions. How to solve the curse of dimensionality is still an open research question, which has not been addressed properly. Fortunately, in most real CBIR applications, approximate retrieval results can sufficiently satisfy users' information needs. This suggests the feasibility of approximate nearest neighbor retrieval. Motivated by this observation, many indexing approaches, such as inverted file [3], tree structure [4] and hashing [5], [6], have been developed in recent years. Inverted file is originally designed for text retrieval, and it can only perform well on indexing high-dimensional sparse feature, such as bag-of-visual-words [3]. Tree structure is competent for indexing low-dimensional features. However, its performance degrades greatly when the dimension of features to be indexed goes high. Furthermore, both inverted file and tree structure consume large amount of memory when storing corresponding data structures. This issue becomes even more serious when image collection scale is large.

As one of the emerging technologies to support fast and accurate image search, visual hashing has received great attention and became a very active research domain in last decade [5], [6]. Its basic idea is to map the raw high-dimensional visual features into binary codes in low-dimensional Hamming space, so that visual similarities of images can be efficiently measured by simple but efficient bit-wise operations. Generally, visual hashing enjoys two major advantages: (1) *Fast query response*—Retrieval process can be completed quickly, because bit-wise operations can be efficiently implemented. (2) *Low storage consumption*—The storage of high-dimensional features can be greatly reduced as the result of binary embedding.

However, due to binary embedding of continuous visual space, semantic information in original visual features may be lost during hashing, which degrades the performance of CBIR. To enrich the semantics of visual hash codes, many machine learning based strategies have been applied and several hashing schemes have been proposed. They include: unsupervised visual hashing [7], [8], [9], supervised visual hashing [10], [11], and semi-supervised visual hashing [12], [13]. Both supervised and semi-supervised visual hashing can improve the semantic discriminative capability of hash codes. However, both paradigms require labeled images in training process. This requirement, actually, may be not

- *L. Zhu and Z. Cheng are with the School of Information Systems, Singapore Management University, Singapore 178902. E-mail: {leizhu0608, jason.zy.cheng}@gmail.com.*
- *J. Shen is with the Department of Computer and Information Sciences, Northumbria University, Newcastle upon Tyne, NE1 8ST, U. K. E-mail: jialie@gmail.com.*
- *L. Xie is with the School of Mathematics, Wuhan University of Technology Hongshan District, Wuhan 430070, China. E-mail: whutxl@hotmail.com.*

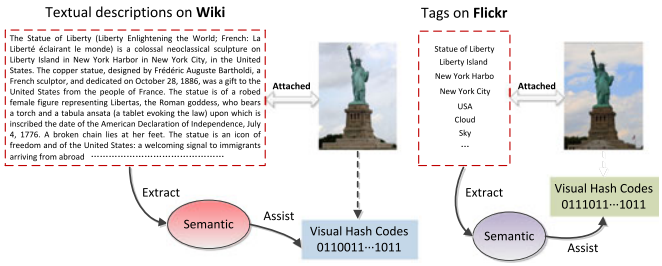


Fig. 1. Image samples. Two similar images of *The Statue of Liberty* collected from Wikipedia and Flickr, respectively. The core idea of SAVH is extracting semantics latently embedded in such associated informative texts to assist visual hashing process.

satisfied in CBIR since good quality labeled images are scarce in practical scenario, while it requires a significant amount of manual efforts and domain expertise. On the other hand, images (such as pictures in social networks) are usually associated with noisy but informative tags or textual descriptions. Fig. 1 shows two image examples about *The Statue of Liberty* from Wikipedia and Flickr respectively. It is not hard to find that both images are accompanied with the texts including valuable semantic elements. These observations inspire us to exploit the auxiliary texts to boost the quality of visual hashing via unsupervised learning. The core challenge is how to develop unsupervised learning scheme to intelligently extract and integrate the semantics from the associated informative texts into visual hash codes.

Generally, according to strategies to leverage different kinds of visual features, existing unsupervised visual hashing schemes can be classified into several independent families: single feature visual hashing (SFVH) [7], [8] and multiple feature visual hashing (MFVH) [14], [15], [16]. They learn hash functions and codes using visual features solely. Due to the lack of discriminative capability on representing high-level semantics, the learned hash codes are not able to characterize semantic correlations of images and the ones between images and semantic topics latently involved in the image database. It should be noted that, one of the visual modality in MFVH can be substituted with text modality. In this case, MFVH becomes multi-modal hashing (MMH) [14], [17]. However, it requires text modality at both stages of offline learning and online hashing (as shown in Fig. 3b). Due to this constraint, the scheme cannot meet the requirement of CBIR in practical retrieval applications, where only visual image is uploaded as query.

On the other hand, unsupervised cross-modal hashing (UCMH) [18], [19], [20], [21], [22], [23] can be applied to CBIR. Inter-media hashing (IMH) [19], linear cross-modal hashing (LCMH) [24], and collective matrix factorization hashing (CMFH) [25] are typical examples. Its core idea is to discover the semantic correlations of multiple modalities and facilitate cross-modal retrieval by mapping heterogeneous modalities into the common Hamming space. However, the main objective of UCMH focuses on how to support effective cross-modal retrieval between images and texts. They generally suffer from two disadvantages when being applied to CBIR:

- In UCMH, hash codes and functions are learned to preserve intra-modality and inter-modality correlations. The semantics in the discovered common semantic space shared by heterogeneous modalities could be limited. Thus, the valuable semantic information originally owned by visual features may not be comprehensively preserved as result of mandatory correlation (as shown in Fig. 2).
- UCMH is specifically designed for the task of cross-modal retrieval. To achieve the goal, it usually treats the involved modalities equally, which generally ignores the differences between modalities on contributions for search performance.[1] Therefore, the semantics in the associated texts cannot be fully exploited for hashing.

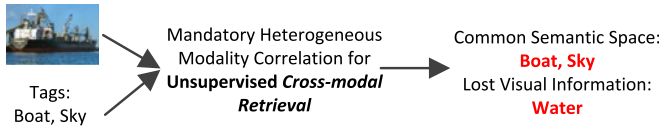


Fig. 2. Important visual information may be lost in unsupervised cross-modal hashing due to mandatory heterogeneous modality correlation.

In this work, we propose a novel unsupervised visual hashing scheme, termed as semantic-assisted visual hashing (SAVH), to effectively perform visual hashing learning with semantic assistance. The key idea is to extract semantics automatically from the noisy associated texts to enhance the discriminative capability of hash codes, and thus facilitate the performance improvement of visual hashing. SAVH works as follows: First, hash code learning is formulated in a unified unsupervised framework, where relaxed hash codes are learned by simultaneously preserving visual similarity of images and considering the assistance of texts. More specifically, our framework integrates two important assistance of auxiliary texts to effectively mitigate the inherent limitations of visual features. The first assistance models high-order semantic relations of images by constructing topic hypergraph, while the second one correlates images and latent shared topics detected via collective matrix factorization. Then, an optimization method based on augmented Lagrangian multiplier (ALM) [26] is proposed to iteratively calculate the optimal solution. We specially preserve bits-uncorrelated constraint during iterative process to facilitate learning and simultaneously reduce information redundancy between hash bits. Finally, hash functions are constructed based on linear regression to enable out-of-sample query extension. Linear projection can support efficient hash code generation in online retrieval.

The key contributions can be summarized as follows:

- Instead of considering only visual feature or equally treating images and texts, SAVH specially exploits the auxiliary texts to assist visual hashing. Two important assistances from auxiliary texts: modeling semantic correlations of images with topic hypergraph, correlating images and latent shared topics via collective matrix factorization, are proposed to effectively incorporate semantics into the hash codes.
- SAVH is designed in a unified unsupervised learning framework, which comprehensively considers visual

1. For example, IMH, CMFH, and LCMH impose the same importance for visual and textual features on learning shared binary space for unsupervised cross-modal retrieval.

similarity preservation of images and semantic-assistance. An effective solution based on ALM is proposed to calculate the optimal hash codes.
- The system architecture of SAVH is developed based on multi-modal learning data but requires only visual image as input query. It meets the practical requirement of CBIR, where database images are usually associated with informative texts, while no text query is provided.
- Comprehensive experiments are conducted on several publicly available image databases. Results highlight various advantages of SAVH and demonstrate that SAVH significantly outperforms several state-of-the-art content-based and cross-modal hashing methods from various perspectives.

The rest of the paper is structured as follows. Section 2 presents literature review. Details about SAVH are introduced in Section 3. Experimental configuration is presented in Section 4. In Section 5, we present experimental results and analysis. Section 6 concludes the paper with summary and future work.

2 Related Work

2.1 Single Feature Visual Hashing

According to the way about how to generate hash function, existing SFVH can be further categorized into two major families: data-independent [27], and data-dependent hashing [7], [8]. Locality sensitive hashing (LSH) [27] is one of the most typical data-independent hashing schemes, which is based on random vectors from specific distribution, e.g., standard Gaussian distribution, to map similar points into Hamming space with high probability. On the other side, data-dependent hashing schemes are proposed to learn the hash functions according to the characteristics of underlying data distribution by using machine learning methods. Spectral hashing (SPH) [7], anchor graph hashing (AGH) [9], and self-taught hashing (STH) [8] are typical unsupervised hashing approaches. SPH learns the hash functions by preserving the similarities of images in the mapped hash codes, while STH extends it for out-of-sample queries via linear support vector machine (LinearSVM) [28] training in Hamming space. AGH approximates the affinity graph with low-rank matrix, and learns the hash functions by binarizing the Nystrom eigen-functions [29]. Iterative quantization (ITQ) [30] is proposed to reduce the quantization loss by rotating the learned hash codes. Besides, sparse learning, manifold learning, and deep learning are applied to hashing. With the trend, sparse embedding hashing [31], [32], manifold based hashing [33], deep learning based hashing [13] are proposed to learn effective binary hash codes.

2.2 Multiple Feature Visual Hashing

Multiple features integration is very important to comprehensively interpret visual contents and achieve optimal learning performance [34], [35], [36]. Many researchers are motivated to design various schemes to conduct hashing while considering multiple feature fusion [14], [16], [37], [38], [39], [40], [41] for different purposes. For example, sequential update for multi-view spectral hashing (SUMVSH) [42] is proposed to sequentially learn hash functions by solving the successive maximization of local variances. In order to achieve the goal, multiple features are integrated by minimizing its α-divergence from view-specific distance matrices. Kim and Choi [37] present multi-view anchor graph hashing (MVAGH) by extending AGH to handle multiple image representations. The integrated binary codes are calculated as the subset of eigenvectors of a fused similarity matrix. Multiple feature hashing (MFH) [14], [17] formulates the hashing learning by simultaneously preserving the local structural information in each modality and considering all the local structures. By using the learned hashing hyperplane, MFH concatenates all features into single vector and maps it into binary hash codes. Cheng et al. [16] formulates the hashing learning on multiple visual features within multi-graph framework, where multiple visual features are integrated with proper weights. More recently, multi-view latent hashing (MVLH) [38] is proposed to incorporate multi-modal features in binary representation learning by discovering the latent factors shared among multiple views. The weights for multiple feature fusion are learned according to the reconstruction error with each view. Multi-view alignment hashing (MVAH) [39] learns hash codes with regularized kernel non-negative matrix factorization. It considers both the hidden semantics and joint probability distribution of multiple visual features.

The most significant limitation of SFVH and MFVH is that they only takes the features from visual modality into account. Due to the semantic gap, image relations characterized by low-level visual feature cannot effectively describe rich image semantics, consequently making the hash codes less semantically meaningful.

2.3 Unsupervised Cross-Modal Hashing

The core idea of UCMH is to map heterogeneous modalities into the common Hamming space, where similarities are computed to return cross-modal retrieval results. One of the typical examples is cross-view hashing (CVH) [18]. It is proposed to extend SPH to cross-modal retrieval by jointly minimizing Hamming distances of similar samples and maximizing that of dissimilar samples. In inter-media hashing [19], a framework is proposed to facilitate cross-modal hashing learning, aiming to preserve intra-similarity of each individual modality and inter-correlation between heterogeneous modalities. Linear cross-modal hashing [24] is proposed to enable scalable multimedia search across different modalities via efficient intra-modality similarity preserving. In [23], latent semantic sparse hashing (LSSH) is proposed to perform cross-modal similarity search in a joint abstract semantic space by employing sparse coding and matrix factorization. Collective matrix factorization hashing [25] learns hash codes using collective matrix factorization with latent factor model from multiple modalities of one sample.

It seems that, UCMH can improve the performance of CBIR, as the projected space may embed more semantics than low-level visual feature space. However, the main design aim of various UCMH approaches is to enable multimedia retrieval *across* heterogeneous modalities. It assumes that each type of the involved modality contributes equally to cross-modal retrieval. This assumption makes the common Hamming space shared by them less

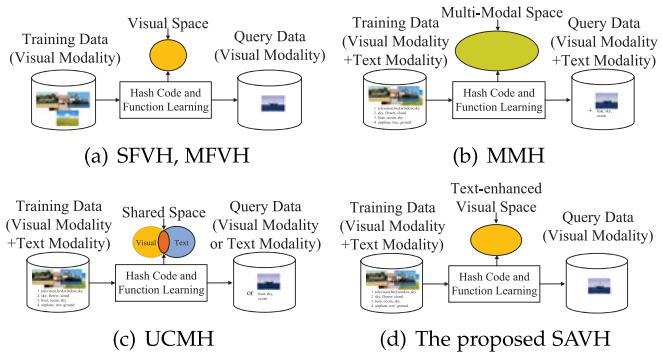


Fig. 3. Basic structures of main unsupervised visual hashing schemes and the proposed SAVH.

discriminative and effective. In addition, discriminative information inherently involved in original visual feature may be lost accordingly in hashing process due to mandatory heterogeneous modality correlation.

Fig. 3 and Table 1 summarize key characteristics of state-of-the-art hashing methods and the proposed SAVH. Based on analysis given above, we can find that it is important to specifically design an intelligent hashing strategy to effectively leverage the associated modality (e.g., informative texts) to assist visual hashing.

3 Semantic-Assisted Visual Hashing

This section gives a detailed introduction of SAVH scheme. We first present system overview and problem formulation in Sections 3.1 and 3.2 respectively. After that, we introduce the scheme used in SAVH which preserves visual similarity in Section 3.3. Then, Section 3.4 illustrates the semantic assistance of auxiliary text in SAVH. Next, we formulate the overall objective function for hash code learning and present an effective optimization approach. We also extend SAVH to out of sample via hash function learning. Finally, Section 3.7 analyzes computational complexity.

3.1 System Overview

Fig. 4 describes the basic framework of the SAVH-based CBIR system. The system mainly includes two core components: offline learning and online hashing.

- *Offline learning*. This component aims to learn hash codes of database images and simultaneously generate hash function for query image. It consists of four main steps. First, visual and text features of images are extracted to transform image pixels to mathematical vector representations. Then, a text-enhanced visual graph is constructed with the assistance of topic hypergraph, and latent semantic topics are detected under guidance of text information. Next, hash codes of database images are learned in a framework which preserves correlations of images and that between images and semantic topics. Finally, hash functions are generated with respect to the hash codes within a linear regression model.
- *Online hashing*. Visual feature of query image is extracted. Then, it is mapped into binary codes with hash functions. Finally, the similarities between query image and database images are calculated in Hamming space, and database images are returned in order of distance ascending.

3.2 Notations and Problem Formulation

In this study, we use boldface uppercase letters to represent matrices, boldface lowercase letters to represent vectors, and calligraphic letters to represent scales. The transpose of matrix $\mathbf{X}$ is denoted as $\mathbf{X}^{\mathrm{T}}$. The inverse of a matrix $\mathbf{X}$ is denoted as $\mathbf{X}^{-1}$. The trace operator on a matrix $\mathbf{X}$ is denoted

TABLE 1
Characteristics of Main Unsupervised Visual Hashing Methods and SAVH

Methods	Query	Learning Feature	Learning Space	Semantic Enhancement	CBIR
Single Feature Visual Hashing (SFVH)	Visual	Visual	Visual	No	Yes
Multiple Feature Visual Hashing (MFVH)	Visual	Visual	Visual	No	Yes
Multi-modal Hashing (MMH)	Visual+Text	Visual+Text	Multi-modal	Yes	No
Unsupervised Cross-modal Hashing (UCMH)	Visual or Text	Visual+Text	Shared	Limited	Partly
The proposed SAVH	Visual	Visual+Text	Text-enhanced	Yes	Yes

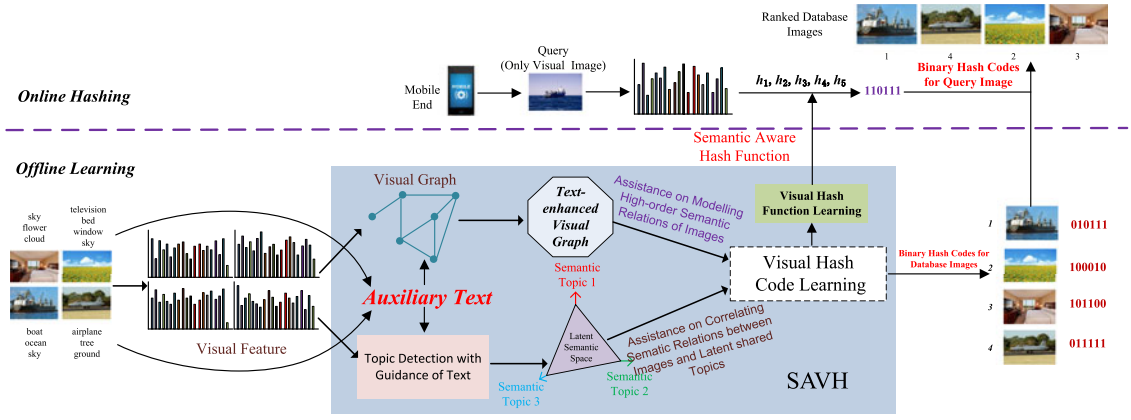


Fig. 4. Framework of the SAVH-based CBIR system.

TABLE 2
Summary of Main Notations

Symbols	Explanations
$\mathbf{X}^{(1)}, \mathbf{X}^{(2)}$	visual and text feature matrix of database images
d_1, d_2	dimension of feature in visual and text modality
N	number of database images
$\mathbf{Y}$	hash codes of database images
L	length of hash codes
$\mathbf{F}$	hash functions
$\mathbf{S}$	affinity matrix of visual graph
k	number of nearest neighbors in visual graph
$\mathbf{T}$	text topics
$\mathbf{H}$	incidence matrix of topic hypergraph
$\mathbf{L}_G$	Laplacian matrix of visual graph
$\mathbf{L}_{THG}$	Laplacian matrix of topic hypergraph
$\mathbf{D}_v$	diagonal matrix of vertex degrees in topic hypergraph
$\mathbf{D}_e$	diagonal matrix of edge degrees in topic hypergraph
$\mathbf{D}_w$	diagonal matrix of edge weights in topic hypergraph
$\mathbf{W}$	projection matrix in hash functions
$\mathbf{V}$	shared topic distributions
$\mathbf{U}^{(1)}, \mathbf{U}^{(2)}$	basis matrix of visual and text space

as $\mathbf{Tr}(\mathbf{X})$. $||\cdot||_F$ denotes Frobenius norm. $\exp(\cdot)$ is exponential function. $\mathtt{sgn}(\cdot)$ is Sign function. $\mathbf{I}$ denotes identity matrix and $\mathbf{1}$ denotes a vector with all 1 elements. The corresponding dimension of them can be inferred from the context. $\mathbf{0}$ denotes a vector or matrix of all 0 elements.

More specifically, we define $\mathbf{X}^{(m)} = [\mathbf{x}_1^{(m)}, \ldots, \mathbf{x}_N^{(m)}] \in \mathbb{R}^{d_m \times N}, m = 1, 2$ as the feature representations of database images extracted from visual and text modality respectively, d_m denotes the dimension of the corresponding feature, N is the number of database images. The aim of visual hashing is to learn hash codes of database images $\mathbf{Y} = [\mathbf{y}_1, \ldots, \mathbf{y}_N] \in \mathbb{R}^{L \times N}$, and a group of hash functions $\mathbf{F} = \{\mathbf{f}_1, .., \mathbf{f}_L\}$, where $\mathbf{y}_i = [y_{1i}, \ldots, y_{Li}]^{\mathrm{T}} \in \mathbb{R}^{L \times 1}$ are the hash codes of the ith image, each hash function $\mathbf{f}_l$ is a mapping: $\mathbb{R}^{d_m} \mapsto \{-1, 1\}, l = 1, \ldots, L$, L is the length of the visual hash codes.[2] Main notations used in the study are listed in Table 2.

3.3 Visual Similarity Preservation

Effectively preserving visual similarities of images in binary hash codes is essential to visual hashing. In this study, we resort to graph model [43], [44], [45] to address the problem. Actually, the inner structure of visual graph can be simply represented with affinity matrix $\mathbf{S}$. In SAVH, we choose local similarity for graph construction, considering its good property on characterizing visual similarities of images [8], [46], [47]. More specifically, we calculate visual similarities between images and their k nearest neighbors, and preserve them in matrix $\mathbf{S}$. Formally, the element at ith row, jth column (visual similarity between image i and j) is calculated as

$$\mathbf{S}_{ij} = \begin{cases} \exp(-||\mathbf{x}_i^{(1)} - \mathbf{x}_j^{(1)}||_F^2 / \theta^{(1)})) \\ \qquad\qquad \text{if } \mathbf{x}_i^{(1)} \in \mathcal{N}_k(\mathbf{x}_j^{(1)}) \text{ or } \mathbf{x}_j^{(1)} \in \mathcal{N}_k(\mathbf{x}_i^{(1)}) \\ 0 \qquad\qquad \text{otherwise} \end{cases} \tag{1}$$

where $\mathcal{N}_k(\mathbf{x})$ denotes the set of k nearest neighbors of $\mathbf{x}$, $\theta^{(1)}$ is normalization factor which is calculated as the average visual distances of images. To preserve visual similarity, we seek to minimize the sum of weighted Hamming distances

$$\min_{\{\mathbf{y}_i\}_{i=1}^N} \sum_{i=1}^N \sum_{j=1}^N \mathbf{S}_{ij} ||\mathbf{y}_i - \mathbf{y}_j||_F^2 \Leftrightarrow \min_{\mathbf{Y}} \ \mathbf{Tr}(\mathbf{Y}\mathbf{L}_G\mathbf{Y}^{\mathrm{T}}) \tag{2}$$

where $\mathbf{L}_G = \mathbf{S}\mathbf{1}_N - \mathbf{S}$ is the Laplacian matrix of visual graph. The behind idea of the above formulation is to incur a heavy penalty if two similar images are mapped far apart. Hence, visually similar images can be mapped into hash codes with short Hamming distances.

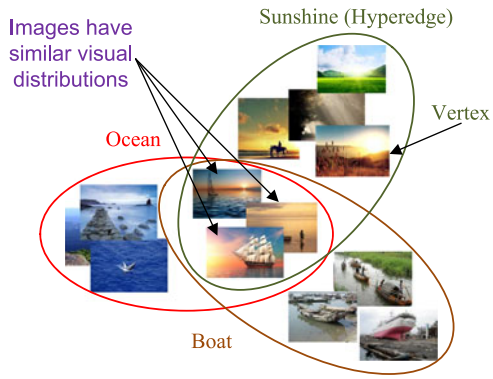


Fig. 5. A typical example of topic hypergraph. Images and latent semantic topics are considered as vertices and hyperedges, respectively. Images that belong to more identical hyperedges share more similar visual distributions.

3.4 Semantic-Assistance of Auxiliary Text

The core idea of SAVH is to effectively leverage the semantics embedded in the associated informative texts around images to assist visual hashing. With the assistance of texts, the generated visual hash codes and functions can aware high-level semantics, and thus they will be more discriminative. In this study, we consider two important assistance.

1. *Assistance on modelling semantic relations of images.* Due to the well-known semantic gap, visual feature inherently has limitations on representing high-level semantics. Hence, the built visual graph usually fails to effectively characterize latent semantic correlations of images. On the other hand, database images are usually associated with informative tags or textual descriptions (as shown in Fig. 1). These texts generally have better semantic descriptive capability than pure image pixels. In addition, text and visual image belong to different modalities. They have discriminative information that may be complement with each other [14]. Therefore, it is promising to leverage the informative texts to assist visual graph on characterizing semantic correlations of images.

   Actually, the latent semantic correlations of images are high-order. It is common that a single image will describe multiple semantic topics, and a topic may be shared by multiple images. In this case, images that share more semantic topics will possess the similar visual contents with greater probability. Inspired by the observation, this study proposes a topic hypergraph (a typical example is shown in Fig. 5) constructed on auxiliary texts to model the

2. The length of hash code is equal to the dimension of the mapped Hamming space.

high-order semantic correlations of images. Different from existing hypergraphs [48], [49], in topic hypergraph, visual images are determined as vertices. Semantic topics detected from text features associated with images are considered as hyperedges. In this case, a hyperedge connects several images and an image belongs to several hyperedges. Hence, the high-order semantic correlations of images are effectively modelled.

We first leverage k-means to partition texts into L groups. Their centers are considered as semantic topics $\mathbf{T} = \{\mathbf{t}_1, \ldots, \mathbf{t}_L\}$ latently embedded in texts. The topic hypergraph can be represented with an $L \times N$ incidence matrix $\mathbf{H}$. The incidence value between hyperedge $\mathbf{t}_l$ and vertex $\mathbf{x}_i^{(2)}$ in $\mathbf{H}$ is

$$\mathbf{H}(\mathbf{t}_l, \mathbf{x}_i^{(2)}) = \exp(-||\mathbf{t}_l - \mathbf{x}_i^{(2)}||_F^2 / \theta^{(2)}) \tag{3}$$

where $\theta^{(2)}$ has similar meaning with $\theta^{(1)}$ and it is calculated as the average text distances of images. Each element in $\mathbf{H}$ measures the probability that a vertex belongs to a hyperedge. With $\mathbf{H}$, the degree of hyperedge $\mathbf{t}_l$ is calculated as

$$\delta(\mathbf{t}_l) = \sum_{i=1}^{N} \mathbf{H}(\mathbf{t}_l, \mathbf{x}_i^{(2)}). \tag{4}$$

In this study, we assume that topics are evenly distributed in database. Thus, we set all weights of hyperedges to 1, $w(\mathbf{t}_l) = 1$. The degree of each vertex is defined as

$$d(\mathbf{x}_i^{(2)}) = \sum_{l=1}^{L} w(\mathbf{t}_l)\mathbf{H}(\mathbf{t}_l, \mathbf{x}_i^{(2)}) = \sum_{l=1}^{L} \mathbf{H}(\mathbf{t}_l, \mathbf{x}_i^{(2)}). \tag{5}$$

Principally, images that belong to more same hyperedges will describe identical semantic concepts with greater probability. Therefore, they should be mapped into near points in Hamming space. Formally, to generate effective hash codes, which can accurately measure image distance with semantic similarity, we derive the following formula

$$\begin{aligned} &\min_{\mathbf{Y}} \ \mathbf{Tr}(\mathbf{Y}\mathbf{L}_{THG}\mathbf{Y}^{\mathrm{T}}) \\ &s.t.\ \mathbf{Y} \in [-1, 1]^{L \times N}, \mathbf{Y}\mathbf{Y}^{\mathrm{T}} = N\mathbf{I}, \mathbf{Y}\mathbf{1} = \mathbf{0} \end{aligned} \tag{6}$$

where $\mathbf{Y}$ is the hash codes of images. The constraint $\mathbf{Y}\mathbf{Y}^{\mathrm{T}} = N\mathbf{I}$ is to guarantee that the learned hash bits to be uncorrelated, and $\mathbf{Y}\mathbf{1} = \mathbf{0}$ enforces each bit to appear with equal possibility as positive or negative. $\mathbf{L}_{THG} \in \mathbb{R}^{N \times N}$ denotes the Laplacian matrix of topic hypergraph, which can be calculated as

$$\mathbf{L}_{THG} = \mathbf{I} - \mathbf{D}_v^{-1/2}\mathbf{H}\mathbf{D}_w\mathbf{D}_e^{-1}\mathbf{H}^T\mathbf{D}_v^{-1/2} \tag{7}$$

where $\mathbf{D}_v$, $\mathbf{D}_e$, and $\mathbf{D}_w$ are the diagonal matrices of the vertex degrees, edge degrees, and hyperedge weights, respectively.

2. *Assistance on correlating images and latent shared semantic topics*. Images and their auxiliary texts jointly describe the same latent topics. These shared topics can be detected effectively with the assistance of auxiliary texts. Generally, semantically similar images will possess similar topic distribution. Therefore, it is reasonable that visual hash codes which measure semantic similarity in Hamming space keep consistent with shared topic distributions. Moreover, we can assume that each hash bit describes a latent shared topic. In this way, hash codes of images can actually reflect latent correlations between images and shared topics, or visual distribution of images on shared topics. Hence, it is promising to preserve correlations of images and shared topics into hash codes to enhance semantic descriptive capability.

In this study, we adopt a simple but effective collective matrix factorization [50] to detect shared semantic topics and directly consider the shared topic distributions as hash codes $\mathbf{Y}$. Collective matrix factorization performs well on discovering common parts across heterogeneous modalities. The formal formulation of this process can be represented as

$$\begin{aligned} \min_{\mathbf{Y},\mathbf{U}^{(1)},\mathbf{U}^{(2)}} \ &||\mathbf{X}^{(1)} - \mathbf{U}^{(1)}\mathbf{Y}||_F^2 + \eta||\mathbf{X}^{(2)} - \mathbf{U}^{(2)}\mathbf{Y}||_F^2 \\ s.t.\ &\mathbf{Y} \in [-1, 1]^{L \times N}, \mathbf{Y}\mathbf{Y}^{\mathrm{T}} = N\mathbf{I}, \mathbf{Y}\mathbf{1} = \mathbf{0}, \eta > 0 \end{aligned} \tag{8}$$

where $\mathbf{U}^{(1)} \in \mathbb{R}^{d_1 \times L}$ and $\mathbf{U}^{(2)} \in \mathbb{R}^{d_2 \times L}$ are basis matrices of visual and text space respectively, $\eta > 0$ is adjustment factor which provides a balance between two terms.

3.5 Overall Objective Function and Optimization

After comprehensively considering visual similarity preservation and the assistance from auxiliary text, we obtain the overall objective function of hash code learning. Its form is

$$\begin{aligned} \min_{\mathbf{Y},\mathbf{U}^{(1)},\mathbf{U}^{(2)}} \ &\Omega(\mathbf{Y}) = ||\mathbf{X}^{(1)} - \mathbf{U}^{(1)}\mathbf{Y}||_F^2 + \eta||\mathbf{X}^{(2)} - \mathbf{U}^{(2)}\mathbf{Y}||_F^2 \\ &+ \lambda\mathbf{Tr}(\mathbf{Y}(\mathbf{L}_G + \alpha\mathbf{L}_{THG})\mathbf{Y}^{\mathrm{T}}) \\ &s.t.\ \mathbf{Y} \in [-1, 1]^{L \times N}, \mathbf{Y}\mathbf{Y}^{\mathrm{T}} = N\mathbf{I}, \mathbf{Y}\mathbf{1} = \mathbf{0}, \eta, \lambda, \alpha > 0 \end{aligned} \tag{9}$$

where η, λ, α are factors which adjust the assistance of auxiliary text on visual hashing. However, solving the above problem is still NP-hard due to the discrete constraints. To make the problem solvable, we relax the discrete constraint and balance constraint $\mathbf{Y}\mathbf{1} = \mathbf{0}$ as recent literature [19], [23], [25]. We first obtain real values of $\mathbf{Y}$, and then binarize it to hash codes via mean thresholding. Note that, we ensure bit-uncorrelated constraint during hash code learning. This design can facilitate learning process (as shown in Eq. (17)) and generate the learned hash codes with less redundancy. With relaxation, the objective formulation is transformed as

$$\begin{aligned} \min_{\mathbf{Y},\mathbf{U}^{(1)},\mathbf{U}^{(2)}} \ &\Omega(\mathbf{Y}) = ||\mathbf{X}^{(1)} - \mathbf{U}^{(1)}\mathbf{Y}||_F^2 + \eta||\mathbf{X}^{(2)} - \mathbf{U}^{(2)}\mathbf{Y}||_F^2 \\ &+ \lambda\mathbf{Tr}(\mathbf{Y}\mathbf{L}\mathbf{Y}^{\mathrm{T}}) \\ &s.t.\ \mathbf{Y}\mathbf{Y}^{\mathrm{T}} = N\mathbf{I}, \mathbf{L} = \mathbf{L}_G + \alpha\mathbf{L}_{THG}, \eta, \lambda, \alpha > 0. \end{aligned} \tag{10}$$

As shown in Eq. (10), due to the orthogonal constraints $\mathbf{Y}\mathbf{Y}^{\mathrm{T}} = N\mathbf{I}$, the objective function is not convex to $\mathbf{Y}$, $\mathbf{U}^{(1)}$,

and $\mathbf{U}^{(2)}$. This study proposes an optimization algorithm based on augmented Lagrangian multiplier [26] to calculate the optimal solution. It has shown desirable efficiency and effectiveness in many matrix-based learning problems. Its core idea is adding auxiliary variables to eliminate equality constraints, and simultaneously minimizing the loss brought by infeasible points. In particular, three auxiliary variables $\mathbf{A}^{(1)}$, $\mathbf{A}^{(2)}$, $\mathbf{B}$ are added,

$$\mathbf{A}^{(1)} = \mathbf{X}^{(1)} - \mathbf{U}^{(1)}\mathbf{Y}, \mathbf{A}^{(2)} = \mathbf{X}^{(2)} - \mathbf{U}^{(2)}\mathbf{Y}, \mathbf{B} = \mathbf{Y}. \tag{11}$$

The objective function is transformed as

$$\begin{aligned}\min_{\mathbf{Y},\mathbf{U}^{(1)},\mathbf{U}^{(2)}} \Omega(\mathbf{Y}) = &||\mathbf{A}^{(1)}||_F^2 + ||\mathbf{A}^{(2)}||_F^2 + \frac{\mu}{2}(||\mathbf{X}^{(1)} - \mathbf{U}^{(1)}\mathbf{Y} \\ &- \mathbf{A}^{(1)} + \frac{\mathbf{E}^{(1)}}{\mu}||_F^2 + \eta||\mathbf{X}^{(2)} - \mathbf{U}^{(2)}\mathbf{Y} - \mathbf{A}^{(2)} \\ &+ \frac{\mathbf{E}^{(2)}}{\mu}||_F^2) + \lambda\mathbf{Tr}(\mathbf{Y}\mathbf{L}\mathbf{B}^{\mathrm{T}}) + \frac{\mu}{2}||\mathbf{Y} - \mathbf{B} + \frac{\mathbf{E}^{(3)}}{\mu}||_F^2 \\ s.t.\ &\mathbf{Y}\mathbf{Y}^{\mathrm{T}} = N\mathbf{I}, \mathbf{L} = \mathbf{L}_G + \alpha\mathbf{L}_{THG}, \eta, \lambda, \alpha, \mu > 0\end{aligned} \tag{12}$$

where $\mathbf{E}^{(1)} \in \mathbb{R}^{d_1 \times N}$, $\mathbf{E}^{(2)} \in \mathbb{R}^{d_2 \times N}$, $\mathbf{E}^{(3)} \in \mathbb{R}^{L \times N}$ measure the gap between target variables and auxiliary variables, μ adjusts the balance between terms. We adopt alternate optimization to solve the above problem iteratively. In particular, we optimize the objective function with respective to one variable while fixing other remaining variables. The key steps for solving $\mathbf{Y}$ are summarized in Algorithm 1.

Algorithm 1. Solving $\mathbf{Y}$ via optimizing (12)

Input:
Feature representations of image, $\mathbf{X}^{(1)}$, $\mathbf{X}^{(2)}$.
Output:
Relaxed hash codes $\mathbf{Y}$.
1: Initialize $\mathbf{E}^{(1)}$, $\mathbf{E}^{(2)}$, $\mathbf{E}^{(3)}$, $\mathbf{Y}$, $\mathbf{U}^{(1)}$, $\mathbf{U}^{(2)}$;
2: **while** not convergence **do**
3: Optimize $\mathbf{A}^{(i)}, i = 1, 2$ while fixing the others;
4: Optimize $\mathbf{U}^{(i)}, i = 1, 2$ while fixing the others;
5: Optimize $\mathbf{B}$ while fixing the others;
6: Optimize $\mathbf{Y}$ while fixing the others;
7: Update $\mathbf{E}^{(1)}$, $\mathbf{E}^{(2)}$, $\mathbf{E}^{(3)}$, μ while fixing the others;
8: **end while**

In the below, **Step 3-7** are introduced in detail.

Step 3. Optimize $\mathbf{A}^{(i)}, i = 1, 2$. The objective function with respective to $\mathbf{A}^{(i)}$ can be represented as

$$\min_{\mathbf{A}^{(i)}} ||\mathbf{A}^{(i)}||_F^2 + \frac{\mu}{2}||\mathbf{X}^{(i)} - \mathbf{U}^{(i)}\mathbf{Y} - \mathbf{A}^{(i)} + \frac{\mathbf{E}^{(i)}}{\mu}||_F^2. \tag{13}$$

By calculating the derivative of the objective function with respective to $\mathbf{A}^{(i)}$, and setting it to $\mathbf{0}$, we can obtain that

$$\begin{aligned}&2\mathbf{A}^{(i)} - \mu\left(\mathbf{X}^{(i)} - \mathbf{U}^{(i)}\mathbf{Y} - \mathbf{A}^{(i)} + \frac{\mathbf{E}^{(i)}}{\mu}\right) = \mathbf{0} \\ &\Rightarrow \mathbf{A}^{(i)} = \frac{\mu\mathbf{X}^{(i)} - \mu\mathbf{U}^{(i)}\mathbf{Y} + \mathbf{E}^{(i)}}{2 + \mu}.\end{aligned} \tag{14}$$

Step 4. Optimize $\mathbf{U}^{(i)}, i = 1, 2$. The objective function with respective to $\mathbf{U}^{(i)}$ can be represented as

$$\min_{\mathbf{U}^{(i)}} ||\mathbf{X}^{(i)} - \mathbf{U}^{(i)}\mathbf{Y} - \mathbf{A}^{(i)} + \frac{\mathbf{E}^{(i)}}{\mu}||_F^2. \tag{15}$$

By calculating the derivative of the objective function with respective to $\mathbf{U}^{(i)}$, and setting it to $\mathbf{0}$, we can obtain

$$\begin{aligned}&2\left(\mathbf{X}^{(i)} - \mathbf{U}^{(i)}\mathbf{Y} - \mathbf{A}^{(i)} + \frac{\mathbf{E}^{(i)}}{\mu}\right)\mathbf{Y}^{\mathrm{T}} = \mathbf{0} \\ &\Rightarrow \mathbf{U}^{(i)}\mathbf{Y} = \mathbf{X}^{(i)} - \mathbf{A}^{(i)} + \frac{\mathbf{E}^{(i)}}{\mu}.\end{aligned} \tag{16}$$

Since $\mathbf{Y}\mathbf{Y}^{\mathrm{T}} = N\mathbf{I}$, we can derive that

$$\mathbf{U}^{(i)} = \frac{1}{N}\left(\mathbf{X}^{(i)} - \mathbf{A}^{(i)} + \frac{\mathbf{E}^{(i)}}{\mu}\right)\mathbf{Y}^{\mathrm{T}}. \tag{17}$$

Step 5. Optimize $\mathbf{B}$. The objective formulation for optimizing with respective to $\mathbf{B}$ can be represented as

$$\min_{\mathbf{B}} \lambda\mathbf{Tr}(\mathbf{Y}\mathbf{L}\mathbf{B}^{\mathrm{T}}) + \frac{\mu}{2}||\mathbf{Y} - \mathbf{B} + \frac{\mathbf{E}^{(3)}}{\mu}||_F^2. \tag{18}$$

With transformation, the objective function for optimizing $\mathbf{B}$ can be rewritten as

$$\begin{aligned}&\min_{\mathbf{B}} \lambda\mathbf{Tr}(\mathbf{Y}\mathbf{L}\mathbf{B}^{\mathrm{T}}) + \frac{\mu}{2}\mathbf{Tr}\left(\left(\mathbf{Y} - \mathbf{B} + \frac{\mathbf{E}^{(3)}}{\mu}\right)^{\mathrm{T}}\left(\mathbf{Y} - \mathbf{B} + \frac{\mathbf{E}^{(3)}}{\mu}\right)\right) \\ &\Leftrightarrow \min_{\mathbf{B}} \frac{\lambda}{\mu}\mathbf{Tr}(\mathbf{Y}\mathbf{L}\mathbf{B}^{\mathrm{T}}) + \mathbf{Tr}\left(-\mathbf{Y}\mathbf{B}^{\mathrm{T}} - \frac{\mathbf{E}^{(3)}}{\mu}\mathbf{B}^{\mathrm{T}} + \mathbf{B}^{\mathrm{T}}\mathbf{B}\right) \\ &\Leftrightarrow \min_{\mathbf{B}} \mathbf{Tr}\left(\left(\frac{\lambda}{\mu}\mathbf{Y}\mathbf{L} - \mathbf{Y} - \frac{\mathbf{E}^{(3)}}{\mu} + \mathbf{B}\right)\mathbf{B}^{\mathrm{T}}\right) \\ &\Leftrightarrow \min_{\mathbf{B}} \mathbf{Tr}\left(\mathbf{B} - \left(\mathbf{Y} + \frac{\mathbf{E}^{(3)}}{\mu} - \frac{\lambda}{\mu}\mathbf{Y}\mathbf{L}\right)\right)^{\mathrm{T}}\left(\mathbf{B} - \left(\mathbf{Y} + \frac{\mathbf{E}^{(3)}}{\mu} - \frac{\lambda}{\mu}\mathbf{Y}\mathbf{L}\right)\right) \\ &\Leftrightarrow \min_{\mathbf{B}} \left\|\mathbf{B} - \left(\mathbf{Y} + \frac{\mathbf{E}^{(3)}}{\mu} - \frac{\lambda}{\mu}\mathbf{Y}\mathbf{L}\right)\right\|_F^2.\end{aligned} \tag{19}$$

The optimal solution of $\mathbf{B}$ can be derived as

$$\mathbf{B} = \mathbf{Y} + \frac{\mathbf{E}^{(3)}}{\mu} - \frac{\lambda}{\mu}\mathbf{Y}\mathbf{L}. \tag{20}$$

Step 6. Optimize $\mathbf{Y}$. The objective function with respective to $\mathbf{Y}$ can be represented as

$$\begin{aligned}\min_{\mathbf{Y}} \frac{\mu}{2}&\left(\left\|\mathbf{X}^{(1)} - \mathbf{U}^{(1)}\mathbf{Y} - \mathbf{A}^{(1)} + \frac{\mathbf{E}^{(1)}}{\mu}\right\|_F^2 + \eta\left\|\mathbf{X}^{(2)} - \mathbf{U}^{(2)}\mathbf{Y}\right. \right. \\ &\left. \left. - \mathbf{A}^{(2)} + \frac{\mathbf{E}^{(2)}}{\mu}\right\|_F^2\right) + \lambda\mathbf{Tr}(\mathbf{Y}\mathbf{L}\mathbf{B}^{\mathrm{T}}) + \frac{\mu}{2}\left\|\mathbf{Y} - \mathbf{B} + \frac{\mathbf{E}^{(3)}}{\mu}\right\|_F^2 \\ s.t.\ &\mathbf{Y}\mathbf{Y}^{\mathrm{T}} = N\mathbf{I}.\end{aligned} \tag{21}$$

With transformation, the objective function for optimizing $\mathbf{Y}$ can be rewritten as

$$\min_{\mathbf{Y}} -\mu\mathbf{Tr}\left(\mathbf{Y}^{\mathrm{T}}\mathbf{U}^{(1)}\left(\mathbf{X}^{(1)}-\mathbf{A}^{(1)}+\frac{\mathbf{E}^{(1)}}{\mu}\right)\right)-\mu\eta\mathbf{Tr}\left(\mathbf{Y}^{\mathrm{T}}\mathbf{U}^{(2)}\left(\mathbf{X}^{(2)}-\mathbf{A}^{(2)}+\frac{\mathbf{E}^{(2)}}{\mu}\right)\right)+\lambda\mathbf{Tr}(\mathbf{Y}^{\mathrm{T}}\mathbf{B}\mathbf{L})-\mu\mathbf{Tr}\left(\mathbf{Y}^{\mathrm{T}}\left(\mathbf{B}-\frac{\mathbf{E}^{(3)}}{\mu}\right)\right)$$
$$\Leftrightarrow \min_{\mathbf{Y}} -\mathbf{Tr}(\mathbf{Y}^{\mathrm{T}}\mathbf{C}) \tag{22}$$

where $\mathbf{C}=\mathbf{B}-\frac{\mathbf{E}^{(3)}}{\mu}-\frac{\lambda}{\mu}\mathbf{B}\mathbf{L}+(\mathbf{U}^{(1)})^{\mathrm{T}}(\mathbf{X}^{(1)}-\mathbf{A}^{(1)}+\frac{\mathbf{E}^{(1)}}{\mu})+\eta(\mathbf{U}^{(2)})^{\mathrm{T}}(\mathbf{X}^{(2)}-\mathbf{A}^{(2)}+\frac{\mathbf{E}^{(2)}}{\mu})$. The Eq. (21) is equivalent to the following optimization problem

$$\max_{\mathbf{Y}} \ \mathbf{Tr}(\mathbf{Y}^{\mathrm{T}}\mathbf{C}) \quad s.t.\ \mathbf{Y}\mathbf{Y}^{\mathrm{T}}=N\mathbf{I}. \tag{23}$$

With singular value decomposition [51], $\mathbf{C}$ can be decomposed as

$$\mathbf{C}=\mathbf{P}\Lambda\mathbf{Q}^{\mathrm{T}} \tag{24}$$

where Λ is rectangular diagonal matrix and its diagonal entries are singular values of $\mathbf{C}$, the columns of $\mathbf{P}$ and $\mathbf{Q}$ are left-singular vectors and right-singular vectors of $\mathbf{C}$, respectively. Then, the optimizing formulation for $\mathbf{Y}$ can be transformed to

$$\max_{\mathbf{Y}} \ \mathbf{Tr}(\mathbf{Y}^{\mathrm{T}}\mathbf{P}\Lambda\mathbf{Q}^{\mathrm{T}}) \Leftrightarrow \max_{\mathbf{Y}} \ \mathbf{Tr}(\Lambda\mathbf{Q}^{\mathrm{T}}\mathbf{Y}^{\mathrm{T}}\mathbf{P}). \tag{25}$$

Theorem 1. *Given any matrix* $\mathbf{Z}\mathbf{Z}^{\mathrm{T}}=N\mathbf{I}$ *and diagonal matrix* $\Lambda\geq\mathbf{0}$, *the optimal solution of* $\max_{\mathbf{Z}}\mathbf{Tr}(\Lambda\mathbf{Z})$ *is* $\mathbf{Z}=\mathtt{diag}(\sqrt{N})$.

Proof. Assuming λ_{ii} and z_{ii} are the ith diagonal entry of Λ and $\mathbf{Z}$ respectively, $\mathbf{Tr}(\Lambda\mathbf{Z})=\sum_i\lambda_{ii}z_{ii}$. Since $\mathbf{Z}\mathbf{Z}^{\mathrm{T}}=N\mathbf{I}$, $z_{ii}\leq\sqrt{N}$. $\mathbf{Tr}(\Lambda\mathbf{Z})=\sum_i\lambda_{ii}z_{ii}\leq\sqrt{N}\sum_i\lambda_{ii}$. The equality holds only when $z_{ii}=\sqrt{N}, z_{ij}=0,\forall i,j)$. This is to say, $\mathbf{Tr}(\Lambda\mathbf{Z})$ reaches its maximum when $\mathbf{Z}=\mathtt{diag}(\sqrt{N})$. □

$\Lambda\geq\mathbf{0}$ as Λ is calculated by Eq. (24). On other hand, we can easily derive that $\mathbf{Q}^{\mathrm{T}}\mathbf{Y}^{\mathrm{T}}\mathbf{P}\mathbf{P}^{\mathrm{T}}\mathbf{Y}\mathbf{Q}=N\mathbf{I}$. Therefore, according to **Theorem** 1, the optimal $\mathbf{Y}$ in Eq. (25) can only be obtained when $\mathbf{Q}^{\mathrm{T}}\mathbf{Y}^{\mathrm{T}}\mathbf{P}=\mathtt{diag}(\sqrt{N})$. Hence, the optimal solution of $\mathbf{Y}$ can be represented as

$$\mathbf{Y}=\sqrt{N}\mathbf{P}\mathbf{Q}^{\mathrm{T}}. \tag{26}$$

Step 7. Updating $\mathbf{E}^{(1)}$, $\mathbf{E}^{(2)}$, $\mathbf{E}^{(3)}$, μ. The update rules are

$$\begin{aligned}\mathbf{E}^{(i)}&=\mathbf{E}^{(i)}+\mu(\mathbf{X}^{(i)}-\mathbf{U}^{(i)}\mathbf{Y}-\mathbf{A}^{(i)}), i=1,2\\ \mathbf{E}^{(3)}&=\mathbf{E}^{(3)}+\mu(\mathbf{Y}-\mathbf{B}),\\ \mu&=\rho\mu\end{aligned} \tag{27}$$

where $\rho>1$ is learning rate which controls the convergence.

It is worth mentioning that, the above objective function and optimization strategy differ from the traditional multi-graph regularized non-negative matrix factorization approaches [52], [53]: (1) The graph regularizer is constructed in our study with different motivations and intrinsic meanings. The visual graph is used to preserve the visual information, which is important to the performance of CBIR. The topic hypergraph is proposed to incorporate the auxiliary semantics to assist visual hashing. (2) In our study, an orthogonal constraint $\mathbf{Y}\mathbf{Y}^{\mathrm{T}}=N\mathbf{I}$ is guaranteed in the whole optimization process. The advantage of this design on reducing the redundancy of hash bits is validated in our experiment. In contrast, [52] is designed without imposing any orthogonal constraint, [53] transforms the orthogonal constraint to a soft one. (3) [52] and [53] follow the traditional way of non-negative matrix factorization to solve the problem. Different from them, our study proposes an effective optimization approach based on ALM to iteratively calculate the optimal solution. (4) Our formulation is specially designed to leverage the semantics in auxiliary texts to assist unsupervised visual hashing in CBIR. In contrast, [52] and [53] are proposed for general data representation in continuous feature space and cross-modal hashing, respectively.

3.6 Hash Function Learning

We leverage linear projection to learn hash functions for its high efficiency on online retrieval. The learning objective is

$$\min_{\mathbf{W}}\Phi(\mathbf{W})=\min_{\mathbf{W}}||\mathbf{Y}-\mathbf{W}^{\mathrm{T}}\mathbf{X}^{(1)}||_F^2+\xi||\mathbf{W}||_F^2 \tag{28}$$

where $\mathbf{W}\in\mathbb{R}^{d_1\times L}$ denotes the projection matrix. Note that, only visual feature $\mathbf{X}^{(1)}$ is used in (28). The main objective is to reduce the loss between the hash codes and the projected ones. $||\mathbf{Y}-\mathbf{W}^{\mathrm{T}}\mathbf{X}^{(1)}||_F^2$ is the loss term and $||\mathbf{W}||_F^2$ is to avoid over-fitting. $\xi>0$ balances these two terms. By calculating the derivative of $\Phi(\mathbf{W})$ with respect to $\mathbf{W}$ and set it to $\mathbf{0}$, we can obtain

$$\mathbf{W}=\left(\mathbf{X}^{(1)}(\mathbf{X}^{(1)})^{\mathrm{T}}+\xi\mathbf{I}\right)^{-1}\mathbf{X}^{(1)}\mathbf{Y}^{\mathrm{T}}. \tag{29}$$

The hash functions can be constructed as

$$\mathbf{F}(\mathbf{x})=\frac{\mathtt{sgn}(\mathbf{W}^{\mathrm{T}}\mathbf{x}-\mathbf{b})+1}{2}, \mathbf{b}=\frac{\mathbf{W}^{\mathrm{T}}\mathbf{X}^{(1)}\mathbf{1}}{N}. \tag{30}$$

Algorithm 2. Summary of Semantic-Assisted Visual Hashing

Input:
Database images: $\{\mathbf{I}_n\}_{n=1}^N$, query image q.
Output:
Hash codes of database images: $\mathbf{Y}$, hash functions: $\mathbf{F}$.
Image retrieval results for image query q.
Offline Learning
1: Extract features of database images, obtaining $\mathbf{X}^{(1)}, \mathbf{X}^{(2)}$;
2: Compute visual graph Laplacian matrix $\mathbf{L}_G$;
3: Compute topic hypergraph Lalacian matrix $\mathbf{L}_{THG}$ via Eq. (7);
4: Learn relaxed hash codes via Algorithm 1;
5: Construct hash functions $\mathbf{F}$ via Eq. (30);
6: Project database images into binary hash codes with $\mathbf{F}$;
Online Hashing
7: Extract visual feature of query image;
8: Project query visual feature into hash codes via Eq. (30);
9: Calculate the Hamming distances between hash codes of query image and that of database images;
10: Rank Hamming distances and return retrieval results.

3.7 Complexity Analysis

This section provides time complexity analysis of the hashing learning. The main procedures of SAVH-based CBIR are summarized in Algorithm 2. In offline training, the computations of graph Laplacian matrix $\mathbf{L}_G$ and topic

TABLE 3
Statistics of Test Collections

Datasets	Wiki	MIR Flickr	NUS-WIDE
Database Size	2,866	25,000	186,643
Query Size	144	250	1,867
Training Size	287	750	5,540
Visual Modality	BoVW (128-Dim)	BoVW (1,000-Dim)	BoVW (500-Dim)
Text Modality	Text Topics (10-Dim)	BoTW (457-Dim)	BoTW (1,000-Dim)

BoVW and BoTW denote bag-of-visual-words and bag-of-textual-words, respectively.

hypergraph Laplacian matrix $\mathbf{L}_{THG}$ consume $O(N^2)$. It is worth noting that these processes are conducted in offline part and consume one-time cost. Besides, they enjoy high degree of parallelism and can be efficiently implemented with advanced parallel computing techniques. To learn the relaxed hash codes, the computational complexity is $O(iter \times (d_1 \times N + d_2 \times N + d_1 \times L + d_2 \times L + L \times N))$, where $iter$ denotes the number of iterations in Algorithm 1. Given $N \gg d_1(d_2) > L$, this process scales linearly with N. The computation of hash functions solves a linear system, whose time complexity is $O(N)$. Calculation of hash codes of database images costs $O(N)$. In online hashing, generating hash code for a query can be completed in $O(d_1 L + L)$, which is quite efficient.

4 Experimental Configuration

4.1 Experimental Datasets

In this study, comprehensive experiments are conducted on three publicly available image datasets: *Wiki* [54], *MIR Flickr* [55], and *NUS-WIDE* [56], to empirically validate the effectiveness of SAVH. All datasets consist of image-text pairs, and they are widely used for evaluating performance of multimedia retrieval in the past work [23], [19], [25]. Following the same setting, each dataset is partitioned into query set, learning set, and database set.[3] This experimental setting matches the practical application scenarios of CBIR, where queries are out of database, and continuously flowing into database when time passes by. Table 3 summarizes key statistics of the test collections.

- *Wiki*[4] consists of 2,866 multimedia documents in 10 semantic categories, which are collected from Wikipedia.[5] Visual contents are represented by 128 dimensional SIFT [57] histogram and text contents are represented by 10 dimensional topic vector generated by latent Dirichlet allocation [58]. For *Wiki* dataset, since images are labelled into 10 independent categories, images in this dataset are considered to be relevant only if they belong to the same class.
- *MIR Flickr*[6] contains 25,000 images from 38 categories from the Flickr.[7] Each image is associated with tags. The tags that appear less than 50 times are removed, resulting in a vocabulary of 457 tags. Visual contents of images in *MIR Flickr* are represented by 1,000 dimensional dense SIFT histogram. Text contents are represented by 457 dimensional binary vector, and each dimension describes the presence of 457 tags. Since images in *MIR Flickr* are generally labeled by several tags, they are considered to be relevant only if they share at least one concept.
- *NUS-WIDE*[8] is comprised of 269,648 images labeled into 81 concepts. In experiments, we preserve 10 most common concepts and the corresponding 186,577 pairs. Each image is associated with tags. On *NUS-WIDE* dataset, images are represented by 500 dimensional SIFT histogram. Text features are 1,000 dimensional binary vectors describing the presence of 1,000 tags. Similarly, images are considered to be relevant if they share at least one concept.

4.2 Evaluation Metrics

In our experimental study, mean average precision (mAP) is adopted as the evaluation metric. The metric has been widely used in literature [23], [19]. For a given query, average precision (AP) is calculated as $AP = \frac{1}{NR}\sum_{r=1}^{R} \psi(r)\varphi(r)$, where R is the total number of retrieved images, NR is the number of relevant images in retrieved set, $\psi(r)$ denotes the precision of top r retrieval images, which is defined as the ratio between the number of the relevant images and the number of retrieved images r, and $\varphi(r)$ is indicator function which equals to 1 if the rth image is relevant to query, and vice versa. mAP is defined as the average of the AP of all queries. Larger mAP means the retrieval performance is better. In experiments, we set R as 100 to collect results. Furthermore, *Precision-Scope* curve is also reported to reflect the retrieval performance variations with respect to the number of retrieved images.

4.3 Compared Approaches

SAVH is specially designed for CBIR without any labeled images. Therefore, for comparison fairness, we compare SAVH with several state-of-the-art unsupervised SFVH and UCMH approaches. More specifically, SFVH approaches used for comparison include: spectral hashing [7], shift-invariant kernel locality sensitive hashing (SKLSH) [27], anchor graph hashing (1-layer) [9], self-taught hashing [8], iterative quantization [30], UCMH approaches used for evaluation include[9]:

- Cross-view hashing [18]. It learns hash functions by jointly minimizing Hamming distances of similar samples and maximizing that of dissimilar samples.
- Composite hashing with multiple information sources (CHMIS) [46]. It integrates multiple modalities into the binary hash codes with proper weights. For comparison fairness, text input is removed and only visual input is preserved in CHMIS. In this case, CHMIS can also be considered as UCMH.

3. In SAVH, the hash codes learned on learning set are all discarded after hash function learning. It leverages the constructed hash functions to generate hash codes for query and database images.
4. http://www.svcl.ucsd.edu/projects/crossmodal/
5. https://www.wikipedia.org/
6. http://lear.inrialpes.fr/people/guillaumin/data.php
7. https://www.flickr.com/
8. http://lms.comp.nus.edu.sg/research/NUS-WIDE.htm
9. For implementation of CVH, we use the code provided by [59]. For SPH, SKLSH, STH, AGH, ITQ, CHMIS, IMH, LSSH, CMFH, we directly download implementation codes from author websites.

TABLE 4
mAP of Compared Unsupervised Hashing Approaches

Methods	Wiki				MIR Flickr				NUS-WIDE			
	16 bits	32 bits	64 bits	128 bits	16 bits	32 bits	64 bits	128 bits	16 bits	32 bits	64 bits	128 bits
SPH	0.1644	0.1740	0.1777	0.1829	0.5976	0.6093	0.6269	0.6433	0.3430	0.3910	0.4415	0.4583
SKLSH	0.1542	0.1671	0.1664	0.1776	0.5624	0.5927	0.6084	0.6280	0.3685	0.3695	0.3680	0.3673
AGH	0.1699	0.1833	0.1828	0.1809	0.6204	0.6231	0.6321	0.6311	0.4646	0.4681	0.4711	0.4731
STH	0.1614	0.1752	0.1837	0.1819	0.6196	0.6258	0.6336	0.6313	0.4120	0.4368	0.4443	0.4606
ITQ	0.1607	0.1649	0.1813	0.1817	0.6442	0.6403	0.6542	0.6555	0.4482	0.4667	0.4860	0.4859
CVH	0.1651	0.1716	0.1766	0.1754	0.5511	0.5583	0.5566	0.5663	0.4447	0.4300	0.4233	0.4149
CHMIS	0.1492	0.1626	0.1755	0.1783	0.5585	0.5612	0.5659	0.5851	0.4419	0.4347	0.4302	0.4265
IMH	0.1676	0.1827	0.1760	0.1831	0.6285	0.6338	0.6454	0.6586	0.4475	0.4618	0.4634	0.4879
LSSH	0.1658	0.1722	0.1749	0.1870	0.5917	0.5875	0.6038	0.6378	0.4449	0.4615	0.4849	0.5002
CMFH	0.1612	0.1677	0.1696	0.1672	0.5733	0.5614	0.5723	0.5735	0.4703	0.4883	0.4942	0.4882
SAVH	**0.1748**	**0.1880**	**0.1914**	**0.1991**	**0.6450**	**0.6511**	**0.6680**	**0.6704**	**0.4962**	**0.5103**	**0.5193**	**0.5281**

The best result in each column is marked with bold.

- Inter-media hashing [19]. It formulates hash function learning in a framework where intra-similarity of each individual modality and inter-correlations between different modalities are both preserved in hash codes.
- Latent semantic sparse hashing [23] performs cross-modal similarity search in a joint abstract semantic space learned by employing sparse coding and matrix factorization for semantic projection.
- Collective matrix factorization hashing [25] learns a latent semantic subspace shared by multiple modalities. In CMFH, both visual and text features are mapped into a unified hash codes.

Note that, CVH, CHMIS, IMH, LSSH, and CMFH can generate hash codes for both query visual image and text. Since the aim of experiment is to test the performance of CBIR, we remove hash codes of text. In this case, the whole retrieval process of CBIR in all compared approaches is performed in binary visual Hamming space. All parameters in compared approaches are adjusted according to the relevant literature and report the best performance.

4.4 Implementation Details

In experiments, we adopt five folds cross-validation to choose parameters. More specifically, the best performance of SAVH is achieved when k is set to 7, 5, 8 on *Wiki*, *MIR Flickr*, and *NUS-WIDE* respectively. Furthermore, SAVH has three parameters: α, λ, and η in Eq. (9), which are used to adjust the assistance of auxiliary text on visual hashing. In particular, the best performance is achieved when $\{\alpha = 1000, \lambda = 1, \eta = 10\}$, $\{\alpha = 0.01, \lambda = 1, \eta = 5\}$, $\{\alpha = 1, \lambda = 0.0001, \eta = 100\}$ on *Wiki*, *MIR Flickr*, and *NUS-WIDE* respectively. The parameters μ and ρ in Eq. (12) and Eq. (27) are used for ALM optimization. The optimal performance is obtained when $\{\mu = 0.01, \rho = 5\}, \{\mu = 0.001, \rho = 2\}, \{\mu = 0.0001, \rho = 2\}$ on *Wiki*, *MIR Flickr*, and *NUS-WIDE* respectively. ξ is used in Eq. (28) to learn hash functions. The optimal performance is obtained when ξ is set to 0.1, 100, 100 on *Wiki*, *MIR Flickr*, and *NUS-WIDE* respectively.

In experiments, hash code length L on all datasets is varied in the range of $[16, 32, 64, 128]$ to observe the performance. The retrieval scope on *Wiki* and *MIR Flickr* is set from 100 to 1,000 with step size 100, that on *NUS-WIDE* is set from 500 to 5000 with step size 500. In the step 1 of Algorithm 1, the initial values of $\mathbf{E}^{(1)}$, $\mathbf{E}^{(2)}$, $\mathbf{E}^{(3)}$ are set to $\mathbf{0}$. The value of $\mathbf{U}^{(1)}$, $\mathbf{U}^{(2)}$, and $\mathbf{Y}$ are obtained by solving a simple matrix factorization problem: $\min_{\mathbf{Y}} ||\mathbf{X}^{(1)} - \mathbf{U}^{(1)}\mathbf{Y}||_F^2 + \eta||\mathbf{X}^{(2)} - \mathbf{U}^{(2)}\mathbf{Y}||_F^2$. All the experiments are conducted on a computer with Intel Xeon(R) CPU E5-2620 2.0 GHz and 32 GB RAM.

5 Results and Discussions

5.1 Comparisons with Unsupervised Hashing

The mAP results of SAVH and all compared approaches on different code lengths and datasets are reported in Table 4. The *Precision-Scope* curves on three datasets are shown in Figs. 6, 7, and 8 respectively. According to the presented results, we can clearly observe that SAVH outperforms the compared approaches. For example, on *NUS-WIDE*, the highest mAP of SAVH is 52.81 percent, which is more than 2.5 percent better than the second best mAP 50.02 percent achieved by LSSH. Besides, we can obtain several insightful observations:

- In SAVH, retrieval performance increases steadily with hash code length. However, for many compared

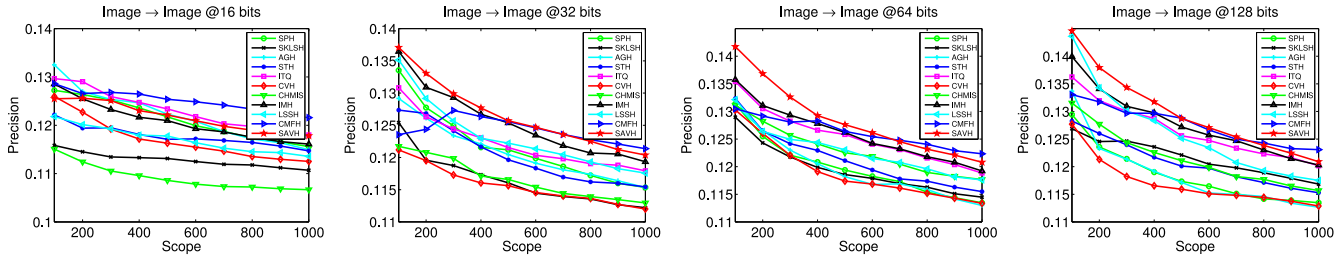


Fig. 6. *Precision-Scope* curves on *Wiki* varying code length.

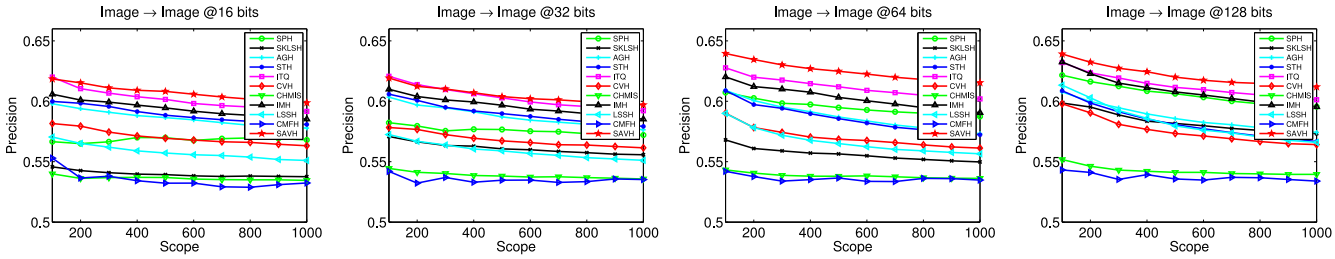


Fig. 7. *Precision-Scope* curves on *MIR Flickr* varying code length.

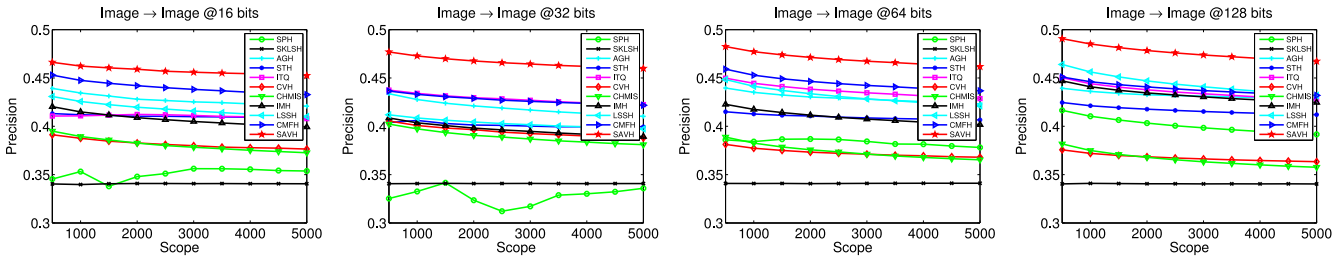


Fig. 8. *Precision-Scope* curves on *NUS-WIDE* varying code length.

TABLE 5
The Times Measured in Seconds for Hash Code Generation

	SPH	SKLSH	AGH	STH	ITQ	CVH	CHMIS	IMH	LSSH	CMFH	SAVH
Wiki	0.0521	0.0028	0.0296	0.0652	**0.0010**	0.0011	0.0402	0.0089	0.5487	**0.0010**	**0.0007**
MIR Flickr	0.1893	0.0059	0.0130	0.8676	0.0028	0.0037	4.0084	0.0048	0.9886	**0.0022**	**0.0018**
NUS-WIDE	0.4561	0.0159	0.0603	5.2052	**0.0073**	0.0140	27.1756	0.0200	7.4780	0.0105	**0.0092**

The best two results in each row are marked with bold.

approaches, the stable performance improvement with hash code length cannot be easily observed. This is because SAVH ensures bits-uncorrelated constraint in hash code learning. The design forces the learned hash bits to have less information redundancy. In this case, more hash bits will bring more new valuable information. Besides, we can find that, with less hash bits, SAVH can achieve better performance than many compared approaches with longer hash codes. The reason is that, with semantic assistance, SAVH can compress more semantics into short hash codes. In practice, it means that CBIR based on SAVH can enjoy faster retrieval process and less storage cost under the same performance level.

- On several code lengths and datasets, it is interesting to find that UCMH approaches even perform worse than SFVH (For example, ITQ and AGH). This experimental phenomenon validates our analysis on UCMH presented in Section 1. Actually, UCMH aims to achieve fast retrieval *across* heterogeneous modalities. Therefore, seeking the shared space of heterogeneous modalities is the main objective (as shown in Fig. 3c). In this way, the discovered common semantic space of heterogeneous modalities can principally preserve semantic correlations of different modalities. But, in some cases, it may even lose the valuable semantics besides the common part in original visual features. UCMH may not be the best suited for CBIR. This observation also motivates us to design SAVH to effectively leverage the auxiliary text to assist visual hashing.
- On most of code lengths and datasets, CMFH, LSSH, and IMH perform better than CVH and CHMIS. This experimental phenomenon is consistent with the result observed in [23], [25]. This is because, in addition to simply preserve intra-similarity within each modality and inter-similarity *across* modalities, CMFH, LSSH, and IMH impose more constraints on hashing learning, which discovers low-dimensional Hamming subspace with more semantics.
- IMH achieves better performance than LSSH and CMFH in many cases. It demonstrates that, IMH performs better than LSSH and CMFH on preserving semantic correlations of original visual feature in process of shared space discovery. More importantly, it shows that performance on cross-modal retrieval obtained by UCMH approaches may be not consistent with the performance on CBIR.[10] The advantages of several UCMH approaches on cross-modal retrieval may not be held on CBIR. It also validates the importance of specially considering assistance of auxiliary texts when developing visual hashing.

Besides the retrieval precision comparison, we also evaluate the efficiency of online image retrieval which impacts user experience most in real practice. As indicated in Algorithm 2, the online retrieval process is comprised of four subsequent steps 7-10. Since step 7, 9, 10 are identical for all hashing approaches, we only compare the hash code generation efficiency in step 8. In particular, we compare the hash

10. LSSH and CMFH consistently perform better than IMH on cross-modal retrieval, as reported in [23], [25].

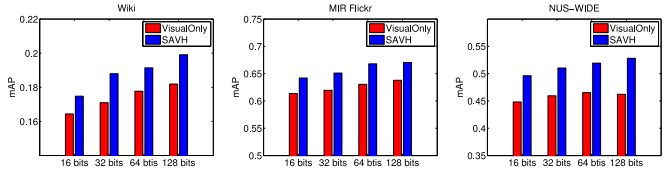


Fig. 9. Effects of semantic assistance on three datasets.

code generation time of all query images when hash code length is fixed to 128 bits. Table 5 presents the main experimental results. From it, we can easily find that, SAVH takes the least time on *Wiki* and *MIR Flickr*, and the second least time on *NUS-WIDE*. It has desirable hashing efficiency. This is because that SAVH adopts simple linear projection for hash code generation. This desirable advantage can well support the application of SAVH to CBIR.

5.2 Effects of Semantic Assistance on Visual Hashing

In this section, we conduct empirical experiments to validate the effectiveness of semantic assistance from auxiliary text on visual hashing. More specifically, we compare the performance of SAVH with the one which ignores discriminative information of texts and only considers visual features.

Fig. 9 presents the detailed experimental results. The key observation we gain is: First, retrieval performance of CBIR can be improved by auxiliary text. The reason for better performance is that, with semantic assistance, relations between images and that between images and latent shared topics can be better modelled and correlated. The valuable extracted semantics can be effectively encoded in the binary hash codes. Second, performance gap is varied on different datasets and lengths of hash code. The largest performance gap is more than 6 percent. The variations of performance gap is mainly caused by the different effectiveness of text on assisting visual hashing.

5.3 Effects of Training Size

This section investigates the performance variations with the training size on *NUS-WIDE*. We fix hash code length to 128 bits and record performance variations when training size is changed from 1,000 to 10,000. Table 6 demonstrates the main results. We can observe that mAP of SAVH increases when more training data is leveraged. However, mAP scores are not improved significantly. This phenomenon illustrates the stabilization of the hash functions learned by SAVH with reasonably small training set. Besides, it should be noted that mAP of SAVH can achieve 0.5006 when training size is 1,000. It is a bit higher than the best performance of compared approaches 0.5002 obtained by LSSH (*NUS-WIDE*, 128 bits). The observation further validates that, the assistance of auxiliary text can effectively mitigate the semantic shortage of hash codes when training data is limited.

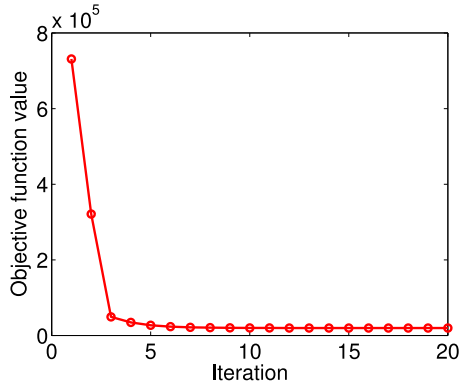


Fig. 10. Variations of objective function value in Eq. (12) with the number of iterations on *NUS-WIDE*.

5.4 Convergence Analysis

This section conducts empirical experiments to analyze the convergence of SAVH. Fig. 10 presents the variations of objective function value in Eq. (12) with the number of iterations on *NUS-WIDE*. We can observe from the figure that, on three datasets, objective function value first decreases with the number of iterations and becomes steady after certain iterations. This experimental results demonstrate that the convergence of SAVH can be guaranteed with augmented Lagrangian multiplier approach.

5.5 Parameter Sensitivity

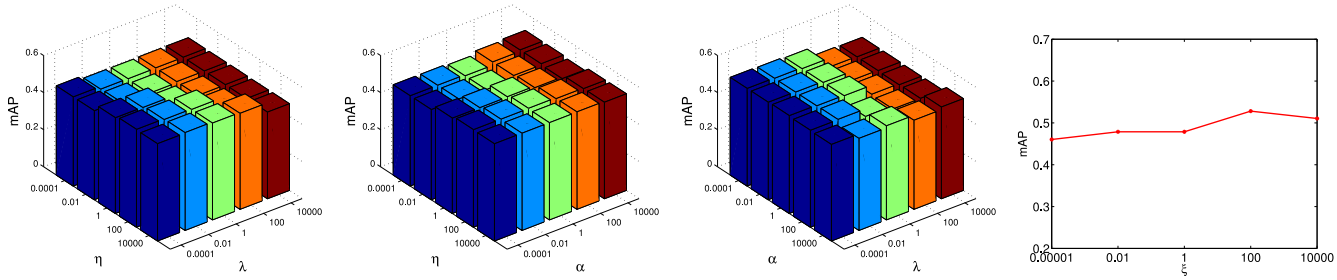
In this section, empirical experiments are conducted to observe the performance variations with parameters α, λ, η, ξ in SAVH. α, λ, and η are used in Eq. (9) to play trade-off between regularization term and empirical loss, while ξ is used in Eq. (28) with similar aims. We fix hash code length to 128 bits and report results on *NUS-WIDE*. Similar results can be found on other code lengths and datasets. We test the results when four parameters are varied from $\{0.0001, 0.01, 1, 100, 10000\}$. For α, λ, η, since they are equipped in the same equation, we observe the performance variations with respect to two parameters while fixing the remaining one parameter. For ξ, we observe the performance variations by fixing α, λ, η. Detailed experimental results are presented in Fig. 11. From (a),(b),(c), we can find that the performance is relatively stable in a wide range of α, λ, η variations. From (d), the best performance can be achieved at certain point ($\xi = 100$).

5.6 Further Comparison with Supervised Hashing

In this section, we conduct experiment to further compare SAVH with several state-of-the-art supervised hashing approaches. The main objective is to validate the effectiveness of SAVH on extracting valuable semantics for visual hashing even with unsupervised learning. The compared supervised hashing approaches include, binary reconstructive embeddings (BRE) [10], kernel based supervised hashing (KSH) [11], semantic correlation

TABLE 6
Performance Variations with Training Size on *NUS-WIDE*

Training size	1K	2K	3K	4K	5K	6K	7K	8K	9K	10K
SAVH	0.5066	0.5111	0.5128	0.5218	0.5281	0.5286	0.5288	0.5294	0.5328	0.5343



(a) $\alpha = 1, \xi = 100$ (b) $\lambda = 0.0001, \xi = 100$ (c) $\eta = 100, \xi = 100$ (d) $\alpha = 1, \lambda = 0.0001, \eta = 100$

Fig. 11. Performance variations with α, λ, η, ξ in SAVH on *NUS-WIDE*.

TABLE 7
mAP of Compared Supervised Hashing Approaches

Methods	Wiki				MIR Flickr				NUS-WIDE			
	16 bits	32 bits	64 bits	128 bits	16 bits	32 bits	64 bits	128 bits	16 bits	32 bits	64 bits	128 bits
BRE	0.1650	0.1559	0.1773	0.1758	0.6050	0.5988	0.6002	**0.6339**	0.3565	0.4018	0.3875	0.4290
KSH	**0.1785**	0.1853	**0.1990**	**0.2221**	0.5993	0.5964	0.6007	0.6160	**0.5312**	**0.5441**	**0.5443**	**0.5536**
SCM-sql	0.1614	**0.1861**	0.1742	0.1757	**0.6509**	**0.6563**	**0.6446**	0.6255	0.4898	**0.5192**	0.5140	0.4792
SCM-orth	0.1650	0.1681	0.1701	0.1780	0.6402	0.6389	0.6220	0.6174	0.4813	0.4542	0.4307	0.4181
SAVH	**0.1748**	**0.1880**	**0.1914**	**0.1991**	**0.6450**	**0.6511**	**0.6680**	**0.6704**	**0.4962**	0.5103	**0.5193**	**0.5281**

The best two results in each column are marked with bold.

maximization (SCM) [60] (Two effective approaches: SCM-sql and SCM-orth are proposed, and we use both of them for comparison.). All their implementation codes are downloaded directly from authors' websites. Their reported experimental results are maximized by adjusting the involved parameters according to relevant literature. Note that, in this experiment, the training data in three datasets are all labeled for supervised learning.

Table 7 presents the main results. From the table, we can easily observe that, on *Wiki* and *MIR Flickr*, our proposed approach SAVH can achieve comparable or even better performance than the best performance achieved by the compared supervised hashing approaches. On *NUS-WIDE*, SAVH achieves better performance than most of the compared supervised hashing approaches. KSH demonstrates superior performance due to the kernel hash function design. However, the generation of kernel matrix in online hashing of SKH needs large amount of computations, which is not easily applicable in large-scale real-time CBIR. These experimental results demonstrate that, even on unlabeled images (with unsupervised learning), SAVH can still generate discriminative hash codes and functions by effectively leveraging the valuable semantics involved in the associated texts to assist visual hashing.

6 Conclusions and Future Work

Most existing single feature and multiple feature hashing approaches for CBIR build their schemes with only visual features. They ignore the valuable semantics involved in the associated texts. Although unsupervised cross-modal hashing approaches can leverage text for retrieval task across heterogeneous modalities, they equally treat visual and text, and still fail to fully take advantages of text. Different from them, this study proposes an effective hashing framework, SAVH. Our idea is leveraging the associated texts of images to assist the visual hashing using unsupervised learning. SAVH can integrate extra discriminative information into the generated visual hash codes and functions. Moreover, SAVH has an important advantage that its offline learning can effectively leverage semantics involved in text, while its online hashing requires only visual image as input. This desirable property matches the requirements of real application scenarios of CBIR. Comprehensive experiments on several standard image datasets validate that the performance of visual hashing can be improved with the assistance of text, and SAVH can achieve superior performance compared with several state-of-the-art methods.

This research opens up several promising directions for further exploration. Notably, it is interesting to further validate the effectiveness of SAVH when more associated modalities are involved. For example, geographical location of image, social correlation of images, and etc. Moreover, it would be also interesting to investigate the effectiveness of visual image on assisting hashing for text retrieval.

Acknowledgments

Jialie Shen is the corresponding author.

References

[1] J.-H. Su, W.-J. Huang, P. S. Yu, and V. S. Tseng, "Efficient relevance feedback for content-based image retrieval by mining user navigation patterns," *IEEE Trans. Knowl. Data Eng.*, vol. 23, no. 3, pp. 360–372, Mar. 2011.

[2] P. Wu, S. C. H. Hoi, P. Zhao, C. Miao, and Z. Liu, "Online multimodal distance metric learning with application to image retrieval," *IEEE Trans. Knowl. Data Eng.*, vol. 28, no. 2, pp. 454–467, Feb. 2016.

[3] J. Sivic and A. Zisserman, "Video Google: A text retrieval approach to object matching in videos," in *Proc. IEEE Int. Conf. Comput. Vis. Pattern Recog.*, 2003, pp. 1470–1477.

[4] P. Ciaccia, M. Patella, and P. Zezula, "M-tree: An efficient access method for similarity search in metric spaces," in *Proc. Int. Conf. Very Large Data Bases*, 1997, pp. 426–435.

[5] J. Wang, H. T. Shen, J. Song, and J. Ji, "Hashing for similarity search: A survey," *CoRR*, vol. abs/1408.2927, 2014.

[6] L. Gao, J. Song, X. Liu, J. Shao, J. Liu, and J. Shao, "Learning in high-dimensional multimedia data: The state of the art," *Multimedia Syst.*, pp. 1–11, 2015, Doi: 10.1007/s00530-015-0494-1.

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

[8] D. Zhang, J. Wang, D. Cai, and J. Lu, "Self-taught hashing for fast similarity search," in *Proc. Int. ACM SIGIR Conf. Res. Develop. Inf. Retrieval*, 2010, pp. 18–25.

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

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

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

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

[13] L. Gao, J. Song, F. Zou, D. Zhang, and J. Shao, "Scalable multimedia retrieval by deep learning hashing with relative similarity learning," in *Proc. ACM Int. Conf. Multimedia*, 2015, pp. 903–906.

[14] J. Song, Y. Yang, Z. Huang, H. T. Shen, and J. Luo, "Effective multiple feature hashing for large-scale near-duplicate video retrieval," *IEEE Trans. Multimedia*, vol. 15, no. 8, pp. 1997–2008, Dec. 2013.

[15] X. Liu, J. He, and B. Lang, "Multiple feature kernel hashing for large-scale visual search," *Pattern Recogn.*, vol. 47, no. 2, pp. 748–757, 2014.

[16] J. Cheng, C. Leng, P. Li, M. Wang, and H. Lu, "Semi-supervised multi-graph hashing for scalable similarity search," *Comput. Vis. Image Understanding*, vol. 124, no. 0, pp. 12–21, 2014.

[17] J. Song, Y. Yang, Z. Huang, H. T. Shen, and R. Hong, "Multiple feature hashing for real-time large scale near-duplicate video retrieval," in *Proc. ACM Int. Conf. Multimedia*, 2011, pp. 423–432.

[18] S. Kumar and R. Udupa, "Learning hash functions for cross-view similarity search," in *Proc. Int. Joint Conf. Artif. Intell.*, 2011, pp. 1360–1365.

[19] J. Song, Y. Yang, Y. Yang, Z. Huang, and H. T. Shen, "Inter-media hashing for large-scale retrieval from heterogeneous data sources," in *Proc. ACM Int. Manage. Data*, 2013, pp. 785–796.

[20] L. Xie, L. Zhu, and G. Chen, "Unsupervised multi-graph cross-modal hashing for large-scale multimedia retrieval," *Multimedia Tools Appl.*, pp. 1–20, 2016, Doi: 10.1007/s11042-016-3432-0.

[21] L. Xie, L. Zhu, P. Pan, and Y. Lu, "Cross-modal self-taught hashing for large-scale image retrieval," *Signal Process.*, vol. 124, pp. 81–92, 2016.

[22] L. Xie, J. Shen, and L. Zhu, "Online cross-modal hashing for web image retrieval," in *Proc. AAAI Conf. Artif. Intell.*, 2016, pp. 294–300.

[23] J. Zhou, G. Ding, and Y. Guo, "Latent semantic sparse hashing for cross-modal similarity search," in *Proc. Int. ACM SIGIR Conf. Res. Develop. Inf. Retrieval*, 2014, pp. 415–424.

[24] X. Zhu, Z. Huang, H. T. Shen, and X. Zhao, "Linear cross-modal hashing for efficient multimedia search," in *Proc. ACM Int. Conf. Multimedia*, 2013, pp. 143–152.

[25] G. Ding, Y. Guo, and J. Zhou, "Collective matrix factorization hashing for multimodal data," in *Proc. IEEE Int. Conf. Comput. Vis. Pattern Recog*, 2014, pp. 2083–2090.

[26] Z. Lin, M. Chen, and Y. Ma, "The augmented lagrange multiplier method for exact recovery of corrupted low-rank matrices," *arXiv e-prints*, 2010, https://arxiv.org/abs/1009.5055

[27] M. Raginsky and S. Lazebnik, "Locality-sensitive binary codes from shift-invariant kernels," in *Proc. Adv. Neural Inf. Process. Syst.*, 2009, pp. 1509–1517.

[28] R.-E. Fan, K.-W. Chang, C.-J. Hsieh, X.-R. Wang, and C.-J. Lin, "Liblinear: A library for large linear classification," *J. Mach. Learn. Res.*, vol. 9, pp. 1871–1874, 2008.

[29] Y. Bengio, O. Delalleau, N. L. Roux, J. Paiement, P. Vincent, and M. Ouimet, "Learning eigenfunctions links spectral embedding and kernel PCA," *Neural Comput.*, vol. 16, no. 10, pp. 2197–2219, 2004.

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

[31] X. Zhu, L. Zhang, and Z. Huang, "A sparse embedding and least variance encoding approach to hashing," *IEEE Trans. Image Process.*, vol. 23, no. 9, pp. 3737–3750, Sep. 2014.

[32] X. Zhu, Z. Huang, H. Cheng, J. Cui, and H. T. Shen, "Sparse hashing for fast multimedia search," *ACM Trans. Inf. Syst.*, vol. 31, no. 2, p. 9, 2013.

[33] F. Shen, C. Shen, Q. Shi, A. van den Hengel, Z. Tang, and H. T. Shen, "Hashing on nonlinear manifolds," *IEEE Trans. Image Process.*, vol. 24, no. 6, pp. 1839–1851, Jun. 2015.

[34] X. Zhu, X. Li, and S. Zhang, "Block-row sparse multiview multilabel learning for image classification," *IEEE Trans. Cybern.*, vol. 46, no. 2, pp. 450–461, Feb. 2016.

[35] L. Zhu, J. Shen, H. Jin, R. Zheng, and L. Xie, "Content-based visual landmark search via multimodal hypergraph learning," *IEEE Trans. Cybern.*, vol. 45, no. 12, pp. 2756–2769, Dec. 2015.

[36] L. Zhu, J. Shen, H. Jin, L. Xie, and R. Zheng, "Landmark classification with hierarchical multi-modal exemplar feature," *IEEE Trans. Multimedia*, vol. 17, no. 7, pp. 981–993, Jul. 2015.

[37] S. Kim and S. Choi, "Multi-view anchor graph hashing," in *Proc. Int. Conf. Acoust., Speech, Signal Process.*, 2013, pp. 3123–3127.

[38] X. Shen, F. Shen, Q.-S. Sun, and Y.-H. Yuan, "Multi-view latent hashing for efficient multimedia search," in *Proc. ACM Int. Conf. Multimedia*, 2015, pp. 831–834.

[39] L. Liu, M. Yu, and L. Shao, "Multiview alignment hashing for efficient image search," *IEEE Trans. Image Process.*, vol. 24, no. 3, pp. 956–966, Mar. 2015.

[40] L. Zhu, J. Shen, and L. Xie, "Topic hypergraph hashing for mobile image retrieval," in *Proc. ACM Int. Conf. Multimedia*, 2015, pp. 843–846.

[41] Y. Mu, J. Shen, and S. Yan, "Weakly-supervised hashing in kernel space," in *Proc. IEEE Int. Conf. Comput. Vis. Pattern Recog.*, 2010, pp. 3344–3351.

[42] S. Kim, Y. Kang, and S. Choi, "Sequential spectral learning to hash with multiple representations," in *Proc. Eur. Conf. Comput. Vis.*, 2012, pp. 538–551.

[43] B. Ni, S. Yan, and A. A. Kassim, "Learning a propagable graph for semisupervised learning: Classification and regression," *IEEE Trans. Knowl. Data Eng.*, vol. 24, no. 1, pp. 114–126, Jan. 2012.

[44] B. Xu, J. Bu, C. Chen, C. Wang, D. Cai, and X. He, "EMR: A scalable graph-based ranking model for content-based image retrieval," *IEEE Trans. Knowl. Data Eng.*, vol. 27, no. 1, pp. 102–114, Jan. 2015.

[45] L. Gao, J. Song, F. Nie, Y. Yan, N. Sebe, and H. T. Shen, "Optimal graph learning with partial tags and multiple features for image and video annotation," in *Proc. IEEE Int. Conf. Comput. Vis. Pattern Recog.*, 2015, pp. 4371–4379.

[46] D. Zhang, F. Wang, and L. Si, "Composite hashing with multiple information sources," in *Proc. Int. ACM SIGIR Conf. Res. Develop. Inf. Retrieval*, 2011, pp. 225–234.

[47] J. Song, Y. Yang, X. Li, Z. Huang, and Y. Yang, "Robust hashing with local models for approximate similarity search," *IEEE Trans. Cybern.*, vol. 44, no. 7, pp. 1225–1236, Jul. 2014.

[48] M. Wang, X. Liu, and X. Wu, "Visual classification by l_1-hypergraph modeling," *IEEE Trans. Knowl. Data Eng.*, vol. 27, no. 9, pp. 2564–2574, Sep. 2015.

[49] J. Yu, D. Tao, and M. Wang, "Adaptive hypergraph learning and its application in image classification," *IEEE Trans. Image Process.*, vol. 21, no. 7, pp. 3262–3272, Jul. 2012.

[50] A. P. Singh and G. J. Gordon, "Relational learning via collective matrix factorization," in *Proc. ACM SIGKDD Int. Conf. Knowl. Discovery Data Mining*, 2008, pp. 650–658.

[51] M. E. Wall, A. Rechtsteiner, and L. M. Rocha, "Singular value decomposition and principal component analysis," *A Practical Approach Microarray Data Anal.*, Springer US, Boston, MA, pp. 91–109, 2003, Doi: 10.1007/0-306-47815-3_5.

[52] J. J.-Y. Wang, H. Bensmail, and X. Gao, "Multiple graph regularized nonnegative matrix factorization," *Pattern Recog.*, vol. 46, no. 10, pp. 2840–2847, 2013.

[53] H. Liu, R. Ji, Y. Wu, W. Liu, and G. Hua, "Supervised matrix factorization for cross-modality hashing," *CoRR*, vol. abs/1603.05572, 2016.

[54] N. Rasiwasia, J. Costa Pereira, E. Coviello, G. Doyle, G. R. Lanckriet, R. Levy, and N. Vasconcelos, "A new approach to cross-modal multimedia retrieval," in *Proc. ACM Int. Conf. Multimedia*, 2010, pp. 251–260.

[55] M. J. Huiskes and M. S. Lew, "The mir flickr retrieval evaluation," in *Proc. ACM Int. Multimedia Inf. Retrieval*, 2008, pp. 39–43.

[56] T.-S. Chua, J. Tang, R. Hong, H. Li, Z. Luo, and Y. Zheng, "Nus-wide: A real-world web image database from national university of singapore," in *Proc. ACM Int. Conf. Image Video Retrieval*, 2009, pp. 48:1–48:9.
[57] D. G. Lowe, "Distinctive image features from scale-invariant keypoints," *Int. J. Comput. Vis.*, vol. 60, no. 2, pp. 91–110, 2004.
[58] D. M. Blei, A. Y. Ng, and M. I. Jordan, "Latent dirichlet allocation," *J. Mach. Learn. Res.*, vol. 3, pp. 993–1022, 2003.
[59] Y. Zhen and D.-Y. Yeung, "A probabilistic model for multimodal hash function learning," in *Proc. ACM SIGKDD Int. Conf. Knowl. Discovery Data Mining*, 2012, pp. 940–948.
[60] D. Zhang and W. Li, "Large-scale supervised multimodal hashing with semantic correlation maximization," in *Proc. AAAI Conf. Artif. Intell.*, 2014, pp. 2177–2183.

Lei Zhu received the BS degree from the Wuhan University of Technology in 2009 and the PhD degree from the Huazhong University of Science and Technology in 2015. He is currently a research fellow at Singapore Management University. His research interests include the area of large-scale image retrieval and classification.

Jialie Shen received the PhD degree in computer science (large-scale media retrieval and database access methods) from the University of New South Wales, Kensington, NSW, Australia. He is a senior lecturer (associate professor) with the Department of Computer and Information Sciences, Northumbria University, Newcastle upon Tyne, UK. His recent research has been published or is forthcoming in leading journals and international conferences, including ACM SIGIR, ACM Multimedia, ACM SIGMOD, ICDE, *IEEE Transaction on Circuits and Systems for Video Technology*, the *IEEE Transactions on Multimedia*, the *IEEE Transactions on Image Processing, ACM Multimedia Systems Journal, ACM Transactions on Internet Technology, and ACM Transactions on Information Systems*. His current research interests include information retrieval, multimedia systems, and economic-aware media analytics. Prof. Shen is an area editor of *Electronic Commerce Research and Applications* and also the chair, a PC member, a reviewer, and a guest editor for several leading information system journals and conferences.

Liang Xie received the BS degree from the Wuhan University of Technology, China, in 2009 and the PhD degree from the Huazhong University of Science and Technology, China, in 2015. He is currently an lecturer in the School of Science, Wuhan University of Technology. His current research interests include image semantic learning and cross-modal and multi-modal multimedia retrieval.

Zhiyong Cheng received the BS degree in thermal energy and power engineering from the Huazhong University of Science and Technology, China, in 2007 and the MS degree in power machinery and engineering from Xian Jiaotong University, China, in 2010. Currently, he is working toward the PhD degree in the School of Information Systems, Singapore Management University, Singapore. His research focuses on multimedia retrieval and recommendation.