Multimedia Systems
DOI 10.1007/s00530-013-0330-4

REGULAR PAPER

Visual word expansion and BSIFT verification for large-scale image search

Wengang Zhou · Houqiang Li · Yijuan Lu · Meng Wang · Qi Tian

Abstract Recently, great advance has been made in large-scale content-based image search. Most state-of-the-art approaches are based on the bag-of-visual-words model with local features, such as SIFT, for image representation. Visual matching between images is obtained by vector quantization of local features. Feature quantization is either performed with hierarchical *k*-NN which introduces severe quantization loss, or with ANN (approximate nearest neighbors) search such as *k*-d tree, which is computationally inefficient. Besides, feature matching by quantization ignores the vector distance between features, which may cause many false-positive matches. In this paper, we propose constructing a supporting visual word table for all visual words by visual word expansion. Given the initial quantization result, multiple approximate nearest visual words are identified by checking supporting visual word table, which benefits the retrieval recall. Moreover, we present a matching verification scheme based on binary SIFT (BSIFT) signature. The L_2 distance between original SIFT descriptors is demonstrated to be well kept with the metric of Hamming distance between the corresponding binary SIFT signatures. With the BSIFT verification, false-positive matches can be effectively and efficiently identified and removed, which greatly improves the precision of large-scale image search. We evaluate the proposed approach on two public datasets for large-scale image search. The experimental results demonstrate the effectiveness and efficiency of our scheme.

Keywords Visual word expansion · Binary SIFT · Matching verification · Image search

1 Introduction

In recent years, great advance has been made in large-scale content-based image retrieval [1–4, 6–8, 24, 25, 27–30]. Two kinds of work make major contribution to it. The first one is the introduction of local invariant feature, involving interest point detector and local patch descriptor. Popular interest point detectors include difference of Gaussian (DoG) [5], MSER [14], Hessian affine [15], etc. Local patch descriptors make a representation of the local appearance around interest points. Well-acknowledged descriptors include SIFT [5], SURF [16], etc. The second work is the bag-of-visual-words (BoW) model [1] leveraged from information retrieval [11]. With the BoW model, local features in images are quantized to visual words by vector quantization. Then, an image is compactly represented with a "bag" of visual words, and can be efficiently indexed with an inverted file structure for online query.

W. Zhou (✉) · Q. Tian
Department of Computer Science, University of Texas at San Antonio, Texas, TX 78249, USA
e-mail: zhwgeeis@gmail.com

Q. Tian
e-mail: qitian@cs.utsa.edu

H. Li
Department of EEIS, University of Science and Technology of China, Hefei 230027, People's Republic of China
e-mail: lihq@ustc.edu.cn

Y. Lu
Department of Computer Science, Texas State University, Texas, TX 78666, USA
e-mail: yl12@txstate.edu

M. Wang
School of Computer and Information, Hefei University of Technology, Hefei 230009, People's Republic of China
e-mail: eric.mengwang@gmail.com

Published online: 10 July 2013

In essence, image search has to address the problem of visual matching between images. When images are represented with local SIFT features, image matching is realized by visual matching of local features. In large-scale image search, two features from different images are considered as a match, if they are quantized to the same visual word. However, there is a dilemma which involves two problems in this strategy. On one hand, even if their distance is small enough, the two features will not be regarded as a match when they are quantized into different visual words. As a result, many relevant features of database images are missed, which consequently reduce the retrieval recall.

On the other hand, as frequently observed, SIFT features quantized to the same visual word may still have large Euclidean distance between each other, which causes many false-positive feature matches and consequently degrades the precision of image search. This is due to the fact that, since the dimension of SIFT feature space is as high as 128, the sub-spaces corresponding to some visual words are still likely to be large even with millions of visual words, i.e., the SIFT feature space is divided into millions of sub-spaces. Generally, in visual matching with SIFT descriptors [5], it is the vector distance between two SIFT descriptors that should be used to determine whether they are likely to be a true match. Therefore, it is necessary to further check the distance between SIFT features after vector quantization. However, it is infeasible to store original SIFT descriptors in an inverted index file, as it will involve excessive cost in memory. Besides, it is also not efficient to compute vector distance with the original SIFT descriptors. Therefore, compact representations of SIFT descriptor are desired.

In literature, there are some methods, such as k-d tree [23] and random forest [10], which can be used to address the first problem discussed above. Although better quantization results can be obtained, the sacrifice in efficiency is non-negligible. To deal with the second problem, some other works, such as Hamming embedding [8, 12], convert SIFT descriptor to binary signature to remove false SIFT matches. However, to our best knowledge, none of them explicitly demonstrate that the Hamming distance from binary signature is consistent with the L_2 distance of the SIFT descriptor. As a result, the improvement from the current works [7, 8, 12] is limited.

In this paper, we propose a novel visual word expansion approach to improve the quantization accuracy and boost the retrieval recall. Our visual word expansion scheme is based on the observation that the expected nearest visual word to a test feature is always close to the approximate nearest visual word which can be efficiently identified by the hierarchical k-NN search. Moreover, we present a new scheme to transform a SIFT descriptor to a binary bit stream, called binary SIFT signature. Extensive study with large-scale (trillion) sample pairs reveal that the generated binary SIFT effectively keeps the distance metric of the original SIFT descriptor. We apply the binary SIFT to large-scale image search. To adapt to the classic BoW model for large-scale image search, the binary SIFT signature is stored in the inverted file list. During the online retrieval, for each query feature quantized to a visual word, we further compare its binary SIFT signature with those in the inverted file list following the visual word. Only those features with sufficiently small enough Hamming distance from the query feature are regarded as true matches. Since the main computation in Hamming distance is logic operation, the computational cost is low. Experiments on image search in million-scale dataset demonstrate the effectiveness of the proposed scheme.

The rest of the paper is organized as follows. Section 2 reviews related work in literature. Section 3 discusses our method in details. Section 4 shows the experimental results. Finally, the conclusion is made in Sect. 5.

2 Related work

In literature, there are lots of works on large-scale content-based image retrieval. Many of them are based on the BoW model and utilize local invariant features, such as SIFT [5], for image representation. Since local features are high dimensional and an image may contain hundreds or thousands of local features, vector quantization is popularly applied to quantize a local feature to a visual word. Consequently, an image is compactly represented by a "bag" of visual word ID, which effectively adapts to the classic inverted index structure for scalable real-time retrieval. To date, lots of algorithms have been proposed to improve different stages of the classic image retrieval framework. In the following, we make a review of related work on feature quantization, hashing and post-processing.

In feature quantization, k-means is widely used to cluster feature samples to generate visual words for feature quantization [1]. In [3], a hierarchical visual vocabulary tree structure is adopted to greatly increase the quantization efficiency. In [6], instead of quantizing one feature to one visual word, each SIFT is mapped to and represented by multiple nearest visual words. It effectively alleviates the quantization loss, but with high computational cost. In [8], to reduce the quantization error, a binary signature is used to verify features quantized to the same visual word. The binary signature is generated with a thresholding vector computed with large training samples for each visual word, respectively. In [12], an asymmetric version of Hamming embedding is developed by exploiting the precise query location instead of the binarized query vector. In [7], the high-dimensional SIFT descriptor space is partitioned into

regular lattices. In [10], a novel quantization method based on randomized trees is introduced to build visual vocabulary. The conjunction of randomized k-d trees creates an overlapping partition of the SIFT feature space and helps to mitigate quantization error. In [24], code words are generated with the first 32 bits from scalar quantization. Zhang et al. [25] considered local features in groups to model the spatial context and proposed to leverage group distance to generate contextual visual vocabulary. In [26], a novel Fisher kernel framework is proposed as an alternative to the classic BoW model. In [27], sparse coding is incorporated to encode visual appearance as a weighted sum of dictionary elements and a novel mixed-norm regularization scheme is proposed to learn the concept membership distribution of visual appearance.

Distinguished from the above approaches, Jegou et al. [13, 28] proposed a novel quantization strategy which jointly optimizes the dimension reduction and indexing. All local descriptors of an image are aggregated into a uniform and compact representation, which ensures excellent scalability in retrieval. Without the ignorance of individual local features, it cannot well handle partial-duplicate image search where the object of interest only takes a small image patch with cluttered background.

Some algorithms exploit better hashing techniques for visual word vectors. In [17, 20], an interesting min-Hash scheme is proposed to independently select a set of visual words from an image as global descriptors and define image similarity as the set overlap. It is based on the philosophy that the more common are the features in the two images, the higher is the probability of having the same min-Hash result. Such scheme is very effective and efficient in detection of near identical images and video shots. In [18], geometric min-Hash exploits local spatial context to construct repeatable hash keys and increase the discriminability of the description.

Some other schemes improve the image search performance in the post-processing stage. In [1], local spatial consistency is imposed to filter visual word matches with low support. In [8], weak geometric consistency of SIFT orientation and scale is used to remove potential false matches. In [10], global spatial verification is performed to estimate an affine model [19] to filter local matches. In [2], geometric context is represented with coding maps. It recursively removes geometrically inconsistent matches by analyzing those coding maps. In [21], geometric coding improves [2] by generating coding maps with full use of SIFT orientation, scale and key point location. The obtained coding maps are invariant to translation, rotation and scale changes. Besides the above spatial verification techniques, query expansion is another important post-processing scheme. It reissues the initial highly ranked results to generate new queries so as to improve the recall performance. General techniques, such as average query expansion, transitive closure expansion and resolution expansion, are discussed in [4]. In [9], two novel expansion strategies, i.e., intra-expansion and inter-expansion, are proposed. Intra-expansion expands more target feature points similar to those in the query, while inter-expansion explores those feature points co-occurring with the search targets, but not present in the query.

Our approach is related to [24] in that the binary codes of SIFT are adopted. However, our strategy differs greatly from [24]. In [24], it first ensures a high precision rate by binary code generation and then boosts the recall rate by flipping the binary code words. The binary code hashing is very efficient, but the time complexity of flipping the binary code words is exponential to the tolerant bit number. As a result, even with a trade-off between accuracy and efficiency, the time cost of [24] is much more time-consuming than the proposed approach on large-scale image search. In contrast, in this paper, we first boost the recall rate of candidate feature matches with our visual word expansion scheme. After that, we improve the precision rate by binary signature verification. Both stages are very efficient with competitive retrieval accuracy.

In this paper, our focus is on the feature quantization stage. We propose a visual word expansion and binary SIFT verification approach for large-scale image search. With minor increase in time cost, our visual word expansion can effectively boost the recall performance of candidate features. Moreover, we adopt a binary SIFT signature for efficient and effective feature matching verification. Unlike the binary signature in [8], our binary SIFT is independent of image collection and is demonstrated to keep the vector distance of the SIFT descriptor. Our approach can also be integrated with those hashing algorithms and post-processing approaches discussed above to achieve better retrieval performance.

3 Method

In Sect. 3.1, we introduce the idea of visual word expansion which refines feature quantization. In Sect. 3.2, we discuss how to generate binary SIFT signature and demonstrate that the vector distance of SIFT descriptor is kept with the Hamming distance of binary SIFT signature. In Sect. 3.3, we introduce other details of the retrieval framework and summarize our algorithm.

3.1 Visual word expansion

In the BoW model, feature quantization is usually performed in a hard-decision mode or with a soft assignment strategy. In the hard-decision quantization, given a new

feature, we traverse from the root node and go down along the nearest child node, until reaching a leaf node. However, such leaf node does not necessarily correspond to the nearest visual word, especially when the test feature is near the cell boundary of visual words, as the instance shows in Fig. 1b. In fact, approximate nearest neighbor search algorithms, such as k-d tree, can be used to find a better visual word, which is more likely to be the nearest visual word of the codebook to the test feature. However, such approaches are usually computationally expensive. To address this dilemma, we propose a visual word expansion scheme, which reduces quantization loss but introduces little computational burden.

Our visual word expansion scheme is based on the observation that, given a test feature vector, the expected nearest visual word is always near the approximate visual words obtained by hierarchical k-NN search [3]. Therefore, we can build a table to record the nearest visual words of each visual word in our visual vocabulary beforehand. That is, for each visual word, we find the top p nearest visual words, called supporting visual words, in the codebook by k-d tree [23]. Each visual word itself is also considered as its supporting visual word. Such processing can be efficiently performed off-line.

Given a test feature $f^{(i)}$, after finding the approximate visual word v_i by traversing the hierarchical vocabulary tree [3], we further compare the test feature vector with the p supporting visual words $v_{i,j}$ ($j = 1 \sim p$) of the v_i. Then, the p supporting visual words are sorted by their distances from the test feature vector in ascending order. When the test feature is an indexed feature of a database image, we just take the nearest supporting visual word $v_{i,k}$ as the quantization result of the test feature, which satisfies:

$$\left\|v_{i,k} - f^{(i)}\right\|_2 < \left\|v_{i,j} - f^{(i)}\right\|_2, \quad \forall j : 1 \le j \le p, j \ne k \tag{1}$$

On the other hand, when the test feature is from a query image in the online search stage, we continually compare the test feature $f^{(i)}$ with the supporting visual words of visual word $v_{i,k}$. Then, we select the top k supporting visual words of $v_{i,k}$ as the quantization results of $f^{(i)}$ and regard those indexed features in the inverted image list following those supporting visual words as candidate matches to the query feature $f^{(i)}$. Such processing will increase the retrieval recall performance. The impact of parameter p and k will be studied in Sect. 4.1.

Intuitively, we can also adopt the soft assignment strategy of index each database feature with multiple nearest supporting visual words. However, such processing will increase multiple times memory cost per indexed feature. Therefore, in our implementation, we only index each feature into the inverted feature list of the nearest supporting visual words.

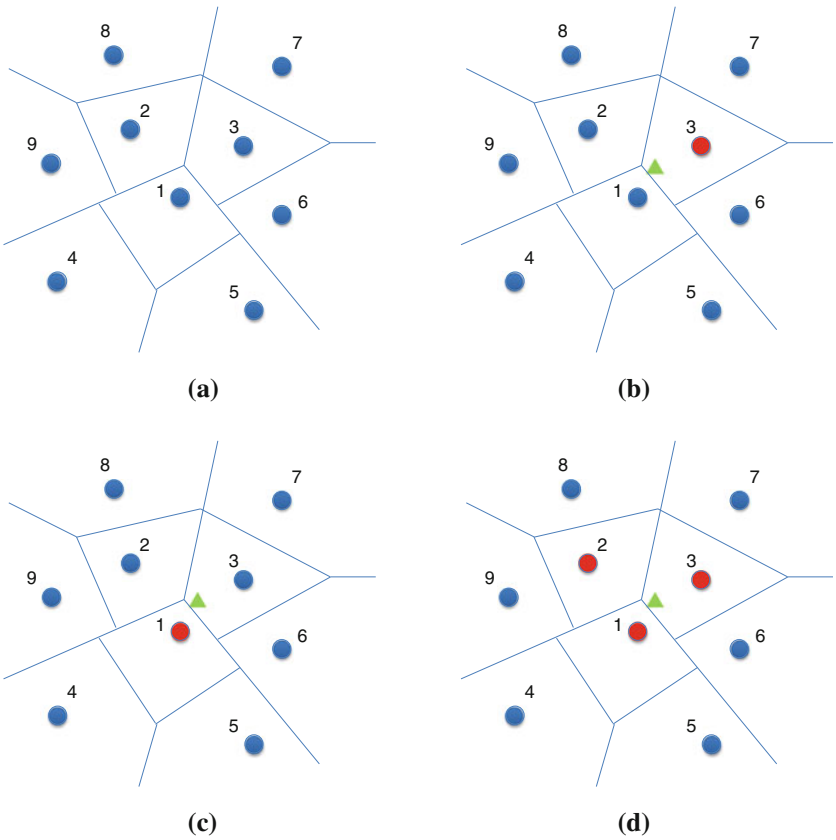

Fig. 1 A toy example of visual word expansion with $p = 4$ and $k = 3$. **a** Nine visual words are obtained by hierarchically clustering training feature samples. Each *blue circle* denotes a visual word. **b** A new test feature (*green triangle*) is quantized to visual word 3 by hierarchical k-NN search. **c** Visual word 1 is found as the nearest visual word by checking the four nearest neighbors of visual word 3. **d** Visual word 1, 2 and 3 are identified as the nearest visual words to the test feature by checking the four nearest neighbors of visual word 1 (best viewed in *color* PDF)

Figure 1 illustrates a toy example of our visual word expansion idea for quantization. Figure 1a shows a tree, which is hierarchically built with the branch number as 3 and the depth as 2, and the 2D feature space is represented by nine visual words. In Fig. 1b, given a test feature (green triangle), we quantize it to visual word 3 by traversing the vocabulary tree. Then, by comparing the distance between the test feature and the four supporting visual words of visual word 3, we can easily identify that the nearest visual word is visual word 1, which is used to index the test feature. When the test feature is a query feature of a query image, we will check the inverted image lists of the top three nearest supporting visual words (visual word 1, 2, and 3) of visual word 1, as highlighted in red.

3.2 Matching verification

The standard SIFT descriptor is extracted from a local image patch by concatenating 8-D orientation histograms of all 16 (4 by 4) sub-patches. We observe that the coefficients in most bins of SIFT descriptor vector are very stable even under various changes in rotation and scaling or noise addition. Such property accounts for its discriminative power in visual identification. In other words, the differences between bins and a predefined threshold are stable for most bins. Based on such observation, we convert a standard SIFT descriptor to a binary signature.

Given a SIFT descriptor vector $\mathbf{F} = (c_1, c_2, \ldots, c_d)^T \in R^d$, $d = 128$, we transform $\mathbf{F}$ to a bit vector (binary SIFT signature) $\mathbf{B} = (b_1, b_2, \ldots, b_d)^T$, as follows:

$$b_i = \begin{cases} 1 & \text{if } c_i > \hat{f} \\ 0 & \text{if } c_i \leq \hat{f} \end{cases} \quad (i = 1, 2, \ldots, d) \tag{2}$$

where $\hat{f}$ is a scalar and is selected as the median value of the element set $\{c_1, c_2, \ldots, c_d\}$ [29]. The binary SIFT signature generation does not involve any training step. It is independent of image collection and it is simple and computationally efficient.

To demonstrate that the discriminative power of SIFT descriptors is well kept in the transformed binary SIFT, we made a statistical study of over 400 billion pairs of SIFT descriptors, taking every SIFT pair extracted from image pairs randomly sampled from a large image dataset. For each descriptor pair, its L_2 distance on the standard SIFT and Hamming distance on the binary SIFT are calculated. From Fig. 2, we can observe that the Hamming distance between binary SIFT is consistent with the average L_2 distance. (The drop in the L_2 distance after the Hamming distance grows over 114 is due to the fact that there is no binary SIFT features with that large a Hamming distance.) In other words, the Hamming distance between binary SIFT can be used to approximate the Euclidean distance between the corresponding SIFT descriptors. Therefore, we can use the binary SIFT instead of the original SIFT descriptor to check the distance between two candidate features. Another advantage of binary SIFT is that the memory cost of binary SIFT is low, making it feasible to store the whole binary SIFT in the index list.

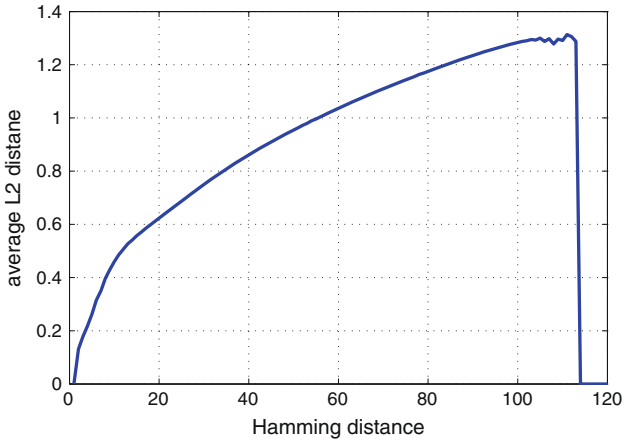

Fig. 2 The average L_2 distance versus Hamming distance. The statistics are obtained based on 400 billion pairs of SIFT descriptors. The L_2 distance is computed with SIFT descriptors unit normalized

In the traditional approach [1–3], two SIFT features from two images are considered as a match if they are quantized to the same visual word. In our implementation, a further verification of binary SIFT is performed. That is, the Hamming distance between two binary SIFT features is no greater than a threshold t. The impact of threshold t will be studied in Sect. 4.1.

Figure 3 shows two examples of feature matching based on binary SIFT signature. It can be observed that false local matches exist on both relevant and irrelevant image pairs after feature quantization. With further verification on binary SIFT, most false matches can be identified and removed.

3.3 Index and retrieval

Our image search method is based on the bag-of-visual-words model. We construct a quantizer by hierarchically clustering large-scale SIFT descriptor samples. The clustering leaf nodes are considered as visual words. The obtained visual vocabulary tree with the supporting visual words (Sect. 3.1) is used to quantize a SIFT feature to a visual word.

In our image search scheme, the binary SIFT is indexed with an inverted file structure for large-scale image database, as illustrated in Fig. 4. Each visual word is followed by a list of entries and each entry contains the ID of images in which the visual word appears. Besides, for each indexed feature, we also store its binary SIFT signature.

(a) (b)

Fig. 3 Matching results verification by binary SIFT signature. The initial matches are obtained with vector quantization. The *red line* segments denote those false matches identified by binary SIFT verification, while the *blue* ones denote true matches passing matching verification. Both images in (**a**) contain the duplicate patch of Mona Lisa face (best viewed in *color* PDF)

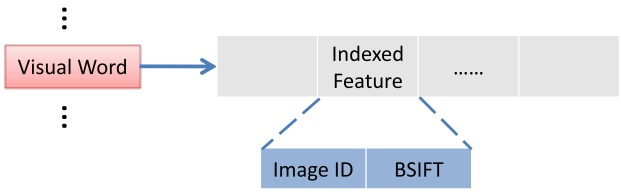

Fig. 4 Inverted file structure

With the inverted index structure, we only need check those images sharing common visual words with the query image and therefore achieve real-time response.

We formulate the image search as a match voting scheme. For each SIFT feature in the query image, each verified feature match will cast a vote to the corresponding image of the database. The similarity between images is defined by the cardinality of matched feature set. Consequently, the database images are ranked by their similarity scores and returned as the image retrieval results. The summary of our retrieval algorithm is shown in Fig. 5.

4 Experiments

We perform the evaluation on two public datasets: DupImage dataset [21] and UKBench dataset [3]. We build a distractor dataset containing 1 million images crawled from the Web. Images in the basic dataset are used as distracters. The DupImage dataset contains 1,104 images from 33 groups, including "Mona Lisa", "American Gothic Painting", "Seven-eleven logo", etc. From the ground truth dataset, 108 representative query images are randomly selected for evaluation comparison. The mean average precision (mAP) is selected to measure the accuracy performance of all methods. The UKBench dataset contains 10,200 images from 2,550 object/scene groups, each group containing four images. We take each image in the UKBench dataset as query and check the number of relevant images in the top-4 retrieval results. The retrieval performance is measured by the N-S score, which is the average four times top-4 accuracy of all 10,200 queries.

We select the standard SIFT feature [1] implemented with an open-source library [22] for image representation. Key points are detected with the difference-of-Gaussian (DoG) detector. A 128-D orientation histogram (SIFT descriptor) is extracted to describe the visual appearance of the local patch around each key point. Before extracting SIFT features, large images are scaled to have a maximum axis size of 400.

In Sect. 4.1, we study the impact of four key parameters in our algorithm on the DupImage dataset mixed with the 1-million distractor dataset. In Sect. 4.2, we compare the proposed approach with four other retrieval algorithms in terms of accuracy, efficiency and memory cost, respectively.

4.1 Parameter impact

There are four key parameters in our approach: visual codebook size, Hamming distance threshold t, supporting visual word parameter k and p. As stated in Sect. 2, the two key components, i.e, visual word expansion and BSIFT verification, are complementary to each other in our system. The first component is focused on improving the recall rate, while the second component contributes to boosting the precision rate. Therefore, we evaluate the impact of those parameters on the two components collaboratively. We first study the impact of visual codebook size and Hamming distance threshold t simultaneously, and select the corresponding parameter setting. After that, the impact of k is investigated to select the optimal value. Finally, we illustrate the impact of p on mean average precision.

To evaluate the impact of the first two parameters on search accuracy and efficiency, we test different settings with all query images on the 1-million image database.

Fig. 5 Summary of our image search algorithm

Algorithm 1: Image search with visual word expansion and BSIFT verification

Input: A query image I

Output: A list of candidate images

Procedure:

1: Extract SIFT features form image I;
2: **For** each SIFT feature $\boldsymbol{f}_i$ in I
3: Compute the binary SIFT of $\boldsymbol{f}_i$ as $\boldsymbol{b}_i$;
4: Quantize $\boldsymbol{f}_i$ to a leaf node $\boldsymbol{n}_i$ by a pre-trained vocabulary tree;
5: Compute the distance between $\boldsymbol{f}_i$ and the supporting visual words $\boldsymbol{v}_{i,j}$ (j=1~p) of $\boldsymbol{n}_i$;
6: Find visual word $\boldsymbol{v}_{i,k}$: $\|\boldsymbol{v}_{i,k} - \boldsymbol{f}_i\|_2 < \|\boldsymbol{v}_{i,j} - \boldsymbol{f}_i\|_2$, for all $1 \leqslant j \leqslant p$, $k \neq j$;
7: Compute the distance between $\boldsymbol{f}_i$ and the supporting visual words $\boldsymbol{s}_{i,j}$ (j=1~p) of $\boldsymbol{v}_{i,k}$;
8: Determine the top k nearest supporting visual words from $\boldsymbol{s}_{i,j}$ (j=1~p);
9: **For** each of top k nearest supporting visual words from Step 8
10: **For** each database feature $\boldsymbol{d}$ in the inverted list of supporting visual word
11: Compute Hamming distance between $\boldsymbol{b}_i$ and the binary SIFT of $\boldsymbol{d}$;
12: **If** the Hamming distance is no greater than a threshold t
13: Increase the vote of the corresponding image;
14: **End**
15: **End**
16: **End**
17: **End**
18: Return database images sorted by their vote scores in descending order.

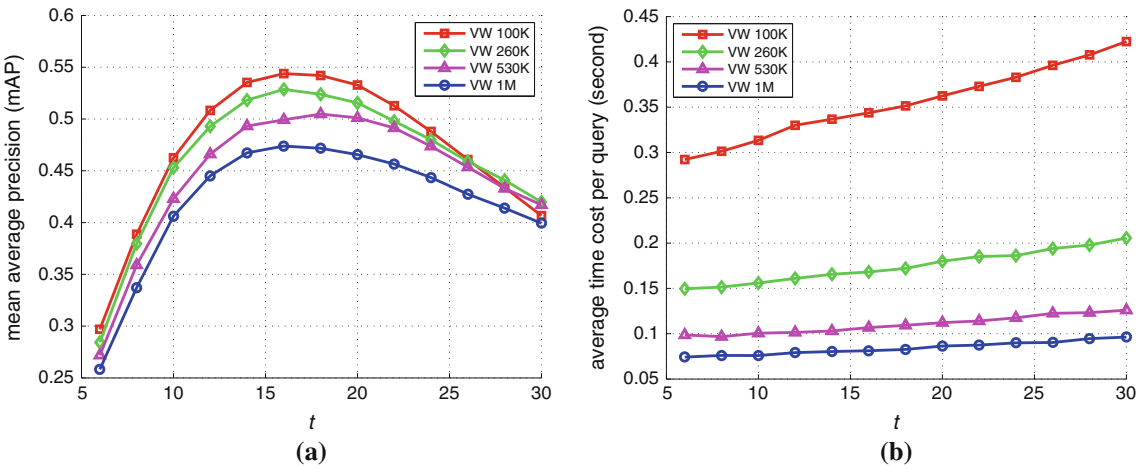

Fig. 6 Parameter impact of codebook size and Hamming distance threshold on the DupImage dataset mixed with the 1-million-image distractor dataset, with $k = 1$ and $p = 50$. **a** Mean average precision, **b** average time cost per query

Four visual codebooks with different sizes, i.e., VW 100 K, VW 260 K, VW 530 K and VW 1 M, are involved for evaluation. Here, VW 100 K represents a visual codebook with 100 thousand visual words. The results are shown in Fig. 6.

From Fig. 6a, it is observed that, when the visual codebook size decreases, the search accuracy increases. This is because smaller visual codebook will keep more true matches, and the false matches can be effectively removed with our binary SIFT verification. On the other hand, when the Hamming threshold t increases, the accuracy first increases to a peak and then drops gradually. This is because smaller t introduces more false negatives, while larger t incurs more false positives. From

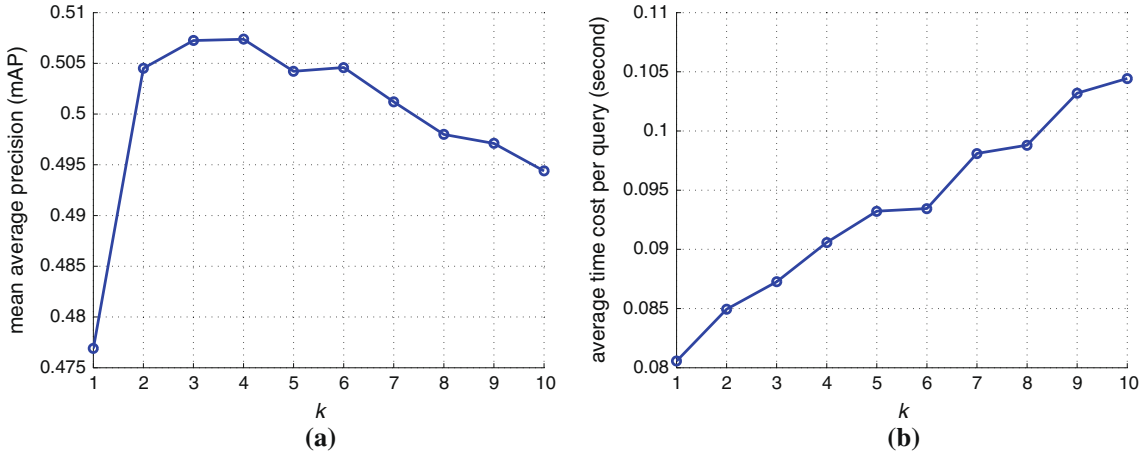

Fig. 7 Parameter impact of different values of k with the 1-M visual codebook, $t = 16$, $p = 50$. **a** Mean average precision, **b** average time cost per query

Fig. 6b, it is shown that the time cost increases sharply when the visual codebook size decreases. This is because more indexed features have to be checked with smaller visual codebook. Considering efficiency, in the following experiments, we select the visual codebook with 1 million visual words and choose the Hamming threshold t as 16.

The third parameter k controls the number of expanded inverted image lists following the supporting visual words. As shown in Fig. 7a, when k increases, the mAP first sharply grows to a peak and then decreases gradually. This is because, when k is relatively small, many relevant database features are included, which increases the retrieval recall and boosts the mAP performance. However, when k becomes larger than 4, more and more database noise features are also included, which consequently degrades retrieval accuracy. On the other hand, more time cost is involved when k takes larger value, as revealed in Fig. 7b. In the following experiments, we select $k = 4$.

The fourth parameter p represents the pool size of supporting visual words. As illustrated in Fig. 8, when p increases, the mAP first rapidly grows and then remains stable after p becomes larger than 60. This is due to the fact that, when p increases from a relatively small value, the pool of supporting visual words becomes larger, which will assist in identifying those nearest visual words in the visual vocabulary and consequently benefit retrieval accuracy. However, when p becomes larger than 60, the additionally included visual words are relatively far away from the query feature and few relevant database features are identified, which will not significantly boost the retrieval accuracy. On the other hand, with a larger p, more candidate feature lists will be verified, which will increase the time cost. To make a trade-off, we select $p = 60$ in our experiments.

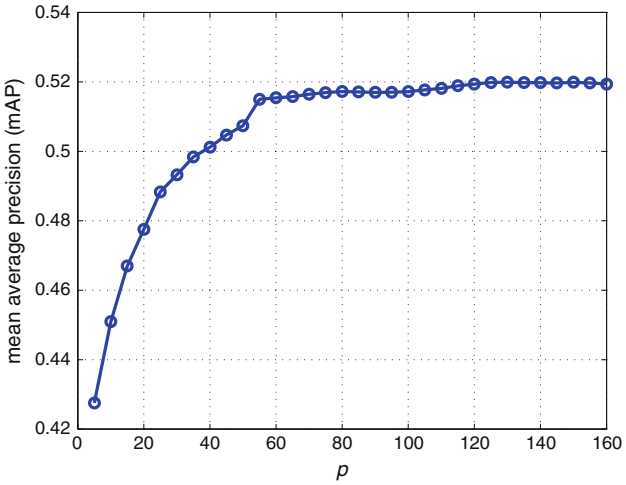

Fig. 8 Mean average precision on different values of p with the 1-M visual codebook, $t = 16$, and $k = 4$

4.2 Evaluation

4.2.1 Comparison algorithms

We compare our approach with four state-of-the-art feature quantization algorithms in large-scale image search. The BoW approach with classic visual vocabulary tree [3] is selected as the "baseline" method. We test various sizes of visual vocabulary for the baseline and find the one with 1 million visual words gives the best overall performance. To enhance the baseline, two other algorithms, i.e., soft assignment (SA) [6] and Hamming embedding (HE) [8], are also included for comparison. The fourth algorithm scalar quantization (SQ) [24] is codebook-training free and indexes images with 32-bit code words. We implement those four comparison algorithms based on the original

papers. Since our focus in this paper is on feature quantization, the weak geometric consistency verification in [8] is not included in the implementation of HE. In the implementation of soft assignment approach [6], the approximate nearest neighbor search is developed with the ANN library [23].

4.2.2 Accuracy

From Table 1, it can be observed that our approach outperforms the three visual codebook-based methods on large image databases. On the 1-million dataset, the mAP of the baseline is 0.38. Our approach hits 0.52, a relatively 36 % improvement over the baseline. Since Hamming codes can effectively filter false features, the Hamming embedding approach achieves a mAP of 0.43, but still 0.09 lower than our approach. The mAP improvement of soft assignment approach is higher than HE. It reaches a mAP of 0.48. Compared with soft assignment, our approach still enjoys a relatively 5.2 % improvement. The SQ approach [24] achieves slightly better mAP performance (0.54) than our approach, but with much higher time complexity.

On the UKBench dataset, the retrieval performance of the comparison algorithms is slightly different from the above observation. As shown in Table 2, the SA approach [6] achieves the highest N-S score over all other methods with the price of high computational cost. Our approach gets an N-S score of 3.18, which is still much better than the other three comparison algorithms. Compared to the four comparison algorithms, our approach makes the best trade-off between accuracy and efficiency.

4.2.3 Efficiency

The experiments are performed on a server with 3.4 GHz CPU and 16 GB memory. Table 1 shows the average online search time cost per query of all four approaches for million-scale image search. The time cost on SIFT feature extraction is not included. It takes the baseline 0.12 s in average to perform one query. Although HE is the most time-efficient one and costs only 0.05 s to finish one online query on average, it suffers more expensive off-line training process since it has to additionally train thresholding vectors for each visual word. Soft assignment [6] is the most time-consuming approach, consuming 0.52 s in average per query. The efficiency of SQ [24] is comparable to the soft assignment approach [6]. The efficiency of our approach is better than the baseline, with 0.09 s on average per query. It is much faster than the soft assignment and the scalar quantization approach and saves off-line training cost compared with HE [8].

The retrieval performance on the UKBench dataset is shown in Table 2. On this dataset, the soft assignment (SA) approach [6] is witnessed as the most time-consuming, while the other four approaches take the comparable time cost for each query. This is due to the fact that this dataset is relatively small with only 10,200 images, and the time cost in retrieval is mostly spent on the feature quantization.

4.2.4 Memory cost

The memory cost on the index file is linear to the number of features to be indexed. Therefore, we compare memory

Table 1 Comparison on mAP and efficiency of different methods on the DupImage dataset with 1-million distractor images

	Baseline [3]	HE [8]	SA [6]	SQ [24]	Our approach
mAP	0.38	0.43	0.48	0.54	0.52
Average time cost per query (second)	0.12	0.05	0.52	0.48	0.09

Not including the time cost for SIFT feature extraction

Table 2 Comparison on N-S score and efficiency of different methods on the UKBench dataset

	Baseline [3]	HE [8]	SA [6]	SQ [24]	Our approach
N-S score	2.90	3.04	3.26	2.99	**3.18**
Average time cost per query (second)	0.04	0.04	0.63	0.05	0.04

Not including the time cost for SIFT feature extraction

Table 3 Memory cost per indexed feature for four approaches

	Baseline [3]	HE [8]	SA [6]	SQ [24]	Our approach
Memory cost per feature (byte)	8	12	24	32	20

cost per feature on all four approaches, as shown in Table 3. For each feature, the baseline approach needs 4 bytes to store image ID and another 4 bytes to store the *tf-idf* weight. The soft assignment has to store each indexed feature in three visual word lists. Therefore it costs 24 bytes, three times the memory cost of the baseline approach. In Hamming embedding approach, for each feature it allocates 4 bytes on image ID and 8 bytes on the 64-bit binary signature. In our approach, besides the 4 bytes for image ID, 16 more bytes are needed to store the 128-bit binary SIFT.

5 Conclusion

In this paper, we present a visual word expansion approach to reduce quantization loss and improve the retrieval recall of candidate features. Moreover, we adopt binary SIFT signature for matching verification to boost retrieval precision. Inverted file structure is used for large-scale indexing and scalable retrieval. Experiments on image search with large-scale database reveal the efficiency and effectiveness of the proposed approach.

In our next work, we will study the gap between vector quantization and visual matching in large-scale image search. We will also investigate better strategies to transform SIFT to binary version preserving the quality of vector comparison.

Acknowledgments This work was provided support as follows: Dr. Li was supported in part by NSFC under contract No. 61272316; Dr. Lu in part by Research Enhancement Program (REP), start-up funding from the Texas State University and DoD HBCU/MI grant W911NF-12-1-0057; Dr. Tian in part by ARO grant W911NF-12-1-0057, NSF IIS 1052851, Faculty Research Awards by Google, NEC Laboratories of America, FXPAL and UTSA START-R award.

References

1. Sivic, J., Zisserman, A.: Video Google: a text retrieval approach to object matching in videos. In Proceedings of ICCV (2003)
2. Zhou, W., Lu, Y., Li, H., Song, Y., Tian, Q.: Spatial coding for large scale partial-duplicate Web image search. In: Proceedings of ACM Multimedia (2010)
3. Nister, D., Stewenius, H.: Scalable recognition with a vocabulary tree. In: Proceedings of CVPR (2006)
4. Chum, O., Philbin, J., Sivic, J., Isard, M., Zisserman, A.: Total recall: automatic query expansion with a generative feature model for object retrieval. In: Proceedings of ICCV (2007)
5. Lowe, D.: Distinctive image features form scale-invariant keypoints. IJCV **20**(2), 91–110 (2004)
6. Philbin, J., Chum, O., Isard, M., Sivic, J., Zisserman, A.: Lost in quantization: improving particular object retrieval in large scale image databases. In: Proceedings of CVPR (2008)
7. Tuytelaars, T., Schmid, C.: Vector quantizing feature space with a regular lattice. In: Proceedings of ICCV (2010)
8. Jegou, H., Douze, M., Schmid, C.: Hamming embedding and weak geometric consistency for large scale image search. In: Proceedings of ECCV (2008)
9. Kuo, Y., Chen, K., Chiang, C., Hsu, W.H.: Query expansion for hash-based image object retrieval. In: Proceedings of ACM Multimedia (2009)
10. Philbin, J., Chum, O., Isard, M., Sivic J., Zisserman, A.: Object retrieval with large vocabularies and fast spatial matching. In: Proceedings of CVPR (2007)
11. Baeza-Yates, R., Ribeiro-Neto, B.: Modern information retrieval. ACM Press, New York (1999). ISBN 020139829
12. Jain, M., Jegou, H., Gros, P.: Asymmetric Hamming embedding: taking the best of our bits for large scale image search. In: Proceedings of ACM Multimedia (2011)
13. Jegou, H., Douze, M., Schmid, C., Pérez, P.: Aggregating local descriptors into a compact image representation. In: Proceedings of CVPR (2010)
14. Matas, J., Chum, O., Martin, U., Pajdla, T.: Robust wide baseline stereo from maximally stable extremal regions. In: Proceedings of BMVC (2002)
15. Mikolajczyk, K., Schmid, C.: Scale and affine invariant interest point detectors. IJCV **1**(60), 63–86 (2004)
16. Bay, H., Tuytelaars, T., Gool, L.V.: SURF: speeded up robust features. In: Proceedings of ECCV (2006)
17. Chum, O., Philbin, J., Zisserman, A.: Near duplicate image detection: min-Hash and tf-idf weighting. In: Proceedings of BMVC (2008)
18. Chum, O., Perdoch, M., Matas, J.: Geometric min-Hashing: finding a (thick) needle in a haystack. In: Proceedings of CVPR (2009)
19. Fischler, M.A., Bolles, R.C.: Random sample consensus: a paradigm for model fitting with applications to image analysis and automated cartography. Comm ACM **24**, 381–395 (1981)
20. Chum, O., Philbin, J., Isard, M., Zisserman, A.: Scalable near identical image and shot detection. In: Proceedings of CIVR (2007)
21. Zhou, W., Li, H., Lu, Y., Tian, Q.: Large scale image search with geometric coding. In: Proceedings of ACM Multimedia (2011)
22. Hess, R.: An open-source SIFT library. In: Proceedings of ACM Multimedia (2010)
23. Arya, S., Mount, D.: Ann: Library for approximate nearest neighbor searching. http://www.cs.umd.edu/~mount/ANN/
24. Zhou, W., Lu, Y., Li, H., Tian, Q.: Scalar quantization for large scale image search. In: Proceedings of ACM Multimedia (2012)
25. Zhang, S., Huang, Q., Hua, G., Jiang, S., Gao, W., Tian, Q.: Building contextual visual vocabulary for large-scale image applications. In: Proceedings of ACM Multimedia, pp. 501–510 (2010)
26. Perronnin, F., Liu, Y., Sandnchez, J., Poirier, H.: Large-scale image retrieval with compressed fisher vectors. In: Proceedings of CVPR, pp. 3384–3391 (2010)
27. Li, L., Jiang, S., Huang, Q.: Learning hierarchical semantic description via mixed-norm regularization for image understanding. IEEE Trans. Multimedia **14**(5), 1401–1413 (2012)
28. Jegou, H., Perronnin, F., Douze, M., Sanchez, J., Perez, P., Schmid, C.: Aggregating local images descriptors into compact codes. IEEE Trans. Pattern Anal. Mach. Intell. (2011)
29. Zhou, W., Li, H., Wang, M., Lu, Y., Tian, Q.: Binary sift: towards efficient feature matching verification for image search. In: Proceedings of ICIMCS, pp. 1–6 (2012)
30. Zhang, S., Tian, Q., Hua, G., Huang, Q., Wen, G.: Generating descriptive visual words and visual phrases for large-scale image applications. IEEE Trans. Image Process. **20**(9), 2664–2677 (2011)