Heterogeneous Graph Propagation for Large-Scale Web Image Search

Lingxi Xie, Qi Tian, *Senior Member, IEEE*, Wengang Zhou, and Bo Zhang

***Abstract*— State-of-the-art web image search frameworks are often based on the bag-of-visual-words (BoVWs) model and the inverted index structure. Despite the simplicity, efficiency, and scalability, they often suffer from low precision and/or recall, due to the limited stability of local features and the considerable information loss on the quantization stage. To refine the quality of retrieved images, various postprocessing methods have been adopted after the initial search process. In this paper, we investigate the online querying process from a graph-based perspective. We introduce a heterogeneous graph model containing both image and feature nodes explicitly, and propose an efficient reranking approach consisting of two successive modules, i.e., incremental query expansion and image-feature voting, to improve the recall and precision, respectively. Compared with the conventional reranking algorithms, our method does not require using geometric information of visual words, therefore enjoys low consumptions of both time and memory. Moreover, our method is independent of the initial search process, and could cooperate with many BoVW-based image search pipelines, or adopted after other postprocessing algorithms. We evaluate our approach on large-scale image search tasks and verify its competitive search performance.**

***Index Terms*— Large-scale web image search, postprocessing, heterogeneous graph propagation, incremental query expansion, image-feature voting.**

I. Introduction

IMAGE search with its self-owning visual contents, known as content based image retrieval (CBIR), has become a fundamental topic in the multimedia community and implies a wide range of real-world applications. To search among a large-scale image corpus, the Bag-of-Visual-Words (BoVW) model with the inverted index structure [1] is widely adopted. Conventional image search framework is often composed of two major stages, *i.e.*, offline indexing and online searching. On the offline stage, local descriptors [2], [3] are extracted on each image, quantized based on a large visual vocabulary [4], [5], and organized with an inverted index. On the online stage, local descriptors are also extracted on the query image, quantized into visual words, and used to access the corresponding entries in the inverted index. Finally, the retrieved image candidates are aggregated as the search results. The flowchart could also be applied to fine-grained image search challenges [6].

Although the BoVW-based image search pipelines are simple, efficient and scalable, they often suffers from low precision and/or recall. The main reasons include the unsatisfied repeatability of local descriptors and the loss of information through the quantization process. In fact, accurate matches between local features could be highly sensitive especially in the cases of manual editing and geometric deformation or stretching, meanwhile there also exist incorrect feature matches between some totally irrelevant images. This may cause some relevant images to be ranked after other irrelevant candidates. Various post-processing techniques are proposed to improve the quality of initial search results using additional clues such as geometric location of local features [7], extra features from the top-ranked images [8], and affinity values propagated between images [9]. Most of these proposed re-ranking approaches have been verified to improve the precision and/or recall of image search to some extent.

In this paper, we inherit a graph-based perspective to formulate the post-processing stage. For each query, we partition retrieved images into four categories, *i.e.*, true-positive (highly ranked, relevant), false-negative (lowly ranked, relevant), false-positive (highly ranked, irrelevant) and true-negative (lowly ranked, irrelevant) sets, and reveal that we can make full use of the relationship between (global) images and (local) features to boost the search performance. For this, we construct a **heterogeneous** graph containing two types of nodes, *i.e.*, images and features. Based on the graph structure, we propose two efficient algorithms, *i.e.*, Incremental Query Expansion (IQE) and Image-Feature Voting (IFV), to improve the recall and precision of initial search results, respectively. The proposed image search framework is illustrated in Figure 1. It is worth noting that the proposed

Manuscript received July 24, 2014; revised December 25, 2014; accepted April 16, 2015. Date of publication May 13, 2015; date of current version August 14, 2015. This work was supported in part by the Beijing Natural Science Foundation under Grant 4132046, in part by the Tsinghua University Initiative Scientific Research Program under Grant 20121088071, in part by the National Basic Research Program (973 Program) of China under Grant 2012CB316301 and Grant 2013CB329403, and in part by the National Natural Science Foundation of China under Grant 61332007, Grant 61273023, Grant 91120011, Grant 61128007, Grant 61429201, and Grant 61472378. The work of Q. Tian was supported by the U.S. Army Research Office under Grant W911NF-12-1-0057 and in part by Faculty Research Awards through NEC Laboratories America, Inc., Princeton, NJ, USA. The associate editor coordinating the review of this manuscript and approving it for publication was Prof. Marios S. Pattichis.

L. Xie and B. Zhang are with the State Key Laboratory of Intelligent Technology and Systems, Tsinghua National Laboratory for Information Science and Technology, Department of Computer Science and Technology, Tsinghua University, Beijing 100084, China (e-mail: 198808xc@gmail.com; dcszb@mail.tsinghua.edu.cn).

Q. Tian is with the Department of Computer Science, University of Texas at San Antonio, San Antonio, TX 78249 USA (e-mail: qitian@cs.utsa.edu).

W. Zhou is with the Department of Electronic Engineering and Information Science, University of Science and Technology of China, Hefei 230027, China (e-mail: zhwg@mail.ustc.edu.cn).

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.2015.2432673

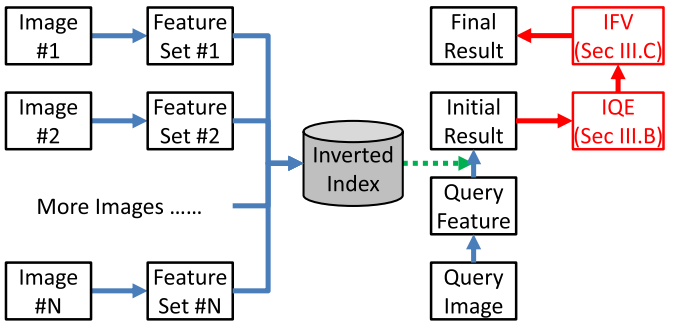


Fig. 1. The general image search framework consisting of offline indexing (left) and online querying (right) stages. The proposed IQE and IFV algorithms are marked in red (best viewed in color PDF).

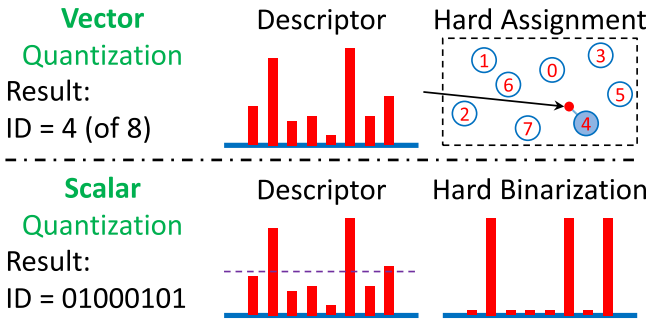


Fig. 2. Vector Quantization (result is a single ID of the nearest codeword in the feature space) and Scalar Quantization (result is a binary vector obtained by hard binarization). This figure is best viewed in color PDF.

re-ranking algorithms are geometry-free, *i.e.*, we do not need to store and use geometric information such as location, scale and orientation of local features, therefore both time and memory costs on the online querying stage are significantly reduced. Our algorithm also enjoys the advantage that being independent of the initial search process. Therefore, one might easily transplant the proposed modules onto many other image search pipelines. It is also possible to apply our method after other post-processing methods for even better results. We show in experiments that our approach achieves competitive search performance on a variety of image search tasks.

The remainder of this paper is organized as follows. We first review the state-of-the-art image search pipeline in Section II. Then Section III introduces the main algorithm, including intuition (in Section III-A) and two modules (IQE in Section III-B and IFV in Section III-C). After extensive experiments are shown in Section IV, we draw the conclusion in Section V.

II. The Image Search Pipeline

In this section, we give a brief overview of the image search pipeline based on the Bag-of-Visual-Words (BoVW) model and the inverted index structure.

A. Descriptor Extraction

We start from an image $\mathbf{I} = \left(a_{ij}\right)_{W \times H}$, where a_{ij} is the pixel on position (i, j). To make a robust representation, handcrafted descriptors are extracted on the regions-of-interest (ROI) of images. For ROI detection, popular algorithms include DoG [2], MSER [10], Hessian Affine [11] operators and dense interest points [12]. For local patch description, SIFT [2], [13], SURF [3], BRIEF [14], BRISK [15], FREAK [16] or USB [17] descriptors could be used. Any combination of ROI detection and patch description yields a set of local descriptors:

$$\mathcal{D} = \{(\mathbf{d}_1, \mathbf{l}_1), (\mathbf{d}_2, \mathbf{l}_2), \ldots, (\mathbf{d}_M, \mathbf{l}_M)\} \tag{1}$$

where $\mathbf{d}_m$ and $\mathbf{l}_m$ denote the m-th description vector and its geometric information (location, scale, orientation, *etc.*), respectively. M is the total number of descriptors.

B. Quantization

After descriptors have been extracted, they are often quantized to be compact. Generally, there are two ways of compressing descriptors into visual words.

The Vector Quantization (VQ) method requires training a codebook with descriptors sampled beforehand. Since the codebook size in image search is often quite large, say, one million, the hierarchical [4] or approximate [5] versions of K-Means are often adopted for acceleration. The descriptors are then encoded by the nearest codeword using hard quantization [4], or into a weighted combination of several codewords with soft quantization [18].

As an alternative choice of feature quantization, the Scalar Quantization (SQ) [19] method does not require training an explicit codebook. A D-dimensional descriptor $\mathbf{d}_m = \left(d_{m,1}, d_{m,2}, \ldots, d_{m,D}\right)$ can be transformed into a D-dimensional bit vector (elements are either 0 or 1) directly by simply setting a threshold, say, the median of descriptor values, and binarizing according to the threshold. The descriptors could also be encoded into longer bit vectors to preserve richer information. In Scalar Quantization, similarity between quantized features is formulated by Hamming distance. There also exist other codebook-free quantization methods [20], [21].

Figure 2 illustrates a comparison between Vector Quantization and Scalar Quantization. Denote the set of quantized local features as $\mathcal{F}$:

$$\mathcal{F} = \{(\mathbf{f}_1, \mathbf{l}_1), (\mathbf{f}_2, \mathbf{l}_2), \ldots, (\mathbf{f}_M, \mathbf{l}_M)\} \tag{2}$$

Here, each feature $\mathbf{f}_m$ is linked to a quantization unit (codeword), therefore it is convenient to say that two descriptors are matched, which means that they are quantized onto the same codeword in the visual vocabulary.

C. Feature Indexing

Large-scale image search often requires finding features' nearest or approximate nearest neighbors in a very short time, therefore the inverted index [1], [5] is often adopted as an efficient and scalable data structure for storing a large number of images and features. In essence, the inverted index is a compact linked list representation of a sparse matrix, in which rows and columns denote features and images, respectively. In the inverted index, each feature entry is followed by a list of image IDs to record its occurrences. Some other clues, such as geometric information of features, can also be stored for verification. On the online retrieval stage, we need only to check those images sharing common features with the query image, and the number of enumerated candidate images is greatly reduced.

The ways of indexing visual words after Vector Quantization and Scalar Quantization are also different. Vector Quantization produces a single ID or a weighted group of IDs for each descriptor, which could be used as the address of the inverted index entry directly. In contrast, the output of Scalar Quantization [19] is a D'-bit binary vector. We can take its first t bits as the indexing address, access the entry by hashing, and store the remaining $D'-t$ bits as well as image ID in the indexed feature. Although the amount of possible codewords is 2^t, which could be as many as 4G when $t = 32$, the number of appeared codewords, *i.e.*, with non-empty indexed feature lists, is much smaller, *e.g.*, less than 80M in practise. It is possible to allocate an entry for each of them.

D. Online Querying

Given a query image, local descriptors are also extracted, quantized and used to look up the inverted index. The retrieved image candidates are then ranked according to their frequencies of occurrence. Sometimes, visual words are weighted for better discrimination [22], [23].

In Scalar Quantization, a quite different online querying process is performed. When a descriptor is quantized (without training a codebook explicitly) into a D'-bit binary vector, we take out its t first bits as the natural address to visit the inverted file structure. In real practise, the search performance could be significantly improved with a soft expansion, which allows to visit all the addresses with at most d bits different with the querying binary vector among first t bits, and to retrieve those features in the index with at most κ different bits overall. d and κ are named *codeword expansion threshold* and *Hamming threshold*, respectively [19]. Given t and d, the number of enumerated inverted lists is $1+d+\frac{d(d-1)}{2}+\ldots+\binom{t}{d}$. Increasing d might result in a higher recall and, simultaneously, a more time-consuming querying process.

E. Post Processing

To improve search accuracy, initial search results are often re-ranked using various post-processing modules. Among these, **query expansion** [8], [24], [25] reissues initial top-ranked results to find valuable features which are not present in the original query; **spatial verification** [5], [26] filters those false-positives by checking geometric consistency of matched features, build efficient representation for false match filtering [7], [27], or extracts visual phrases [28] to verify matches on the more robust feature groups; and **diffusion-based algorithms** [29], including those based on graphs [9], [30]–[32] propagate affinities or beliefs via a graph structure to capture the high-level connections between images. Also there are other methods aimed at selecting high-quality features [33], discovering feature co-occurrence [34], [35], extracting contextual information [36]–[38], incorporating nearest-neighbor information [39], [40], adopting an alternative matching kernel [41], combining different searching results [42], or dealing with similarity between features [43]. Other recent proposed post-processing methods include [44], [45]. All the approaches have been verified to boost the precision or recall of search results to some extent.

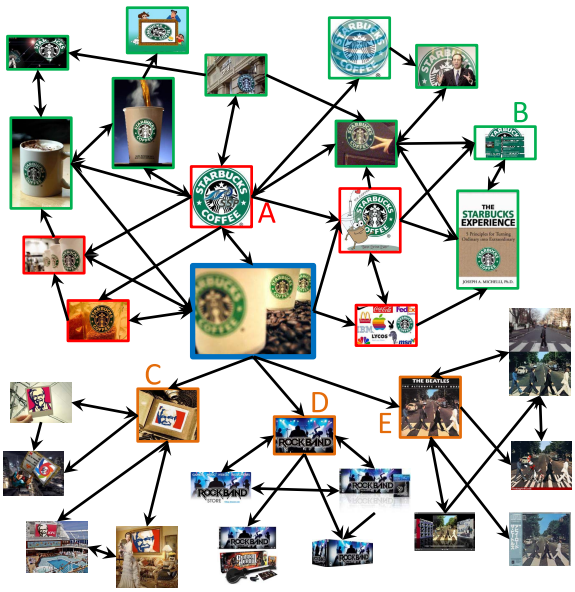


Fig. 3. Example of a toy image graph with four types of nodes (best viewed in color PDF). See the texts for detailed explanations.

The major contribution of this paper lies on the post-processing stage. Based on a graph-based perspective, we propose two geometry-free algorithms to improve the precision and recall of initial search results, respectively.

III. Heterogeneous Graph Propagation

In this section, we investigate the image search and re-ranking problem from an alternative perspective. We build a heterogeneous graph containing both image and feature nodes explicitly, and propose two geometry-free post-processing algorithms to improve precision and recall, respectively.

A. From Homogeneous to Heterogeneous

First of all, we visualize initial image search results using a graph structure in Figure 3. Each node in the graph denotes an image and there is a directed edge from image **X** to **Y** if and only if **Y** is ranked among the top-10 candidates when **X** is the query. We take a blurred and curved logo of *Starbucks Coffee* (in the blue frame) as the query. According to the results in Figure 3, other images in the graph could be categorized into four exclusive groups:

1) **true-positives** (in red frames, *e.g.*, **A**). These relevant images share a large number of common features with the query, and are naturally ranked among the top.
2) **false-negatives** (in green frames, *e.g.*, **B**). These relevant images do not have enough feature matches with the query, therefore are not ranked among the top. We want to promote the ranking of these images.
3) **false-positives** (in orange frames, *e.g.*, **C**, **D** and **E**). These irrelevant images somehow share a few common features with the query and are ranked among the top. We aim at filtering false matches on these images.
4) **true-negatives** (without colored frames). We could simply ignore these irrelevant images.

Obviously, it is the incorrect ranking of false-negatives and false-positives that causes the unsatisfied search performance. An ideal re-ranking algorithm should promote the false-negatives to improve the **recall**, meanwhile filter the false-positives to improve the **precision**. Intuitively, initial search results are obtained by simply counting the number of feature matches, but not all the features are equally important for a specified query image. If a feature, denoted as $\mathbf{f}$, is located on the semantic region of the query image, it can be used to retrieve the false-negative images containing $\mathbf{f}$, otherwise we shall decrease the weighting of this feature in order to weaken the false-positive images containing $\mathbf{f}$.

Due to the essential importance of features, we explicitly add them as another type of nodes into the image graph, turning the originally homogeneous (containing only image nodes) graph into heterogeneous (containing both image and feature nodes). In formal, we define a **heterogeneous graph** $\mathbf{G} = \{\mathcal{I}, \mathcal{F}, \mathcal{E}, \mathcal{S}, \mathcal{W}\}$. Here, $\mathcal{I} = \{\mathbf{I}_1, \mathbf{I}_2, \ldots, \mathbf{I}_N\}$ and $\mathcal{F} = \{\mathbf{f}_1, \mathbf{f}_2, \ldots, \mathbf{f}_M\}$ are the sets of image and feature nodes, and $\mathcal{E}$ is the set of undirected edges. There is a link between image $\mathbf{I}_n$ and feature $\mathbf{f}_m$ iff $\mathbf{f}_m$ appears in $\mathbf{I}_n$, *i.e.*, $\mathbf{I}_n$ contains at least one feature that shares the same quantization unit as $\mathbf{f}_m$. $\mathcal{S}$ and $\mathcal{W}$ denote the sets of **scores** for images, and **weights** for features. A sample graph is shown in Figure 5 (c). Based on the heterogeneous graph, we can perform efficient propagation algorithms to calculate the scores of images and weights of features, and use the scores to re-rank the images for better search quality.

B. Boosting Recall: Incremental Query Expansion

We start from the initial search process. Denote the query image as $\mathbf{I}_{q,0}$, and $\mathbf{f}_m \in \mathbf{I}_n$ (or equivalently $\mathbf{I}_n \ni \mathbf{f}_m$) if the feature $\mathbf{f}_m$ appears on the image $\mathbf{I}_n$. To be clear, we are working on the quantized features rather than the original descriptors, therefore $\mathbf{f}_m$ appears on $\mathbf{I}_n$ means that at least one feature in $\mathbf{I}_n$ shares the same quantization unit with $\mathbf{f}_m$. The initial search process starts by setting the weights of all features appeared in $\mathbf{I}_{q,0}$ to 1:

$$w(\mathbf{f}_m) = \mathbb{I}(\mathbf{f}_m \in \mathbf{I}_{q,0}) \tag{3}$$

and calculating the image scores by summing the weights of the related features:

$$s(\mathbf{I}_n) = \sum_{m, \mathbf{f}_m \in \mathbf{I}_n} w(\mathbf{f}_m) \tag{4}$$

Sorting the scores yields the initial image ranking:

$$\mathcal{I}_0 = \{\mathbf{I}_{0,1}, \mathbf{I}_{0,2}, \ldots, \mathbf{I}_{0,N}\} \tag{5}$$

However, due to the limited representative power of local features, such a simple method might produce a number of false-negatives. An intuitive solution arises from the observation of Figure 3, in which we can find a **true-positive path** connecting the query and most of the false-negatives. It suggests adding features in the initial top-ranked images into the search process. Here, we propose to expand the query with simple operations in the heterogeneous graph. Formally, we denote $\mathbf{I}_{0,1}$ in Eqn (5) as $\mathbf{I}_{q,1}$, which means that we take the top-ranked candidate, *i.e.*, the image most likely to be true-positive, as an extra query image. Then we update the feature weights according to the new query image set:

$$w(\mathbf{f}_m) = \sum_{r'=0}^{r} \mathbb{I}(\mathbf{f}_m \in \mathbf{I}_{q,r'}) \tag{6}$$

In the first round $r = 1$. The updated weights are used in Eqn (4) to calculate the image scores for re-ranking.

This iterative process is repeated for R rounds. In each round, we expand the query set with the highest-ranked image which is not considered before, and update feature weights and image scores orderly to re-rank the candidates. The proposed algorithm is named **Incremental Query Expansion** (IQE). R is the **maximal expansion rounds** and will be discussed in Section IV-B.

We illustrate the IQE algorithm with a real example shown in Figure 4. The query image is a faked sample of *Mona Lisa* on with only the face region is preserved. Obviously, only a few relevant features could be extracted on this image, and the initial search result is of low recall. With 3 iterative rounds of expansion, we find much more useful features to enrich the query information, and in this way we significantly improve the recall of search results.

C. Boosting Precision: Image-Feature Voting

It is worth noting that the above query expansion process might also bring in false-positives, *e.g.*, the third round of the example illustrated in Figure 4. For this reason, the precision of search results might still be unsatisfied. To filter out the irrelevant images, we make full use of the intrinsic relationship between images and features. A natural observation is that true-positive images often contain relevant features, and the relevant features are often found on true-positive images. Therefore we follow the affinity propagation algorithms on bipartite graphs [46] and design the **Image-Feature Voting** (IFV) process to iteratively update the scores and weights.

Recall that we have obtained the scores of images $s(\mathbf{I}_n)$ and the weights of features $w(\mathbf{f}_m)$ after the IQE algorithm. Each voting round consists of four stages, *i.e.*, image score normalization, image-to-feature voting, feature-to-image voting, and image re-ranking. The image score normalization process calculates a **belief function** $\Phi(\cdot)$ for each image:

$$\Phi(\mathbf{I}_n) = \exp(-\sigma \times \gamma(\mathbf{I}_n)) \tag{7}$$

where $\sigma = 0.5$ is the smoothing parameter and $\gamma(\mathbf{I}_n)$ is the current ranking of $\mathbf{I}_n$, *i.e.*, the top-ranked image has $\gamma(\mathbf{I}_n) = 1$, the runner-up has $\gamma(\mathbf{I}_n) = 2$, *etc.* Next, we perform the image-to-feature voting process in which we update the features' weights by accumulating images' scores:

$$w(\mathbf{f}_m) = \sum_{n, \mathbf{I}_n \ni \mathbf{f}_m} \Phi(\mathbf{I}_n) \tag{8}$$

and the feature-to-image voting process in which we reversely calculate images' scores by collecting weights of features:

$$s(\mathbf{I}_n) = \sum_{m, \mathbf{f}_m \in \mathbf{I}_n} w(\mathbf{f}_m) \tag{9}$$

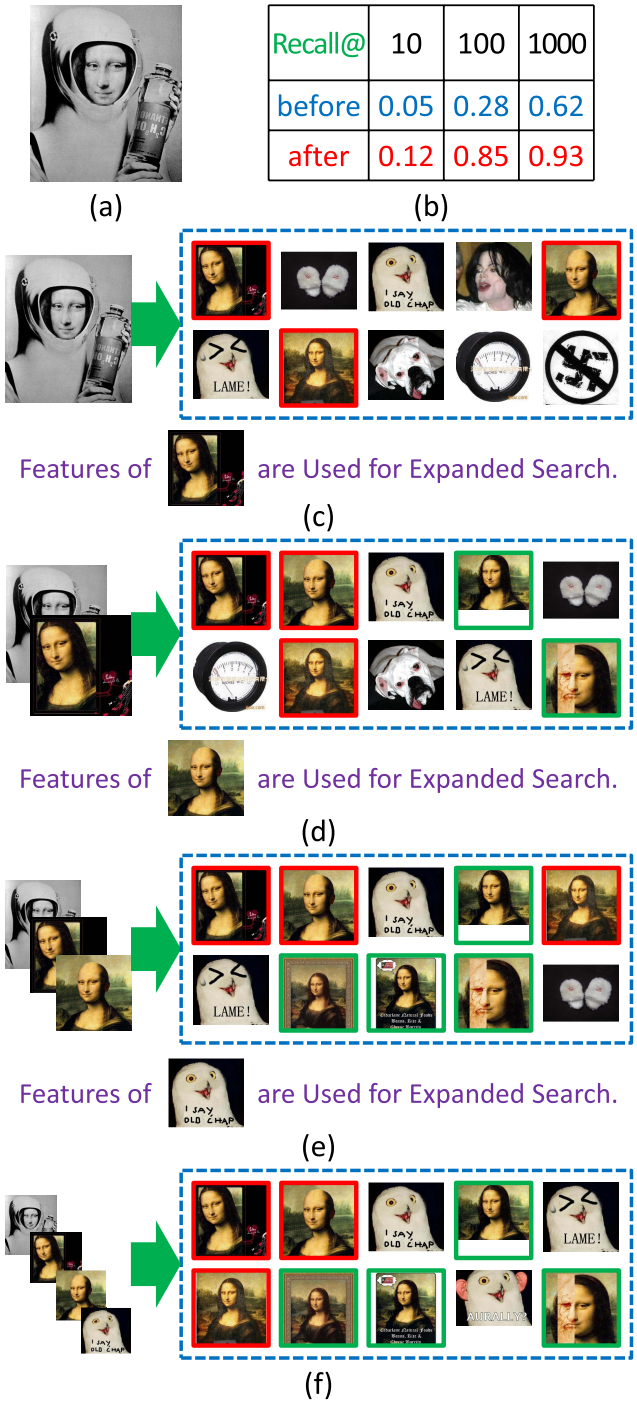


Fig. 4. A real example of Incremental Query Expansion (best viewed in color PDF). (a) The query image, with a relatively small semantic region (face). Due to the low quality, few images in the initial search results (c) are true-positives (marked with red solid frames). (b) Comparison of recall at top-10, top-100 and top-1000 returned images before and after the query expansion. (c) Initial search result (in the dashed frame). The features of the top-ranked image are extracted and used for expanded search. (d) The 1st expanded search process. (e) The 2nd expanded search process. (f) The 3rd expanded search (the expanded query is false-positive), which brings noises into the search result. Nevertheless, we retrieve much more true-positives which are not found initially (marked with green solid frames). The false-positives introduced in expansion could be efficiently filtered on the next stage, Image-Feature Voting.

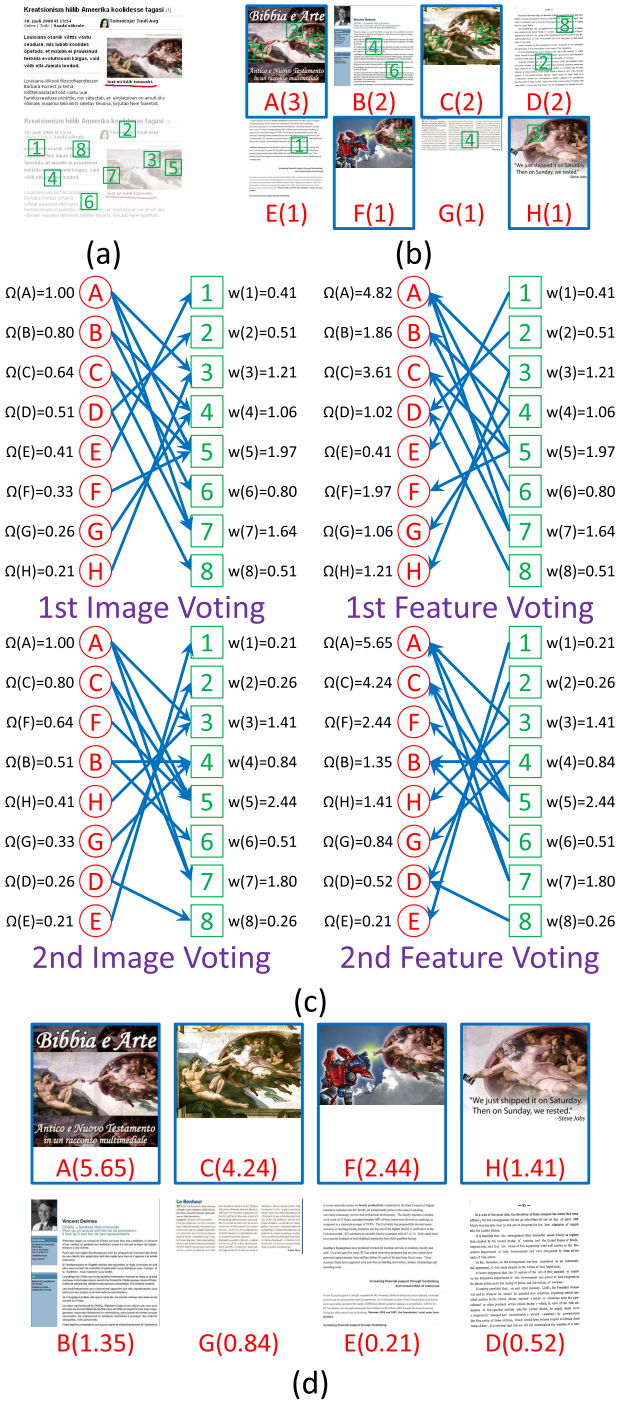


Fig. 5. A toy example to illustrate the Image-Feature Voting process (best viewed in color PDF). (a) The query image containing 8 features, among which only 3 are located on the semantic region (the paint). Here, we believe that the user's search intention is the picture (*The Creation of Adam*) instead of the surrounding texts. (b) The initial search result, where blue frames indicate the true-positives. The mAP value of the initial searching is only $0.67 = \frac{1}{4}(1/1 + 2/3 + 3/6 + 4/8)$. (c) Two voting rounds propagating the affinity values back and forth between images and features. After one round, we have obtained much improved search result (mAP is 0.95). After two rounds, all the true-positives are top-ranked (mAP is 1, see (d)) and the voting process converges. The top-3 weighted features (5, 7, 3) are just those ones on the semantic region of the query image. Numbers in parenthesis are the corresponding image scores.

Finally, we re-rank images according to the updated scores. The iteration would continue until convergence or the **maximal voting rounds** V is reached.

Please note that the scoring function $\Phi(\cdot)$ is very important in our approach. It serves as a regularizer after each round of voting. We do not make efforts to normalize the weights and

Algorithm 1 Heterogeneous Graph Propagation

Input: A set of images $\mathcal{I} = \{\mathbf{I}_1, \mathbf{I}_2, \ldots, \mathbf{I}_N\}$, a set of features $\mathcal{F} = \{\mathbf{f}_1, \mathbf{f}_2, \ldots, \mathbf{f}_M\}$, a query image $\mathbf{I}_{q,0}$, parameters R, V and σ.
procedure INCREMENTAL QUERY EXPANSION (IQE)
  Initialize weights $w(\mathbf{f}_m)$ by Eqn (3).
  Initialize scores $s(\mathbf{I}_n)$ by Eqn (4).
  for $r = 1, 2, \ldots, R$ **do**
    Find the expanded query image $\mathbf{I}_{q,r}$.
    Update weights and scores by Eqn (6) and (4).
  end for
end procedure
procedure IMAGE-FEATURE VOTING (IFV)
  Take the results produced by IQE as the initial ranking.
  for $v = 1, 2, \ldots, V$ **do**
    Compute the belief function by Eqn (7).
    Update feature weights by Eqn (8).
    Update feature scores and (9).
    Re-rank images according to their scores.
  end for
end procedure
Output: Re-ranked results, which is the ranklist after the final round of IFV.

scores of features and images, as the constant normalization factor is ignorable in the sorting operations.

Figure 5 shows a toy example containing 8 images and 8 features. One might observe that IFV improves the precision of search results significantly. Algorithm 1 illustrates an overall flowchart of IQE and IFV. Both procedures only involve linear array operations and could be implemented conveniently and efficiently in real practise.

D. Discussions

We propose two modules, *i.e.*, IQE and IFV, which play different but equally important roles in post-processing. Intuitively, IQE is aimed at discovering new connections between the query and originally false-negatives, whereas IFV is focused on re-calculating the weights of images and features for filtering false-positives. According to the results observed in Section IV-B, IQE and IFV help to boost the recall and precision of the initial search results, respectively, which comes up to the expectation for which we design the algorithms. Moreover, IQE and IFV cooperate with each other very well. As observed in Section IV-C, using one of them alone produces worse results than integrating them together.

Since both IQE and IFV algorithms include a quantization stage, and the accumulated scores are manually defined, it is difficult to provide a strict mathematical estimation on their convergence rates. However, according to the application of Random Walk theory on the HITS algorithm [46], convergence could be mostly guaranteed if the higher ranked elements (images or features) are always assigned with larger values. Other successful applications of Random Walk in information retrieval [9], [47] also provide evidences that such affinity propagation algorithms have good mathematical properties. In experiments, convergence is observed in every single case (each query in each dataset, see Section IV-B).

E. Acceleration

The proposed online querying stage consists of three parts, *i.e.*, initial search, IQE and IFV. The time cost of initial search depends on the baseline system.

For the IQE algorithm, we are actually performing R extra search processes, which would naturally require R-fold time of the baseline algorithm. We can reduce the time complexity of this stage by sacrificing the expanded search accuracy to some extent. For the Vector Quantization based methods such as [5] and [23], we can only take 20% most frequent visual words in the top-ranked image for expanded search. For Scalar Quantization [19], the time used for each expanded search could also be reduced to about 20% by adopting a tighter expansion threshold $d = 1$ instead of $d = 2$ (see Section II-D). Although this strategy causes the search accuracy drop slightly, it helps to reduce the time cost of a 10-round query expansion ($R = 10$, see Section IV-B) to just about twice of the initial search process.

Each round of the IFV algorithm requires propagating affinities between images and features. Suppose there are N images in the database, and each image contains M features in average, then a voting process with V rounds would require $2VNM$ arithmetic calculations in total. Typically we have $N = 10^6$, $M = 500$ and $V = 5$ (see Section IV-B), and the total number of floating operations is 5×10^9, requiring about 8s in a single 3.0GHz core. To accelerate, we can reduce the number of images in re-ranking by considering only top-U candidates after query expansion, where $U \ll N$ is named the **maximal voting candidates**. Consequently, the total calculation count is reduced to $2VUM$. Experimental results in Section IV-B reveal that in a database containing one million images, we could obtain very good results when we take merely $U = 1000$ top-ranked candidates. In this way, we could finish the voting process within 100ms (see Section IV-E).

F. Comparison to Previous Works

It is also instructive to discuss the difference between our work and previous post-processing methods for image search. We compare IQE and IFV with query expansion and diffusion-based re-ranking algorithms, respectively.

We perform IQE, an incremental process for query expansion, which is different with previous methods [8], [25], [48] in which all the images are considered and added simultaneously. IQE allows us to update the image ranking after each iteration and select the most competitive candidate for expansion. Consequently, it increases the probability that each expansion is made on a true-positive. Let us go back to the real example illustrated in Figure 4. If we simply take the top-10 candidates in the initial result for expansion, only 3 of them are true-positives. However, the number could increase to 7 when we perform the expansion in an incremental manner. We also count the above numbers over all the queries of the same dataset, and the average percentage of true-positives used for expansion (10 rounds) increases from 37% to 75% when we use the incremental expansion strategy. This helps us to bring

in more useful candidates and prevents from introducing too many noises.

On the other hand, the voting process between images and features is also a key innovation. Compared to previous works [9], [29], [49], our method enjoys simplicity (one can simply implement the voting process based on initial search results), stability (the convergence of voting process is guaranteed by the Random Walk theory [46]), and scalability (it is easy to scale up to the Web scale, *e.g.*, billions of images). Our method is also based on the assumption that each query shall have enough number of true-positives (see Section IV-F for details).

Another difference between our post-processing approach (IQE followed by IFV) and other competitors, such as [27] and [41], is that we do not need to use geometric information such as location, scale and orientation of visual words. By which, we reduce time and memory consumptions as shown in Section IV-E. The absence of geometric information makes it very difficult to perform spatial verification. Fortunately, we verify that the geometry-free image-feature voting process is also capable of filtering false-positives. Compared to the relevant works [39], [48] which is also geometry-free, we report higher search accuracy (see Section IV-C) thanks to the cooperation between IQE and IFV. We have performed several different strategies with [48] which is probably the most similar work to ours, such as adding the expanded images one by one, selecting features by image-feature voting, *etc*. In experiments, we observe higher accuracy and lower time complexity compared to [48].

Moreover, we propose a heterogeneous graph structure, which is different from the homogeneous one in [50], so that we can formulate both the IQE and IFV algorithms within the framework of affinity propagation. Although heterogeneous graph is previously adopted in very similar tasks [51], our work serves as a pioneer to represent images and features as nodes explicitly in a graph structure. The generalized formulation makes our approach very easy to implement yet efficient to carry out. It could be applied onto many other BoVW-based image search algorithms, even when other post-processing algorithms have been adopted in advance.

IV. Experiments

A. Datasets and Implementation Details

We conduct two parts of experiments on four publicly available datasets.

The first part is performed on two near-duplicate Web image datasets, *i.e.*, the DupImage dataset [52] containing 1104 images from 33 groups, and the CarLogo-51 dataset [32] containing 51 popular car logos and 11903 images. We also crawl one million irrelevant images from the Web as distractors. To evaluate performance with respect to the number of distractors, we construct 3 smaller subsets (100K, 200K and 500K) from the basic (1M) set by random sampling. All the greyscale images are resized with their aspect ratio preserved so that the larger axis has 300 pixels. We extract RootSIFT descriptors [41] on the regions-of-interest detected by the DoG operator [2], and use Scalar Quantization [19] to encode descriptors into 256-bit binary codes. The first 32 bits

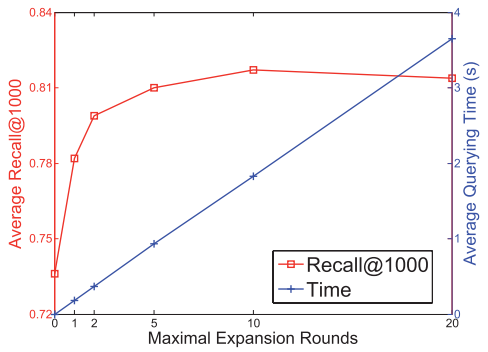


Fig. 6. The average recall-at-1000 value and average querying time (s) with respect to R in the IQE algorithm.

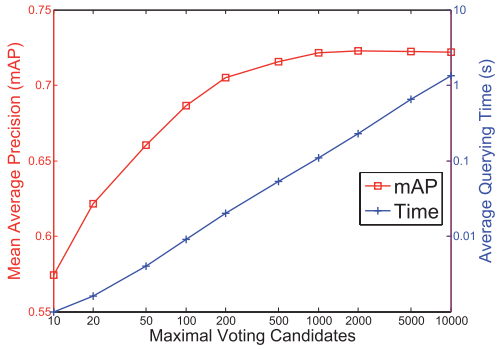


Fig. 7. The mAP value and average querying time (s) with respect to U in the IFV algorithm. We fix $R = 10$ and $V = 5$.

of quantized codes are taken as the address in the inverted index and the remaining 224 $(256 - 32)$ bits and image IDs are stored in the linked list. On the online querying stage, the inverted index is visited with codeword expansion threshold $d = 2$ ($d = 1$ in the IQE process for efficiency) and Hamming threshold $\kappa = 24$.

The second part is performed on two object retrieval datasets, *i.e.*, the Oxford buildings (Oxford5K) dataset [5] with 5063 images and 55 queries, and the Paris buildings (Paris6K) dataset [18] with 6391 images and 55 queries. Both datasets are mixed with 100K Flickr distractor images, which are also released by the same authors. All the images are turned into greyscale and not resized. RootSIFT descriptors [41] are extracted on the regions-of-interest detected by the Hessian Affine operators [11]. The Approximate K-Means (AKM) clustering [5] with one million codewords are performed, and descriptors are encoded with hard quantization. We use the ℓ_p-norm IDF [23] for computing feature weights in the initial search process.

It is worth noting that we have used two different approaches [19], [23] for initial search. We aim at evaluating our post-processing algorithm on different baseline systems to reveal its generalization ability.

B. Impact of Parameters

We study the impact of parameters in our approach, *i.e.*, maximal expansion rounds R, maximal voting candidates U and maximal voting rounds V. We evaluate the recall-at-1000 (in IQE) and the mAP value using the DupImage dataset with one million distractors, and plot the results in Figure 6, 7 and 8, respectively. One can see that both IQE and IFV significantly improve the quality of initial search results.

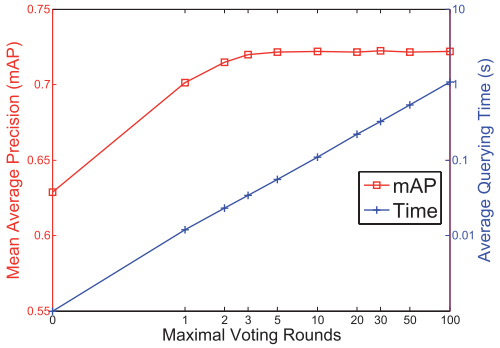


Fig. 8. The mAP value and average querying time (s) with respect to V in the IFV algorithm. We fix $R = 10$ and $U = 1000$.

TABLE I
THE mAP VALUES ON THE DUPIMAGE DATASET WITH DIFFERENT NUMBERS OF DISTRACTOR IMAGES

Dataset Size	100K	200K	500K	1M
HVVT [4]	0.5093	0.4727	0.4233	0.3776
HE [7]	0.5526	0.5125	0.4646	0.4287
SA [18]	0.6401	0.5963	0.5317	0.4804
GC [52]	0.6071	0.5868	0.5570	0.5294
SQ [19]	0.6289	0.6067	0.5736	0.5417
SQ + IQE	0.7643	0.7472	0.6971	0.6659
SQ + IFV	0.6873	0.6698	0.6469	0.6202
SQ + IQE + IFV	**0.8038**	**0.7822**	**0.7504**	**0.7216**

TABLE II
THE mAP VALUES ON THE CARLOGO-51 DATASET WITH DIFFERENT NUMBERS OF DISTRACTOR IMAGES

Dataset Size	100K	200K	500K	1M
HVVT [4]	0.1860	0.1755	0.1610	0.1500
HE [7]	0.2165	0.2036	0.1865	0.1733
SA [18]	0.2448	0.2326	0.2149	0.2004
SQ [19]	0.2449	0.2350	0.2238	0.2109
SQ + IQE	0.3187	0.3074	0.2954	0.2812
SQ + IFV	0.2819	0.2693	0.2578	0.2446
SQ + IQE + IFV	**0.3431**	**0.3314**	**0.3187**	**0.3014**

TABLE III
THE mAP VALUES ON THE OXFORD5K AND PARIS6K DATASETS, WITH OR WITHOUT FLICKR100K DISTRACTORS

Dataset	Oxford5K	–105K	Paris6K	–106K
RANSAC [5]	0.647	0.541	–	–
Soft Assignment [18]	0.825	0.719	0.718	0.605
Total-Recall [25]	0.827	0.767	0.805	0.710
Local Geometry [27]	0.916	–	0.885	–
Fine Vocabulary [53]	0.849	0.795	0.824	0.773
K-Reciprocal [39]	0.814	0.803	0.767	–
Three things [41]	**0.929**	**0.891**	**0.910**	–
Hamming-QE [48]	0.838	0.804	0.828	–
Hamming-QE+SP [48]	0.880	0.840	0.828	–
ℓ_p-IDF [23]	0.746	0.704	0.622	0.594
ℓ_p-IDF + IQE	0.825	0.806	0.789	0.740
ℓ_p-IDF + IFV	0.793	0.776	0.702	0.685
ℓ_p-IDF + IQE + IFV	**0.877**	**0.858**	**0.845**	**0.812**

We have also recorded the average querying time in each figure. It can be observed that the time complexity grows almost linearly with the parameters R, U and V, whereas using large parameters, say, $R = 20$ or $V = 10$, helps little to improve the search performance. We also observe quite similar regularity on other three datasets, either with or without distractor images, therefore we believe that the accuracy-complexity tradeoff is a general technique that boosts online querying efficiency. Therefore, we choose the following set of parameters in later experiments: maximal expansion rounds $R = 10$, maximal voting candidates $U = 1000$ and maximal voting rounds $V = 5$. Of course, the above parameters could be determined automatically by, for example, defining a stopping threshold. We do not develop an automatic parameter tuning algorithm, since the fixed parameters work well in other cases, *i.e.*, produce satisfied mAP values with reasonable time/memory consumptions in **all** the datasets. The observation above also provides an intuitive illustration of the post-processing modules' convergence properties. For every single case (each query in each dataset), the difference in mAP between using ($R = 20, V = 10$) and ($R = 50, V = 50$) is less than 0.01.

C. Performance Evaluation

We report the performance of our approach and other competitors on the near-duplicate experiments in Table I and II, respectively. On the DupImage and CarLogo-51 datasets with one million distractors, our approach reports the mAP value of 0.72 and 0.30, giving relatively 33.2% and 42.9% improvement over the baseline, Scalar Quantization [19].

We also report the performance of our approach and other competitors on the Oxford5K and Paris6K datasets with 100K distractors in Table III. One can observe that our algorithm also boosts the performance of the baseline retrieval system [23]. Moreover, our approach significantly outperforms another geometry-free post-processing algorithm [39], and works slightly better than a very related work, Hamming query expansion [48]. Although the reported accuracy of our approach is a little lower than some algorithms with complicated post-processing stages [27], [41], it is still promising considering it does not use any geometric information of local features, which enjoys the advantage of low time and memory consumptions as shown in Section IV-E. Our algorithm could also be appended after those algorithms to further improve the search performance with little extra computation.

Based on above two experiments, we can conclude that we have designed a generalized re-ranking algorithm, which could be applied after more than one image search baselines, and on both near-duplicate and object retrieval datasets.

D. Sample Results

Figure 9 shows a representative query in the DupImage dataset with one million distractors. Due to the low quality of the query image (the main part of the query image contains random noises), the numbers of efficient matches between the query image and some true-positives are even less than 3 (see the examples in blue or green frames for examples). For such poor matched images, both baseline algorithms ([4] and [19]) fail to retrieve them from the large database. With IQE and IFV, we successfully find most hard true-positives. It is worth noting that in this sample, we improve

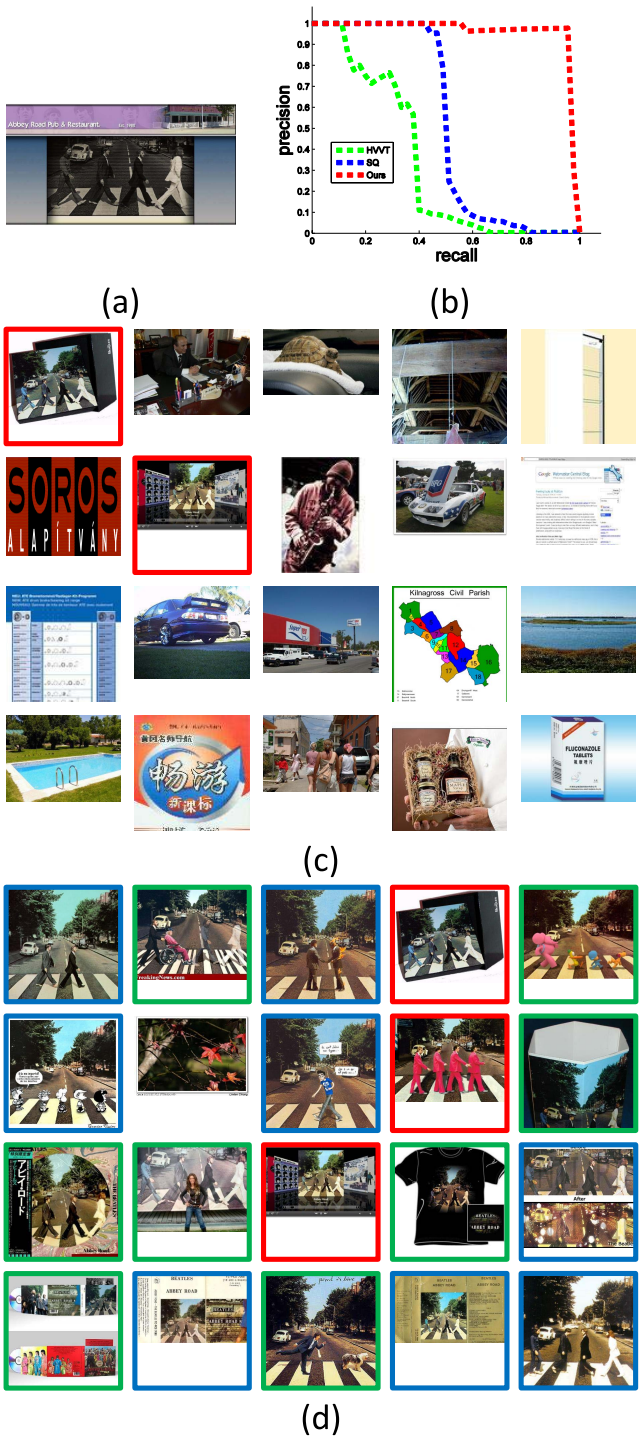


Fig. 9. A representative query and search results on the DupImage dataset with one million distractors (best viewed in color PDF). (a) A difficult query image with random noises. (b) Precision-recall curves on Hierarchical Visual Vocabulary Tree [4], Scalar Quantization [19] and our work. We obtain significant improvement on the mAP value (areas under the precision-recall curves). (c) (d) Top ranked (21 to 40) images returned by Scalar Quantization and our approach (the first 20 images are true-positives for both algorithms and therefore are not shown). We label true-positives with colored frames. In which, blue frames indicate images with only one or two feature match with the query, and green frames indicate those with no matches: it is difficult for the baseline algorithms ([4] and [19]) to rank these images among the top or even retrieve them.

the mAP (area under the precision-recall curve) impressively to 0.952, which is much better than 0.337 by [4] and 0.521 by [19].

TABLE IV
AVERAGE QUERYING TIME ON THE DUPIMAGE AND CARLOGO-51 DATASETS WITH DIFFERENT SETS OF DISTRACTORS. SCALAR QUANTIZATION [19] SERVES AS THE INITIAL SEARCH ALGORITHM. WE ONLY USE A SINGLE 3.0GHZ CORE. LISTED TIME IS IN MILLISECONDS

Dataset Size	100K	200K	500K	1M
Baseline (BL) [19]	77	132	276	477
IQE (this work)	150	255	532	932
IFV (this work)	18	31	59	110
HGP (BL+IQE+IFV)	245	418	867	1519

TABLE V
AVERAGE QUERYING TIME ON THE OXFORD5K AND PARIS6K DATASETS, WITH OR WITHOUT FLICKR100K DISTRACTORS. AKM WITH ℓ_p-IDF [23] SERVES AS THE INITIAL SEARCH ALGORITHM. WE ONLY USE A SINGLE 3.0GHZ CORE. LISTED TIME IS IN MILLISECONDS

Dataset Size	5K	105K	6K	106K
Baseline (BL) [23]	23	124	34	139
IQE (this work)	48	256	71	270
IFV (this work)	7	30	10	36
HGP (BL+IQE+IFV)	78	410	115	445

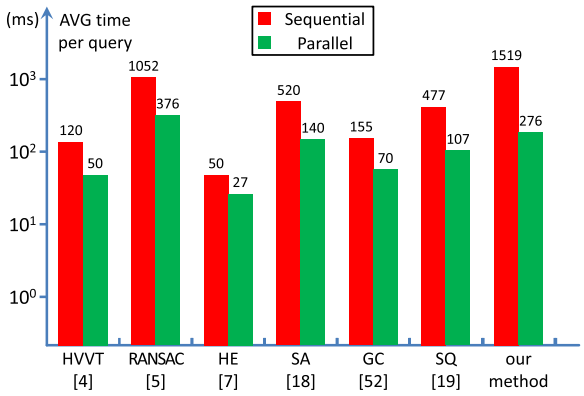


Fig. 10. Comparison of average querying time of different methods on the DupImage dataset with one million distractors (not including the time cost for descriptor extraction, quantization and indexing). The parallelized versions are implemented with all 8 cores.

E. Time and Memory Costs

We evaluate our algorithm on a CPU with 8 × 3.0GHz cores. A theoretical analysis of time complexity could be found in Section III-E.

The time cost of each separate module for near-duplicate and object retrieval datasets is listed in Table IV and V, respectively. All the techniques discussed in III-E have been adopted for acceleration. It is shown that our algorithm is highly scalable, as the time cost grows sub-linearly with the dataset size. For the DupImage and CarLogo-51 datasets with one million distractors, time cost of different approaches for a single query is summarized in Figure 10. For the Oxford105K and Paris106K datasets, our algorithm requires about 450ms to return search results, which is significantly faster than those complicated methods (both [27] and [41] requires more than

TABLE VI
PIECEWISE STATISTICS ON THE mAP GAIN OVER SCALAR QUANTIZATION [19]. RESULTS ARE REPORTED ON THE DUPIMAGE DATASET WITH ONE MILLION DISTRACTORS

mAP in SQ	Percentage	Average mAP Gain
0.0 – 0.2	23.4%	−0.0572
0.2 – 0.5	21.5%	+0.4327
0.5 – 0.8	23.4%	+0.3242
0.8 – 1.0	31.7%	+0.0769
Overall	100.0%	+0.1799

1s to process a query). Comparing with the most similar work [48] which requires 955ms for a query with spatial matching, our method works slightly better but requires only half time consumption.

An inverted file needs 4 bytes to store the image ID for each visual word for Vector Quantization [5], and 32 bytes to store the image ID with remaining binary codes for Scalar Quantization [19]. If the geometric information is also stored, at least 4 more bytes (for spatial coordinates) are required for a visual word. Our post-processing algorithms do not use geometric information, which saves 50% and 11% of the total storage in the above cases, respectively.

In one word, our algorithm beats the baseline performance significantly with two efficient post-processing modules. The time cost of these algorithms, as we have reported in the previous part, is about twice of the baseline algorithm. Compared to complicated post-processing algorithms such as [27], [41], and [48], our method requires much less time or memory consumptions. Therefore, it is convenient and worthwhile to transplant our model onto Web-scale image search applications.

F. Where Does the Improvement Come From?

Finally we conduct an interesting comparison between Scalar Quantization [19] and our work on the DupImage dataset with one million distractors. We partition query images into four groups according to the initial mAP value using baseline algorithm (Scalar Quantization), and calculate the averaged mAP gain in each group respectively. From the results shown in Table VI, we can observe that our approach improves the mAP value mainly on the queries with medium difficulties, *i.e.*, with initial mAP values between 0.2 and 0.8. For those very difficult samples with initial mAPs less than 0.2, our approach works even worse than the baseline algorithm, simply because we can find few true-positives in the top-ranked candidates for IQE, and IFV is also polluted by a dominant number of irrelevant features. Therefore, it is **not** reasonable to adopt our approach in the case that only few samples could be matched to each query image, such as in the UK-Bench dataset [4] (only 4 images in each near-duplicate group). Our approach fits very well for the large-scale Web image search problem, in which we can expect a large number of visually similar samples for each querying image.

V. CONCLUSIONS

The major innovation of this paper lies in the post-processing stage, on which we investigate the search process from a graph-based perspective. With an intuition to emphasize the importance of features, we introduce a **heterogeneous** graph consisting of both image and feature nodes explicitly, and propose two novel algorithms, *i.e.*, Incremental Query Expansion (IQE) and Image-Feature Voting (IFV), to boost the recall and precision of the initial search results, respectively. Since the proposed algorithms do not require using geometric information, time and memory consumptions on the online querying stage are significantly reduced. The independency between our method and initial search processes also makes it convenient to be adopted in many other tasks. Experiments on large-scale image datasets reveal that the proposed approach enjoys great advantages over the baseline methods [19], [23], and are more efficient in time and memory consumptions compared to complicated post-processing algorithms [27], [41].

REFERENCES

[1] J. Sivic and A. Zisserman, "Video Google: A text retrieval approach to object matching in videos," in *Proc. 9th IEEE Int. Conf. Comput. Vis.*, Oct. 2003, pp. 1470–1477.

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

[3] H. Bay, A. Ess, T. Tuytelaars, and L. Van Gool, "Speeded-up robust features (SURF)," *Comput. Vis. Image Understand.*, vol. 110, no. 3, pp. 346–359, 2008.

[4] D. Nister and H. Stewenius, "Scalable recognition with a vocabulary tree," in *Proc. IEEE Comput. Vis. Pattern Recognit.*, Jun. 2006, pp. 2161–2168.

[5] J. Philbin, O. Chum, M. Isard, J. Sivic, and A. Zisserman, "Object retrieval with large vocabularies and fast spatial matching," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jun. 2007, pp. 1–8.

[6] L. Xie, J. Wang, B. Zhang, and Q. Tian, "Fine-grained image search," *IEEE Trans. Multimedia*, vol. 17, no. 5, pp. 636–647, May 2015.

[7] H. Jegou, M. Douze, and C. Schmid, "Hamming embedding and weak geometric consistency for large scale image search," in *Proc. 10th Eur. Conf. Comput. Vis.*, 2008, pp. 304–317.

[8] O. Chum, J. Philbin, J. Sivic, M. Isard, and A. Zisserman, "Total recall: Automatic query expansion with a generative feature model for object retrieval," in *Proc. IEEE 11th Int. Conf. Comput. Vis.*, Oct. 2007, pp. 1–8.

[9] Y. 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.

[10] J. Matas, O. Chum, M. Urban, and T. Pajdla, "Robust wide-baseline stereo from maximally stable extremal regions," *Image Vis. Comput.*, vol. 22, no. 10, pp. 761–767, 2004.

[11] K. Mikolajczyk and C. Schmid, "Scale & affine invariant interest point detectors," *Int. J. Comput. Vis.*, vol. 60, no. 1, pp. 63–86, 2004.

[12] T. Tuytelaars, "Dense interest points," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jun. 2010, pp. 2281–2288.

[13] L. Xie, Q. Tian, and B. Zhang, "Max-SIFT: Flipping invariant descriptors for Web logo search," in *Proc. IEEE Int. Conf. Image Process.*, Oct. 2014, pp. 5716–5720.

[14] M. Calonder, V. Lepetit, M. Ozuysal, T. Trzcinski, C. Strecha, and P. Fua, "BRIEF: Computing a local binary descriptor very fast," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 34, no. 7, pp. 1281–1298, Jul. 2012.

[15] S. Leutenegger, M. Chli, and R. Y. Siegwart, "BRISK: Binary robust invariant scalable keypoints," in *Proc. IEEE Int. Conf. Comput. Vis.*, Nov. 2011, pp. 2548–2555.

[16] A. Alahi, R. Ortiz, and P. Vandergheynst, "FREAK: Fast retina keypoint," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jun. 2012, pp. 510–517.

[17] S. Zhang, Q. Tian, Q. Huang, W. Gao, and Y. Rui, "USB: Ultrashort binary descriptor for fast visual matching and retrieval," *IEEE Trans. Image Process.*, vol. 23, no. 8, pp. 3671–3683, Aug. 2014.

[18] J. Philbin, O. Chum, M. Isard, J. Sivic, and A. Zisserman, "Lost in quantization: Improving particular object retrieval in large scale image databases," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jun. 2008, pp. 1–8.

[19] W. Zhou, Y. Lu, H. Li, and Q. Tian, "Scalar quantization for large scale image search," in *Proc. 20th ACM Int. Conf. Multimedia*, 2012, pp. 169–178.

[20] W. Zhou, M. Yang, H. Li, X. Wang, Y. Lin, and Q. Tian, "Towards codebook-free: Scalable cascaded hashing for mobile image search," *IEEE Trans. Multimedia*, vol. 16, no. 3, pp. 601–611, Apr. 2014.

[21] W. Zhou, H. Li, R. Hong, Y. Lu, and Q. Tian, "BSIFT: Toward data-independent codebook for large scale image search," *IEEE Trans. Image Process.*, vol. 24, no. 3, pp. 967–979, Mar. 2015.

[22] H. Jegou, M. Douze, and C. Schmid, "On the burstiness of visual elements," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jun. 2009, pp. 1169–1176.

[23] L. Zheng, S. Wang, and Q. Tian, "$\mathcal{L}_p$-norm IDF for scalable image retrieval," *IEEE Trans. Image Process.*, vol. 23, no. 8, pp. 3604–3617, Aug. 2014.

[24] Y.-H. Kuo, K.-T. Chen, C.-H. Chiang, and W. H. Hsu, "Query expansion for hash-based image object retrieval," in *Proc. 17th ACM Int. Conf. Multimedia*, 2009, pp. 65–74.

[25] O. Chum, A. Mikulik, M. Perdoch, and J. Matas, "Total recall II: Query expansion revisited," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jun. 2011, pp. 889–896.

[26] O. Chum, M. Perdoch, and J. Matas, "Geometric min-hashing: Finding a (thick) needle in a haystack," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jun. 2009, pp. 17–24.

[27] M. Perd'och, O. Chum, and J. Matas, "Efficient representation of local geometry for large scale object retrieval," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jun. 2009, pp. 9–16.

[28] S. Zhang, Q. Huang, G. Hua, S. Jiang, W. Gao, and Q. Tian, "Building contextual visual vocabulary for large-scale image applications," in *Proc. ACM Int. Conf. Multimedia*, 2010, pp. 501–510.

[29] R. Fergus, P. Perona, and A. Zisserman, "A visual category filter for Google images," in *Proc. Eur. Conf. Comput. Vis.*, 2004, pp. 242–256.

[30] G. Kim and A. Torralba, "Unsupervised detection of regions of interest using iterative link analysis," in *Proc. Adv. Neural Inf. Process. Syst.*, 2009, pp. 961–969.

[31] W. Liu, Y.-G. Jiang, J. Luo, and S.-F. Chang, "Noise resistant graph ranking for improved Web image search," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jun. 2011, pp. 849–856.

[32] L. Xie, Q. Tian, W. Zhou, and B. Zhang, "Fast and accurate near-duplicate image search with affinity propagation on the ImageWeb," *Comput. Vis. Image Understand.*, vol. 124, pp. 31–41, Jul. 2014.

[33] P. Turcot and D. G. Lowe, "Better matching with fewer features: The selection of useful features in large database recognition problems," in *Proc. IEEE 12th Int. Conf. Comput. Vis. Workshops*, Sep./Oct. 2009, pp. 2109–2116.

[34] O. Chum and J. Matas, "Unsupervised discovery of co-occurrence in sparse high dimensional data," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jun. 2010, pp. 3416–3423.

[35] L. Xie, Q. Tian, M. Wang, and B. Zhang, "Spatial pooling of heterogeneous features for image classification," *IEEE Trans. Image Process.*, vol. 23, no. 5, pp. 1994–2008, May 2014.

[36] H. Jegou, C. Schmid, H. Harzallah, and J. Verbeek, "Accurate image search using the contextual dissimilarity measure," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 32, no. 1, pp. 2–11, Jan. 2010.

[37] X. Wang, M. Yang, T. Cour, S. Zhu, K. Yu, and T. Han, "Contextual weighting for vocabulary tree based image retrieval," in *Proc. IEEE Int. Conf. Comput. Vis.*, Nov. 2011, pp. 209–216.

[38] W. Zhou, Q. Tian, Y. Lu, L. Yang, and H. Li, "Latent visual context learning for Web image applications," *Pattern Recognit.*, vol. 44, nos. 10–11, pp. 2263–2273, 2011.

[39] D. Qin, S. Gammeter, L. Bossard, T. Quack, and L. Van Gool, "Hello neighbor: Accurate object retrieval with k-reciprocal nearest neighbors," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jun. 2011, pp. 777–784.

[40] X. Shen, Z. Lin, J. Brandt, S. Avidan, and Y. Wu, "Object retrieval and localization with spatially-constrained similarity measure and k-NN re-ranking," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jun. 2012, pp. 3013–3020.

[41] R. Arandjelovic and A. Zisserman, "Three things everyone should know to improve object retrieval," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jun. 2012, pp. 2911–2918.

[42] S. Zhang, M. Yang, T. Cour, K. Yu, and D. N. Metaxas, "Query specific fusion for image retrieval," in *Proc. Eur. Conf. Comput. Vis.*, 2012, pp. 660–673.

[43] D. Qin, C. Wengert, and C. Van Gool, "Query adaptive similarity for large scale object retrieval," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jun. 2013, pp. 1610–1617.

[44] Z. Liu, H. Li, W. Zhou, R. Zhao, and Q. Tian, "Contextual hashing for large-scale image search," *IEEE Trans. Image Process.*, vol. 23, no. 4, pp. 1606–1614, Apr. 2014.

[45] L. Zheng, S. Wang, Z. Liu, and Q. Tian, "Packing and padding: Coupled multi-index for accurate image retrieval," in *Proc. IEEE Comput. Vis. Pattern Recognit.*, Jun. 2014, pp. 1947–1954.

[46] J. M. Kleinberg, "Authoritative sources in a hyperlinked environment," *J. ACM*, vol. 46, no. 5, pp. 604–632, 1999.

[47] S. Brin and L. Page, "The anatomy of a large-scale hypertextual Web search engine," *Comput. Netw. ISDN Syst.*, vol. 30, nos. 1–7, pp. 107–117, 1998.

[48] G. Tolias and H. Jegou, "Visual query expansion with or without geometry: Refining local descriptors by feature aggregation," *Pattern Recognit.*, vol. 47, no. 10, pp. 3466–3476, 2014.

[49] M. Donoser and H. Bischof, "Diffusion processes for retrieval revisited," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jun. 2013, pp. 1320–1327.

[50] C. Deng, R. Ji, W. Liu, D. Tao, and X. Gao, "Visual reranking through weakly supervised multi-graph learning," in *Proc. IEEE Int. Conf. Comput. Vis. (ICCV)*, 2013, pp. 2600–2607.

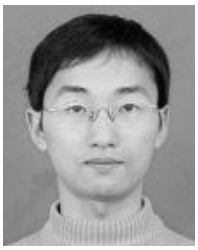
[51] H. Ma, J. Zhu, M. R.-T. Lyu, and I. King, "Bridging the semantic gap between image contents and tags," *IEEE Trans. Multimedia*, vol. 12, no. 5, pp. 462–473, Aug. 2010.

[52] W. Zhou, H. Li, Y. Lu, and Q. Tian, "Large scale image search with geometric coding," in *Proc. 19th ACM Int. Conf. Multimedia*, 2011, pp. 1349–1352.

[53] A. Mikulik, M. Perdoch, O. Chum, and J. Matas, "Learning vocabularies over a fine quantization," *Int. J. Comput. Vis.*, vol. 103, no. 1, pp. 163–175, 2013.

Lingxi Xie received the B.E. degree from the Department of Computer Science and Technology, Tsinghua University, China, in 2010, where he is currently pursuing the Ph.D. degree. He was a Research Intern with Microsoft Research Asia from 2013 to 2014, and from 2014 to present. He was a Visiting Researcher with the Department of Computer Science, University of Texas at San Antonio, in 2014. His research interests include computer vision, multimedia information retrieval, and machine learning.

Qi Tian (M'96–SM'03) received the B.E. degree in electronic engineering from Tsinghua University, China, in 1992, the M.S. degree in electrical and computer engineering from Drexel University, in 1996, and the Ph.D. degree in electrical and computer engineering from the University of Illinois, Urbana–Champaign, in 2002. He is currently a Professor with the Department of Computer Science at the University of Texas at San Antonio. His research interests include multimedia information retrieval and computer vision. He has authored over 290-refereed journal and conference papers. His research projects were funded by NSF, ARO, DHS, SALSI, CIAS, and UTSA, and he received faculty research awards from Google, NEC Laboratories of America, FXPAL, Akiira Media Systems, and HP Labs. He took a one-year faculty leave at Microsoft Research Asia (MSRA) from 2008 to 2009. He was the co-author of an ACM ICMR 2015 Best Paper, ICMCS 2012 Best Paper, a MMM 2013 Best Paper, a PCM 2013 Best paper, a Top 10% Paper Award in MMSP 2011, a Best Student Paper in ICASSP 2006, and co-author of a Best Paper Candidate in PCM 2007. He received 2010 ACM Service Award. He has been serving as Program Chairs, Organization Committee Members, and TPCs of numerous IEEE and ACM Conferences, including ACM Multimedia, SIGIR, ICCV, and ICME. He is in the Editorial Board of IEEE TRANSACTIONS ON MULTIMEDIA, the IEEE TRANSACTIONS ON CIRCUIT AND SYSTEMS FOR VIDEO TECHNOLOGY, *Multimedia System Journal*, the *Journal of Multimedia*, and the *Journal of Machine Visions and Applications*. He is also the Guest Editor of the IEEE TRANSACTIONS ON MULTIMEDIA, the *Journal of Computer Vision and Image Understanding*, *Pattern Recognition Letters*, the *EURASIP Journal on Advances in Signal Processing*, and the *Journal of Visual Communication and Image Representation*.



Wengang Zhou received the B.E. degree in electronic information engineering from Wuhan University, China, in 2006, and the Ph.D. degree in electronics engineering and information science from the University of Science and Technology of China (USTC), China, in 2011. He was a Research Intern with the Internet Media Group, Microsoft Research Asia, from 2008 to 2009. From 2011 to 2013, he was a Post-Doctoral Researcher with the Computer Science Department, University of Texas at San Antonio. He is currently an Associate Professor with the Department of Electronic Engineering and Information Science, USTC. His research interest is mainly focused on multimedia content analysis and retrieval.

Bo Zhang received the B.E. degree from the Department of Automatic Control, Tsinghua University, China, in 1958. From 1980 to 1982, he visited the University of Illinois at Urbana–Champaign, USA, as a Scholar. He is currently a Professor with the Department of Computer Science and Technology, Tsinghua University. His research interests include artificial intelligence, machine learning, pattern recognition, knowledge engineering, intelligent robotics, and intelligent control. He is a member of the Chinese Academy of Sciences. He won the Prize of European Artificial Intelligence in 1984. He won the 1st-class of Science and Technology Progress Prize three times in 1987, 1993, and 1998, and the 3rd-class Prize two times in 1995 and 1999, respectively.