Joint Hypergraph Learning for Tag-Based Image Retrieval

Yaxiong Wang, Li Zhu, Xueming Qian, *Member, IEEE*, and Junwei Han

***Abstract*— As the image sharing websites like Flickr become more and more popular, extensive scholars concentrate on tag-based image retrieval. It is one of the important ways to find images contributed by social users. In this research field, tag information and diverse visual features have been investigated. However, most existing methods use these visual features separately or sequentially. In this paper, we propose a global and local visual features fusion approach to learn the relevance of images by hypergraph approach. A hypergraph is constructed first by utilizing global, local visual features, and tag information. Then, we propose a pseudo-relevance feedback mechanism to obtain the pseudo-positive images. Finally, with the hypergraph and pseudo relevance feedback, we adopt the hypergraph learning algorithm to calculate the relevance score of each image to the query. Experimental results demonstrate the effectiveness of the proposed approach.**

***Index Terms*— Tag-based image retrieval, hypergraph, feature fusion, visual feature, pseudo relevance feedback.**

I. Introduction

With the development of social media based on Web 2.0, huge amounts of images spring up everywhere on the Internet, which makes many online tasks such as image retrieval [4]–[9], [22], [23], [32]–[34], [38]–[43], image recommendation [72], [73] very challenging. The large-scale web images demand the researchers to develop efficient algorithms for more accurate indexing and retrieval. Compared with content-based image retrieval (TBIR), tag-based image search is more commonly used in social media [32], [50]. In the last few decades, extensive efforts have been dedicated to image relevance retrieval. However, many algorithms can't achieve satisfactory results for tag mismatch, noisy tags and query ambiguity problems [50]. Thus, more and more researchers attempt to utilize visual features and user relevance feedback to improve the retrieval accuracy.

There are several visual features designed to express images such as color feature [29], shape feature [36], textural feature [37], edge feature [1], SIFT [16] and deep feature [56], [60]. Different visual features describe different aspects of an image. Therefore, some algorithms try to fuse multiple visual features to improve the image retrieval precision [2], [4], [5], [33]. However, most existing methods usually explore multiple visual features separately. For example, Yang *et al.* [2] first construct a graph for every feature. Then they apply random walk model to get a relevance score according to each constructed graph. Finally, they re-rank the images by the linear combination of the relevance scores of different features. Zhang *et al.* [4] first select training samples, then they apply multiple visual features by simpleMKL to train the classification function for image ranking. Yang *et al.* [5] learn the Mahalanobis matrix for different visual features and calculate the distance of images by the Mahalanobis distance of corresponding visual feature. Yu *et al.* [33] construct five hypergraphs for five visual features, and integrate the visual consistency constrains of these hypergraphs to learn a linear model for ranking. Gao *et al.* [34] construct hypergraph by local visual feature only and abandon the global visual information of images. However, different visual features have district emphasis on describing the content of an image, therefore, separately or sequentially using these information is suboptimal for social image retrieval.

Many TBIR algorithms are designed based on graph model aiming at utilizing multiple visual features [2], [46]. Graph-based approaches are based on the assumption that neighboring images in a graph having close relevant scores. Usually, a similarity graph is constructed first, where the vertex is the image and edge weight is the similarity between vertices. Then some link structure analysis technologies are employed to exploit the vertex relations. However, the edge of conventional graph only associates with two vertices, that is to say, one edge in graph can only capture the relationship of two vertices. Fortunately, hypergraph can overcome this limitation. The hypergraph can be regarded as a generalization of the graph. Compared to conventional graph, hypergraph can model the relationship of more than two vertices and more complex

Manuscript received July 21, 2017; revised March 19, 2018 and May 2, 2018; accepted May 2, 2018. Date of publication May 16, 2018; date of current version June 11, 2018. This work was supported in part by NSFC under Grants 61732008, 61772407, 61332018, and u1531141, in part by the the National Key R&D Program of China under Grant 2017YFF0107700, and in part by the Guangdong Provincial Science and Technology Plan Project under Grant 2017A010101006. The associate editor coordinating the review of this manuscript and approving it for publication was Prof. Xiaochun Cao. *(Corresponding authors: Li Zhu; Xueming Qian.)*

Y. Wang is with the School of Software, Xi'an Jiaotong University, Xi'an 710049, China, and also with the SMILES Laboratory, Xi'an Jiaotong University, Xi'an 710049, China (e-mail: wangyx15@stu.xjtu.edu.cn).

L. Zhu is with the School of Software, Xi'an Jiaotong University, Xi'an 710049, China (e-mail: zhuli@mail.xjtu.edu.cn).

X. Qian is with the Key Laboratory for Intelligent Networks and Network Security, Ministry of Education, Xi'an Jiaotong University, Xi'an 710049, China, also with the SMILES Laboratory, Xi'an Jiaotong University, Xi'an 710049, China, and also with Zhibian Technology Co., Ltd., Taizhou 317000, China (e-mail: qianxm@mail.xjtu.edu.cn).

J. Han is with the School of Automation, Northwestern Polytechnical University, Xi'an 710072, China (e-mail: junweihan2010@gmail.com).

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

Digital Object Identifier 10.1109/TIP.2018.2837219

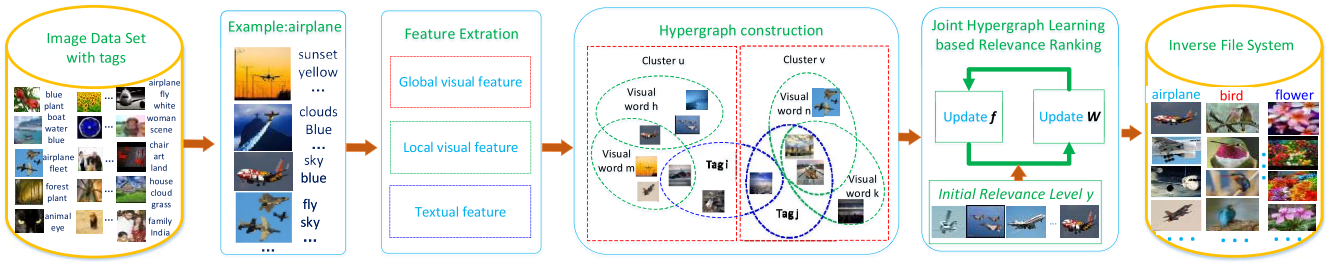

Fig. 1. The framework of proposed method, *PRF* represents pseudo relevance feedback. Take image set with tag "airplane" as an example. Select the image list in inverted file system and conduct our algorithm in this image list. We save each tag-sorted image list pair to form the sorted file system, which is directly used in online retrieval.

relationship between objects [3]. Several papers have shown the superiority of hypergraph [14]–[17].

Hypergraph not only takes pairwise relationship into consideration, but also models the higher order relationship among three or more vertices containing grouping information. Hypergraph method is widely used in data mining and information retrieval tasks [6], [7]. Cai *et al.* [9] first train attribute ion classifiers, then construct hypergraph based on these classifiers, finally they obtain the relevance score by hypergraph learning. Jing *et al.* [10] request users' relevance feedback, then they propagate relevance of feedback images to other images, finally a hypergraph is constructed based on the k-nearest mechanism. Gao *et al.* [34] construct hypergraph by tags and local visual feature, and get the final relevance score of images by hypergraph learning. Yu *et al.* [33] construct hypergraphs based on different visual features separately, then learn a linear model for ranking by these hypergraphs.

In this paper, we propose a hypergraph-based approach to simultaneously utilize different visual features and tags for image relevance learning. Fig. 1 illustrates the schedule of our framework. In our system, each image is represented by both a global feature and a local feature, like color moment and SIFT. Besides visual content, semantic information, i.e. the tags associated with image are also employed in our method.

We construct a hypergraph for query tag, in which the vertices denote the images for ranking and hyperedges are subsets of these images. Our constructed hypergraph contains semantic and visual hyperedges. The semantic hyperedge is generated by the co-occurrence tags of query. Global and local visual features are simultaneously utilized to construct the visual hyperedges. In the learning process, we identify a set of relevance scores of images by iteratively updating them and the weights of hyperedges.

The contributions of this paper can be summarized as follows:

1) We present a novel joint learning approach for tag based web image retrieval (JHR), which utilizes the global, local visual features and textual feature simultaneously. Compared to using global, local visual feature or textual feature alone or separately, the joint hypergraph learning approach can capture more reliable relationships between images.
2) We propose a new pseudo relevance feedback mechanism for tag-based image retrieval. First, we conduct clustering on the co-occurrence tags. Then we assign images to clusters. Finally, we estimate the relevance between image and query by fusing the image cluster relevance and image relevance to query. The introducing of cluster relevance is an assistant for calculating image initial relevance score, which is superior to using image relevance only.
3) We build the inverted file system for tags in offline part. All steps of our algorithm are conducted offline. In online retrieval, we only match the query tag to get the retrieval results, thus the online search is very efficient.

From a broader perspective, this paper exhibits a novel method of utilizing multiple visual features to capture more reliable relation between images for tag-based image retrieval task. Different from using these features separately or sequentially [2], [4], [5], [33], our proposed mechanism can effectively achieve the fusion of multiple features.

The remainder of this paper is organized as follows. In section II, we review the related work of the tag-based image retrieval. In section III, we briefly introduce the hypergraph learning model. The system overview is illustrated in section IV. We present the feature extraction in section V. Section VI elaborates the details of each process in our system. Experiments are shown in section VII, and discussions are stated in Section VIII. Finally, conclusion and future work are given in section IX.

II. Related Work

Social image share websites like Flick usually ask the users for several tags when they upload their sharing images. The online retrieval can be conducted by key words match. However, the retrieval results are not satisfactory for the unreliable tags. Therefore, a series of methods are proposed aiming at incorporating visual factors into image ranking over the last decades. Hypergraph has shown its ascendency in information retrieval task [14]–[17]. Many scholars designed their algorithms based on hypergraph for image retrieval. The following subsections present the existing works related to the above two aspects respectively.

A. Image Visual Re-Ranking

The massive available images in internet make the retrieval task challenging. There are lots of researches done on the tag based image retrieval. Visual re-ranking is one of important

methods to improve the retrieval results. The existing visual re-ranking methods can be classified into three categories: clustering based, classification based and graph based approaches.

Clustering based methods are based on the truth that the relevant images to query share high visual similarity. In clustering based methods, images in the initial list are first grouped into different clusters and then sorted based on the cluster conditional probability. Duan *et al.* [38] first cluster the images by textual and visual features respectively and then treat each cluster as a word (textual or visual). Finally, the ranking problem is modeled as a multi-instance learning problem in which the pseudo-positive samples are the top ranked images and negative samples are randomly selected. Tang *et al.* [39] propose an intent based search approach that aims at solving the query ambiguity in TBIR. They ask the user to click one query image, by which they capture the user's search intent. Then, the images from a group which is obtained by text-based search are re-ranked based on both visual and textual information.

The classification based image retrieval approach consists of three steps in general: the positive and negative samples from the initial retrieval list are selected first, then classifiers are trained and finally the initial images are ranked according to the scores from the trained classifier. Tian *et al.* [40] propose a re-ranking method with user interaction, which first selects images according to an active sample selection method and then asks the user to label them. Finally, it learns a discriminative sub-manifold by the label information. Lekshmi and John [55] first request a feedback image from user and select positive images, then they train a perceptron based on the selected samples. Instead of requiring user's effort, obtaining training samples by click information is more practical. Several papers have shown that the user's click is a reliable clue for revealing images relevant or not [2], [4], [5], [41]–[43]. Zhang *et al.* [4] treat the click information as implicit relevant feedback and select the top clicked images as the relevant samples. The re-ranking processing is conducted by the learned simpleMKL model. Xioufis *et al.* [43] also select the top clicked images as relevance samples, then they extract multiple visual features to train multiple classifiers. Finally, they fuse the results of these classifiers to re-rank images. Yan and Hauptmann [44] use conventional idea of pseudo-relevance feedback that treats top ranked images as the pseudo-positives and bottom as the pseudo-negatives.

In graph based methods, a graph is used to capture the relations between images. The graph is constructed with images or tags as nodes and the edges are weighted by visual or textual likeness. Image re-ranking is performed by graph learning algorithm. In these methods, the graph construction plays the key role, since the relations and associations of all nodes are represented by the graph. Jing and Baluja [45] treat the re-ranking problem as random walk on an affinity graph. The final retrieval list is obtained by the learned weights of nodes. Hou *et al.* [47] first construct two graphs based on semantic and visual information of images in initial list. Then they apply random walk to get the relevance scores by graph learning. Finally, they combine the semantic and visual relevance scores to re-rank the images. Liu *et al.* [48] generate two matrices based on visual information and "social clue". The final transition matrix is the combination of visual matrix and "social clue" matrix. They apply random walk to get the final relevance score. Instead of using only one modality, many scholars integrate multi-modality to improve the performance [2], [4], [5], [43], [46]. Yang *et al.* [2] first construct a graph for every visual feature. Then, they model the re-ranking as an optimization problem by the results of multi-modality graph learning. Wang *et al.* [46] also construct graph using multiple features, and the final relevance score is learned by a joint optimization framework. Wang *et al.* [68] design a semi-supervised multiple kernel learning approach for image re-ranking and categorization, and multiple features are made use of to enhance the generalization power of semi-supervised learning.

B. Hypergraph Based Applications

Hypergraph has shown its effectiveness of higher-order sample relationship modeling in many mechanism learning tasks such as data mining and information retrieval [3], [34]. Yu *et al.* [35] propose an adaptive hypergraph learning method for transduction image classification. Zhou *et al.* [3] propose a general hypergraph framework that can be applied in clustering, classification and embedding tasks. Liu *et al.* [58] design a hash method based on hypergraph model, they first utilize hypergraph to capture the relation of vertices and generate the binary code by spectral hashing. Wang *et al.* [59] propose a dimensionality reduction framework based on hypergraph. Fang *et al.* [60] utilize hypergraph designing a topic-sensitive influencer mining approach for interest-based social media networks. Wong and Lu [52] propose a 3-D object description method. They denote the vertices the surface patches of an object and the hyperedges represent the connection of the pair of boundary segments. Hong *et al.* [62] construct hypergraph by k-nearest mechanism, then they decompose the hypergraph Laplacian to obtain the fused feature vector, by which they train an autoencoder network for 3D pose recovery. Hong *et al.* [63] utilize hypergraph Laplacian to preserve the local similarity to recover 3-D human pose from silhouettes. Hong *et al.* [64] fuse multi-view data to recognize 3D object, hypergraph is used to better capture the connectivity among views. Hypergraph is also employed in multi-label learning [14], image/video segmentation [13], [69]–[71] and even music recommendation [53].

For image retrieval, many algorithms are designed based on hypergraph technique. Liu *et al.* [7] propose a soft hypergraph, which assigns each vertex to a hyperedge in a soft way. The image retrieval task is formulated as the problem of hypergraph ranking. Zhu *et al.* [65] learn hash codes for mobile image retrieval based on hypergraph, they utilize hypergraph to model the high-order semantics of images. Zhu *et al.* [66] construct a topic hypergraph and utilize the hypergraph Laplacian to integrate the textual information into a unified framework for content based image retrieval. Xie *et al.* [67] present a dynamic hash method that constructs hash code by a dynamic dictionary for online image retrieval. Jouili and Tabbone [51]

design a hypergraph based on the graph representation of image and transform the image retrieval task to the problem of indexing graph. Huang *et al.* [6] construct the hypergraph according to a probability value rather than the binary. An optimization framework is applied to get the relevance score of images. Cai *et al.* [9] select some general attributions and train SVM classifier for each attribution. They construct hypergraph based on the scores from trained classifiers. The final relevance score of image is obtained by hypergraph learning. Jing *et al.* [10] first ask the user to mark an image as the relevant feedback. With the visual assistant clues, they choose other relevant images based on the feedback images. Then, a hypergraph is constructed based on the labeled images by k-nearest mechanism. Wang *et al.* [8] only consider the visual and textual hybrid hyperedge and ignore single modality hyperedge. They construct hyperedges for every visual word and tag pair, which suffer from the enormous number of hyperedges. Gao *et al.* [34] first remove the noisy tags of images. Then they generate hyperedge by visual words and the selected tags respectively. Re-ranking is conducted by the score from hypergraph learning. Gao *et al.* [34] also fuse different features by hypergraph model, however it has considerable differences with this work. First, we fuse three different features, i.e. the global, local and textual features to improve the relevance, while [34] only take local and textual features into consideration. Second, how to fuse the global feature efficiently is an important problem and innovation in our paper, while [34] doesn't need to pay attention to this problem. Third, [34] employ all the visual words to generate the hyperedge without filtering, while we only choose the higher frequency visual words in image cluster to generate hyperedge. Fourth, our relevance feedback mechanism is different from the [34]'s. In [34], the images are initially ranked by the average Flickr Distance and they select the top ranked images as the pseudo relevant images, while we rank the images initially by fusing the cluster score and the average google distance of images to query and top ranked images are selected as the pseudo-relevant images. Fifth, [34] first remove the noisy tags by a tag refinement algorithm, while we use the original tags directly. Sixth, [34] select the tags for hyperedge construction based on the TF-IDF (Term Frequency-Inverse Document Frequency) value, while we select the tags with TF value. Seventh, an inverted file system is finally built for fast online retrieval in our paper, while [34] pays no attention to the online retrieval.

However, most of the above literatures use only one visual modality or use multiple visual modalities separately. In our proposed method, we fuse the global, local visual and textual features simultaneously by hypergraph model to improve the retrieval performance.

III. Adaptive Hypergraph Learning Model

Before presenting our approach, we first briefly introduce the hypergraph learning model.

A. Hypergraph Definition

Hypergraph is a generalization of traditional graph in which the edges, called hyperedges, are arbitrary nonempty subsets of the vertex set [11]. A hypergraph $\mathcal{G} = (\mathcal{V}, \mathcal{E}, \omega)$ is composed by a vertex set $\mathcal{V}$, an edge set $\mathcal{E}$, and the weights of edges ω. Each edge e is given a weight $\omega(e)$. In image retrieval task, the images to be sorted are the vertices and a hyperedge is a subset of these images, as shown in Fig.1. A hypergraph $\mathcal{G}$ can be represented by a $|\mathcal{V}| \times |\mathcal{E}|$ incident matrix H with entries defined as follows:

$$h(v, \mathrm{e}) = \begin{cases} 1 & if\ v \in e \\ 0 & if\ v \notin e. \end{cases} \tag{1}$$

where $h(v, e) = 1$ indicates that the vertex v is associated with edge e, otherwise $h(v, e) = 0$.

The degree of a vertex $v \in \mathcal{V}$ is defined as the sum of edge weights associated with v:

$$d(v) = \sum_{e \in \mathcal{E}} \omega(e)\, h(v, e) \tag{2}$$

For a hyperedge $\mathrm{e} \in \mathcal{E}$, its hyperedge degree can be defined as the number of vertices within the hyperedge:

$$\delta(e) = \sum_{v \in \mathcal{V}} h(v, e) \tag{3}$$

Let W denote diagonal matrix of the hyperedge weights:

$$W(i, j) = \begin{cases} \omega(i) & if\ i = j \\ 0 & if\ i \neq j \end{cases} \tag{4}$$

B. Hypergraph Learning Model

For each tag, adaptive hypergraph learning based retrieval is formulated as a regularization framework as follows:

$$\arg\left\{\lambda R_{emp}(f) + \Omega(f) + \mu \Psi\omega)\right\} \tag{5}$$

where λ, μ are the regularization parameters, f is the relevance score vector that needs to be learned. $\Omega(f)$ is the normalized cost function, which is defined as:

$$\Omega(f) = \frac{1}{2} \sum_{e \in \mathcal{E}} \sum_{u, v \in \mathcal{V}} \frac{\omega(e) h(u, e) h(v, e)}{\delta(e)} \left(\frac{f(u)}{\sqrt{d(u)}} - \frac{f(v)}{\sqrt{d(v)}} \right)^2 \tag{6}$$

This term means that the relevance score of vertices in the same hyperedge should be close.

For image retrieval task, the images in the same hyperedge share similar group information. For example, images in a hyperedge all contain a common tag. Thus, $\Omega(f)$ makes the relevance scores of these images close.

$R_{emp}(f)$ is empirical loss:

$$R_{emp}(f) = ||f - y||^2 = \sum_{u \in \mathcal{V}} (f(u) - y(u))^2 \tag{7}$$

where $y \in R^{|\mathcal{V}|}$ and the component $y(u), u \in \mathcal{V}$ represents the relevance level of image u to tag q. In this paper, we divide the relevance level into two classes: relevant (with score 1) and irrelevant (with score 0).

In the hypergraph learning based image retrieval, our aim is to minimize the regularized loss function Eq. (5). Images with larger f are more relevant to query. Thus, for a query

tag q, we can sort the images by descending order according to the learned relevance score f.

The empirical loss $R_{emp}(f)$ guarantees that the final relevance score f are not far away from the initial label information y.

$\Psi(\omega)$ is an l_2 norm regularizer on the weights, i.e. $\Psi(\omega) = ||\omega||_{l_2}^2$. This strategy is popular to avoid overfitting [35].

C. Optimization Solution

For adaptive hypergraph learning, the scores of vertices f and the weights of hyperedges ω are two parameters that need to be learned. Let D_v and D_e denote the diagonal matrices of vertex degree and the hyperedge degree respectively. Let $\Theta = D_v^{-\frac{1}{2}} H W D_e^{-1} H^T D_v^{-\frac{1}{2}}$, and $\Delta = I - \Theta$, then the cost function can be rewritten as:

$$\Omega(f) = f^T \Delta f \tag{8}$$

where Δ is a positive semi-definite matrix called hypergraph Laplacian.

Then the regularization framework can be rewritten as:

$$\begin{aligned} &\arg\min_{f,\omega} \ \{f^T \Delta f + \lambda ||f - y||^2 + \mu \sum_{\mathrm{e}\in\mathcal{E}} \omega^2(e)\} \\ &\text{s.t.} \ \sum_{\mathrm{e}\in\mathcal{E}} \omega\,(\mathrm{e}) = 1 \end{aligned} \tag{9}$$

optimization problem as shown in Eq.(9) can be efficiently solved by alternating optimization strategy [34].

First, we fix ω and optimize f. Differentiate costfunction with respect to f, we can obtain:

$$\Delta f + \lambda(f - y) = 0 \tag{10}$$

Following some simple algebraic steps, we have:

$$f = \frac{1}{1-\xi}(I - \xi\Theta)^{-1} y \tag{11}$$

where $\xi = \frac{1}{1+\lambda}$.

Next, we fix f and optimize ω, and the equation becomes:

$$\begin{aligned} &\arg\min_{\omega} \{f^T \Delta f + \mu \sum_{e\in\mathcal{E}} \omega^2(e)\} \\ &\text{s.t.} \ \sum_{e\in\mathcal{E}} \omega(e) = 1 \end{aligned} \tag{12}$$

This optimization problem can be solved by Lagrangian multiplier method. Update ω by the following equation:

$$\omega(i) = \frac{1}{n_e} - \frac{f^T \Gamma D_e^{-1} \Gamma^T f}{2 n_e \mu} + \frac{f^T \Gamma_i D_e^{-1}(i,i) \Gamma_i^T f}{2\mu} \tag{13}$$

where n_e denotes the number of hyperedge, Γ is defined as:

$$\Gamma = D_v^{-\frac{1}{2}} H \tag{14}$$

and Γ_i is the i-th column of Γ.

The two steps above continue until convergence. When the objective function reaches its optimal value, we get the final score of the images to the query tag.

IV. System Overview

In this section, we roughly show the main procedures of our proposed method. As shown in Fig. 1, all parts in our system can be conducted offline. We take query "*airplane*" as an example to illustrate the main procedures in our system. We first select the images with tag "airplane" as the dataset to be sorted. Then, for each image, we extract its global feature, local feature and textual feature. Third, a hypergraph is constructed by fusing these three types of features. Next, we obtain the initial label vector by pseudo relevance feedback. Finally, we enter hypergraph incident matrix and initial label vector into the hypergraph learning algorithm to obtain the final relevance scores of images. Afterwards, we sort the images by descending order according to the learned relevance scores. We process all tags in dataset by above steps and store the tag-ranked image list pairs to form the sorted inverted file system. For online retrieval, when user issues a query, we return the image list in the sorted inverted file system by keywords matching.

V. Feature Extraction

In this section, we introduce the visual feature and textual feature extraction. In section V-A, we present visual feature extraction and textual feature extraction is introduced in section V-B.

A. Visual Feature Extraction

In our proposed method, global and local visual features are fused to improve the retrieval accuracy. The following two subsections introduce the two types of features we apply in this paper.

1) Global Feature Extraction: In this paper, color and texture features are selected as the global visual features. Color feature is one of the most widely used visual features in image retrieval, for its invariance with respect to image scaling, rotation, and translation. Texture feature describes the structure arrangement of surfaces and their relationship to the environment, such as fruit skin, clouds, tree and fabric. For each image in image set, we extract the 225-dimensional color moment and 125-dimension texture feature. In our experiment, we splice these two types of features into a 353-dimensional vector and normalize it as the final global visual feature.

2) Local Feature Extraction: For the local visual feature, we obtain it by BOW (bag-of-visual-words) model which is trained based on local SIFT feature. We first extract local SIFT descriptors of all images in data set. Then train a visual vocabulary with o visual words by k-means clustering. Thus, we can represent image by an o-D visual word frequency vector. In our experiment, we use the hierarchical k-means [35] to speed the clustering procedure. This paper, we set $o = 1000$.

B. Textual Feature Extraction

In this paper, we treat the tags associated with image as its textual feature. In order to express tag effectively, we employ the word2vec model [32], [57].

We train word vector model based on vocabulary of English Wikipedia dataset [28] by word2vec algorithm. To generate the word vectors well, we employ the skip-gram model. After training finished, each tag can be represented by a 100-D vector.

Besides training word vector, we also use the English Wikipedia words to filter the tags and remove the tags of image that are not in Wikipedia wordlist in our experiment.

VI. Joint Hypergraph Learning Based Image Retrieval

In this section, we elaborate our proposed method in detail. In section VI-A, we introduce the semantic hypergraph construction and present the visual hypergraph construction by global and local features in section VI-B. Section VI-C presents the pseudo relevance feedback method, and the complexity analysis is given in Section VI-D. Appendix A lists the main notations and definitions in this paper.

A. Semantic Hyperedge Construction

In this subsection, we elaborate the semantic hypergraph construction, which is based on the co-occurrence tags of query q. We choose tags at first and each selected tag generates a semantic hyperedge.

We denote the image dataset with tag q by $\mathcal{X} = \{x_1, x_2, \ldots, x_n$ and n is the image number. Thus, for query q, we only need to conduct image search on $\mathcal{X}$. We regard each social image in image set $\mathcal{X}$ as a vertex in the hypergraph $\mathcal{G} = (\mathcal{V}, \mathcal{E}, \omega)$, $\mathcal{V}$ is the images in $\mathcal{X}$, $\mathcal{E}$ is the hyperedge set and ω is the hyperedge weight.

In the semantic hyperedge construction procedure, we first choose co-occurrence tags with query q and the tags with higher co-occurrence frequency are left for further hypergraph construction.

Let $T = t_1, t_2, \ldots, t_m$ be the co-occurrence tag set of query q, from which we select the high frequency tags for hyperedge construction. We denote the selected tag set by $T^* = \{t_1, t_2, \ldots, t_{m^*}$, $m^* \leq m$. Then, each tag in T^* is used to generate a hyperedge: the social images with the same tags are connected. That is, a semantic hyperedge consists of the images with a common tag in T^*. Thus, we can obtain m^* textual hyperedges.

Semantic hyperedge construction is based on the truth that the tags with higher co-occurrence frequency with query q is more possible to be relevant. For example, in NUS-Wide dataset, there are 18,936 images containing tag "sky", in which 8,525 images contain tag "clouds" and only 6 images contain tag "tortoise", so it is reasonable to believe that "clouds" is more relevant to "sky" than "tortoise". Therefore, we connect the images with tag "clouds" in $\mathcal{X}$ to generate a hyperedge while abandon the tag "tortoise".

B. Visual Hyperedge Construction

In this subsection, we present our visual hypergraph construction. We construct the first layer hypergraph by global feature and the second layer is constructed based on the first layer by local feature.

First of all, we cluster the image set X based on global feature by mean-shift or k-means clustering approach. Let B $= \{b_1, b_2, \ldots, b_{|B|}$ be the clustering results, where $b_i, i \in [1, |B|]$ represents the i-th cluster and $|B|$ represents the number of clusters. Then, we generate hyperedge by connecting the images within the same cluster, which is the first layer visual hypergraph.

Next, we use the local feature, i.e. SIFT feature [16], to construct the second layer hypergraph. We employ the BOW to represent the images [17]–[20] beforehand.

We construct the second layer hypergraph based on the BOW representation of images. Let $I = \{I_1, I_2, \ldots, I_{|b_i|}$ be the images in cluster $b_i \in B$, we count the frequency of all the visual words in cluster b_i and only retain top K visual words with highest frequency. Each term of the K visual words generates a visual hyperedge. That is to say, the images containing the same visual word forms a hyperedge and every cluster can generate K hyperedges. Thus we obtain $|B| \times K$ visual hyperedges in total.

In this construction process, we select the images in the first layer to construct the second layer hyperedge through the local visual clues, i.e. BOW. From the process of the first layer visual hyperedge construction, the images in the second layer hyperedge are not only globally similar but also locally similar.

We choose the high frequency BOWs based on the fact that relevance images share highly visual similarity [54]. That is to say, there are some visual modalities appearing repeatedly in relevant images. In turn, the images with high frequency visual modality are more possible to be relevant. In our method, the visual words are the instance of visual modality. Thus, we tie the images sharing the high frequency BOWs by hyperedges and assign these images similar scores by hypergraph learning (see explanation of Eq. (6)).

C. Pseudo Relevance Feedback

Our pseudo relevance feedback mechanism consists of five steps. 1) tag clustering, we cluster all the co-occurrence tags of query q. 2) relevance of cluster to query, the relevance of each cluster to the query is calculated. 3) semantic relevance, we compute the semantic relevance score of image to query in this step. 4) relevance of image to query, semantic relevance and cluster relevance are fused as the relevance estimation of image to query. 5) pseudo-relevance feedback, relevance level of image is labeled according to its order determined by the relevance estimation in step 4.

1) Tag Clustering: With the trained word vector, cosine similarity is introduced to measure the similarity between tags:

$$s_{tag}(t_i, t_j) = \frac{< v_i, v_j >}{||v_i|| \times ||v_j||} \tag{15}$$

where v_i is the word vector of tag t_i and $||\cdot||$ represents the vector norm. Thus, we can obtain the similarity matrix of tags. We conduct clustering on the co-occurrence tag set T by AP-clustering algorithm [27]. Similar tags can be assigned into the same cluster. We choose AP-Cluster is mainly because that AP-Clustering algorithm is not necessary to assign the number of cluster center [32]. We denote the clustering results

by $C = \{c_1, c_2, \ldots, c_{|C|}$, where $|C|$ is the cluster number we obtain.

2) Relevance of Cluster to Query: We take a cluster $c \in C$ as an example to explain the relevance computing. Let $T^c = \{t_1^c, t_2^c, \ldots, t_{|T^c|}^c$ represent the tags in cluster c. The relevance between cluster c and the query q is defined as:

$$r = \frac{1}{|T^c|} \sum_{i=1}^{|T^c|} \log(1 + h_i^c) \tag{16}$$

where h_i^c is the tag t_i^c co-occurrence frequency with query q. Here we utilize the log transform to mitigate the big difference between tag frequencies in dataset.

3) Semantic Relevance: We simply estimate the semantic relevance of an image x_i to the query tag q as the average semantic similarity between q and all tags of image x_i as follows:

$$s(x_i, q) = \frac{1}{|T_i|} \sum_{t \in \mathcal{T}_i} GD(q, t) \tag{17}$$

where $\mathcal{T}_i$ is the tag set of image x_i, $|T_i|$ is the number of tags associated to x_i and$GD(q, t)$is the Google distance of tag q and tag t:

$$GD(q, t) = exp(-\frac{\max\{\log R(q), \log R(t)\} - \log R(q, t)}{\log G - \min\{\log R(q), \log R(t)\}}) \tag{18}$$

where G is the total number of image in the dataset X, $R(q, t)$ represents the number of image tagged by query q and t simultaneously.

4) Relevance of Image to Query: First, we assign the image x_i to a unique tag cluster that shares maximum tags with the image.

Then, we determine relevance of tag q to image x_i by the linear combination of the corresponding cluster relevance and the semantic relevance as follows:

$$F(x_i, q) = \alpha r_i + (1 - \alpha) s(x_i, q) \tag{19}$$

where r_i is relevance score of cluster which image x_i belongs to, it's computed by Eq. (16). $\alpha \in [0, 1]$ is a constant.

5) Pseudo Relevance Feedback: Based on the relevance score F, we sort all the web images that associate with the tag q in descending order. In our pseudo relevance feedback approach, the images in the top A are selected as the relevant images (with their relevance level y is 1) and the left image are considered as the irrelevant images (with their relevance level y is 0).

When we obtain the label vector y by above steps and the hypergraph by Section VI-A, B. $R_{emp}(f)$ in Eq. (7) and $\Omega(f)$ in Eq. (6) can be specified. We can model our joint hypergraph learning based image retrieval problem based on Eq. (5). Next, we conduct the iteration algorithm introduced in Section III-C. Then, we obtain the relevance score f of all images containing tag q. Finally, we sort the images according to learned f in descending order.

Thus, for each query tag, we can obtain its ranked image list, and our algorithm doesn't require user's interaction with the help of our pseudo-relevance feedback. Therefore, the proposed algorithm can be conducted offline completely and the inverted file system can be built offline.

For a tag t in the dataset, we conduct our proposed algorithm and rank the images with tag t according to the learned relevance score. Thus, we can obtain the corresponding ranked image list ℓ for the tag t. We traverse all the tag and save the tag-ranked image list pair $< \mathrm{t}, \mathrm{l} >$ to form the inverted file system. The online retrieval will be very simple. When user issues a query p, we obtain the corresponding image list ℓ_p by key words matching in our inverted file system and return ℓ_p as the retrieval results.

D. Algorithm Complex Analysis

Our computational time cost is mainly from three parts: hypergraph construction, hypergraph learning and our relevance feedback mechanism. The time cost of hypergraph construction is caused by the mean-shift clustering algorithm applied in our first-layer hypergraph generation, whose complexity is $\mathrm{O}(\mathcal{C}_1 n^2 l)$, where n and $\mathcal{C}_1$ are the image(vertex) number and iteration times of all points on average respectively, l is the feature length. In mean-shift clustering, the number of iterations $\mathcal{C}_1$is inversely proportional to the bandwidth h. In the hypergraph learning process, we first fix the weights of hyperedges ω and optimize the relevance score f by Eq. (11), the complexity is $\mathrm{O}(\mathcal{C}_2 n^3)$, where $\mathcal{C}_2$ is the number of iterations of hypergraph learning. We next fix f and update the weights of hyperedges ω according to the Eq. (13), whose complexity is $\mathrm{O}(\mathrm{n}M^2)$, then the total complexity of update all hyperedge weights is $\mathrm{O}(\mathcal{C}_2 \mathrm{n} M^3)$, where M is the total number of hyperedge. The time cost of pseudo-relevance feedback is mainly from the applied AP clustering algorithm, whose complexity is $\mathrm{O}(\mathcal{C}_3 n^3)$, where $\mathcal{C}_3$ is the number of iterations of AP clustering.

From the analysis above, the computational cost of our proposed algorithm is $\mathrm{O}(\mathcal{C}_1 \mathrm{n}^2 l + (\mathrm{C}_2 + \mathcal{C}_3)\mathrm{n}^3 + \mathcal{C}_2 \mathrm{n} M^3)$, where n and $M$are the image (vertex) number and hyperedge number respectively, $\mathcal{C}_1, \mathcal{C}_2$ and $\mathcal{C}_3$.are the number of iterations for mean-shift clustering, hypergraph learning and AP clustering respectively.

VII. Experiments

In order to demonstrate the effectiveness of our global local joint hypergraph learning based image retrieval approach (denoted by JHR[1]), we conduct experiments on the NUS-Wide image set by utilizing following 20 tags as query: animal, birds, boats, bridge, buildings, clouds, dancing, flowers, grass, house, lake, moon, mountain, ocean, person, plane, plants, reflection, sports and tower. We systematically make comparisons for the proposed JHR and following six tag-based image retrieval approaches:

a) RR: Relevance-based ranking [22], an optimization framework is applied to automatically re-rank images based on visual and semantic information.
b) DR: Diverse ranking [23]. In this algorithm, the topic coverage of each image is calculated. Then, apply the

[1]The code can be found in page: https://github.com/wangyxxjtu/JHR-Code.

PageRank model based on the topic coverage to re-rank the initial retrieval results.

c) GBR: Global-based semi-supervised learning [21]. Semi-supervised learning has been widely applied in multimedia, such as image/object retrieval [25]–[27]. Here we adopt pseudo relevance feedback in Section VI-C.

d) PHR: Probability Hypergraph ranking [6]. Hyperedge adscription of a vertex is represented by a probability value rather than binary, then an optimization framework is applied to get the relevance scores of images.

e) HRPP: Hypersphere-based relevance preserving projection [61]. A hypersphere-based dimension-reduction algorithm is proposed, and the reranking is conducted by the new image features and estimated hypersphere center.

f) MGFR: Multi-graph fusion ranking [54]. Greedy selection is conducted first based on seed images, and multi-graph fusion mechanism is then applied to re-rank image.

In our baseline approach, we set the parameter K as 3 and set the hypergraph learning parameters $\lambda = 1$ and $\mu = 0.01$. Section VIII-B will discuss parameter K and discuss parameter λ and μ in section VIII-C. We set $\alpha = 0.1$, $A = 100$ respectively and discuss these two parameters in Section VIII-G and Section VIII-F respectively.

To make fair comparisons for seven methods, we use the parameters that the corresponding paper suggests for RR [22], DR [23], GBR [21] PHR [6], HRPP [61] and MGFR [54].

A. Dataset

In order to evaluate the performances of different approaches, we conduct experiment on NUS-Wide dataset. It contains 269648 images and 5018 unique tags from Flickr. Furthermore, NUS-Wide provides six types of low-level visual features including: color histogram (CH-64D), color correlation histogram (CORR-73D), edge-detection histogram (EDH-73D), block-wise color moments (CM-256D), and wavelet textures (WT-128D). Thus, we can directly use these features in our experiment.

B. Performance Evaluation

In the NUS-Wide dataset, each image is manually labeled into two relevance levels for test tags: 1-relevant and 0-irrelevant. Thus, we can evaluate the seven comparison methods objectively.

1) Criteria of Performance Evaluation: We use the NDCG [32] and average precision under depth n (denoted as $AP@n$) as the relevance performance evaluation which are expressed as follows:

$$NDCG@n = \frac{1}{W}\sum_{i=1}^{n}\frac{2^{rel_i}-1}{log(1+i)} \tag{20}$$

$$AP@n = \frac{1}{n}\sum_{i=1}^{n}\left(\sum_{j=1}^{i}\frac{rel_j}{i}\right) \tag{21}$$

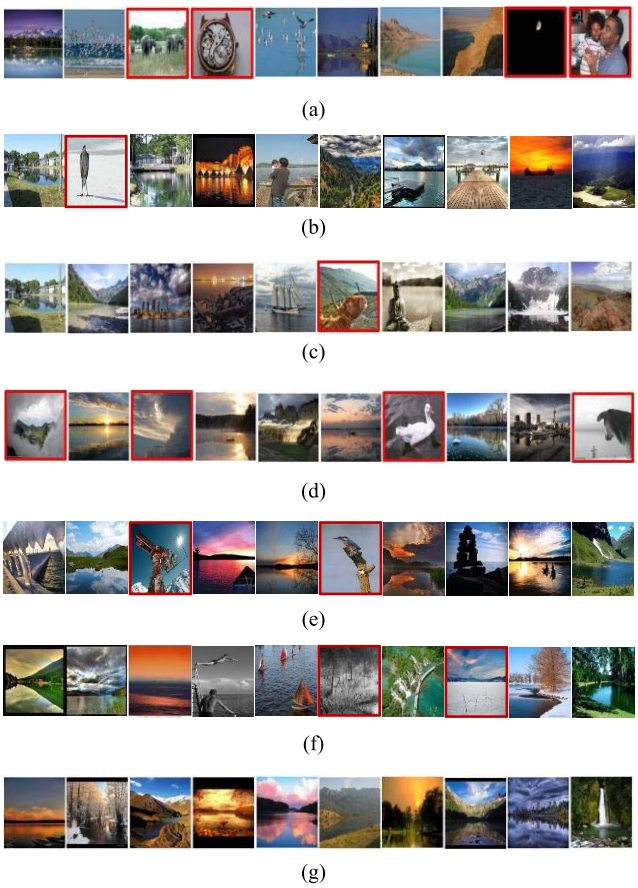

Fig. 2. Top 10 retrieval results for query "lake", red frame indicates the irrelevant images. (a) RR. (b) DR. (c) GBR. (d) PHR. (e) HRPP. (f) MGFR. (g) JHR.

where W is a normalization constant that is chosen so that the optimal ranking's NDCG score is 1, rel_i indicates the relevant level of image x_i to the query tag q, which is defined as:

$$rel_i = \begin{cases} 1, & if\ x_i\ is\ relevant\ to\ q \\ 0, & if\ x_i\ is\ irrelevant\ to\ q \end{cases} \tag{22}$$

2) Performance Analysis: Let MAP@n and MNDCG@n denote the mean values of AP@n and NDCG@n for all the 20 query tags. The MNDCG@n and MAP@n with depth=1, 10, 20, 50 and 100 are shown in Fig. 4 and Fig. 5. For example, the MNDCG@20 of seven methods are 0.4783, 0.609, 0.678, 0.709, 0.747, 0.713, and 0.789.

From Fig. 4 and Fig. 5, we can see that our JHR method achieves better performance than all the six competing methods. From Fig. 4, when the depth is 100, the MNDCG of JHR can reach 0.774, while RR, DR, GBR, PHR, HRPP and MGFR are 0.5102, 0.658, 0.682, 0.684, 0.702 and 0.689 respectively. MGFR can reach the same performance of top 1 with our JHR. However, with the depth deepening, our JHR performs much better than MGFR. This indicates that our proposed joint retrieval method is effective. From Fig.5, we can obtain the same conclusion.

Fig. 2 and Fig. 3 show the top 10 results for example query *lake* and *flowers* for all the seven methods respectively. The irrelevant results are marked by red frames. As shown

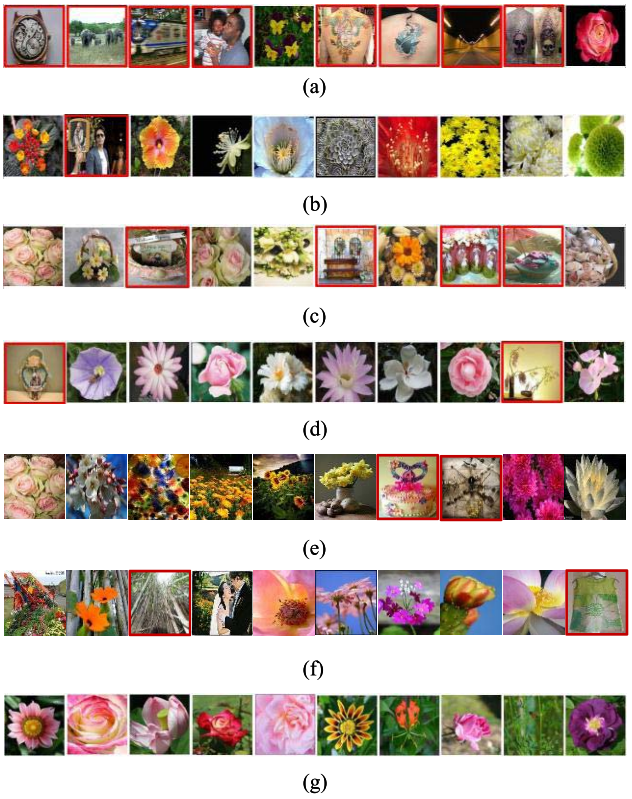

Fig. 3. Top 10 retrieval results for query "flowers", red frame indicates the irrelevant images. (a) RR. (b) DRR. (c) GBR. (d) PHR. (e) HRPP. (f) MGFR. (g) JHR.

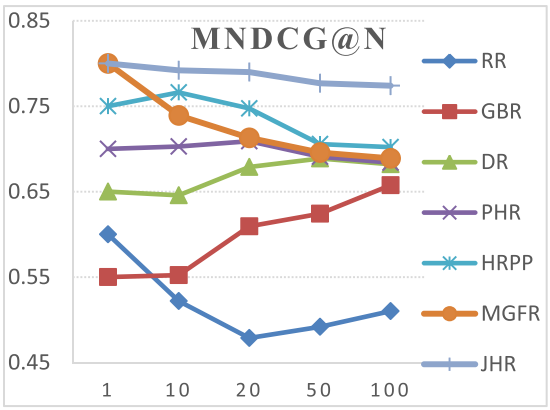

Fig. 4. The MNDCG of all 7 ranking methods under different depths.

in Fig.2 and Fig.3, the top ranked images determined by RR, DR GBR PHR, HRPP and MGFR all suffer from the lack of accuracy, their retrieval results all contain irrelevant images in top 10 retrieval results. From Fig.2 (a), we find that the third, the fourth images, the ninth and the tenth images of RR are irrelevant. The DR, GBR, PHR, HRPP and MGFR methods also introduce the irrelevant images. While results of our JHR method are all relevant.

VIII. Discussion

In this section, we completely discuss the impact of different parameters and the metric methods involved in our proposed

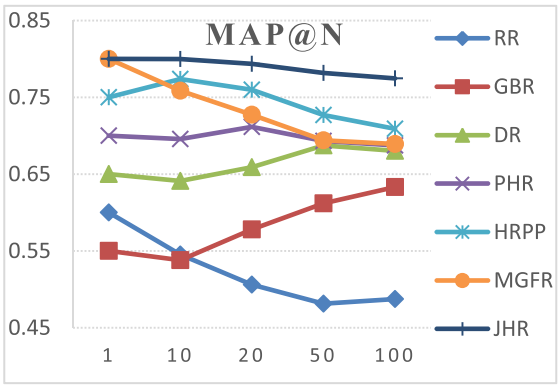

Fig. 5. The MAP of all 7 ranking methods under different depths.

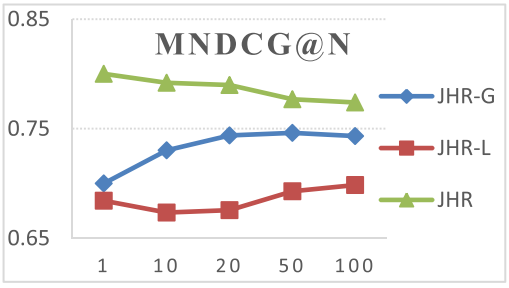

Fig. 6. MNDCG of three methods.

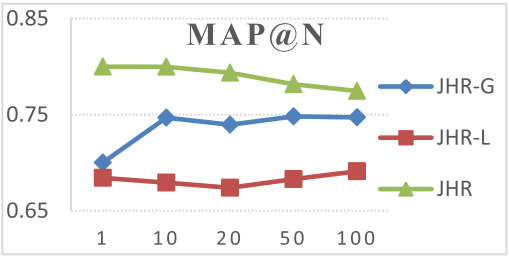

Fig. 7. MAP of three methods.

JHR method. We will validate our visual feature fusion method and discuss the number of visual words for the second hyperedge construction and the parameter in hypergraph learning. For metric methods, we will discuss about the relevance of image to query defined by Eq. (17). Besides parameters and metric methods, clustering algorithms in our method are also discussed in this section.

A. Discussion About the Visual Feature Fusion

In this subsection, we validate the proposed visual feature fusion method in this paper. Let JHR-G and JHR-L denote our method with only global and local visual features respectively. Fig. 6 and Fig.7 show the MNDCG and MAP of different methods respectively.

From Fig. 6 and Fig. 7, we can find that JHR performs much better than JHR-G and JHR-L. This means that using global and local features simultaneously is better than using any of them alone and our proposed global-local fusion mechanism is effective.

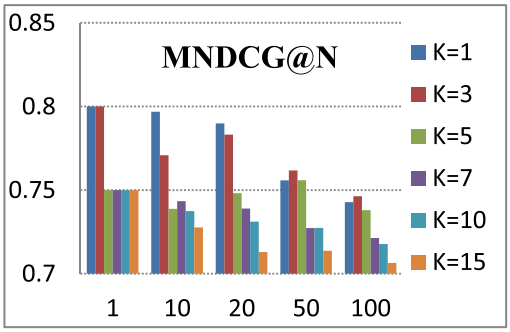

Fig. 8. The MNDCG of JHR under $K = \{1, 3, 5, 7, 10, 15\}$.

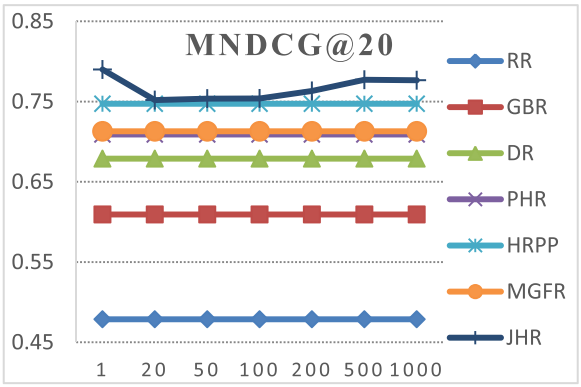

Fig. 9. The MNDCG of JHR under $\lambda = \{1, 20, 50, 100, 200, 500, 1000\}$.

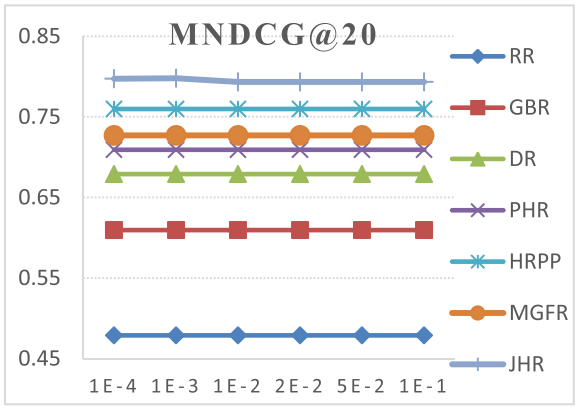

Fig. 10. The MNDCG of JHR under $\mu = \{0.0001, 0.001, 0.01, 0.02, 0.05, 0.1\}$.

We can also find that the JHR-G performs better than JHR-L. This is because that the global visual feature contains more information and is more reliable than the local visual words.

What needs to be mentioned is that the JHR-L method is very similar to the approach proposed in reference [34]. Gao *et al.* [34] first revise the tags, then they employ selected tags and the visual words to generate hyperedges. The re-ranking is conducted by the hypergraph learning. If we skip the tag refinement step and replace the pseudo-relevance feedback with ours, the JHR-L is equivalent to the method in [34].

B. Discussion About the Number of Visual Words in Second Layer Hyperedge Construction

In this part, we will discuss the parameter K in the second layer visual hyperedge construction. The parameter K represents the number of high frequency visual words that we select to generate the second layer hyperedge. The other parameters are the same as in Section VII.

Fig. 8 shows the MNDCG@n, and we can observe that under $K = 1$, 3, the MNDCG@1 can reach 0.8. In Fig.8, the MNDCG of $K = 1$ performs best under depth 1, 10, 20, when the depth is over 20, MAP of $K = 3$ is the best. With the depth deepening, JHR with $K = 3$ performs better than the others.

As we expressed in Section VI-B, the high frequency visual words are the instance of visual modalities appearing repeatedly among relevant images. From Fig. 8, we can see that there are 3 visual modalities shared by relevant images on average. Too lower K will miss the true visual modality and too higher will introduce the false visual modality, both situations will drop the performance.

C. Discussions About Weight λ and μ Selection

In this part, we discuss the impact of the regularization parameter λ and μ, set the other parameters be the same as in Section VII.

Fig. 9 and Fig. 10 show the MNDCG@20 performance curves with respect to the variation of λ and μ respectively. In Fig. 9, we fix $\mu = 0.01$ and vary λ from 1 to 1000 and In Fig. 10, we fix λtobe1 and vary μ from 0.0001 to 0.1. From Fig. 9 and Fig. 10, we can see that our method outperforms the comparison methods when λ and μ vary in a wide range and compared with parameter λ, μ have smaller effect on the experiment results.

D. Discussion About the Clustering Algorithm

In our algorithm, both the construction of first layer hypergraph and the pseudo relevance feedback use clustering algorithm. In this part, we discuss the impact of different clustering algorithms.

1) Clustering Algorithm in First Layer Hypergraph Construction: We choose the mean-shift clustering algorithm for the first layer visual hypergraph construction by global visual feature and the images in the same cluster form a hyperedge, see Section VI-B for details.

In this part, we conduct experiments using AP and k-means clustering algorithms and discuss the performance difference.

In Fig. 11, Global-A denotes the JHR with applying AP-clustering algorithm [27], Global-K denotes the JHR with applying k-means clustering and Global-M is the JHR with mean-shift clustering. From Fig. 11, Global-M outperforms Global-A and Global-K. Under different depths, the performances of JHR with three different clustering algorithms are close.

In our method, we map the visual similar images into the same group by clustering and give them close score

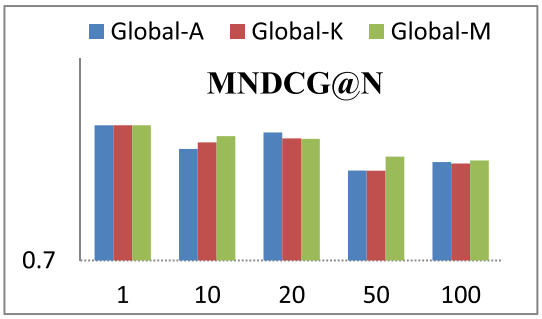

Fig. 11. The MNDCG of JHR applying different clustering algorithms in visual hypergraph construction.

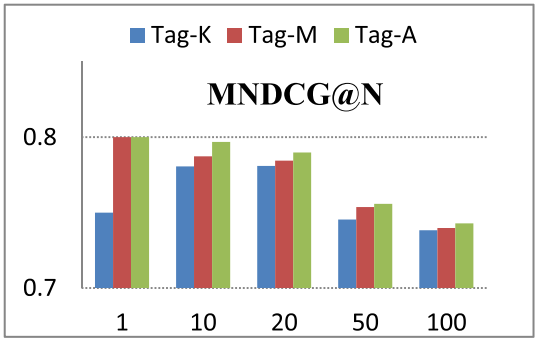

Fig. 12. The MNDCG of JHR applying different clustering algorithms in tag clustering.

by hypergraph learning. From Fig. 11, although JHR with mean-shift cluster is the best, the other two situations, JHR with AP cluster and K-means, also perform much better than the competing methods and are close to the JHR with mean-shift.

2) Clustering Algorithm in Pseudo-Relevance Feedback: In our pseudo feedback mechanism, we choose the AP clustering algorithm to cluster the co-occurrence tags of query q and estimate the relevance between image and query q based on these tag clusters, see Section VI-C for details.

In this part, we also apply mean-shift and k-means clustering for comparing. Fig. 12 shows the MNDCG of JHR with different clustering algorithms in pseudo relevance feedback. Tag-A, Tag-M, Tag-K denote the JHR with AP-clustering, mean-shift and k-means clustering for tag clustering in pseudo relevance feedback respectively.

From Fig. 12, we can observe that Tag-M and Tag-A perform better than Tag-K, Tag-M and Tag-A are very competitive.

In the relevance feedback, we choose the clustering algorithm to give a baseline score of images in the same cluster. From Fig. 12, we can find that performances of JHR with different clustering algorithms are close.

From the above discussion, we find that our JHR with different clustering algorithms all perform similarly and much better than the comparison methods. This means that our clustering idea is the key rather than the clustering algorithm we apply.

TABLE I
THE PERFORMANCES OF DIFFERENT MEASURES IN EQ. (17)

	MNDCG@20	MAP@20
Google Distance	0.78987	0.79364
Gaussian kernel	0.78999	0.78926
Cosine similarity	0.74211	0.73889

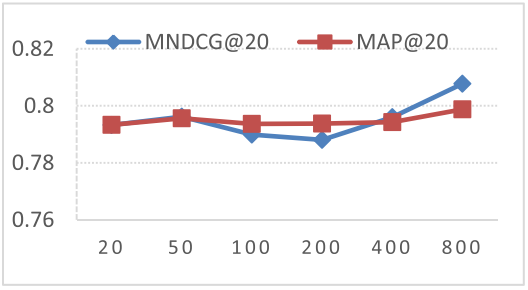

Fig. 13. performance under $A = \{20, 50, 100, 200\ 400, 800\}$.

E. Discuss the Relevance Measure of Image to Query

We use Eq. (17) to measure the similarity between query q and image x_i and $s(x_i, q)$ measures the semantic similarity between image x_i itself and q, which is calculated based on Google Distance.

In this part, we compare the impact on performance of JHR with different similarity measures. We choose the Gaussian kernel measure and cosine similarity for comparing. The similarity based on Gaussian kernel can be written as:

$$s_G(x_i, q) = \frac{1}{|T_i|} \sum_{t \in \mathcal{T}_i} \exp\left(-\frac{||v_q - v_t||^2}{2\sigma^2}\right) \tag{23}$$

where v_q and v_t are the word vectors of query q and tag t, T_i is the tag set of image x_i and $||\cdot||$ is the vector norm, $|T_i|$ is the number of tags in T_i, σ is a constant, we simply set it as 0.5.

The cosine similarity is defined as:

$$s_c(x_i, q) = \frac{1}{|T_i|} \sum_{t \in \mathcal{T}_i} \frac{< v_q, v_t >}{||v_q||||v_t||} \tag{24}$$

where v_t is the word vector of tag t.

Table 1 shows the performance comparison and we can observe that the Google distance and Gaussian kernel perform better than the cosine similarity, the former two are very competitive. Under the measure of average NDCG20, Gaussian kernel is better, while under average AP@20, Google distance is more outstanding.

F. Discussion About the Parameter A

In this subsection, we discuss the impact of parameter A,which deermines the number of pseudo-relevant images in our pseudo relevance feedback method. Please see Section VI-C for details.

Fig.13 shows the MAP@20 and MNDCG@20 for different A. We can see from Fig. 13 that the MAP and

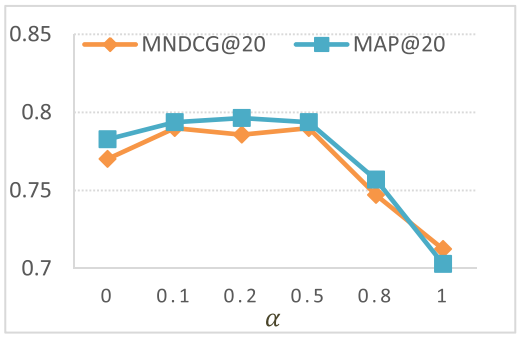

Fig. 14. performance under $\alpha = \{0, 0.1, 0.2, 0.5, 0.8, 1\}$.

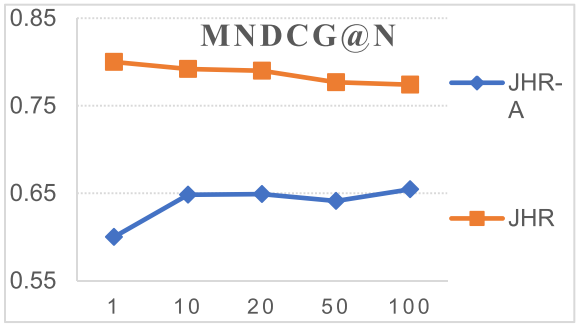

Fig. 15. The performance of two learning algorithms.

MNDCG exhibit the coincident tendency and the MAP performs more smoothly when the Avaries from 20 to 800. If $A = 20$, which means we only mark the top 20 images as the relevant and the left are the irrelevant, the performance is not that good. This may be too many truly relevant images are marked as irrelevant, the case of $A = 50$ can validate this explanation. When $A = 100$ and 200, the poor performance maybe because the top 100 and 200 introduce many irrelevant images that we mark as the relevant. when A is over 200, we can see that the MAP and MNDCG are consistent growth, this because we mark the most images as the correct relevant level.

It seems that the MNDCG experiences a sharp shock when $A = 100$, in fact, it only changes less than 0.006, which is acceptable. Thus, the whole performances shown in Fig.13 change smoothly.

G. Discussion About the Parameter α

In this part, we discuss the performance of JHR under different α. In Eq. (19), parameter α is the weight of image cluster score in pseudo relevance feedback. Fig. 14 shows the performance under different α.

In Fig. 14, the average NDCG and AP show the coincident tendency. We can find that the if $\alpha = 1$, which means that we mark images only based on the relevance score of cluster that the image belong to (the first term in Eq.(19)) and pay no attention to the relevance score of image itself to the tag (the second term in Eq.(19)). In this case, the performance is the worst. This is because the relevance score of cluster in Eq. (19) aims at giving a baseline score of images in the same cluster and is only the assistant of relevance score of image itself. While $\alpha = 0$ only focus on the relevance score of image itself to the tag, it also doesn't achieve the best performance. This reveals that the introducing of cluster relevance score cluster makes sense. We also observe that performance under $\alpha = 0.1, 0.2, 0.5$ are very close and $\alpha = 0.1$ performs best.

From the above discussions, we can see that our proposed method not only outperforms the comparing methods but exhibit relatively smooth change under different learning parameters, clustering methods and the metric methods. This means that our feature fusing mechanism is the key rather than the parameters and processing methods we apply

H. Discussion About the Learning Algorithm

Reference [35] also proposes an adaptively hypergraph learning algorithm. In this subsection, we conduct experiment to discuss the performance difference.

Fig.15 shows the MNDCG of two learning algorithms, where JHR-A is JHR with the learning algorithm proposed by [35]. From Fig. 15, the adaptive learning algorithm applied in this paper performs much better than the learning algorithm in [35]. This may be because:

1) The algorithm proposed by [35] is designed for classification task. While the learning algorithm in our paper is a learning to rank framework. The algorithm in [35] has greater demands on the labels. However, the image labels in our paper are marked by the pseudo-relevance feedback algorithm and are not the ground truth actually.
2) The images labeled as irrelevant are from more than one class. As introduced in our pseudo-relevance feedback algorithm, only a small part of images are labeled as relevant. The images labeled as irrelevant contains many noisy images, and these images are dissimilar with each other. Regarding these image as one class (the irrelevant) may confuse the learning procedure.

As a result, the learning algorithm in [35] exhibits poor performance. For example, the MNDCG@1 of JHR-A is only 0.6, while JHR can reach 0.8. And JHR is much more outstanding than JHR-A when the depth varies from 1 to 100. From the theoretical analysis and experiment results, the learning algorithm applied in this paper is more appropriate for our ranking task.

I. Discussion About the Deep Feature

In this paper, we employ the traditional features, i.e. color moment and texture feature, as the global features to represent images. While deep feature is widely used recently. In this subsection, we extract the 4096-D deep feature using FC-layer of pre-trained VGG-16 network, and conduct experiments to investigate the performance difference between traditional and deep features.

Fig. 16 shows the experiment results, where JHR-D and JHR-T are JHR with deep feature and traditional feature respectively, JHR-$\mathcal{G}_t$ and JHR-$\mathcal{G}_d$ represents JHR with only (global) traditional and deep features respectively while ignoring the local feature, JHR-L represents JHR with only local feature.

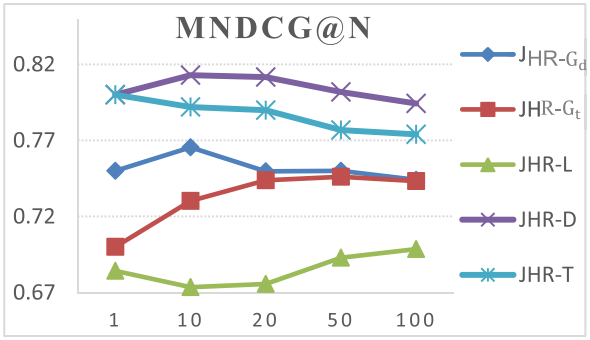

Fig. 16. The MNDCG of five methods.

TABLE II
THE TIME COST OF ALL METHODS (UNIT: SECOND)

Methods	Response time
RR	23351.9
DR	25332.3
GBR	8744.9
PHR	2469.9
MGFR	12574.8
HRPP	8700.2
JHR	36432.5

Fig. 16 shows the MNDCG of five methods. There are three important observations from Fig. 16:

1) JHR-D and JHR-$\mathcal{G}_d$ perform better than JHR-T and JHR-$\mathcal{G}_t$ respectively, which means that deep feature is better than traditional feature in our JHR algorithm.
2) JHR-T is more outstanding than JHR-$\mathcal{G}_D$. This reveals that although deep feature performs better than the traditional, only utilizing it alone can't reach the performance of utilizing traditional global and local features simultaneously, which indicates the high efficiency of our feature fusion mechanism.
3) The most important observation is that JHR-D performs much better than JHR-$\mathcal{G}_d$, JHR-L, and JHR-T is better than JHR-$\mathcal{G}_t$ and JHR-L. This indicates that the whether it's traditional feature or deep feature, using global and local features simultaneously is better than using any of them alone.

From above discussion, we can obtain two conclusions: 1) our feature fusion mechanism is the key instead of the feature we apply. 2) deep feature can also be used in our algorithm and can improve our performance further.

J. On the Response Time Comparison

In this subsection, we give the response time comparison for all the methods. We conduct all the test queries using Matlab R2015b on Windows 10 (X64) system with Intel Xeon E5 CPU and 32G memory. The results are shown in Table 2.

In order to finish the hypergraph learning on all the test tags, the necessary time is around 10 hours. In fact, the hypergraph construction and learning can be conducted offline, this is because we design the pseudo-relevance feedback algorithm to replace the user interaction and our algorithm can be conducted automatically and offline. The online retrieval is

TABLE III
NOTATIONS AND DEFINITIONS

The Symbol	Meaning
q	The query tag.
$\mathcal{X} = \{x_1, x_2, \dots, x_n\}$	$\mathcal{X}$ is the image set with q and x_i denotes the i-th image.
n	The number of images with q.
$\mathcal{G} = (\mathcal{V}, \mathcal{E}, \omega)$	A hypergraph $\mathcal{G}$, $\mathcal{V}, \mathcal{E}, \omega$ are the vertex set, hyperedge set and weights of hyperedge.
$H \in R^{n\times\lvert\mathcal{E}\rvert}$	Hypergraph incident matrix.
$v, d(v)$	A vertex and its corresponding degree.
$e, \delta(e), \omega(e)$	A hyperedge, corresponding degree and weight of the hyperedge respectively.
f	The relevance score vector that needs to be learned.
$y \in R^{n\times1}$	The relevance label vector of image to q.
$D_v \in R^{n\times n}$	The diagonal matrix of the vertex degree.
$D_e \in R^{\lvert\mathcal{E}\rvert\times\lvert\mathcal{E}\rvert}$	The diagonal matrix of the hyperedge degree.
$T = \{t_1, t_2, \dots, t_m\}$	T indicates co-occurrence tag set of query q, and t_i indicates the i-th tag.
$H = \{h_1, h_2, \dots, h_m\}$	H indicates the corresponding co-occurrence frequency of tag in T.
m	The number of co-occurrence tag with tag q.
T^*	The tag set selected for hyperedge construction.
$\mathrm{B} = \{b_1, b_2, \dots, b_{\lvert B\rvert}\}$	B indicates the clustering results by global visual feature, b_i indicates the i-th cluster.
K	The number of visual words selected to generate hyperedge in each cluster.
s_{tag}	The similarity of two tags.
v_i	The word vector of tag t_i.
$C = \{c_1, c_2, \dots, c_{\lvert C\rvert}\}$	C indicates the tag clustering results, and c_i indicates the i-th tag cluster.
$T^c = \{t_1^c, t_2^c, \dots, t_{\lvert T^c\rvert}^c\}$	T^c indicates the tag set in a tag cluster c, t_i^c indicates the i-th tag in cluster c.
h_i^c	The co-occurrence frequency of i-th tag in a tag cluster c.
r	The relevance score of cluster
$s(x_i, q)$	The semantic relevance of image x_i to tag q.
$\mathcal{T}_i$	The tag set of image x_i.
GD	The google distance.
X, G	X denotes the whole image dataset, G indicates the image number in X.
$F(x_i, q)$	The relevance score of image x_i to tag q estimated by pseudo relevance feedback.

only a key word matching process. Therefore, the time cost is accepted. The time cost mentioned above is counted by conducting proposed algorithm on the test tags sequentially. In fact, the learning process of each tag is independent and the program can be designed parallel, therefore the time cost can be further reduced.

IX. Conclusion and Future Work

In this paper, we propose a new joint re-ranking method for social image retrieval, in which we simultaneously utilize global, local visual features and textual feature to improve the retrieval accuracy. Experiment results on NUS-Wide dataset show that combing the global and local visual features is much better than using any of them alone and also more efficient than the comparison methods. The discussions in experiment show that our method has lighter dependence on the learning parameters, clustering methods and the metric methods we apply.

However, in our method, we only consider the relevance of result and ignore the diversity. In our future work, we will investigate the diversity by multiple visual features.

Appendix A
Notations and Definitions

See Table III.

References

[1] C. Xi, H. Guang, and X. Shunli, "An image registration method based on similarity of edge information," in *Proc. IEEE Int. Symp. Ind. Electron*. Piscataway, NJ, USA: IEEE Press, May 2012, pp. 217–224.

[2] X. Yang, Y. Zhang, T. Yao, C.-W. Ngo, and T. Mei, "Click-boosting multi-modality graph-based reranking for image search," *Multimedia Syst.*, vol. 21, no. 2, pp. 217–227, Mar. 2015.

[3] D. Zhou, J. Huang, and B. Schölkopf, "Learning with hypergraphs: Clustering, classification, and embedding," in *Proc. NIPS*, vol. 19, 2006, pp. 1–8.

[4] Y. Zhang, X. Yang, and T. Mei, "Image search reranking with query-dependent click-based relevance feedback," *IEEE Trans. Image Process.*, vol. 23, no. 10, pp. 4448–4459, Oct. 2014.

[5] X. Yang, T. Mei, Y. Zhang, J. Liu, and S. Satoh, "Web image search re-ranking with click-based similarity and typicality," *IEEE Trans. Image Process.*, vol. 25, no. 10, pp. 4617–4630, Oct. 2016.

[6] Y. Huang, Q. Liu, S. Zhang, and D. N. Metaxas, "Image retrieval via probabilistic hypergraph ranking," in *Proc. IEEE CVPR*, Jun. 2010, pp. 3376–3383.

[7] Q. Liu, Y. Huang, and D. N. Metaxas, "Hypergraph with sampling for image retrieval," *Pattern Recognit.*, vol. 44, nos. 10–11, pp. 2255–2262, 2011.

[8] L. Wang, Z. Zhao, and F. Su, "Tag-based social image search with hyperedges correlation," in *Proc. IEEE Vis. Commun. Image Process. Conf.*, Dec. 2014, pp. 330–333.

[9] J. Cai, Z.-J. Zha, M. Wang, S. Zhang, and Q. Tian, "An attribute-assisted reranking model for Web image search," *IEEE Trans. Image Process.*, vol. 24, no. 1, pp. 261–272, Jan. 2015.

[10] P. Jing, Y. Su, C. Xu, and L. Zhang, "HyperSSR: A hypergraph based semi-supervised ranking method for visual search reranking," *Neurocomputing*, vol. 274, pp. 50–57, Jan. 2016.

[11] Y. Xiang, X. Zhou, T. Chua, and C.-W. Ngo, "A revisit of generative model for automatic image annotation using Markov random fields," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jun. 2009, pp. 1153–1160.

[12] S. Agarwal, J. Lim, L. Zelnik-Manor, P. Perona, D. Kriegman, and S. Belongie, "Beyond pairwise clustering," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jun. 2005, pp. 838–845.

[13] Y. Huang, Q. Liu, and D. Metaxas, "Video object segmentation by hypergraph cut," in *Proc. IEEE Comput. Vis. Pattern Recognit.*, Jun. 2009, pp. 1738–1745.

[14] L. Sun, S. Ji, and J. Ye, "Hypergraph spectral learning for multi-label classification," in *Proc. SIG KDD*, 2008, pp. 668–676.

[15] Z. Tian, T. Hwang, and R. Kuang, "A hypergraph-based learning algorithm for classifying gene expression and arrayCGH data with prior knowledge," *Bioinformatics*, vol. 25, no. 21, pp. 2831–2838, Nov. 2009.

[16] D. G. Lowe, "Distinctive image features from scale-invariant keypoints," *Int. J. Comput. Vis.*, vol. 60, no. 2, pp. 91–110, Nov. 2004.

[17] R. Ji, H. Yao, X. Sun, B. Zhong, and W. Gao, "Towards semantic embedding in visual vocabulary," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jun. 2010, pp. 918–925.

[18] J. Yang, Y.-G. Jiang, A. G. Hauptmann, and C.-W. Ngo, "Evaluating bag-of-visual-words representations in scene classification," in *Proc. ACM SIGMM Workshop Multimedia Inf. Retrieval*, Sep. 2007, pp. 197–206.

[19] R. Ji *et al.*, "Location discriminative vocabulary coding for mobile landmark search," *Int. J. Comput. Vis.*, vol. 96, no. 3, pp. 290–314, Feb. 2012.

[20] Y. Jiang, C.-W. Ngo, and J. Yang, "Towards optimal bag-of-features for object categorization and semantic video retrieval," in *Proc. ACM Int. Conf. Image Video Retrieval*, 2007, pp. 494–501.

[21] D. Zhou, O. Bousquet, T. N. Lal, J. Weston, and B. Schölkopf, "Learning with local and global consistency," in *Proc. Adv. Neural Inf. Process. Syst.*, vol. 16. 2004, pp. 321–328.

[22] D. Liu, X.-S. Hua, M. Wang, and H. Zhang, "Boost search relevance for tag-based social image retrieval," in *Proc. IEEE Int. Conf. Multimedia Expo*, Jun./Jul. 2009, pp. 1636–1639.

[23] K. Song, Y. Tian, W. Gao, and T. Huang, "Diversifying the image retrieval results," in *Proc. 14th Annu. ACM Int. Conf. Multimedia*, 2006, pp. 707–710.

[24] M. Wang, X.-S. Hua, R. Hong, J. Tang, G.-J. Qi, and Y. Song, "Unified video annotation via multigraph learning," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 19, no. 5, pp. 733–746, May 2009.

[25] Y. Yang, F. Nie, D. Xu, J. Luo, Y. Zhuang, and Y. Pan, "A multimedia retrieval framework based on semi-supervised ranking and relevance feedback," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 34, no. 4, pp. 723–742, Apr. 2012.

[26] Y. Gao, M. Wang, D. Tao, R. Ji, and Q. Dai, "3-D object retrieval and recognition with hypergraph analysis," *IEEE Trans. Image Process.*, vol. 21, no. 9, pp. 4290–4303, Sep. 2012.

[27] B. J. Frey and D. Dueck, "Clustering by passing messages between data points," *Science*, vol. 315, no. 5814, pp. 972–976, Feb. 2007.

[28] *English Wikipedia Dataset*. Accessed: May 2015. [Online]. Available: https://dumps.wikimedia.org/enwiki/latest/enwiki-latest-pages-articles.xml.bz2

[29] H. Yu, M. Li, H.-J. Zhang, and J. Feng, "Color texture moments for content-based image retrieval," in *Proc. IEEE Int. Conf. Image Process.*, Sep.2002, pp. 929–932.

[30] T. Ojala, M. Pietikäinen, and D. Harwood, "Performance evaluation of texture measures with classification based on Kullback discrimination of distributions," in *Proc. IAPR Int. Conf. Pattern Recognit. (ICPR)*, vol. 1. 1994, pp. 582–585.

[31] Y. Cheng, "Mean shift, mode seeking, and clustering," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 17, no. 8, pp. 779–790, Aug. 1995.

[32] X. Qian, D. Lu, Y. Wang, L. Zhu, Y. Y. Tang, and M. Wang, "Image re-ranking based on topic diversity," *IEEE Trans. Image Process.*, vol. 26, no. 8, pp. 3734–3747, Aug. 2017.

[33] J. Yu, D. Tao, M. Wang, and Y. Rui, "Learning to rank using user clicks and visual features for image retrieval," *IEEE Trans. Cybern.*, vol. 45, no. 4, pp. 767–779, Apr. 2015.

[34] Y. Gao, M. Wang, Z. Zha, J. Shen, X. Li, and X. Wu, "Visual-textual joint relevance learning for tag-based social image search," *IEEE Trans. Image Process.*, vol. 22, no. 1, pp. 363–376, Jan. 2013.

[35] 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.

[36] S. Belongie, J. Malik, and J. Puzicha, "Shape matching and object recognition using shape contexts," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 24, no. 4, pp. 509–522, Apr. 2002.

[37] N. Dalal and B. Triggs, "Histograms of oriented gradients for human detection," in *Proc. IEEE Int. Conf. Comput. Vis. Pattern Recognit.* Piscataway, NJ, USA: IEEE Press, Jun. 2005, pp. 886–893.

[38] L. Duan, W. Li, I. W.-H. Tsang, and D. Xu, "Improving Web image search by bag-based reranking," *IEEE Trans. Image Process.*, vol. 20, no. 11, pp. 3280–3290, Nov. 2011.

[39] X. Tang, K. Liu, J. Cui, F. Wen, and X. Wang, "IntentSearch: Capturing user intention for one-click Internet image search," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 34, no. 7, pp. 1342–1353, Jul. 2012.

[40] X. Tian, D. Tao, X.-S. Hua, and X. Wu, "Active reranking for Web image search," *IEEE Trans. Image Process.*, vol. 19, no. 3, pp. 805–820, Mar. 2010.

[41] J. Yu, X. Yang, F. Gao, and D. Tao, "Deep multimodal distance metric learning using click constraints for image ranking," *IEEE Trans. Cybern.*, vol. 47, no. 12, pp. 4014–4024, Dec. 2017.
[42] J. Yu, Y. Rui, and D. Tao, "Click prediction for Web image reranking using multimodal sparse coding," *IEEE Trans. Image Process.*, vol. 23, no. 5, pp. 2019–2032, May 2014.
[43] E. Spyromitros-Xioufis, S. Papadopoulos, A. Ginsca, A. Popescu, Y. Kompatsiaris, and I. Vlahavas, "Improving diversity in image search via supervised relevance scoring," in *Proc. ICMR*, 2015, pp. 323–330.
[44] R. Yan and A. G. Hauptmann, "Query expansion using probabilistic local feedback with application to multimedia retrieval," in *Proc. ACM CIKM*, 2007, pp. 361–370.
[45] F. Jing and S. Baluja, "VisualRank: Applying pagerank to large-scale image search," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 30, no. 11, pp. 1877–1890, Nov. 2008.
[46] M. Wang, H. Li, D. Tao, K. Lu, and X. Wu, "Multimodal graph-based reranking for Web image search," *IEEE Trans. Image Process.*, vol. 21, no. 11, pp. 4649–4661, Nov. 2012.
[47] H.-M. Hou, X.-S. Xu, G. Wang, and X.-L. Wang, "Joint-rerank: A novel method for image search reranking," *Multimedia Tools Appl.*, vol. 74, no. 4, pp. 1423–1442, Feb. 2015.
[48] S. Liu, P. Cui, H. Luan, W. Zhu, S. Yang, and Q. Tian, "Social visual image ranking for Web image search," in *Proc. MMM*, 2013, pp. 239–249.
[49] K. Arai and R. Ali, "Hierarchical K-means: An algorithm for centroids initialization for K-means," *Rep. Faculty Sci. Eng. Saga Univ.*, vol. 36, no. 1, pp. 25–31, 2007.
[50] D. Lu, X. Liu, and X. Qian, "Tag-based image search by social re-ranking," *IEEE Trans. Multimedia*, vol. 18, no. 8, pp. 1628–1639, Aug. 2016.
[51] S. Jouili and S. Tabbone, "Hypergraph-based image retrieval for graph-based representation," *Pattern Recognit.*, vol. 45, no. 11, pp. 4054–4068, Nov. 2012.
[52] A. K. C. Wong, S. W. Lu, and M. Rioux, "Recognition and shape synthesis of 3-D objects based on attributed hypergraphs," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 11, no. 3, pp. 279–290, Mar. 1989.
[53] J. Bu *et al.*, "Music recommendation by unified hypergraph: Combining social media information and music content," in *Proc. ACM Int. Conf. Multimedia*, 2010, pp. 391–400.
[54] B. Lin, A. Wei, and X. Tian, "Visual re-ranking through greedy selection and rank fusion," in *Proc. Int. Conf. Multimedia Modeling*, 2016, pp. 289–300.
[55] V. L. Lekshmi and A. John, "Bridging the semantic gap in image search via visual semantic descriptors by integrating text and visual features," in *Advances in Intelligent Systems and Computing*, vol. 42. Singapore: Springer, Dec. 2015.
[56] A. Gordo, J. Almazan, J. Revaud, and D. Larlus, "Deep image retrieval: Learning global representations for image search," in *Proc. ECCV*, 2016, pp. 241–257.
[57] T. Mikolov, L. Sutskever, K. Chen, G. Corrado, and J. Dean, "Distributed representations of words and phrases and their compositionality," in *Proc. NIPS*, 2013, pp. 3111–3119.
[58] Y. Liu, J. Shao, J. Xiao, F. Wu, and Y. Zhuang, "Hypergraph spectral hashing for image retrieval with heterogeneous social contexts," *Neurocomputing*, vol. 119, pp. 49–58, Nov. 2013.
[59] L. Wang, Z. Zhao, and F. Su, "Efficient multi-modal hypergraph learning for social image classification with complex label correlations," *Neurocomputing*, vol. 171, pp. 242–251, Jan. 2016.
[60] Q. Fang, J. Sang, C. Xu, and Y. Rui, "Topic-sensitive influencer mining in interest-based social media networks via hypergraph learning," *IEEE Trans. Multimedia*, vol. 16, no. 3, pp. 796–812, Apr. 2014.
[61] J. Zhong, Y. Pang, and X. Li, "Relevance preserving projection and ranking for Web image search reranking," *IEEE Trans. Image Process.*, vol. 24, no. 11, pp. 4137–4147, Nov. 2015.
[62] C. Hong, J. Yu, J. Wan, D. Tao, and M. Wang, "Multimodal deep autoencoder for human pose recovery," *IEEE Trans. Image Process.*, vol. 24, no. 12, pp. 5659–5670, Dec. 2015.
[63] C. Hong, J. Yu, D. Tao, and M. Wang, "Image-based three-dimensional human pose recovery by multiview locality-sensitive sparse retrieval," *IEEE Trans. Ind. Electron.*, vol. 62, no. 6, pp. 3742–3751, Jun. 2015.
[64] C. Hong, J. Yu, J. You, X. Chen, and D. Tao, "Multi-view ensemble manifold regularization for 3D object recognition," *Inf. Sci.*, vol. 320, pp. 395–405, Nov. 2015.
[65] L. Zhu, J. Shen, L. Xie, and Z. Cheng, "Unsupervised topic hypergraph hashing for efficient mobile image retrieval," *IEEE Trans. Cybern.*, vol. 47, no. 11, pp. 3941–3954, Nov. 2017.
[66] L. Zhu, J. Shen, L. Xie, and Z. Cheng, "Unsupervised visual hashing with semantic assistant for content-based image retrieval," *IEEE Trans. Knowl. Data Eng.*, vol. 29, no. 2, pp. 472–486, Feb. 2017.
[67] L. Xie, J. Shen, J. Han, L. Zhu, and L. Shao, "Dynamic multi-view hashing for online image retrieval," in *Proc. IJCAI*, 2017, pp. 3133–3139.
[68] S. Wang, Q. Huang, S. Jiang, and Q. Tian, "S^3MKL: Scalable semi-supervised multiple kernel learning for real-world image applications," *IEEE Trans. Multimedia*, vol. 14, no. 4, pp. 1259–1274, Aug. 2012.
[69] J. Han, D. Zhang, G. Cheng, N. Liu, and D. Xu, "Advanced deep-learning techniques for salient and category-specific object detection: A survey," *IEEE Signal Process. Mag.*, vol. 35, no. 1, pp. 84–100, Jan. 2018.
[70] J. Han, R. Quan, D. Zhang, and F. Nie, "Robust object co-segmentation using background prior," *IEEE Trans. Image Process.*, vol. 27, no. 4, pp. 1639–1651, Apr. 2018.
[71] X. Yao, J. Han, D. Zhang, and F. Nie, "Revisiting co-saliency detection: A novel approach based on two-stage multi-view spectral rotation co-clustering," *IEEE Trans. Image Process.*, vol. 26, no. 7, pp. 3196–3209, Jul. 2017.
[72] X. Qian, X. Lu, J. Han, B. Du, and X. Li, "On combining social media and spatial technology for poi cognition and image localization," *Proc. IEEE*, vol. 105, no. 10, pp. 1937–1952, Oct. 2017.
[73] X. Qian, C. Li, K. Lan, X. Hou, Z. Li, and J. Han, "POI summarization by aesthetics evaluation from crowd source social media," *IEEE Trans. Image Process.*, vol. 27, no. 3, pp. 1178–1189, Mar. 2018.

Yaxiong Wang received the B.S. degree from Lanzhou University, Lanzhou, China, in 2015. He is currently pursuing the Ph.D. degree with the School of Software, Xi'an Jiaotong University, Xi'an, China. He is currently a Post-Graduate at the SMILES Laboratory, School of Electronic and Information Engineering, Xi'an Jiaotong University. His current research interests include tag-based image retrieval.

Li Zhu received the B.S. degree from Northwestern Polytechnical University in 1989 and the M.S. and Ph.D. degrees from Xi'an Jiaotong University in 1995 and 2000, respectively. He is currently an Associate Professor with the School of Software, Xi'an Jiaotong University. His main research interests include multimedia processing and communication, parallel computing, and networking.

Xueming Qian (M'10) received the B.S. and M.S. degrees from the Xi'an University of Technology, Xi'an, China, in 1999 and 2004, respectively, and the Ph.D. degree from the School of Electronics and Information Engineering, Xi'an Jiaotong University, in 2008. He was a Visiting Scholar with Microsoft Research Asia from 2010 to 2011. He was an Assistant Professor with Xi'an Jiaotong University, where he was an Associate Professor from 2011 to 2014, where he is currently a Full Professor. He is also the Director of the SMILES Laboratory, Xi'an Jiaotong University. His research is supported by the National Natural Science Foundation of China, Microsoft Research, and the Ministry of Science and Technology. His research interests include social media big data mining and search. He received the Microsoft Fellowship in 2006 and the Outstanding Doctoral Dissertations of Xi'an Jiaotong University and Shaanxi Province, in 2010 and 2011, respectively.

Junwei Han received the B.S. and Ph.D. degrees from Northwestern Polytechnical University, Xi'an China, in 1999 and 2003, respectively. He was a Research Follow with Nanyang Technological University, The Chinese University of Hong Kong, Dublin City University, and the University of Dundee. He was a Visiting Student with Microsoft Research Asia and a Visiting Researcher with the University of Surrey. He is currently a Professor with Northwestern Polytechnical University. His research interests include computer vision and multimedia processing.