Fast Image Retrieval: Query Pruning and Early Termination

Liang Zheng, Shengjin Wang, Member, IEEE, Ziqiong Liu, and Qi Tian, Senior Member, IEEE

Abstract—Efficiency is of great importance for image retrieval systems. For this pragmatic issue, this paper proposes a fast image retrieval framework to speed up the online retrieval process. To this end, an impact score for local features is proposed in the first place, which considers multiple properties of a local feature, including TF-IDF, scale, saliency, and ambiguity. Then, to decrease memory consumption, the impact score is quantized to an integer, which leads to a novel inverted index organization, called Q-Index. Importantly, based on the impact score, two closely complementary strategies are introduced: query pruning and early termination. On one hand, query pruning discards less important features in the query. On the other hand, early termination visits indexed features only with high impact scores, resulting in the partial traversing of the inverted index. Our approach is tested on two benchmark datasets populated with an additional 1 million images to account as negative examples. Compared with full traversal of the inverted index, we show that our system is capable of visiting less than 10% of the “should-visit” postings, thus achieving a significant speed-up in query time while providing competitive retrieval accuracy.

Index Terms—Early termination, image retrieval, impact score, query pruning.

I. INTRODUCTION

Recent years have witnessed dramatic accuracy improvement of image retrieval thanks to the introduction of invariant local features [1], which gives rise to the widely used Bag-of-Words (BoW) model [2]. Inspired from classic text retrieval approaches, the BoW model transforms an image into a histogram of visual words produced by feature quantization. Typically, visual words are defined by a codebook trained on a feature pool with various clustering algorithms [3], [4]. To promote retrieval efficiency in terms of both time and memory, an inverted index is employed.

Efficiency is a central issue in any retrieval system, which should return the relevant documents/images to the user in real time. The online image retrieval procedure can be divided into three steps: feature extraction, feature quantization, and inverted index traversing. For web-scale applications, as the number of indexed images is getting larger, the inverted index traversing is the only one of increasing time out of the three parts. Therefore, this paper aims at improving efficiency of the traversing part. During this process, for each feature in a query image, an inverted list of postings (or indexed features) is identified from the corresponding entry in the inverted index. The scores of these images are increased correspondingly. In essence, two steps have direct impacts on efficiency: a loop over all query features and a full traverse of all inverted lists associated with query.

On one hand, an image is typically described by several thousand visual words in the BoW model. Nevertheless, unlike words in the text, visual words in an image do not have a semantic meaning and are subject to the ambiguity issue [5]. Except for some statistical cues such as TF-IDF [2], [6], we are unlikely to tell whether these words are important for retrieval. A brute-force loop of all query features not only over-uses the less important ones, but also leads to a large amount of requests on the inverted index. In contrast, for a common query in text retrieval, only several query words are submitted. Image retrieval is a typical long query problem and effective query pruning strategy is needed for this task.

On the other hand, a major bottleneck of inverted index traversing is the length of the inverted list. Roughly, it grows linearly with the database size. For a common visual word, its inverted list could be as long as hundreds of MBs or even GBs in a web-scale network. An exhaustive traverse on the inverted index can be extravagantly expensive. This problem appears more severe if we consider a typical case where a query contains hundreds of visual words. It means that a non-trivial fraction in the image collection needs to be visited before the query is solved. For each query feature, a naive full traverse of the postings takes considerable time. As a consequence, it would be desirable if we are able to skip some postings while preserving the retrieval accuracy.

The key to the above two problems consists in measuring the importance of a local feature: query features with low importance are discarded, and indexed postings with low importance are not visited. Overall, this paper makes three major contributions.

Manuscript received September 12, 2014; revised December 30, 2014 and February 22, 2015; accepted February 23, 2015. This work was supported in part by the National High Technology Research and Development Program of China (863 Program) under Grant 2012AA011004, in part by the National Science and Technology Support Program under Grant 2013BAK02B04, and in part by the National Science Foundation of China (NSFC) under Grant 61429201. The work of Q. Tian was supported in part by ARO Grant W911NF-12-1-0057 and in part by the Faculty Research Award from the NEC Laboratories of America. The associate editor coordinating the review of this manuscript and approving it for publication was Dr. Cees Snoek.

L. Zheng, S. Wang, and Z. Liu are with the Department of Electronic Engineering, Tsinghua University, Beijing 100084, China (e-mail: liangzheng06@gmail.com; wgsj@tsinghua.edu.cn; zqiongliu@gmail.com).

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). 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/TMM.2015.2408563
An impact score is introduced. It is computed for each local feature, which takes into account various properties of a feature, i.e., TF-IDF, saliency score, scale, as well as its ambiguity;

- Based on the impact score, the query pruning technique is proposed. Given a query image, the query features are sorted according to their impact scores, with those of low impact scores discarded;
- A complementary strategy, i.e., early termination, is introduced. Based on the quantized impact score, a novel inverted index, called Q-Index is organized. Early termination works in a way that the most promising postings are accessed first, which greatly improves efficiency with accuracy preserved.

The pipeline of the proposed framework is illustrated in Fig. 1. Extensive experiments on two benchmark datasets show that, our method is capable of achieving over 10 times speedup over the baseline settings, while competitive accuracy is still preserved.

The rest of the paper is organized as follows. Firstly, Section II provides a brief review of the related works. Then, the impact score is discussed in Section III, and we introduce the fast image retrieval framework in Section IV. Experiments are presented in Section VI. The datasets and evaluation protocol is presented in Section V. Section VII concludes the paper.

II. RELATED WORK

The visual word is a core component in the BoW model. Typically, a high dimensional feature is quantized to one or a few [4], [7] nearest neighbors in the codebook, represented by a word ID. This process is accompanied by a large information loss, leading to visual word ambiguity [5]. To correct quantization artifacts, hard quantization can be replaced with schemes such as multiple assignment [8], sparse coding [9], etc. To further enhance the discriminative ability of visual words, for example, Mikulik et al. [10] propose to train a large codebook by unsupervised learning and tree balancing. Niu et al. [11] propose a personalized codebook adaptation method to create database-friendly visual words. Meanwhile, SIFT feature can be fused with other complementary features [12]–[14] to filter out false matches. Another popular choice is to model the spatial constraints among local features [15]–[18] or embed descriptor signatures [8], [19] into the inverted index.

The nature of visual word determines its lack of semantic meaning, compared with words in the text. Given a query image, we do not even know which features are more important and given special attention. Moreover, a recent trend is to use more local features obtained by adjusting the threshold of feature detectors [19], or combining feature learning [20]. A few investigations have been conducted on the image query pruning problem. Apart from the classic PCA approach, Swets et al. [21] propose to use discriminative eigenfeatures for an automatic feature selection scheme using a training set, which is not available in a prior-free retrieval system. Another classic criteria involves the TF-IDF weights [3]: features with very low IDF (more common in the database) can be discarded, which is also known as “stop word”. This method is widely used in text retrieval community [22], [23], where TF-IDF can be the primary evidences when estimating word importance. Cai et al. [24] propose to discards visual words that are distant from their cluster centers, while Foo et al. [25] use SIFT keypoints with large contrast values. In [26], Zhang et al. reduce the number of salient points using segmentation as a filter. Gao et al. [27] design an interactive scheme to select the most informative query views, and Banerjee et al. evaluate the importance of local features using a fuzzy entropy based measure. More recent works have included other cues, e.g., saliency value, for the estimation of visual word importance [28], [29] or shape prior knowledge [30]. Our work departs from previous ones on feature selection in that we use multiple cues of a keypoint and provide analysis on how they work together, thus providing a more comprehensive estimation of feature impact score.

Typically, an inverted index is constructed and modified for efficiency. Tao et al. [31] design a point-indexed representation for memory efficiency. Methods such as the multi-index [13], [32] and joint-index [33] make use of multiple codebooks to improve accuracy and efficiency simultaneously. But in many cases, the attempt to improve accuracy is accompanied by efficiency compromise. For example, the geometric visual phrases [15] enable strict spatial constraints, at the cost of more storage of offset table and coordinate computations. The co-indexing approach [34] expands the inverted index to improve recall, somewhat similar to the soft quantization [7]. In text retrieval,
III. IMPACT SCORE

A. Contributing Factors

In this work, an impact score is designed so as to reflect the importance of a feature. We call feature and visual word interchangeably in this paper. The impact score is formulated as the product of several factors associated with a feature, which are itemized below:

1) TF-IDF A classic way to measure the discriminative ability of a feature is to use the TF-IDF criteria. The TF factor characterizes feature’s occurrence in an image, informative about textures such as repetitive structures. One recurrent topic about TF is the burstiness [40] problem. Jégou et al. [40] propose to down-weight the TF score by a \( \sqrt{t_f} \) operator. This method is effective when applied to the images in the database. But for the query image, we find that the original TF is more informative. Therefore, for TF, the query image uses the \( t_f \) representation, while the database image uses the \( \sqrt{t_f} \) counterpart.

On the other hand, IDF determines the contribution of a given feature in terms of its occurrence across the whole database. The presence of a less common feature serves as a better discriminator than that of a more common one. The IDF score for a feature \( t \) is defined as,

\[
idf(t) = \log \frac{N}{n_t}
\]

where \( N \) denotes the total number of images in the collection, and \( n_t \) encodes the number of images where \( t \) occurs. Typically, IDF serves as a classic method for feature selection: visual words with large IDF are considered valuable, and vice versa. Based on it, the concept of “stop word” is proposed, which refers to those visual words with lowest IDF values, similar to the common words such as “a”, “the” in the text. The stop words sometimes are discarded to improve performance.

2) Scale The scale of a local feature can be used to filter out false positive matches by preserving those matches with similar scales. In our case, SIFT keypoints generated at the larger scales are given higher scores than those at the smaller scales. The intuition is that a larger scale is in effect able to filter out noisy keypoints at the smaller scales. Our strategy improves retrieval performance by giving more weight to the least-noisy scales.

3) Saliency The usage of saliency in image retrieval is typically motivated from the following observation. Usually regions that interest the user are not in the nonsalient regions. For the retrieved images, one would expect that at least some parts of the image represent the user’s interested regions. Based on these insights, previous works [28], [29] incorporate saliency information into either feature detector, vocabulary generation, or region representation to improve matching precision. In this work, however, we explicitly use saliency map generated by GBVS algorithm [41] for feature selection.

4) Ambiguity The visual word is subject to the problem of ambiguity, which is defined to consist of visual word plausibility and uncertainty [5]. The basic cause of ambiguity is that features may be far from the corresponding cluster centers. One way to address this problem involves dropping features if their distance to the centers are larger than a threshold [24]. In [7], Philbin et al. propose to use the distance \( d_t \) between a feature and its visual word. For each feature \( t \), its weight is calculated as

\[
w_a(t) = \exp \left( -\frac{d_t^2}{\sigma^2} \right)
\]

where \( \sigma \) is a weighting parameter. In our experiments, we set \( \sigma \) to 0.2. The more distant a feature is to its cluster center, the more ambiguous it will be, and the less important it is. Our work departs from [7], [24] in that, instead of using a single factor or a single weighting strategy, we combine multiple factors and multiple weighting strategies into an impact score. By doing this, comprehensive feature selection effect is achieved.

B. Formulation

Given a local feature \( t \), its TF, IDF (Eq. (1)), scale, saliency, and ambiguity (Eq. (2)) are denoted as \( t_f \), \( \idf \), \( s_e \), \( s_a \), and \( w_a \), respectively. Then, for a query feature, its impact score is defined as

\[
I_q(t) = t_f \cdot \idf \cdot s_e^\beta \cdot s_a^\gamma \cdot w_a^\alpha
\]

where \( \alpha \), \( \beta \), and \( \gamma \) are exponents for \( s_e \), \( s_a \), and \( w_a \), respectively. Here, we set the exponents of \( t_f \) and \( \idf \) to be 1. On the counterpart, for a database image indexed in the inverted index, the impact score of each indexed feature is denoted as

\[
I_d(t) = \sqrt{t_f} \cdot \idf \cdot s_e^\beta \cdot s_a^\gamma \cdot w_a^\alpha
\]

where the square operator is implied by the burstiness handling [40]. The intuition behind is summarized as follows:

- a larger TF encodes more informative description of image structure;
- a larger IDF means a feature is less common, thus more discriminative;
- a larger feature scale covers more area of an image, provides a more discriminative description, and is less affected by noise;
- a larger saliency value of a feature denotes that the feature is located in a more salient region, and is more interesting to user;
- when a feature is close to the cluster center, it has a large \( w_a \), and is less ambiguous.

Note that, for simple feature selection schemes, the impact score typically contains one factor, such as IDF [3], ambiguity [24], keypoint contrast value [25], etc. In comparison, our
method integrates multiple contributing factors in the impact score, by which a more informed and comprehensive selection effect can be achieved (a comparison with single factors is presented in Section VI-C).

C. Quantization

In the classic word-level inverted index, each posting consists of its image ID and some metadata such as the binary signature [8]. In the classic document-level index, each posting stores its TF score as well as image ID. Under both circumstances, the metadata can be stored as small integers amenable to compression. However, the impact scores defined in Eq. (4) are floating-point values. For a dynamic retrieval system, where indexed features are being added constantly, we have to store the impact scores of all indexed features, so that an input indexed feature can be sorted in the inverted list (see Section IV-B2). In this scenario, the floating-point impact scores are very memory-consuming.

To resolve this problem, the impact scores are quantized to an integer ranging from 1 to k. Intuitively, we would expect a small number of terms to receive high weight and a relatively large number of terms to receive low weight, somehow like the IDF scheme. In other words, “ordinary” is shared by many, and “outstanding” is less frequent. Based on this idea, we follow the lead by Anh et al. [36], which also embodies the “many ordinary” principle.

Specifically, a geometric sequence is built to achieve the above idea. Let i be the quantized impact score between 1 and k. Parameter n_i is the number of value i to be assigned in an image. Then, the geometric sequence is featured by \( n_i = R \cdot n_{i+1} \), where \( R \geq 1 \) is the common ratio. Further, we set the initial value \( n_k = R - 1 \). For an image \( d \) with \( u_d \) visual words, the solution to the two constraints is \( R = (u_d + 1)^{\frac{1}{k}} \). When \( R \) is solved, \( n_1, n_2, \ldots, n_k \) can be determined by the rounded integers. In typical cases, \( R \) is between 1 and 3, so that an image contains zero, one, or two visual words receiving an impact score of k. Table II presents an example of how to compute \( R \) and \( n_i \).

<table>
<thead>
<tr>
<th>Symbol</th>
<th>Annotation</th>
</tr>
</thead>
<tbody>
<tr>
<td>t</td>
<td>a detected keypoint, or equivalently, a feature</td>
</tr>
<tr>
<td>tf</td>
<td>term frequency of the corresponding visual word</td>
</tr>
<tr>
<td>idf</td>
<td>inverse document frequency of the corresponding visual word</td>
</tr>
<tr>
<td>se</td>
<td>the scale of a keypoint</td>
</tr>
<tr>
<td>sa</td>
<td>the saliency value of a keypoint</td>
</tr>
<tr>
<td>ua</td>
<td>the ambiguity score of a keypoint</td>
</tr>
<tr>
<td>a</td>
<td>exponent of se</td>
</tr>
<tr>
<td>b</td>
<td>exponent of sa</td>
</tr>
<tr>
<td>c</td>
<td>exponent of ua</td>
</tr>
<tr>
<td>( I_q(t) )</td>
<td>impact score of keypoint t in a query image</td>
</tr>
<tr>
<td>( I_d(t) )</td>
<td>impact score of keypoint t in a database image</td>
</tr>
</tbody>
</table>

Discussions. In the proposed quantization scheme, we adjust the impact scores so that, for all images in the database, the largest (and the smallest) impact scores in each image are roughly the same. In this way, all the images would possess some “important” features, so the retrieval procedure pays roughly equal prior attention to all database images.

Intuitively, another strategy is to quantize the impact score according to global statistics, i.e., the high quantized impact scores are assigned to large values, and vice versa. Nevertheless, for example, consider two images having the same visual word histogram weighted by TF-IDF. One image has many “important” words, while the other has many “ordinary” words. If we quantize the impact score globally, then the first image would obtain a higher rank, and the second image would be even wiped out by query pruning (or feature selection). However, if we stick to the quantization method used in this paper, the two images would both be very competitive, which in turn improves the recall of image retrieval. In literature, this is supported by [36] in text retrieval.

Under current quantization method, for images with a lot of high-impact features, since they tend to be easier to be retrieved already (for similar word histogram with the query), the pruning of some of the high-impact features would be less influential.

On the other hand, for images with a lot of ordinary features, if these features are just discarded, the retrieval results would be affected in a large extent.

By quantizing impact scores within each image (locally), we actually preserve the competitiveness of all database images, and recall is not compromised. In the future study, we will investigate more into this problem, and explore the impact of global quantization scheme in image retrieval.

IV. PROPOSED FRAMEWORK FOR FAST IMAGE RETRIEVAL

After introducing the method for computing feature impact score, in this section, we provide a detailed description on how to perform fast image retrieval. First, query pruning and early termination are discussed in Section IV-A and Section IV-B, respectively. Then, we present the similarity function in Section IV-C. Third, computational complexity analysis is performed in Section IV-D.

A. Query Pruning

With Eq. (3), the query pruning scheme is straightforward. Given a query image and its pool of local features, we calculate the impact score for each feature. Then, we can either employ a threshold for feature selection, or sort the features according to the impact score, after which a percentage is set to filter out those with the lowest impact scores. Both strategies are equivalent, and this paper employs the latter. After the query features with
high impact scores are selected, the image retrieval process can be carried out in the same way as the baseline approach.

Fig. 2 presents an example of query pruning on Oxford5k dataset [4]. In this example, AP (Average Precision) is employed to evaluate retrieval accuracy, which denotes the area under the precision-recall curve. For the classic method (left), the query has 1,990 visual words. After query pruning (right), 67.2% visual words are discarded. The remaining 653 query visual words yield an AP of 49.12% (see Section V-B for its meaning), which is even 1.07% higher than the baseline.

**B. Early Termination**

**Q-Index:** The baseline mechanism for answering a query uses a pre-computed inverted index. Each visual word in the codebook has a corresponding entry in the index, and is associated with a list of postings \(<d_i, f_d>\) recording its image ID and other metadata. In a document sorted index, postings are sorted according to their image IDs. Thus it is problematic for precise control of early termination. In the attempt to solve the problem of early termination, we introduce the Quantized Index (Q-Index) organization. The structure is illustrated in Fig. 3.

The Q-Index belongs to the impact sorted index, where postings are sorted according to their impact scores. In this inverted index category, it is feasible to have access to the important postings first, and then to the less important ones. With the quantized impact score introduced in Section III-C, Q-Index is constructed through three steps:

- Sort the inverted lists in decreasing order of their IDF scores;
- Each inverted list is partitioned into \(k\) impact regions, each corresponding to a quantized impact score;
- Given a feature \(t\), its image ID and metadata will be indexed in the corresponding impact region of inverted list \(t\).

The benefit of Q-Index is two fold. First, it enables prior access to those postings with large impact scores. Second, it hardly introduces any system burden, thus being memory efficient.

**Early Termination Method:** The algorithm for early termination is described as follows. In the offline stage, the Q-Index is constructed. In the online stage, on querying the Q-Index, postings are visited by decreasing impact score. When a certain percentage of postings are processed, the traverse can be stopped. During traversing, the similarity between feature pairs is the same with the baseline, i.e., weighted by TF-IDF scheme.

On the inverted list level, what is the order for traversing? Here, two alternatives can be used. The first one is serial: the inverted lists are traversed one by one. One inverted list is fully traversed before the next list is opened. The second method involves parallel operation: open all inverted lists simultaneously, and then the top-ranked postings are visited first.

On the posting level, another question is: what if postings have the same quantized impact score? Which should be indexed preferentially? For a large-scale and updating system, we use TF to perform this task. To this end, we propose the following two secondary sorting criteria:

- \((impact, IDF)\): within each inverted list, postings with the same impact score are sorted by decreasing order of TF. Higher TF denotes larger impact, as suggested in Eq. (4). Ties are further broken by increasing order of image IDs;
- \((impact, imgID)\): postings with the same impact score are sorted simply by increasing order of image IDs.

Since TF is also a discrete variable, when it is used as the key for secondary sorting, each impact region in the Q-Index is further partitioned into subregions, corresponding to the TF values. In this scenario, we do not store TF explicitly in Q-Index, maintaining unchanged memory cost.

As illustrated in Fig. 1, the inverted lists are partially traversed. Since the most important postings are visited, the retrieval efficiency can be greatly promoted while the accuracy remains stable.

**C. Similarity Function**

In this section, we define two similarity functions used in the baseline and the proposed framework, respectively.

For the baseline method, given a feature \(t\) which belongs to both query \(q\) and a database image \(d\), the matching score can be written as

\[
\phi(t) = 1 \cdot 1 \cdot \text{idf}(t)^2
\]

where \(\text{idf}(\cdot)\) is the Inverse Document Frequency of \(t\). In Eq. (5), the two “1”s represent the Term Frequency (TF) of feature \(t\). Note that here we employ a word-level inverted index [8], so that it is the feature (keypoint) rather than a given visual word that is being matched. So in this case, each feature has a TF of 1. Then, the similarity score \(S_{q,d}\) between images \(q\) and \(d\) is defined as

\[
S_{q,d} = \frac{\sum_{t \in q} \phi(t)}{\ell_2^q \cdot \ell_2^d}
\]

where \(\ell_2^q\) and \(\ell_2^d\) are L2 norms of \(q\) and \(d\), respectively.

For our method, note that, the impact score is used only for sorting the query and indexed features. On the query side, a
feature is counted iff its impact score is higher than a threshold. On the inverted index side, with the quantized impact score, a feature is traversed iff it is located in an impact region which is also higher than a threshold. Therefore, the matching score \( \hat{\phi}(t) \) is re-written as

\[
\hat{\phi}(t) = 1 \cdot 1 \cdot idf(t)^2 \cdot \delta_q(t) \cdot \delta_d(t)
\]

where \( \delta_q(\cdot) \) and \( \delta_d(\cdot) \) indicate whether a query or indexed feature should be counted during query pruning or early termination, respectively. In this scenario, i.e., when query pruning and early termination are employed simultaneously, the similarity function can be defined as

\[
\hat{S}_{q,d} = \sum_{t \in \mathcal{Q} \cap \mathcal{D}} \hat{\phi}(t) / \ell_q^2 \cdot \ell_d^2
\]

In this paper, the baseline and the proposed methods use Eq. (6) and Eq. (8) as similarity functions, respectively.

**D. Computational Complexity Analysis**

*Computational Complexity:* After introducing the algorithms for query pruning and early termination, we present an analysis on the computational complexity of the proposed method. We assume that an image database has \( N \) images, and that each image has \( \bar{n} \) local features on average. Let \( K \) denote the size of the codebook. Then, given a query image with \( n_q \) local features, the computational complexity of the baseline can be approximated as

\[
C_b = \mathcal{O}(N_b) \approx \mathcal{O}\left(\frac{N \cdot \bar{n} \cdot f_q \cdot n_q \cdot f_p}{K}\right)
\]

where \( N_f \) denotes the total number of postings to be traversed under fast image retrieval framework, \( f_e \) encodes on average the fraction of postings to be traversed when early termination is employed, and \( f_p \) is the fraction of features in the query that are preserved after query pruning. In Eq. (10), the product of \( f_e \) and \( f_p \) yields the Percentage of Processed Postings (POPP), which can be calculated as

\[
\text{POPP} = \frac{N_f}{N_b} \times 100\%
\]

Comparing Eq. (9) and Eq. (10), it is clear that the computational complexity of the proposed method has a linear relationship (by POPP) with the baseline. The smaller POPP is, the larger the speedup will be. Nevertheless, what we must keep in mind is the trade-off between speed and accuracy: as following experiments will show, a proper POPP helps produce significant speed up while retaining the retrieval accuracy.

*Discussions:* In the standard image retrieval framework using BoW model with large codebooks and the inverted index, a number of prior arts which aim at improving retrieval accuracy suffer from increased computational complexity [15], [42]–[44]. For these methods, the visual matching process is inevitably accompanied by extra computation such as coordinate subtraction [15], [16], weight estimation [44], [45], etc. For example, for the method of geometry-preserving visual phrases [15], apart from additional memory consumption for offset tables, during inverted list traversal, one has to perform additional plus-minus operation for two times. Ideally, it is
expected that the traversal complexity in [15] is three times that of the baseline.

Moreover, a recent trend lies in the compact image representations such as the Fisher Vector (FV) [46] and the Vector of Locally Aggregated Descriptors (VLAD) [47]. This line of methods, although also relying on local features and the BoW model, are distinguished from others by using small codebooks and by representing an image with a global vector. Then, the dimension of the vectors is reduced with PCA and subsequently quantized using Product Quantization [48] to reduce memory consumption. It is shown [49], [47] that VLAD can achieve one or two orders of magnitude speedup over the BOW [50] using ADC or IVFADC indexing schemes, respectively. Nevertheless, one recurrent issue about VLAD consists in the relatively low retrieval accuracy due to the “aggregation” process.

In addition, for a simple feature selection scheme (mentioned in Section III-A), a threshold is employed to filter out features with low impact scores. The selected features are fixed once a threshold is determined. After feature selection, the retrieval process goes the same way with the baseline system. In this scenario, our method stores the quantized impact score in the Q-Index, which enables dynamic and flexible feature selection.

V. DATASETS AND EVALUATION PROTOCOL

A. Datasets

We evaluate our approach on two benchmark datasets, i.e., the Oxford5k [4] and Paris6k [51] datasets. Along with them, a distractor dataset with 1 million images is used.

Oxford5k This dataset is composed of 5063 images collected from Flickr for 11 groups of famous Oxford architectures. Each group has 5 query images taken under distinct illuminations and angles.

Paris6k This dataset is generated in couple with Oxford5k. It contains 6412 images which is annotated with respect to 55 queries of 11 groups.

MirFlickr1M This dataset [52] is used to test the large scale settings. It contains 1 million images collected from Flickr during the data collection period, for each date, a number of uploaded images with highest interestingness were requested. A small subset (25000) of this dataset is annotated using general topics like “sky”, “water”, “transport”, etc. Considering the randomness of the data collection process, the chance is remote that this dataset contains relevant images to queries in the test datasets. Therefore, images in the MirFlickr1M dataset serve as distractors, i.e., they are considered as negative samples and used to scale up the test datasets. A summarization of the three datasets is presented in Table III.

B. Evaluation Protocol

Mean Average Precision (mAP) To measure the retrieval accuracy, Average Precision (AP) is used for each query, which is calculated as the area under the precision-recall curve. Then, APs of all queries are averaged to obtain the Mean Average Precision (mAP).

VI. EXPERIMENTS

A. Baseline Setup

In this paper, we consider the Bag-of-Words (BoW) baseline proposed by Philbin et al. [4], along with some recent techniques to improve baseline performance. The procedure of the baseline is described as follows.\(^1\)

- Following [20], we use the DoG-Affine (DoGAff) detector, and the scaling factor is set to 12.5. For each image, keypoints are detected using DoG detector, wrapped by affine transformation, and then described by a 128-D SIFT descriptor. We use the rootSIFT variant [53], which is easy to implement and yields higher performance under Euclidean distance.
- The codebook is trained on Oxford5k using Approximate K-Means (AKM) algorithm [4]. Both clustering and quantization schemes employ the FLANN library [54] for Approximate Nearest Neighbor (ANN) search. The FLANN precision of ANN search is set to 0.60 and 0.85, respectively. We use hard quantization for efficiency, and the codebook size is set to 1M.
- A standard inverted index is built in accordance to the codebook, and store all the local features in the image collection.
- We use the \(L_p\)-norm IDF [6] instead of classic IDF, and set \(p\) to 3. We use L2 normalization to normalize the scores.

In Table IV, we show the baseline results (mAP in percentage) on Oxford5k, Oxford5k + 1M, Paris6k, and Paris6k + 1M datasets. With rootSIFT and \(L_p\)-norm IDF, we achieve mAP of 74.27% and 64.74% on the two datasets, respectively. Our baseline outperforms some state-of-the-arts by itself: Zheng et al. produce mAP of 69.6% and 56.2% on the two datasets, respectively; Zhang et al. [15] report mAP of 69.6% on Oxford5k dataset; Zhang et al. [34] obtain mAP of 58.32% on Oxford5k dataset. In comparison with the VLAD representation, [55] reports mAP of 28.9% and 27.0% using the original VLAD and their improved version after Product Quantization (PQ), respectively; [56] reports mAP of 44.8% using 128-D vector without PQ. When comparing results on the 1M dataset, since previous works may use different large-scale datasets, the results are not directly comparable. But generally

Fig. 4. Parameter selection of $\alpha$, $\beta$, and $\gamma$. We vary the exponents of the three variances, i.e., scale, saliency, and ambiguity, respectively. The mAP (%) results are drawn for (a) Oxford5k, POPP = 10%, (b) Paris6k, POPP = 10%, (c) Oxford, POPP = 20%, and (d) Paris, POPP = 20%, respectively. The codebook of size 1M is trained on Oxford5k.

speaking, our results on the 1M datasets are competitive. We also implement the re-ranking step using query expansion [57]. In the experiments, we focus on improving efficiency while preserving initial result accuracy. Most re-ranking methods naturally require a good initial result [4], [43], [58].

B. Parameter Selection

Exponent parameters $\alpha$, $\beta$, and $\gamma$ in Eq. (3) and Eq. (4), $\alpha$, $\beta$, and $\gamma$ are the exponents of feature scale, saliency and ambiguity, respectively. The three parameters are tuned on both Oxford5k and Paris6k datasets: we use 30% query pruning and set the percentage of processed postings (POPP) to be 10% and 20%, respectively. Note that, the impact score is not quantized for early termination here. The parameter tuning results are shown in Fig. 4. We can see that, under 10% or 20% POPP, the curves first rise and then drops after a peak. The value of $\alpha$ is to be determined in the next part. From these results, we set $\alpha = 0.6$, $\beta = 0.8$, and $\gamma = 1.4$ in the following experiments.

Quantization parameter $k$ A critical issue includes determining the optimal value of $k$. To this end, various $k$ values are tested on the benchmark datasets. For the controlled variable, we do not adopt query pruning, which uses unquantized impact score. We set POPP to 10% and 20%, respectively. When $k$ varies, the mAP results are demonstrated in Fig. 5. Across the different datasets, when $k$ takes a small value (2 or 4), we obtain low results because indexed features are not well-sorted according to their impact scores. But as $k$ increases, the accuracy remains somewhat stable. It can be speculated that when $k$ is very large, the performance will converge to the baseline. From these results, we set $k = 8$ in our experiments.

Fig. 5. Parameter selection of $k$. When $k = 2, 4, 6, 8, 10, 12$, mAP results are presented with respect to POPP = 10% and 20% on (a) Oxford5k and (b) Paris6k, respectively. We choose $k = 8$ in the following experiments.

C. Evaluation

Query pruning We evaluate query pruning in Fig. 6. On Oxford5k and Paris6k datasets, we plot mAP against the percentage of pruned query features according to Eq. (3). The dashed black line represents the baseline result for reference. As we can see, among the five factors which are used alone as the impact score, TF has the worst performance when the pruning percentage is small. In fact, since the TF score is typically 1 or 2 for each feature, it does not have much discriminative ability being an impact score. When using IDF alone, mAP improves marginally at 5% pruning point, but drops quickly afterwards. This is consistent with previous findings that using a small number of stop words helps improve accuracy. Similar situation can be observed for scale, but the dropping rate seems to be slower than IDF on Paris6k. This indicates that local features with larger scale possess more discriminative ability. For the saliency and ambiguity factors, there is evident improvement over the baseline. This suggests that both factors are good discriminators among query features.

Moreover, it is obvious from Fig. 6 that the proposed impact score consistently outperforms the others. Specifically, on Oxford5k dataset, we are capable of reproducing the baseline when 40% query visual words are pruned. In this scenario, compared with the baseline, we achieve the best mAP of 62.33% (2.98%) on the drop of 10% visual words. Similar observation can also be found in Fig. 6(b) on Paris6k dataset: the performance is maintained after 30% query visual words are discarded, and the best mAP (53.01%) is obtained at 10% pruning point. These results thus demonstrate the effectiveness of the proposed impact score in query pruning task.

Early termination The results of early termination are presented in Fig. 7. We plot the mAP results against the percentage of processed postings (POPP). Four configurations for both baseline and our method are shown. Overall speaking, the proposed impact-based approach dramatically outperforms the baselines. Specifically, we are able to achieve almost the same performance by only visiting less than 10% of the postings. In this sense, the traversing time is decreased by more than 10 times.

The second finding associates with the secondary sorting strategy using either TF or image ID. The results suggest that using TF as the secondary key yields a performance no worse than using image ID. This is expected because each inverted
Fig. 6. Image retrieval accuracy (mAP in percentage) versus percentage of pruned visual words (%). Results on Oxford5k + 1M and Paris6k + 1M are shown. The dashed line denotes the baseline. The blue curve represents the pruning strategy defined in Eq. (3). Five alternative strategies are also presented, i.e., using only IDF, TF, scale, saliency, and ambiguity as the impact score.

Fig. 7. Early termination on (a) Oxford5k + 1M and (b) Paris6k + 1M datasets. mAP (%) is plotted against the percentage of processed postings (in log scale). Eight methods are compared. The baseline uses Eq. (6) as similarity function, while ours uses Eq. (8). Serial and parallel denote the two kinds of operations described in Section IV-B2. For sorting options, the baseline uses only TF or image ID (imgID) as keys, while our method employ impact score (imp) as the primary key, and TF or image ID as the secondary key.

list is better sorted using a combination like (impact, TF). We notice that under the “serial” option, TF and image ID yield very similar curves. The reason is that, at each percentage of processed postings, only one inverted list is partially traversed, leaving others either fully traversed or not traversed at all. Therefore, the secondary sorting option actually makes little difference under serial operation.

Thirdly, in large-scale settings, we observe that using the parallel option generates higher curves. It means that the inverted lists are simultaneously open, and the top postings in these lists are visited first. Another advantage of parallel processing is that it can be readily parallelized. Based on these results, the early termination mode we recommend is using Eq. (8) as similarity function, (impact, TF)-sorted inverted lists, and parallel processing. Potentially, the recommended Q-Index can be very useful in industrial applications such as web image retrieval, in which the powerful infrastructure can be made full use of: given a query image with thousands of visual words, simultaneously opening the inverted lists and stopping early can greatly reduce response time. For other state-of-the-art methods based on BoW and inverted list, our method can be readily applied to improve the retrieval speed. Nevertheless, one issue that must be paid attention to is the unstable performance of the current system. Since we do not have a stop criteria for pruning and early termination, the system should be dynamic in that user may interactively select a preferred setting to draw a balance between speed and accuracy.

Combination of query pruning and early termination To test the complementarity of the proposed two strategies, we present large-scale results on the two datasets as in Fig. 8. The point on the x-axis where POPP = 100% means that all the postings which should be visited if neither query pruning nor early termination is applied. When using pruning, the curves do not reach the 100% point. From these results, we obtain two major observations. First, as more query features are pruned, the curves lean upward when POPP is small. This indicates that query pruning improves accuracy more quickly in terms of early termination.

Second, on the far end, pruning 10%, 20%, and 30% features yields superior performance to both the baseline and the “no
pruning” case on Oxford5k + 1M dataset. Specifically, the best efficiency performance on Oxford5k + 1M dataset is obtained by “30% pruning + early termination”, i.e., $\text{mAP} = 59.63\%$, $\text{POPP} = 3.89\%$. Similarly, on Paris6k + 1M dataset, when using “20% pruning + early termination”, we obtain $\text{mAP} = 50.25\%$, $\text{POPP} = 7.85\%$. Note that, when combined, query pruning does not have a superior performance to when it is used alone as in Fig. 6. It is expected because too much query pruning will drop some (but not many) useful cues for early termination. Nevertheless, the complementarity is clear from the results.

**Memory cost** For memory usage, since the quantized impact score is employed, we do not have to store them in the inverted index. With the Q-Index, only some pointers recording the impact regions need to be kept in memory, the size of which can be neglected. Moreover, within each impact region, a finer partitioning for TF can be readily constructed. Therefore, for the (impact, tf)-sorted Q-Index, only image IDs need to be stored, along with some negligible pointers. Specifically, for each posting, a total of 4 bytes are allocated to store the image ID. For the Oxford5k+1M dataset as detailed in Table III, the memory consumption is 4.87 GB, which remains the same with the baseline method. In comparison, the VLAD representation [47] would consume 16 bytes for an image, and a total of 15.3M for the Oxford5k + 1M dataset. Although the memory usage of VLAD is two orders of magnitude below the BoW model, the cost of VLAD is a lower retrieval accuracy (see Section VI-A).

**Time cost** Using the above described method, the time-accuracy trade-off is summarized in Table V. As POPP(%) increases, Table V shows how the retrieval accuracy (mAP in percentage) as well as the traversing time (ms) vary. Note that when POPP = 100%, the baseline system is produced. Our experiments are performed using a server with 3.46 GHz CPU and 128 GB memory. The time for feature extraction and quantization is 0.70 s and 0.93 s on average on the 1M dataset. Note that, this paper focuses on speeding up the traversal process.

Fast feature extraction [59] and quantization [60] methods have been proposed in recent years. Table V shows that the time for traversing the inverted index is greatly reduced when query pruning and early termination are coupled. When mAP is required at 59.63%, the traversing time is 9.6 ms, a save of more than ten folds.

A brute force indexing strategy is to index the BoW histograms directly by FLANN [54]. However, such a method requires vast memory (more than 3700 GB for the 1M dataset), and one can imagine that the time-accuracy tradeoff would be extremely undesirable. On the other hand, In [49], the retrieval time in a 10 million dataset is 716 ms for ADC and 46 ms for IVFADC, respectively. Comparing with [49], our method is faster than ADC, and has slightly lower speed than IVFADC. Considering the high accuracy of the BoW model with large codebook and inverted index, this work has made good progress toward real-time application.

**VII. DISCUSSION AND CONCLUSION**

This paper focuses on the efficiency issue of BoW based image retrieval. First, an impact score is introduced, computed for each local feature, which takes into account various attributes of a feature, e.g., TF-IDF, scale, saliency as well as ambiguity. Then, based on the impact score, two complementary techniques, i.e., query pruning and early termination, are proposed. Query pruning deals with the query side, in which query features with low impact scores are discarded. On the
other hand, early termination works on the database features stored in the inverted index. The postings are sorted by decreasing order of impact score, so that those of high impacts are visited first. To save memory usage, impact scores are quantized and implicitly stored in the so-called Q-Index. Experiments are performed on Oxford5k and Paris6k datasets. When applied alone, the two techniques are effective on each side. We further show that the combination produces a significant improvement in retrieval efficiency, while preserving a high accuracy.

We emphasize the difference between this work and the simple feature selection methods on three aspects. First, this paper proposes a novel impact score which integrates multiple properties of a local feature, so our definition of feature impact is more comprehensive (see Section III-A). Second, through Q-Index, we propose two alternatives for inverted index traversal, i.e., serial (depth first search) and parallel (breadth first search) operations. In fact, parallel traversal is quite similar to feature selection on the database images, because it implicitly exerts a threshold on the indexed features. In experiments, we show that serial traversal is only slightly inferior to parallel traversal. Therefore, other than simple feature selection or parallel traversal, we actually find that serial traversal is also competitive for early termination. Third and most importantly, the proposed system is well extendable and flexible. The proposed system can be readily adopted when the database size keeps growing. Newly indexed local features can find their impact region in the Q-Index. In this sense, all the database features can be stored in the Q-Index. However, for feature selection methods, only those local features with impact scores above a threshold are stored. Once selected, the pool of indexed features are fixed and cannot be further pruned (it is expensive to store the raw impacts in the inverted index).

In contrast to the rigid feature selection method, the flexibility of our system is a great advantage. On one hand, users can find their preferred tradeoff between speed and accuracy interactively: On the other hand, for system maintenance, new algorithms, optimizations, or extensions can be added to the system without too much effort, because all the original local features are being stored without extra memory requirement. Compared with retrieval speed, memory usage is of less importance due to the extensive usage of PC Clusters.

In the future work, we will further investigate other properties of a local feature, in order to design more effective impact scores. Meanwhile, other early termination options such as the safe termination or top-k ranking, will be studied. This work focuses on Term-at-a-time scoring mode, and we will study the Document-at-at-time [61] scheme as well.

REFERENCES


Abstract—Efficiency is of great importance for image retrieval systems. For this pragmatic issue, this paper proposes a fast image retrieval framework to speed up the online retrieval process. To this end, an impact score for local features is proposed in the first place, which considers multiple properties of a local feature, including TF-IDF, scale, saliency, and ambiguity. Then, to decrease memory consumption, the impact score is quantized to an integer, which leads to a novel inverted index organization, called Q-Index. Importantly, based on the impact score, two closely complementary strategies are introduced: query pruning and early termination. On one hand, query pruning discards less important features in the query. On the other hand, early termination visits indexed features only with high impact scores, resulting in the partial traversing of the inverted index. Our approach is tested on two benchmark datasets populated with an additional 1 million images to account as negative examples. Compared with full traversal of the inverted index, we show that our system is capable of visiting less than 10% of the “should-visit” postings, thus achieving a significant speed-up in query time while providing competitive retrieval accuracy.

Index Terms—Early termination, image retrieval, impact score, query pruning.

I. INTRODUCTION

Recent years have witnessed dramatic accuracy improvement of image retrieval thanks to the introduction of invariant local features [1], which gives rise to the widely used Bag-of-Words (BoW) model [2]. Inspired from classic text retrieval approaches, the BoW model transforms an image into a histogram of visual words produced by feature quantization. Typically, visual words are defined by a codebook trained on a feature pool with various clustering algorithms [3], [4]. To promote retrieval efficiency in terms of both time and memory, an inverted index is employed.

Efficiency is a central issue in any retrieval system, which should return the relevant documents/images to the user in real time. The online image retrieval procedure can be divided into three steps: feature extraction, feature quantization, and inverted index traversing. For web-scale applications, as the number of indexed images is getting larger, the inverted index traversing is the only one of increasing time out of the three parts. Therefore, this paper aims at improving efficiency of the traversing part. During this process, for each feature in a query image, an inverted list of postings (or indexed features) is identified from the corresponding entry in the inverted index. The scores of these images are increased correspondingly. In essence, two steps have direct impacts on efficiency: a loop over all query features and a full traverse of all inverted lists associated with query.

On one hand, an image is typically described by several thousand visual words in the BoW model. Nevertheless, unlike words in the text, visual words in an image do not have a semantic meaning and are subject to the ambiguity issue [5]. Except for some statistical cues such as TF-IDF [2], [6], we are unlikely to tell whether these words are important for retrieval. A brute-force loop of all query features not only over-uses the less important ones, but also leads to a large amount of requests on the inverted index. In contrast, for a common query in text retrieval, only several query words are submitted. Image retrieval is a typical long query problem and effective query pruning strategy is needed for this task.

On the other hand, a major bottleneck of inverted index traversing is the length of the inverted list. Roughly, it grows linearly with the database size. For a common visual word, its inverted list could be as long as hundreds of MBs or even GBs in a web-scale network. An exhaustive traverse on the inverted index can be extravagantly expensive. This problem appears more severe if we consider a typical case where a query contains hundreds of visual words. It means that a non-trivial fraction in the image collection needs to be visited before the query is solved. For each query feature, a naive full traverse of the postings takes considerable time. As a consequence, it would be desirable if we are able to skip some postings while preserving the retrieval accuracy.

The key to the above two problems consists in measuring the importance of a local feature: query features with low importance are discarded, and indexed postings with low importance are not visited. Overall, this paper makes three major contributions.
• An impact score is introduced. It is computed for each local feature, which takes into account various properties of a feature, i.e., TF-IDF, saliency score, scale, as well as its ambiguity;
• Based on the impact score, the query pruning technique is proposed. Given a query image, the query features are sorted according to their impact scores, with those of low impact scores discarded;
• A complementary strategy, i.e., early termination, is introduced. Based on the quantized impact score, a novel inverted index, called Q-Index is organized. Early termination works in a way that the most promising postings are accessed first, which greatly improves efficiency with accuracy preserved.

The pipeline of the proposed framework is illustrated in Fig. 1. Extensive experiments on two benchmark datasets show that, our method is capable of achieving over 10 times speedup over the baseline settings, while competitive accuracy is still preserved.

The rest of the paper is organized as follows. Firstly, Section II provides a brief review of the related works. Then, the impact score is discussed in Section III, and we introduce the fast image retrieval framework in Section IV. Experiments are presented in Section VI. The datasets and evaluation protocol is presented in Section V. Section VII concludes the paper.

II. RELATED WORK

The visual word is a core component in the BoW model. Typically, a high dimensional feature is quantized to one or a few nearest neighbors in the codebook, represented by a word ID. This process is accompanied by a large information loss, leading to visual word ambiguity. To correct quantization artifacts, hard quantization can be replaced with schemes such as multiple assignment, sparse coding, etc. To further enhance the discriminative ability of visual words, for example, Mikulik et al. propose to train a large codebook by unsupervised learning and tree balancing. Niu et al. propose a personalized codebook adaptation method to create database-friendly visual words. Meanwhile, SIFT feature can be fused with other complementary features to filter out false matches. Another popular choice is to model the spatial constraints among local features or embed descriptor signatures into the inverted index.

The nature of visual word determines its lack of semantic meaning, compared with words in the text. Given a query image, we do not even know which features are more important and given special attention. Moreover, a recent trend is to use more local features obtained by adjusting the threshold of feature detectors, or combining feature learning. A few investigations have been conducted on the image query pruning problem. Apart from the classic PCA approach, Swets et al. propose to use discriminative eigenfeatures for an automatic feature selection scheme using a training set, which is not available in a prior-free retrieval system. Another classic criteria involves the TF-IDF weights: features with very low IDF (more common in the database) can be discarded, which is also known as ’stop word’. This method is widely used in text retrieval community, where TF-IDF can be the primary evidences when estimating word importance. Cai et al. propose to discards visual words that are distant from their cluster centers, while Foo et al. use SIFT keypoints with large contrast values. In [26], Zhang et al. reduce the number of salient points using segmentation as a filter. Gao et al. design an interactive scheme to select the most informative query views, and Banerjee et al. evaluate the importance of local features using a fuzzy entropy based measure. More recent works have included other cues, e.g., saliency value, for the estimation of visual word importance or shape prior knowledge. Our work departs from previous ones on feature selection in that we use multiple cues of a keypoint and provide analysis on how they work together, thus providing a more comprehensive estimation of feature impact score.

Typically, an inverted index is constructed and modified for efficiency. Tao et al. design a point-indexed representation for memory efficiency. Methods such as the multi-index, and joint-index make use of multiple codebooks to improve accuracy and efficiency simultaneously. But in many cases, the attempt to improve accuracy is accompanied by efficiency compromise. For example, the geometric visual phrases enable strict spatial constraints, at the cost of more storage of offset table and coordinate computations. The co-indexing approach expands the inverted index to improve recall, somewhat similar to the soft quantization. In text retrieval,
various dynamic pruning methods are proposed to “stop early” [35]–[37] or “skip” [38], [39]. Our work departs from the prior arts as follows. First, an impact score is defined, which considers various attributes of an image feature. Second, to the best of our knowledge, it is the first time that query pruning and early termination techniques are coupled in a very flexible Q-Index system in image retrieval. Our framework is shown to have great advantage in both memory and time efficiency.

III. IMPACT SCORE

A. Contributing Factors

In this work, an impact score is designed so as to reflect the importance of a feature. We call feature and visual word interchangeably in this paper. The impact score is formulated as the product of several factors associated with a feature, which are itemized below:

1) TF-IDF A classic way to measure the discriminative ability of a feature is to use the TF-IDF criteria. The TF factor characterizes feature’s occurrence in an image, informative about textures such as repetitive structures. One recurrent topic about TF is the burstiness [40] problem. Jégou et al. [40] propose to down-weight the TF score by a \( \sqrt{t_f} \) operator. This method is effective when applied to the images in the database. But for the query image, we find that the original TF is more informative. Therefore, for TF, the query image uses the \( tf \) representation, while the database image uses the \( \sqrt{t_f} \) counterpart.

On the other hand, IDF determines the contribution of a given feature in terms of its occurrence across the whole database. The presence of a less common feature serves as a better discriminator than that of a more common one. The IDF score for a feature \( t \) is defined as,

\[
idf(t) = \log \frac{N}{n_t}
\]

where \( N \) denotes the total number of images in the collection, and \( n_t \) encodes the number of images where \( t \) occurs. Typically, IDF serves as a classic method for feature selection: visual words with large IDF are considered valuable, and vice versa. Based on it, the concept of “stop word” is proposed, which refers to those visual words with lowest IDF values, similar to the common words such as “a”, “the” in the text. The stop words sometimes are discarded to improve performance.

2) Scale The scale of a local feature can be used to filter out false positive matches by preserving those matches with similar scales. In our case, SIFT keypoints generated at the larger scales are given higher scores than those at the smaller scales. The intuition is that a larger scale is in effect able to filter out noisy keypoints at the smaller scales. Our strategy improves retrieval performance by giving more weight to the least-noisy scales.

3) Saliency The usage of saliency in image retrieval is typically motivated from the following observation. Usually regions that interest the user are not in the nonsalient regions. For the retrieved images, one would expect that at least some parts of the image represent the user’s interested regions. Based on these insights, previous works [28], [29] incorporate saliency information into either feature detector, vocabulary generation, or region representation to improve matching precision. In this work, however, we explicitly use saliency map generated by GBVS algorithm [41] for feature selection.

4) Ambiguity The visual word is subject to the problem of ambiguity, which is defined to consist of visual word plausibility and uncertainty [5]. The basic cause of ambiguity is that features may be far from the corresponding cluster centers. One way to address this problem involves dropping features if their distance to the centers are larger than a threshold [24]. In [7], Philbin et al. propose to use the distance \( d_t \) between a feature and its visual word. For each feature \( t \), its weight is calculated as

\[
w_a(t) = \exp \left( -\frac{d_t^2}{\sigma^2} \right)
\]

where \( \sigma \) is a weighting parameter. In our experiments, we set \( \sigma \) to 0.2. The more distant a feature is to its cluster center, the more ambiguous it will be, and the less important it is. Our work departs from [7], [24] in that, instead of using a single factor or a single weighting strategy, we combine multiple factors and multiple weighting strategies into an impact score. By doing this, comprehensive feature selection effect is achieved.

B. Formulation

Given a local feature \( t \), its TF, IDF (Eq. (1)), scale, saliency, and ambiguity (Eq. (2)) are denoted as \( t_f \), \( idf \), \( s_e \), \( s_a \), and \( w_a \), respectively. Then, for a query feature, its impact score is defined as

\[
I_q(t) = t_f \cdot idf \cdot s_e^\alpha \cdot s_a^\beta \cdot w_a^\gamma
\]

where \( \alpha, \beta, \gamma \) are exponents for \( s_e, s_a, \) and \( w_a \), respectively. Here, we set the exponents of \( t_f \) and \( idf \) to be 1. On the counterpart, for a database image indexed in the inverted index, the impact score of each indexed feature is denoted as

\[
I_d(t) = \sqrt{t_f} \cdot idf \cdot s_e^\alpha \cdot s_a^\beta \cdot w_a^\gamma
\]

where the square operator is implied by the burstiness handling [40]. The intuition behind is summarized as follows:

- a larger TF encodes more informative description of image structure;
- a larger IDF means a feature is less common, thus more discriminative;
- a larger feature scale covers more area of an image, provides a more discriminative description, and is less affected by noise;
- a larger saliency value of a feature denotes that the feature is located in a more salient region, and is more interesting to user;
- when a feature is close to the cluster center, it has a large \( w_a \), and is less ambiguous.

Note that, for simple feature selection schemes, the impact score typically contains one factor, such as IDF [3], ambiguity [24], keypoint contrast value [25], etc. In comparison, our
method integrates multiple contributing factors in the impact score, by which a more informed and comprehensive selection effect can be achieved (a comparison with single factors is presented in Section VI-C).

C. Quantization

In the classic word-level inverted index, each posting consists of its image ID and some metadata such as the binary signature [8]. In the classic document-level index, each posting stores its TF score as well as image ID. Under both circumstances, the metadata can be stored as small integers amenable to compression. However, the impact scores defined in Eq. (4) are floating-point values. For a dynamic retrieval system, where indexed features are being added constantly, we have to store the impact scores of all indexed features, so that an input indexed feature can be sorted in the inverted list (see Section IV-B2). In this scenario, the floating-point impact scores are very memory-consuming.

To resolve this problem, the impact scores are quantized to an integer ranging from 1 to \( k \). Intuitively, we would expect a small number of terms to receive high weight and a relatively large number of terms to receive low weight, somehow like the IDF scheme. In other words, “ordinary” is shared by many, and “outstanding” is less frequent. Based on this idea, we follow the lead by Anh et al. [36], which also embodies the “many ordinary” principle.

Specifically, a geometric sequence is built to achieve the above idea. Let \( i \) be the quantized impact score between 1 and \( k \). Parameter \( n_i \) is the number of value \( i \) to be assigned in an image. Then, the geometric sequence is featured by \( n_i = R \cdot n_{i+1} \), where \( R \geq 1 \) is the common ratio. Further, we set the initial value \( n_k = R - 1 \). For an image \( d \) with \( u_d \) visual words, the solution to the two constraints is \( R = (u_d + 1) \frac{1}{2} \). When \( R \) is solved, \( n_1, n_2, \ldots, n_k \) can be determined by the rounded integers. In typical cases, \( R \) is between 1 and 3, so that an image contains zero, one, or two visual words receiving an impact score of \( k \). Table II presents an example of how to compute \( R \) and \( n_i \). After introducing the method for computing feature impact score, in this section, we provide a detailed description on how to perform fast image retrieval. First, query pruning and early termination are discussed in Section IV-A and Section IV-B, respectively. Then, we present the similarity function in Section IV-C. Third, computational complexity analysis is performed in Section IV-D.

### IV. PROPOSED FRAMEWORK FOR FAST IMAGE RETRIEVAL

#### A. Query Pruning

With Eq. (3), the query pruning scheme is straightforward. Given a query image and its pool of local features, we calculate the impact score for each feature. Then, we can either employ a threshold for feature selection, or sort the features according to the impact score, after which a percentage is set to filter out those with the lowest impact scores. Both strategies are equivalent, and this paper employs the latter. After the query features with decreasing order of their impact scores, and then are assigned a quantized impact according to \( n_i \).

**Discussions.** In the proposed quantization scheme, we adjust the impact scores so that, for all images in the database, the largest (and the smallest) impact scores in each image are roughly the same. In this way, all the images would possess some “important” features, so the retrieval procedure pays roughly equal prior attention to all database images.

Intuitively, another strategy is to quantize the impact score according to global statistics, i.e., high quantized impact scores are assigned to large values, and vice versa. Nevertheless, for example, consider two images having the same visual word histogram weighted by TF-IDF. One image has many “high impact” words, while the other has many “ordinary” words. If we quantize the impact score globally, then the first image would obtain a higher rank, and the second image would be even wiped out by query pruning (or feature selection). However, if we stick to the quantization method used in this paper, the two images would both be very competitive, which in turn improves the recall of image retrieval. In literature, this is supported by [36] in text retrieval.

Under current quantization method, for images with a lot of high-impact features, since they tend to be easier to be retrieved already (for similar word histogram with the query), the pruning of some of the high-impact features would be less influential. On the other hand, for images with a lot of ordinary features, if these features are just discarded, the retrieval results would be affected in a large extent.

By quantizing impact scores within each image (locally), we actually preserve the competitiveness of all database images, and recall is not compromised. In the future study, we will investigate more into this problem, and explore the impact of global quantization scheme in image retrieval.

### TABLE I

[Table I not cited within text. Please cite all tables in numerical order.]

<table>
<thead>
<tr>
<th>Symbol</th>
<th>Annotation</th>
</tr>
</thead>
<tbody>
<tr>
<td>( t )</td>
<td>a detected keypoint, or equivalently, a feature</td>
</tr>
<tr>
<td>( tf )</td>
<td>term frequency of the corresponding visual word</td>
</tr>
<tr>
<td>( idf )</td>
<td>inverse document frequency of the corresponding visual word</td>
</tr>
<tr>
<td>( se )</td>
<td>the scale of a keypoint</td>
</tr>
<tr>
<td>( sa )</td>
<td>the saliency value of a keypoint</td>
</tr>
<tr>
<td>( wa )</td>
<td>the ambiguity score of a keypoint</td>
</tr>
<tr>
<td>( \alpha )</td>
<td>exponent of ( se )</td>
</tr>
<tr>
<td>( \beta )</td>
<td>exponent of ( sa )</td>
</tr>
<tr>
<td>( \gamma )</td>
<td>exponent of ( wa )</td>
</tr>
<tr>
<td>( I_q(t) )</td>
<td>impact score of keypoint ( t ) in a query image</td>
</tr>
<tr>
<td>( I_d(t) )</td>
<td>impact score of keypoint ( t ) in a database image</td>
</tr>
</tbody>
</table>

**TABLE II**

**Example of Computing \( R \) and \( n_i \).**

<table>
<thead>
<tr>
<th>( \frac{R_{Raw}}{n_1} )</th>
<th>1.37</th>
<th>3.25</th>
<th>7.12</th>
<th>18.31</th>
<th>43.43</th>
<th>103.0</th>
<th>244.4</th>
<th>579.6</th>
</tr>
</thead>
<tbody>
<tr>
<td>( n_i )</td>
<td>1</td>
<td>3</td>
<td>8</td>
<td>18</td>
<td>43</td>
<td>103</td>
<td>244</td>
<td>580</td>
</tr>
</tbody>
</table>

**IMPORTANT SYMBOLS**

- \( R \): Impact score
- \( n_i \): Quantization number
- \( \alpha, \beta, \gamma \): Exponents
- \( I_q(t), I_d(t) \): Impact scores
- \( se, sa, wa, \alpha, \beta, \gamma \): Parameters

**ACKNOWLEDGMENTS**

The authors would like to thank the anonymous reviewers for their comments and suggestions that helped improve the quality of the paper. This work was supported in part by the National Natural Science Foundation of China under Grant 61602070, in part by the Beijing Municipal Science and Technology Commission under Grant Z171100002417005, and in part by the Beijing Municipal Commission of Education under Grant KZ201810005018.
high impact scores are selected, the image retrieval process can be carried out in the same way as the baseline approach.

Fig. 2 presents an example of query pruning on Oxford5k dataset [4]. In this example, AP (Average Precision) is employed to evaluate retrieval accuracy, which denotes the area under the precision-recall curve. For the classic method (left), the query has 1990 visual words. After query pruning (right), 67.2% visual words are discarded. The remaining 653 query visual words yield an AP of 49.12% (see Section V-B for its meaning), which is even 1.07% higher than the baseline.

B. Early Termination

Q-Index: The baseline mechanism for answering a query uses a pre-computed inverted index, Each visual word in the codebook has a corresponding entry in the index, and is associated with a list of postings $<d_i, f_a>$ recording its image ID and other metadata. In a document sorted index, postings are sorted according to their image IDs. Thus it is problematic for precise control of early termination. In the attempt to solve the problem of early termination, we introduce the Quantized Index (Q-Index) organization. The structure is illustrated in Fig. 3.

The Q-Index belongs to the impact sorted index, where postings are sorted according to their impact scores. In this inverted index category, it is feasible to have access to the important postings first, and then to the less important ones. With the quantized impact score introduced in Section III-C, Q-Index is constructed through three steps:

- Sort the inverted lists in decreasing order of their IDF scores;
- Each inverted list is partitioned into $k$ impact regions, each corresponding to a quantized impact score;
- Given a feature $t$, its image ID and metadata will be indexed in the corresponding impact region of inverted list $d$.

The benefit of Q-Index is two-fold. First, it enables prior access to those postings with large impact scores. Second, it hardly introduces any system burden, thus being memory efficient.

Early Termination Method: The algorithm for early termination is described as follows. In the offline stage, the Q-Index is constructed. In the online stage, on querying the Q-Index, postings are visited by decreasing impact score. When a certain percentage of postings are processed, the traverse can be stopped. During traversing, the similarity between feature pairs is the same with the baseline, i.e., weighted by TF-IDF scheme.

On the inverted list level, what is the order for traversing? Here, two alternatives can be used. The first one is serial: the inverted lists are traversed one by one. One inverted list is fully traversed before the next list is opened. The second method involves parallel operation: open all inverted lists simultaneously, and then the top-ranked postings are visited first.

On the posting level, another question is: what if postings have the same quantized impact score? Which should be indexed preferentially? For a large-scale and updating system, we use TF to perform this task. To this end, we propose the following two secondary sorting criteria:

- $(impact, f)$: within each inverted list, postings with the same impact score are sorted by decreasing order of TF. Higher TF denotes larger impact, as suggested in Eq. (4). Ties are further broken by increasing order of image IDs;
- $(impact, imgID)$: postings with the same impact score are sorted simply by increasing order of image IDs.

Since TF is also a discrete variable, when it is used as the key for secondary sorting, each impact region in the Q-Index is further partitioned into subregions, corresponding to the TF values. In this scenario, we do not store TF explicitly in Q-Index, maintaining unchanged memory cost.

As illustrated in Fig. 1, the inverted lists are partially traversed. Since the most important postings are visited, the retrieval efficiency can be greatly promoted while the accuracy remains stable.

C. Similarity Function

In this section, we define two similarity functions used in the baseline and the proposed framework, respectively.

For the baseline method, given a feature $t$ which belongs to both query $q$ and a database image $d$, the matching score can be written as

$$
\phi(t) = 1 \cdot 1 \cdot idf(t)^2
$$

where $idf(\cdot)$ is the Inverse Document Frequency of $t$. In Eq. (5), the two “1”s represent the Term Frequency (TF) of feature $t$. Note that here we employ a word-level inverted index [8], so that it is the feature (keypoint) rather than a given visual word that is being matched. So in this case, each feature has a TF of 1. Then, the similarity score $S_{q,d}$ between images $q$ and $d$ is defined as

$$
S_{q,d} = \sum_{t \in q} \phi(t)
$$

where $\ell^2_2$ and $\ell^2_2$ are L2 norms of $q$ and $d$, respectively.

For our method, note that, the impact score is used only for sorting the query and indexed features. On the query side, a
feature is counted if its impact score is higher than a threshold. On the inverted index side, with the quantized impact score, a feature is traversed if it is located in an impact region which is also higher than a threshold. Therefore, the matching score \( \hat{\phi}(t) \) is re-written as

\[
\hat{\phi}(t) = 1 \cdot 1 \cdot \text{idf}(t)^2 \cdot \delta_q(t) \cdot \delta_d(t)
\]

where \( \delta_q(\cdot) \) and \( \delta_d(\cdot) \) indicate whether a query or indexed feature should be counted during query pruning or early termination, respectively. In this scenario, i.e., when query pruning and early termination are employed simultaneously, the similarity function can be defined as

\[
\hat{S}_{q,d} = \frac{\sum_{t \in q\cdot d} \hat{\phi}(t)}{l_q^2 \cdot l_d^2}
\]

In this paper, the baseline and the proposed methods use Eq. (6) and Eq. (8) as similarity functions, respectively.

D. Computational Complexity Analysis

\textbf{Computational Complexity:} After introducing the algorithms for query pruning and early termination, we present an analysis on the computational complexity of the proposed method. We assume that an image database has \( N \) images, and that each image has \( n \) local features on average. Let \( K \) denote the size of the codebook. Then, given a query image with \( n_q \) local features, the computational complexity of the baseline can be approximated as

\[
C_b = \mathcal{O}(N_b) \approx \mathcal{O}\left(\frac{N \cdot n}{K} \cdot n_q \cdot f_p\right)
\]

where \( N_b \) denotes the total number of postings to be traversed under fast image retrieval framework, \( f_q \) encodes on average the fraction of postings to be traversed when early termination is employed, and \( f_p \) is the fraction of features in the query that are preserved after query pruning. In Eq. (10), the product of \( f_q \) and \( f_p \) yields the Percentage of Processed Postings (POPP), which can be calculated as

\[
\text{POPP} = \frac{N_f}{N_b} \times 100\% \tag{11}
\]

where \( N_b \) stands for the number of postings to be traversed in the baseline; in our method, when query pruning and early termination are employed, the number of visited postings is encoded by \( N_f \). We aim at visiting as few postings as possible. Therefore, we employ POPP to measure retrieval efficiency.

Comparing Eq. (9) and Eq. (10), it is clear that the computational complexity of the proposed method has a linear relationship (by POPP) with the baseline. The smaller POPP is, the larger the speedup will be. Nevertheless, what we must keep in mind is the trade-off between speed and accuracy: as following experiments will show, a proper POPP helps produce significant speed up while retaining the retrieval accuracy.

\textbf{Discussions:} In the standard image retrieval framework using BoW model with large codebooks and the inverted index, a number of prior arts which aim at improving retrieval accuracy suffer from increased computational complexity \cite{15, 42, 44}. For these methods, the visual matching process is inevitably accompanied by extra computation such as coordinate subtraction \cite{15, 16}, weight estimation \cite{44, 45}, etc. For example, for the method of geometry-preserving visual phrases \cite{15}, apart from additional memory consumption for offset tables, during inverted list traversal, one has to perform additional plus-minus operation for two times. Ideally, it is
expected that the traversal complexity in [15] is three times that of the baseline.

Moreover, a recent trend lies in the compact image representations such as the Fisher Vector (FV) [46] and the Vector of Locally Aggregated Descriptors (VLAD) [47]. This line of methods, although also relying on local features and the BoW model, are distinguished from others by using small codebooks and by representing an image with a global vector. Then, the dimension of the vectors is reduced with PCA and subsequently quantized using Product Quantization [48] to reduce memory consumption. It is shown [49], [47] that VLAD can achieve one or two orders of magnitude speedup over the Bow [50] using ADC or IVFADC indexing schemes, respectively. Nevertheless, one recurrent issue about VLAD consists in the relatively low retrieval accuracy due to the “aggregation” process.

In addition, for a simple feature selection scheme (mentioned in Section III-A), a threshold is employed to filter out features with low impact scores. The selected features are fixed once a threshold is determined. After feature selection, the retrieval process goes the same way with the baseline system. In this scenario, our method stores the quantized impact score in the Q-Index, which enables dynamic and flexible feature selection.

V. DATASETS AND EVALUATION PROTOCOL

A. Datasets

We evaluate our approach on two benchmark datasets, i.e., the Oxford5k [4] and Paris6k [51] datasets. Along with them, a distractor dataset with 1 million images is used.

Oxford5k This dataset is composed of 5063 images collected from Flickr for 11 groups of famous Oxford architectures. Each group has 5 query images taken under distinct illuminations and angles.

Paris6k This dataset is generated in couple with Oxford5k. It contains 6412 images which is annotated with respect to 55 queries of 11 groups.

MirFlick1M This dataset [52] is used to test the large scale settings. It contains 1 million images collected from Flickr. During the data collection period, for each date, a number of uploaded images with high interestingness were requested. A small subset (25000) of this dataset is annotated using general topics like “sky”, “water”, “transport”, etc. Considering the randomness of the data collection process, the chance is remote that this dataset contains relevant images to queries in the test datasets. Therefore, images in the MirFlick1M dataset serve as distractors, i.e., they are considered as negative samples and used to scale up the test datasets. A summarization of the three datasets is presented in Table III.

B. Evaluation Protocol

Mean Average Precision (mAP) To measure the retrieval accuracy, Average Precision (AP) is used for each query, which is calculated as the area under the precision-recall curve. Then, APs of all queries are averaged to obtain the Mean Average Precision (mAP).

VI. EXPERIMENTS

A. Baseline Setup

In this paper, we consider the Bag-of-Words (BoW) baseline proposed by Philbin et al. [4], along with some recent techniques to improve baseline performance. The procedure of the baseline is described as follows.1

1 Following [20], we use the DoG-Affine (DoGAff) detector, and the scaling factor is set to 12.5. For each image, keypoints are detected using DoG detector, wrapped by affine transformation, and then described by a 128-D SIFT descriptor. We use the rootSIFT variant [53], which is easy to implement and yields higher performance under Euclidean distance.

2 The codebook is trained on Oxford5k using Approximate K-Means (AKM) algorithm [4]. Both clustering and quantization schemes employ the FLANN library [54] for Approximate Nearest Neighbor (ANN) search. The FLANN precision of ANN search is set to 0.60 and 0.85, respectively. We use hard quantization for efficiency, and the codebook size is set to 1M.

3 A standard inverted index is built in accordance to the codebook, and store all the local features in the image collection.

4 We use the Lp-norm IDF [6] instead of classic IDF, and set p to 3. We use L2 normalization to normalize the scores.

In Table IV, we show the baseline results (mAP in percentage) on Oxford5k, Oxford5k + 1M, Paris6k, and Paris6k + 1M datasets. With rootSIFT and Lp-norm IDF, we achieve mAP of 74.27% and 64.74% on the two datasets, respectively. Our baseline outperforms some state-of-the-arts by itself: Zheng et al. produce mAP of 69.6% and 56.2% on the two datasets, respectively; Zhang et al. [15] report mAP of 69.6% on Oxford5k dataset; Zhang et al. [34] obtain mAP of 58.32% on Oxford5k dataset. In comparison with the VLAD representation, [55] reports mAP of 28.9% and 27.0% using the original VLAD and their improved version after Product Quantization (PQ), respectively; [56] reports mAP of 44.8% using 128-D vector without PQ. When comparing results on the 1M dataset, since previous works may use different large-scale datasets, the results are not directly comparable. But generally

TABLE III

DETAILS OF THE DATASETS USED IN THE EXPERIMENTS

<table>
<thead>
<tr>
<th>Dataset</th>
<th># images</th>
<th># queries</th>
<th># descriptors</th>
<th>Evaluation</th>
</tr>
</thead>
<tbody>
<tr>
<td>Oxford5k</td>
<td>5063</td>
<td>55</td>
<td>25255297</td>
<td>mAP</td>
</tr>
<tr>
<td>Paris6k</td>
<td>6412</td>
<td>55</td>
<td>26519259</td>
<td>mAP</td>
</tr>
<tr>
<td>MirFlickr1M</td>
<td>1000000</td>
<td>n.a</td>
<td>128211598</td>
<td>n.a</td>
</tr>
</tbody>
</table>

TABLE IV

MAP (%) OF THE BASELINE METHOD ON BENCHMARK DATASETS

<table>
<thead>
<tr>
<th>Methods</th>
<th>Oxford5k</th>
<th>Oxford5k + 1M</th>
<th>Paris6k</th>
<th>Paris6k + 1M</th>
</tr>
</thead>
<tbody>
<tr>
<td>BoW</td>
<td>66.18</td>
<td>50.09</td>
<td>56.02</td>
<td>41.30</td>
</tr>
<tr>
<td>+ rootSIFT</td>
<td>70.35</td>
<td>53.40</td>
<td>60.52</td>
<td>44.85</td>
</tr>
<tr>
<td>+ Lp-norm IDF</td>
<td>74.27</td>
<td>59.35</td>
<td>64.74</td>
<td>50.10</td>
</tr>
<tr>
<td>+ w-ranking</td>
<td>86.19</td>
<td>68.49</td>
<td>77.22</td>
<td>57.08</td>
</tr>
<tr>
<td>Zheng et al. [6]</td>
<td>69.6</td>
<td>62.6</td>
<td>56.2</td>
<td>51.3</td>
</tr>
<tr>
<td>Zhang et al. [15]</td>
<td>69.6</td>
<td>53.2</td>
<td>-</td>
<td>-</td>
</tr>
<tr>
<td>Zhang et al. [34]</td>
<td>58.32</td>
<td>48.29</td>
<td>-</td>
<td>-</td>
</tr>
</tbody>
</table>

speaking, our results on the 1M datasets are competitive. We also implement the re-ranking step using query expansion [57]. In the experiments, we focus on improving efficiency while preserving initial result accuracy. Most re-ranking methods naturally require a good initial result [4], [43], [58].

B. Parameter Selection

Exponent parameters $\alpha$, $\beta$, and $\gamma$. In Eq. (3) and Eq. (4), $\alpha$, $\beta$, and $\gamma$ are the exponents of feature scale, saliency and ambiguity, respectively. The three parameters are tuned on both Oxford5k and Paris6k datasets: we use 30% query pruning and set the percentage of processed postings (POPP) to be 10% and 20%, respectively. Note that, the impact score is not quantized for early termination here. The parameter tuning results are shown in Fig. 4. We can see that, under 10% or 20% POPP, the curves first rise and then drops after a peak. The value of $k$ is to be determined in the next part. From these results, we set $\alpha = 0.6$, $\beta = 0.8$, and $\gamma = 1.4$ in the following experiments.

Quantization parameter $k$. A critical issue includes determining the optimal value of $k$. To this end, various $k$ values are tested on the benchmark datasets. For the controlled variable, we do not adopt query pruning, which uses unquantized impact score. We set POPP to 10% and 20%, respectively. When $k$ varies, the mAP results are demonstrated in Fig. 5. Across the different datasets, when $k$ takes a small value (2 or 4), we obtain low results because indexed features are not well-sorted according to their impact scores. But as $k$ increases, the accuracy remains somewhat stable. It can be speculated that when $k$ is very large, the performance will converge to the baseline. From these results, we set $k = 8$ in our experiments.

C. Evaluation

Query pruning. We evaluate query pruning in Fig. 6. On Oxford5k and Paris6k datasets, we plot mAP against the percentage of pruned query features according to Eq. (3). The dashed black line represents the baseline result for reference. As we can see, among the five factors which are used alone as the impact score, TF has the worst performance when the pruning percentage is small. In fact, since the TF score is typically 1 or 2 for each feature, it does not have much discriminative ability being an impact score. When using IDF alone, mAP improves marginally at 5% pruning point, but drops quickly afterwards. This is consistent with previous findings that using a small number of stop words helps improve accuracy. Similar situation can be observed for scale, but the dropping rate seems to be slower than IDF on Paris6k. This indicates that local features with larger scale possess more discriminative ability. For the saliency and ambiguity factors, there is evident improvement over the baseline. This suggests that both factors are good discriminators among query features.

Moreover, it is obvious from Fig. 6 that the proposed impact score consistently outperforms the others. Specifically, on Oxford5k dataset, we are capable of reproducing the baseline when 40% query visual words are pruned. In this scenario, compared with the baseline, we achieve the best mAP of 62.33% (+2.98%) on the drop of 10% visual words. Similar observation can also be found in Fig. 6(b) on Paris6k dataset: the performance is maintained after 30% query visual words are discarded, and the best mAP (53.01%) is obtained at 10% pruning point. These results thus demonstrate the effectiveness of the proposed impact score in query pruning task.

Early termination. The results of early termination are presented in Fig. 7. We plot the mAP results against the percentage of processed postings (POPP). Four configurations for both baseline and our method are shown. Overall speaking, the proposed impact-based approach dramatically outperforms the baselines. Specifically, we are able to achieve almost the same performance by only visiting less than 10% of the postings. In this sense, the traversing time is decreased by more than 10 times.

The second finding associates with the secondary sorting strategy using either TF or image ID. The results suggest that using TF as the secondary key yields a performance no worse than using image ID. This is expected because each inverted
Fig. 6. Image retrieval accuracy (mAP in percentage) versus percentage of pruned visual words (%). Results on Oxford5k + 1M and Paris6k + 1M are shown. The dashed line denotes the baseline. The blue curve represents the pruning strategy defined in Eq. (3). Five alternative strategies are also presented, i.e., using only IDF, TF, scale, saliency, and ambiguity as the impact score.

Fig. 7. Early termination on (a) Oxford5k + 1M and (b) Paris6k + 1M datasets. mAP (%) is plotted against the percentage of processed postings (in log scale). Eight methods are compared. The baseline uses Eq. (6) as similarity function, while ours uses Eq. (8). Serial and parallel denote the two kinds of operations described in Section IV-B2. For sorting options, the baseline uses only TF or image ID (imgID) as keys, while our method employs impact score (imp) as the primary key, and TF or image ID as the secondary key.

list is better sorted using a combination like (impact, TF). We notice that under the “serial” option, TF and image ID yield very similar curves. The reason is that, at each percentage of processed postings, only one inverted list is partially traversed, leaving others either fully traversed or not traversed at all. Therefore, the secondary sorting option actually makes little difference under serial operation.

Thirdly, in large-scale settings, we observe that using the parallel option generates higher curves. It means that the inverted lists are simultaneously open, and the top postings in these lists are visited first. Another advantage of parallel processing is that it can be readily parallelized. Based on these results, the early termination mode we recommend is: using Eq. (8) as similarity function, (impact, TF)-sorted inverted lists, and parallel processing. Potentially, the recommended Q-Index can be very useful in industrial applications such as web image retrieval, in which the powerful infrastructure can be made full use of: given a query image with thousands of visual words, simultaneously opening the inverted lists and stopping early can greatly reduce response time. For other state-of-the-art methods based on BoW and inverted list, our method can be readily applied to improve the retrieval speed. Nevertheless, one issue that must be paid attention to is the unstable performance of the current system. Since we do not have a stop criteria for pruning and early termination, the system should be dynamic in that user may interactively select a preferred setting to draw a balance between speed and accuracy.

Combination of query pruning and early termination To test the complementarity of the proposed two strategies, we present large-scale results on the two datasets as in Fig. 8. The point on the x-axis where POPP = 100% means that all the postings which should be visited if neither query pruning nor early termination is applied. When using pruning, the curves do not reach the 100% point. From these results, we obtain two major observations. First, as more query features are pruned, the curves lean toward upright when POPP is small. This indicates that query pruning improves accuracy more quickly in terms of early termination.

Second, on the far end, pruning 10%, 20%, and 30% features yields superior performance to both the baseline and the “no
pruning” case on Oxford5k + 1M dataset. Specifically, the best efficiency performance on Oxford5k + 1M dataset is obtained by “30% pruning + early termination”, i.e., mAP = 51.63%, POPP = 3.89%. Similarly, on Paris6k + 1M dataset, when using “20% pruning + early termination”, we obtain mAP = 50.25%, POPP = 7.85%. Note that, when combined, query pruning does not have a superior performance to when it is used alone as in Fig. 6. It is expected because too much query pruning will drop some (but not many) useful cues for early termination. Nevertheless, the complementarity is clear from the results.

Memory cost: For memory usage, since the quantized impact score is employed, we do not have to store them in the inverted index. With the Q-Index, only some pointers recording the impact regions need to be kept in memory, the size of which can be neglected. Moreover, within each impact region, a finer partitioning for TF can be readily constructed. Therefore, for the (impact, tf)-sorted Q-Index, only image IDs need to be stored, along with some negligible pointers. Specifically, for each posting, a total of 4 bytes are allocated to store the image ID. For the Oxford5k + 1M dataset as detailed in Table III, the memory consumption is 4.87 GB, which remains the same with the baseline method. In comparison, the VLAD representation [47] would consume 16 bytes for an image, and a total of 15.3M for the Oxford5k + 1M dataset. Although the memory usage of VLAD is two orders of magnitude below the BoW model, the cost of VLAD is a lower retrieval accuracy (see Section VI-A).

Time cost: Using the above described method, the time-accuracy trade-off is summarized in Table V. As POPP(%) increases, Table V shows how the retrieval accuracy (mAP in percentage) as well as the traversing time (ms) vary. Note that when POPP = 100%, the baseline system is produced. Our experiments are performed using a server with 3.46 GHz CPU and 128 GB memory. The time for feature extraction and quantization is 0.70 s and 0.93 s on average on the 1M dataset. Note that, this paper focuses on speeding up the traversal process. Fast feature extraction [59] and quantization [60] methods have been proposed in recent years. Table V shows that the time for traversing the inverted index is greatly reduced when query pruning and early termination are coupled. When mAP is required at 59.63%, the traversing time is 9.6 ms, a save of more than ten folds.

A brute force indexing strategy is to index the BoW histograms directly by FLANN [54]. However, such a method requires vast memory (more than 3700 GB for the 1M dataset), and one can imagine that the time-accuracy tradeoff would be extremely undesirable. On the other hand, In [49], the retrieval time in a 10 million dataset is 716 ms for ADC and 46 ms for IVFADC, respectively. Comparing with [49], our method is faster than ADC, and has slightly lower speed than IVFADC. Considering the high accuracy of the BoW model with large codebook and inverted index, this work has made good progress toward real-time application.

VII. DISCUSSION AND CONCLUSION

This paper focuses on the efficiency issue of BoW based image retrieval. First, an impact score is introduced, computed for each local feature, which takes into account various attributes of a feature, e.g., TF-IDF, scale, saliency as well as ambiguity. Then, based on the impact score, two complementary techniques, i.e., query pruning and early termination, are proposed. Query pruning deals with the query side, in which query features with low impact scores are discarded. On the
other hand, early termination works on the database features stored in the inverted index. The postings are sorted by decreasing order of impact score, so that those of high impacts are visited first. To save memory usage, impact scores are quantized and implicitly stored in the so-called Q-Index. Experiments are performed on Oxford5k and Paris6k datasets. When applied alone, the two techniques are effective on each side. We further show that the combination produces a significant improvement in retrieval efficiency, while preserving a high accuracy.

We emphasize the difference between this work and the simple feature selection methods on three aspects. First, this paper proposes a novel impact score which integrates multiple properties of a local feature, so our definition of feature impact is more comprehensive (see Section III-A). Second, through Q-Index, we propose two alternatives for inverted index traversal, i.e., serial (depth first search) and parallel (breadth first search) operations. In fact, parallel traversal is quite similar to feature selection on the database images, because it implicitly exerts a threshold on the indexed features. In experiments, we show that serial traversal is only slightly inferior to parallel traversal. Therefore, other than simple feature selection or parallel traversal, we actually find that serial traversal is also competitive for early termination. Third and most importantly, the proposed system is well extendable and flexible. The proposed system can be readily adopted when the database size keeps growing. Newly indexed local features can find their impact region in the Q-Index. In this sense, all the database features can be stored in the Q-Index. However, for feature selection methods, only those local features with impact scores above a threshold are selected. Once selected, the pool of indexed features are fixed and cannot be further pruned (it is expensive to store the raw impacts in the inverted index).

In contrast to the rigid feature selection method, the flexibility of our system is a great advantage. On one hand, users can find their preferred tradeoff between speed and accuracy interactively: On the other hand, for system maintenance, new algorithms, optimizations, or extensions can be added to the system without too much effort, because all the original local features are being stored without extra memory requirement. Compared with retrieval speed, memory usage is of less importance due to the extensive usage of PC Clusters.

In the future work, we will further investigate other properties of a local feature, in order to design more effective impact scores. Meanwhile, other early termination options such as the safe termination or top-k ranking, will be studied. This work focuses on Term-at-a-time scoring mode, and we will study the Document-at-at-time [61] scheme as well.

REFERENCES

IEEE TRANSACTIONS ON MULTIMEDIA


Liang Zheng received the B.E. degree in life science in 2010 from Tsinghua University, Beijing, China, where he is currently working toward the Ph.D. degree in information and communication engineering from the Department of Electronic Engineering. His research interests include multimedia information retrieval and computer vision.

Shengjin Wang (S’96–A’96–M’03) received the B.E. degree from Tsinghua University, Beijing, China, in 1985, and the Ph.D. degree from the Tokyo Institute of Technology, Tokyo, Japan, in 1997. From 1997 to 2003, he was a Research Staff Member with the Internet System Research Laboratories, NEC Corporation, Japan. Since 2003, he has been a Professor with the Department of Electronic Engineering, Tsinghua University, Beijing, China, where he is currently the Director of the Research Institute of Image and Graphics. His current research interests include image processing, computer vision, video surveillance, and pattern recognition.

Ziqiong Liu received the B.E. degree in information engineering from Southeast University, Nanjing, China, in 2011, and is currently pursuing the Ph.D. degree in electronic engineering from Tsinghua University, Beijing, China. Her current research interests include image/video processing and large scale multimedia retrieval.

Qi Tian (S’05–M’06–SM’03) received the B.E. degree in electronic engineering from Tsinghua University, Beijing, China, in 1992, the M.S. degree in electrical and computer engineering from Drexel University, Philadelphia, PA, USA, in 1996, and the Ph.D. degree in electrical and computer engineering from the University of Illinois at Urbana-Champaign, Champaign, IL, USA, in 2002. He took a one-year faculty leave at Microsoft Research Asia from 2008 to 2009. He is currently a Professor with the Department of Computer Science, University of Texas at San Antonio, San Antonio, TX, USA. His research interests include multimedia information retrieval and computer vision.