Joint Optimization Toward Effective and Efficient Image Search

Shikui Wei, Dong Xu, Xuelong Li, *Fellow, IEEE,* and Yao Zhao, *Senior Member, IEEE*

***Abstract*—The bag-of-words (BoW) model has been known as an effective method for large-scale image search and indexing. Recent work shows that the performance of the model can be further improved by using the embedding method. While different variants of the BoW model and embedding method have been developed, less effort has been made to discover their underlying working mechanism. In this paper, we systematically investigate the image search performance variation with respect to a few factors of the BoW model, and study how to employ the embedding method to further improve the image search performance. Subsequently, we summarize several observations based on the experiments on descriptor matching. To validate these observations in a real image search, we propose an effective and efficient image search scheme, in which the BoW model and embedding method are jointly optimized in terms of effectiveness and efficiency by following these observations. Our comprehensive experiments demonstrate that it is beneficial to employ these observations to develop an image search algorithm, and the proposed image search scheme outperforms state-of-the-art methods in both effectiveness and efficiency.**

***Index Terms*—Bag-of-words (BoW), embedding method, high effectiveness, high efficiency, large scale image search.**

I. Introduction

IMAGE SEARCH aims to effectively and efficiently retrieve the desired images from a large image database. However, identifying relevant images is still a challenging task due to appearance variations in illumination, scale, viewpoint, orientation, etc. [1]–[5], especially for large-scale image database. This has been experimentally verified by TRECVID evaluation on automatic/manual/interactive search [6]. Moreover, image search efficiency is also becoming a critical issue with rapid growth of the image database size.

Recently, bag-of-words model (BoW) has become popular in multimedia search area due to its simplicity and effectiveness [7], [8], [9]. The basic idea is to represent an image as a collection of orderless visual words and organize all visual words into an inverted table for efficient image search. The earlier work focuses mainly on the construction of visual vocabulary by using different vector-quantization techniques [1], [9]–[13] to optimally approximate the local image descriptors. Since a visual word is an approximate representation of a local descriptor, the accuracy of image search based on the visual word matching will be inevitably decreased due to the quantization error. To alleviate it, the latest extensions [14], [15] introduce additional embedding codes to compensate the quantization error.

While a lot of BoW-based methods and embedding methods have been proposed, less effort has been made to understand their working mechanism in a quantitative manner. For example, how the search performance is affected by the quantization error? What is the major factor for reducing the quantization error? How can the embedding method improve search performance? Understanding their essentials is critical for developing better image search systems. To this end, we systematically study how some key factors of the BoW model and embedding methods affect the search effectiveness and efficiency, and summarize several observations to guide the system design. Following these observations, we present a new image search method, called PKMLSE, which optimally balances the effectiveness and efficiency of the image search system by combining partitioned *k*-means clustering scheme (PKM) with a linear segment embedding (LSE) method. The main contributions of this paper can be summarized as follows.

1) We discover the working mechanism underlying the BoW model and embedding methods in a quantitative manner, and summarize a few observations from comprehensive experiments. While no strict proof is given for those observations, they indeed work well for guiding the design of better image search engines in terms of effectiveness and efficiency. In addition, these observations are not specific to the proposed PKMLSE

Manuscript received October 2, 2011; revised July 27, 2012 and December 29, 2012; accepted February 4, 2013. Date of publication March 27, 2013; date of current version November 18, 2013. This work was supported in part by the National Basic Research Program of China under Grant 2012CB316400, the National Science Foundation of China under Grants 61125106, 61202241, 61025013, 61210006, 91120302, and 61072093, the Fundamental Research Funds for the Central Universities under Grant 2012JBZ012 and Grant 2013JBM024, the Shaanxi Key Innovation Team of Science and Technology under Grant 2012KCT-04, Program for Changjiang Scholars and Innovative Research Team in University, and the Open Project Program of the National Laboratory of Pattern Recognition (NLPR). This paper was recommended by Associate Editor P. S. Sastry.

S. Wei is with the Institute of Information Science, Beijing Jiaotong University, with the Beijing Key Laboratory of Advanced Information Science and Network Technology, Beijing 100044, China, and also with the School of Computer Engineering, Nanyang Technological University, 639798, Singapore (e-mail: shkwei@bjtu.edu.cn).

D. Xu is with the School of Computer Engineering, Nanyang Technological University, 639798, Singapore (e-mail: dongxu@ntu.edu.sg).

X. Li is with the Center for OPTical IMagery Analysis and Learning, State Key Laboratory of Transient Optics and Photonics, Xi'an Institute of Optics and Precision Mechanics, Chinese Academy of Sciences, Xi'an 710119, China (e-mail: xuelong_li@opt.ac.cn).

Y. Zhao is with the Institute of Information Science, Beijing Jiaotong University, and also with the State Key Laboratory of Rail Traffic Control and Safety, Beijing 100044, China (e-mail: yzhao@bjtu.edu.cn).

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

Digital Object Identifier 10.1109/TCYB.2013.2245890

scheme, and they can be generally applied to optimizing the existing work. To the best of our knowledge, such observations have never been published.

2) We propose a new scheme for efficiently constructing very large visual vocabularies and assigning visual words. Instead of generating auxiliary information of visual words as reported in [15], the PKM method is employed to construct visual vocabulary here. More importantly, the proposed PKM vocabulary construction method greatly meets the requirement of the optimal rules in the proposed observations. As to be shown in experimental part, the PKM scheme provides an extremely efficient mapping process from local descriptors to visual words and an extremely small cost of memory usage without degrading the search effectiveness.
3) We propose a discriminative embedding method, called linear segment embedding. Although this method is very simple, the generated embedding codes provide outstanding capability for distinguishing true matches from false ones. Together with the proposed PKM scheme, the embedding method significantly improves the search accuracy as well as decreases the computational cost by jointly optimizing the parameter settings according to the proposed observations.

The rest of this paper is organized as follows. We first give a brief review about the BoW model and embedding methods in Section II. Then, some observations, which convey the working mechanism of the BoW model and embedding methods, are summarized and validated in Section III. Section IV presents the proposed PKMLSE method, which is designed by following these proposed observations. In Section V, experimental results and performance analysis from real-world image search are given in detail. Finally, we discuss future work and conclude this paper in Section VI.

II. Related Work

The BoW model is first used in the areas of natural language processing and information retrieval, in which a document is represented as a bag of orderless words. In multimedia search area, the BoW model is adopted to represent an image as a collection of orderless visual words [1], where one key issue is how to construct a visual vocabulary [16]. In essence, the construction of visual vocabulary is to vector-quantize the local image descriptors in a training set. A widely used construction scheme is to perform exact k-means clustering (FKM) and treat each cluster centroid as a visual word [9]. In the word assignment stage, a new local descriptor is mapped to a visual word by searching its nearest cluster centroid. Although this method is quite effective, its time and space complexity is extremely high, making it even unpractical for handling a large set of training samples. To improve the computational efficiency, a number of methods [1], [10], [11], [17], [18] have been designed to efficiently construct visual vocabularies. An approximate k-means clustering scheme [11] speeds up the vocabulary construction by replacing the exact computation with an approximate nearest-neighbor method [19]. A hierarchical k-means clustering scheme (HKM) [1] reduces time complexity by recursively performing k-means clustering with a small cluster number. Other efficient vocabulary construction schemes include [10], [17]. However, all these methods construct visual vocabularies by quantizing the training samples in their original feature space, which need a very large training set when constructing a large visual vocabulary, and thus, lead to high computational cost and huge memory usage in the training stage. In addition, since each visual word is separately stored as a high dimensional vector, the memory usage for holding a large visual vocabulary is also nontrivial in the query stage. To address these issues, we construct a visual vocabulary by separately building a series of small visual subvocabularies in the low-dimensional subspaces and combining these subvocabularies by Cartesian product. In this way, the time and space costs are only spent on constructing and storing those subvocabularies, which is trivial. In our latest work, some good features of PKM, such as unbiased property, are discussed in detail [33].

Given a visual vocabulary, a commonly used paradigm for large-scale image search is to organize the database images with an inverted file after representing these images with collections of orderless visual words. In particular, each visual word corresponds to a list or cell. Given a new database descriptor, it is mapped to a certain word by using the nearest neighbor search, and then a new entry containing the image ID of the descriptor is inserted into the corresponding list. Once we have indexed all the local descriptors of the database images to the inverted table in an off-line fashion, the next stage is to query the inverted table when given a query descriptor. The query process is similar to the indexing process. Likewise, the query descriptor is first mapped to a visual word, and then all items in the corresponding list are returned as matches.

Since a visual word is an approximate representation of a local image descriptor, the search accuracy that is based on the visual word matching will be inevitably decreased due to the quantization error. To improve the effectiveness of the standard BoW framework, some efforts have been made on introducing more auxiliary information. In [14], [15], and [20], an embedding method is introduced into the standard BoW-based scheme. The basic idea is to embed the set of objects into points in a low-dimensional embedding space where the distances among points approximate the distances among objects [21], [22]. In this way, a local image descriptor is represented by a visual word together with a compact description in the embedding space. For example, the product quantization based embedding (PQE) method in [15] maps a local image descriptor into a 64-b embedding code. In addition to introducing embedding code, it is possible to improve the image search performance by exploring the spatial information. To this end, Zhang *et al.* [23] directly take the spatial contextual information among local image descriptors into account when constructing visual vocabularies, whereas others introduce the spatial information by pyramid matching [24]–[27]. For existing embedding methods [14], [15], however, most of them are based on an underlying assumption that a local image descriptor is only mapped into one visual word (i.e., $M = 1$), while some of them show performance improve-

ment by employing multiple assignment [28], [29]. In other words, multiple assignment is not fully considered. In addition, the relationship between visual vocabulary size K and the assignment amount M are also not investigated systematically. Note that mapping a local descriptor to M visual words is to find out M nearest neighbors in the visual vocabulary.

In this paper, we aim at discovering the working mechanism underlying the BoW model and embedding methods, and propose some rules for guiding the design of better image search systems. As we will discuss in the following sections, understanding their essentials can benefit the design of optimal system and significantly improve the performance in terms of effectiveness and efficiency.

III. Observations

In a BoW-based image search framework, the search process is the voting process of matched visual words in query image and database images. Ideally, if visual words can perfectly represent local image descriptors, the voting score can exactly reflect the image similarity. However, it is almost not possible to obtain the ideal result, as only part of the items in the matched cell are true matches of a given query descriptor due to the quantization error. Therefore, the image search performance depends heavily on the quantization error. As indicated in existing work [7], two key factors that affect quantization error are visual vocabulary size K and vocabulary construction method. In fact, when K is large enough, even a random quantization can also obtain a good approximation. That is, we can employ a nonoptimal method to construct a visual vocabulary with low quantization error by enlarging the vocabulary size.

An alternative approach of complementing the quantization error is to refine these items in the coarsely matched cell by using the embedding methods. Only the items whose embedding distances to the query descriptor are not more than a threshold are returned as matches. In this way, some false matches can be removed, and the accuracy can be improved accordingly. For the single assignment case, when the false matches in a cell are much more than true matches, the refining process can remove much false matches as well as keep enough true matches. However, when K is large, the ratio of true matches in the matched cell is high since the quantization error is relatively small. In this case, using the embedding method not only prunes false matches but also removes more true matches, which will violate descriptor matching. In other words, the existing embedding methods only work well for some small or medium visual vocabularies when M is set to 1.

To verify the above conclusion, we carry out several experiments from the view of matching exactness and completeness. In our experiments, we use a toy dataset for testing, where 10K SIFT descriptors are used as query descriptors and another 1000K SIFT descriptors are treated as database descriptors. For the original BoW model, all the database descriptors are directly mapped to an inverted table via the preconstructed visual vocabulary, and the searching process is conducted for each query descriptor by returning the database descriptors falling in the matched cells. For embedding methods, additional codes are associated with the items in cells and only the items whose embedding distances to the query meet a threshold are returned as matches. Here, the Hamming embedding (HE) method in [20] is employed with the optimal parameter setting reported in the original paper. In our experiments, the 100 nearest neighbors of each query descriptor are treated as groundtruth, which are obtained by brute-force searching in all database descriptors. To evaluate the exactness and completeness of descriptor matching, we proposed two measurements, i.e., weighted average precision (AP) and weighted average recall (AR), which are defined as

$$WeightedAP = \frac{\sum_{i=1}^{T} tp^i}{\sum_{i=1}^{T} (tp^i + fp^i)} \cdot \left(1 - \frac{T_{null}}{T}\right) \quad (1)$$

$$WeightedAR = \frac{\sum_{i=1}^{T} tp^i}{\sum_{i=1}^{T} (tp^i + fn^i)} \cdot \left(1 - \frac{T_{null}}{T}\right) \quad (2)$$

where tp^i, fp^i, and fn^i denote the true positives, false positives, and false negatives of ith query, respectively. T is the total number of queries, and T_{null} is the number of queries for which no result is returned. Using the weighted AP can avoid not a number problem in case no result is returned for some queries.

The experimental results for the matching exactness and completeness are illustrated in Figs. 1 and 2, respectively. For comparison, the precision and recall curves from the standard BoW-based method are also plotted. As expected, when K is relatively small (not more than 50K in our case), the precision obtained by the embedding method is higher than the standard BoW-based method. That is, the embedding method is indeed effective for pruning some false matches. In fact, after K gets up to a certain value K_{thd} (5K in our case), the precision begins to decrease. When K is large enough (larger than 50K in our case), the precision is even lower than the standard BoW-based method, as shown in Fig. 1. Obviously, the experimental results are consistent with our conclusion. For the completeness of descriptor matching, the recall obtained by the embedding method is absolutely lower than the standard BoW-based method. This is because the embedding method is just a refining step of items in the matched cell and no new items are added. When it attempts to remove false matches, some true matches are unavoidably pruned away due to the unperfect similarity matching, leading to the decrease of recall. Indeed, it is obvious that the recall decreases with increasing accuracy. Giving both recall and precision curves, here, we just want to highlight that image search quality can still be improved by increasing match accuracy of local descriptors, even decreasing the match recall. Now, we can formally introduce the first observation.

Observation 1*:* When $M = 1$, the embedding method like Hamming embedding works only well for some small or medium vocabularies and the main contribution is to improve the precision.

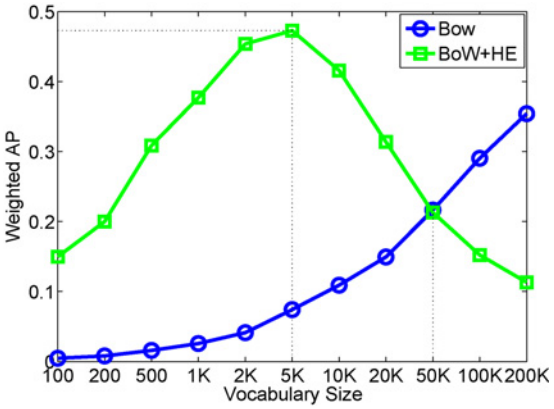

Fig. 1. Impact of K on exactness of descriptor matching before and after using the embedding method.

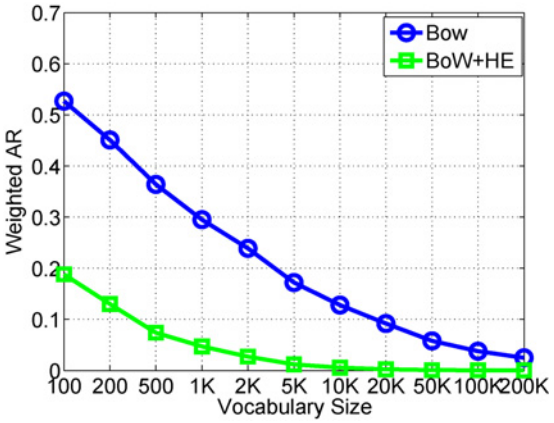

Fig. 2. Impact of K on completeness of descriptor matching before and after using the embedding method.

In addition to the effect of visual vocabulary size, existing embedding methods also pay less attention to the effect of multiple assignment, especially the relationship between multiple assignment and visual vocabulary size. Here, we carry out some experiments to reveal the effect of M with varied K. In this situation, one query descriptor is mapped to M cells, and all these cells are refined by the embedding methods. The experimental results are illustrated in Fig. 3. When K is relatively small (e.g., lower than $5K$), increasing M will significantly decrease the precision for a fixed K. Conversely, when K is large enough (e.g., greater than 50K), increasing M will remarkably improve the precision for a fixed K. It can be explained as follows. When K is small, i.e., using a small vocabulary, there are a large number of items in each cell. Increasing M will dramatically enlarge the set of items that will be refined by the embedding codes. Since more items means more confusion, it is not capable for embedding method to correctly distinguish true matches from false matches in such a huge set. Therefore, the precision will be decreased with increasing M while the recall can be improved further. In another extreme case, when K is extreme large, there are few items in a matched cell and these items are true matches with high probability. Although the embedding method can still further refine these items, it is also possible to remove all

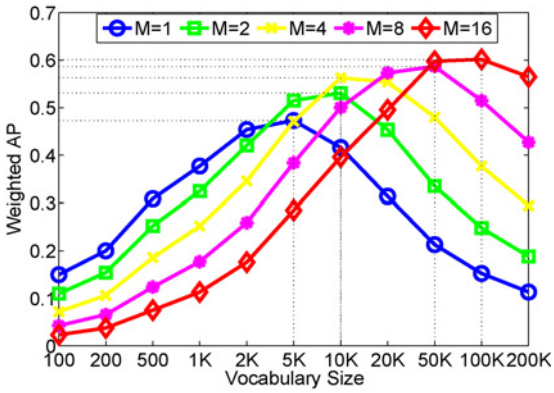

Fig. 3. Impact of M on exactness of descriptor matching using the embedding method.

of them for some queries. In this case, no result is returned for those queries, leading to missed matches. Therefore, the weighted average precision over all queries will be deteriorated even if the precision for some queries is one. Table I shows the statistics on average number of reserved items in a cell before and after refining. The average number decreases with increasing K, and it is not more than one after K gets up to a certain value. Obviously, if the average number is lower than one, there are some queries with zero matched items. The statistic results are consistent with our conclusion. Hence, increasing M makes queries have more chances to get nonzero number of reserved items. In addition, since quantization error has been significantly reduced by using both a large K and the embedding method, the reserved items are usually true matches with high probability. Therefore, increasing M will improve the average precision. Fig. 3 clearly shows the tendency when K is larger than 50K.

As discussed above, there exists a threshold K_{thd} in $M = 1$ case, where the best performance is obtained. Similarly, in the multiple assignment case, there is also a K_{thd} for a fixed M, and the values of K_{thd} are different for different values of $M > 1$. Generally, the larger the value of M is, the larger the value of K_{thd} is. In other words, a large K_{thd} corresponds to a large M. This is because we need more matched cells to compensate the loss of matches caused by the refining process. In addition, we can observe that a larger pair of (M, K_{thd}) generates higher precision of descriptor matching, compared 0.581 at K_{thd} = 100K and M = 16 with 0.449 at K_{thd} = 5K and M = 1. The reason is similar to the case of M = 1. A large K_{thd} means high precision in the refined cells, and a large M means more chance to complement the loss of missed true matches due to the sparseness of matched cells. Note that the sparseness is caused by both the large K and the refining step. Now, we can formally introduce the second observation.

Observation 2*:* The optimal precision can be approached with a relatively big M when K is large enough.

IV. PKMLSE Image Search Framework

The final goal of image search is to provide users with a fast and effective image search engine. In Section III, we introduce some observations to reveal why and how the BoW

TABLE I
STATISTICS ON AVERAGE NUMBER OF RESERVED DESCRIPTORS IN A CELL BEFORE AND AFTER REFINING

Method	M	K										
		100	200	500	1K	2K	5K	10K	20K	50K	100K	200K
BoW	1	11260.44	5652.11	2298.45	1151.31	578.75	232.19	117.25	61.33	26.62	13.12	7.18
Embedding	1	125.230	64.946	23.977	12.465	6.003	2.495	1.276	0.719	0.361	0.236	0.162
	2	249.086	129.704	47.985	25.151	12.023	5.005	2.502	1.386	0.685	0.434	0.301
	4	485.758	254.151	94.632	50.359	24.001	9.928	4.945	2.695	1.310	0.788	0.538
	8	935.408	493.672	183.876	99.009	47.031	19.460	9.600	5.145	2.476	1.427	0.957
	16	1777.264	941.688	354.038	192.841	91.678	37.875	18.647	9.898	4.636	2.678	1.754

model and embedding methods work well. While no strict proof is given for those observations, we will verify that it is beneficial to employ these observations for the development of new image search systems. In this section, a new image search system is designed to verify the correctness of those observations as a special case.

According to Observation 1, existing embedding schemes with default setting can only work well for small or medium visual vocabularies and are sensitive to K. Therefore, we should design an image search scheme by following Observation 2, i.e., jointly optimizing K and M in embedding framework. However, the traditional methods cannot optimally balance the effectiveness and efficiency. When they significantly improve the effectiveness by increasing both K and M, the efficiency is decreased greatly. Therefore, we should pay more attention to search efficiency. According to the report in [7], the vocabulary construction method is even not optimal, we can still achieve promising results by using both a large vocabulary and an effective embedding method.

In brief, we need to find a vocabulary construction method whose mapping complexity is lower than $O(KM)$ and an embedding scheme that can provide enough discriminative capability. To this end, we proposed a PKMLSE based image search framework, which combines a highly efficient vocabulary construction method (i.e., PKM) with a discriminative embedding method (i.e., LSE). As demonstrated in experiments later, since the image search scheme based on PKM and LSE is designed by following the proposed optimal rules, it can optimally balance the effectiveness and efficiency.

A. Partitioned k-means Clustering

As discussed above, most of existing schemes are generally to construct a visual vocabulary by directly clustering the training local image descriptors in their original feature space, resulting in high computation and space complexity. In addition, the efficiency of word assignment is also low.

To address these issues, we introduce the partitioned k-means clustering scheme to construct visual vocabulary, in which the basic idea is to decompose an unmanageably large task into some smaller subtasks [30]. Instead of directly quantizing the input vector in its entire space, PKM first partitions the entire vector into a number of subvectors with lower dimensionality, and then separately quantizes each subvector with its own vocabulary. That is, we need to construct subvocabularies individually in subspaces [31]. In our context, we partition the entire vector into N subvectors and quantize each subvector with its corresponding visual subvocabulary.

Formally, we assume that the input vectors belong to a D-dimensional vector space $\mathbb{R}^D$, which can be the original feature space or its transformed space (e.g., PCA projection). $\mathbb{R}^D$ is divided into a set of subspaces, which is formulated as

$$\mathbb{R}^D = (R^{b_1}_{b_0+1}, R^{b_2}_{b_1+1}, \cdots, R^{b_N}_{b_{N-1}+1}) \tag{3}$$

where $b_{i-1}+1$ and b_i determine the component boundaries of subspace $R^{b_i}_{b_{i-1}+1}$ in the original space $\mathbb{R}^D$. Note that all the subspaces are obtained by uniformly partitioning the space components into several groups. For example, a 128-dimensional space can be divided into two unoverlapped 64-dimensional subspaces. The step-by-step is as follows.

1) Get a training set of images.
2) Extract local descriptors (D-dimensional) from all the training images.
3) Divide uniformly description space into N subspaces $(R^{b_1}_{b_0+1}, R^{b_2}_{b_1+1}, \cdots, R^{b_N}_{b_{N-1}+1})$.
4) Perform independently k-means clustering on all descriptors in each subspace and form subvocabulary V_i with L visual subwords, where $i = 1, 2, \cdots, N$.
5) Build the final vocabulary V by Cartesian product, i.e., $V = V_1 \times V_2 \times \cdots \times V_N$.

After the Cartesian product step, we can obtain a final visual vocabulary with L^N visual words. Note that a final visual word is constructed by concatenating the words in individual subvocabularies.

B. Linear Segment Embedding

As analyzed above, the embedding methods can remarkably improve the precision of descriptor matching by pruning lots of false matches via the embedding codes of descriptors. In essence, the embedding codes are compact representation of descriptors, so the discriminative capability of those embedding codes will heavily impact the pruning quality. Therefore, it is necessary to develop high discriminative embedding codes that can correctly remove false matches as well as keep true matches.

In this section, we present a fast and effective embedding method, which generates a binary representative code by employing the linear segment approximation method. This method includes two main steps, i.e., vector segmentation and binary value calculation. Suppose we have constructed a visual vocabulary $V = \{v_1, v_2, \cdots, v_K\}$. All the binary codes

of original local image descriptors are generated against the visual vocabulary V.

Formally, we assume that the descriptor and the visual word belong to a D-dimensional vector space R^D. Given a local image descriptor $X = (x_1, \cdots, x_D)$, it is uniformly divided into s segments of length d, i.e., $D = s \times d$. Let S_j^x be the jth segment of X, that is

$$S_j^x = (x_{(j-1)\times d+1}, \cdots, x_{j\times d}). \tag{4}$$

Then, we construct a binary vector B of length s for X as

$$B = (b_1, \cdots, b_s) \tag{5}$$

$$b_j = \begin{cases} 1, & \text{if } mean(S_j^x) - mean(S_j^{v^*}) > 0 \\ 0, & \text{otherwise} \end{cases} \tag{6}$$

$$v^* = \min_{v_i \in V} \|X - v_i\|_2 \tag{7}$$

where b_j is a binary bit generated for segment S_j^x, and $S_j^{v^*}$ denotes the jth segment of visual word v^*. Here, Equation (7) means that the given descriptor X is mapped to its nearest visual word v^* in terms of Euclidean distance. The length of binary code is determined by the number of segments. For example, let $X = (1, 3, 5, 7)$ be a 4-D local image descriptor, and $v^* = (2, 4, 6, 4)$ is its nearest visual word. Assume that X is uniformly divided into two segments, i.e., $S_1^x = (1,3)$ and $S_2^x = (5,7)$. Then, b_1 is set to 0 since $mean(S_1^x) = 2$ is lower than $mean(S_1^{v^*}) = 3$. Similarly, b_2 is set to 1 since $mean(S_2^x) = 6$ is greater than $mean(S_2^{v^*}) = 5$.

C. Complexity Analysis

For time complexity, the main computational cost in the searching process is two-fold. The first time-consuming process is to map each descriptor in the query image to the nearest neighbor in the visual vocabulary. For the FKM-based image search system, the computational cost is linearly increasing with respect to the vocabulary size K, leading to a complexity of $O(K)$. In addition to the size of visual vocabulary, the computational cost of the HKM- and PKM-based image search systems is also related to the number of layers l and the number of subspaces N, respectively. For a fixed K, the time complexity of HKM and PKM is $O(K^{1/l})$ and $O(K^{1/N}/N)$, respectively. Therefore, if l and N are same, the efficiency of PKM clearly outperforms HKM.

The second time-consuming process lies in the score voting process. In essence, the voting process is to accumulate the scores between query descriptors and matched database descriptors. Therefore, the computational complexity of score voting is related to the number of query descriptors and the number E of matched descriptors in the database. For a single query descriptor, the expected computing cost C_{ori} is as follows [20]:

$$C_{ori} = \sum_{i=1}^{K} p_i \times (E \times p_i) = E \sum_{i=1}^{K} p_i^2 \tag{8}$$

where p_i is the probability of assigning the query descriptor to the ith visual word. To obtain the minimum cost C_{ori}^*, p_i should be equal to $1/K$, i.e., all lists in the inverted table are of equal length. In this case, $C_{ori}^* = E/K$.

In our context, however, we attempt to improve search effectiveness by significantly enlarging the vocabulary size and using multiple assignment. Therefore, given a singe query descriptor, we define a new computing cost function as

$$C_{new} = M \times E \sum_{i=1}^{n\times K} p_i^2 \tag{9}$$

where M is the number of visual words the query descriptor is mapped to, and n indicates that the vocabulary is enlarged n times, i.e., the size of new vocabulary is $n \times K$. Likewise, we can obtain the minimum cost by setting p_i to $1/(n \times K)$, i.e., $C_{new}^* = \frac{E}{n\times K} \times M = \frac{E}{K} \times \frac{M}{n}$.

Clearly, C_{new}^* is smaller than C_{ori}^* when M is smaller than n. This means that it is theoretically possible to improve effectiveness without decreasing search efficiency by employing the new rules driven from the proposed observations. In practice, it is more possible for FKM method to approach the theoretical minimum cost. For the proposed PKM method, while the probabilities of assigning a given descriptor to different visual words are more unbalanced, it provides extremely mapping efficiency and can speed up the voting process by employing a large n and a relatively small M.

D. Multiple Assignment

Our main principle of designing image search system is to combine a large visual vocabulary with multiple assignment as indicated in Observation 2. Therefore, the PKM scheme discussed in Section III should be extended to facilitate multiple assignment. In essence, multiple assignment is KNN problem rather than NN search, i.e., mapping a given descriptor to several nearest visual words. To this end, we proposed a simple but effective method to perform KNN search.

Suppose we have constructed a visual vocabulary $V = V_1 \times V_2 \times \cdots \times V_N$ by using the PKM scheme. Given a partitioned descriptor $X^D = (X_1^{b_1}, X_{b_1+1}^{b_2}, \cdots, X_{b_{N-1}+1}^{b_N})$, we perform a KNN search for any subvector $X_{b_{i-1}+1}^{b_i}$ against its corresponding visual subvocabulary V_i and get a set V_i^* with k nearest subwords, which is denoted as follows:

$$V_i^* = KNN(X_{b_{i-1}+1}^{b_i}, V_i). \tag{10}$$

Combining V_i^* from different partitions by Cartesian product, we can get a set with k^N nearest visual words of X^D as

$$V_{sub} = V_1^* \times V_2^* \times \cdots \times V_N^*. \tag{11}$$

As indicated in (11), M should be equal to the size of V_{sub}, i.e., $M = k^N$. However, we clearly know the value of M and N in practice, so what we need is to select a proper k. Here, we round $M^{1/N}$ to its nearest integer toward infinity as shown in (12), and assign the integer to k. Clearly, the true M is not less than the expected M. In our experiments, since we split the original feature space into two subspaces (i.e., $N = 2$), the expected M values such as 1 and 16 can be obtained exactly by setting k to 1, 4, respectively. For the case of $M = 2$, the true value of M is set to 4 (i.e., $k = 2$). That is, two

nearest neighbors $\{A_1, A_2\}$ and $\{B_1, B_2\}$ are remained for the first subspace and the second subspace respectively, which form four final matches, i.e., (A_1, B_1), (A_1, B_2), (A_2, B_1) and (A_2, B_2). To give a fair comparison, only (A_1, B_1) and (A_2, B_1) are remained as two best matches in our experiments

$$k = ceil(M^{1/N}). \tag{12}$$

E. Sketch of PKMLSE Scheme

Similarly to the existing image search scheme, the proposed PKMLSE scheme also includes two key components, i.e., the indexing process of database and the searching process of query. Here, we will give a sketch of the proposed scheme by using pseudocode as follows.

1) Train visual subvocabularies $\{V_i\}$ in individual subspaces $R^{b_i}_{b_{i-1}+1}$.
2) Construct the final PKM visual vocabulary $V = V_1 \times V_2 \times \cdots \times V_N$ by Cartesian product.
3) For any database image I:
   a) extract its key points and describe each point with SIFT descriptor;
   b) map each local image descriptor into the nearest visual word and insert the image id to the cell associated with the visual word.
4) Given a query image Q:
   a) extract its key points and describe each point with SIFT descriptor;
   b) map each local image descriptor into the nearest visual words and return all image ids in corresponding cells;
   c) vote image ids returned for all query descriptors and rank them according to their scores.

V. Experiments

A. Experimental Setup

In our experiments, two datasets, i.e., Holidays and Flickr1M datasets, are used for training and testing. The detailed information can be found in [31] and [32]. Note that a set of total 1M descriptors are sampled from Flickr1M dataset to learn the visual vocabularies.

For the performance evaluation, we employ average precision (AP) and mean average precision (MAP) measures. AP corresponds to the area under recall and precision curves. After obtaining the AP for each query, we calculate MAP by averaging APs over all the queries, which indicates the overall performance of all the query images.

B. Evaluation on Observations

According to Observation 1, the embedding method can only work well for small or medium vocabularies when M is set to one. Intuitively, applying embedding method to image search should have similar conclusion. To validate it, we implement four schemes which combine different vocabulary construction and embedding methods, i.e., FKM+HE, PKM+LSE, HKM+LSE, HKM+HE. The performance variation with respect to the vocabulary size K are plotted in Figs. 4–7. We can observe that the performance is not always improved with increasing the size of vocabulary when $M = 1$. When K is lower than a certain value K_{thd}(e.g., $K_{thd} = 20K$ in Fig. 4), increasing K will accordingly improve the image search quality. Nevertheless, when K is larger than K_{thd}, the performance will dramatically degrade with the increasing of K. That is, embedding method does not work well for image matching with a large vocabulary, which is consistent with Observation 1. The explanation is similar to the analysis in Section III. When K is larger than K_{thd}, the precision of descriptor matching will decrease due to zero matches of some query descriptors. Note that the analysis is based on the case of $M = 1$ which is the default setting for most of the existing embedding methods.

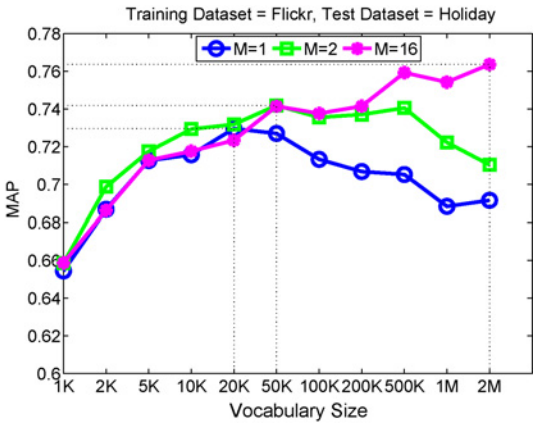

Fig. 4. Impact of K and M on the search performance of FKMHE method.

In Section III, we also point out that for any $M > 1$, there is also a K_{thd} for a fixed M, and a pair of M and K_{thd} with enough large values tends to obtain the optimal precision. We carry out experiments to validate whether these conclusions hold in image matching. The experimental results are also shown in Figs. 4–7. It is clear that the search performance using multiple assignment consistently outperforms the single assignment while using large vocabularies. For a fixed M, there is always a threshold K_{thd} where the best performance is obtained. That is, the conclusions for descriptor matching hold in image matching. In addition, a larger pair of M and K_{thd} always performs better than a smaller pair, compared 0.7685 at $K_{thd} = 500$K and $M = 16$ with 0.6995 at $K_{thd} = 50$K and $M = 1$ in Fig. 5. It is also consistent with the analysis in Section III.

C. Evaluation on Approximation

Although combining a large visual vocabulary with a large M can significantly improve the search accuracy, it leads to an intractable computational cost for vocabulary construction and multiple assignment if using traditional methods. Therefore, we introduce a new and simple method to efficiently perform vocabulary construction and multiple assignment, i.e., PKM. As stated in Section III, the different vocabulary construction methods lead to comparable results when K is large enough. That is, even the vocabulary construction methods are not optimal, we still achieve promising results by using a large

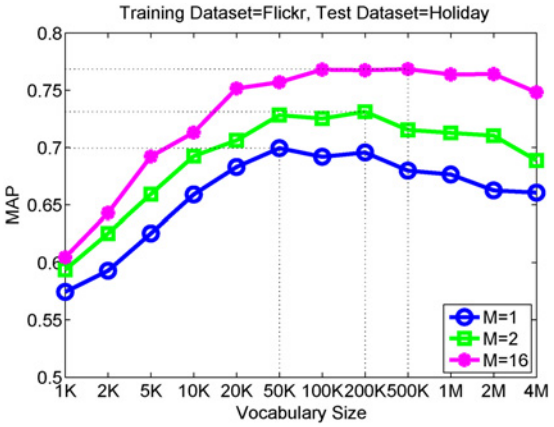

Fig. 5. Impact of K and M on the search performance of the PKMLSE method.

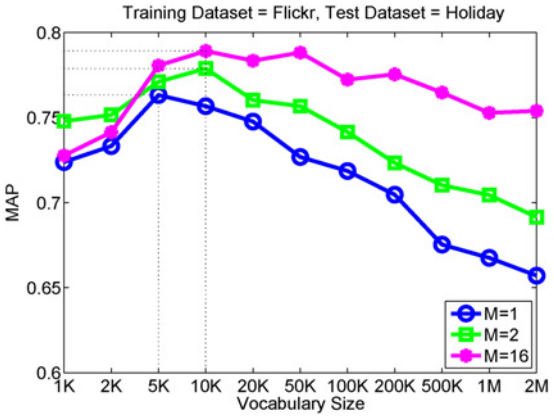

Fig. 6. Impact of K and M on the search performance of HKMLSE method.

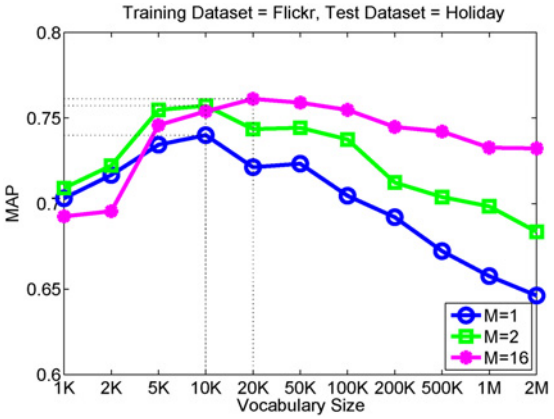

Fig. 7. Impact of K and M on the search performance of HKMHE method.

vocabulary and an effective embedding method. To validate this assumption, we compare the proposed PKM scheme with a commonly used HKM scheme. For fair comparison, both schemes are combined with the same proposed LSE embedding scheme. The experimental results are illustrated in Fig. 8. As expected, the performance of PKM is comparable when K is large enough (e.g., $K > 100K$), while HKM is better than PKM when the vocabulary size is small. Since our purpose is to design a better image search system by combining a large

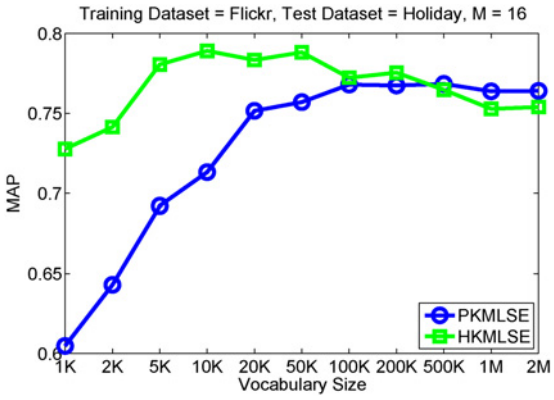

Fig. 8. Evaluation on the effectiveness of the proposed PKMLSE scheme.

vocabulary with a large M, PKM meets our requirement. Note that the discriminative power of PKM scheme is weak when K is small. When K is less than 200K, the precision of PKM is even less than random quantization with a small K due to the effect of marginalization. Fortunately, the PKM scheme can easily generate a very large visual vocabulary and map a local descriptor to visual words with trivial cost. In addition, PKM also has more good properties compared to HKM, such as less memory usage, higher efficiency. We will discuss them in more detail in Section V-F.

D. Evaluation on Embedding Methods

As analyzed in Section III, the embedding methods can remarkably improve the precision of descriptor matching by pruning lots of false matches via the embedding codes of descriptors. In essence, the embedding codes are compact representation of descriptors, and their discriminative capability will significantly impact pruning quality. Therefore, it is necessary to develop high discriminative embedding codes that can correctly remove false matches as well as retain true matches. In this subsection, we compare the proposed LSE embedding method with two state-of-the-art methods, i.e., HE and PQE methods. For fair comparison, we employ the same visual vocabulary construction method for all embedding methods. Intuitively, it is a good choice to treat the proposed PKM scheme as the fixed vocabulary construction method. However, it is not easy to combine Hamming embedding method with the proposed PKM scheme. The main reason lies in that generating the Hamming embedding code requires median vectors that are constructed by using the original training samples associated with individual visual words in the training stage. Given a training set, when we construct the visual vocabulary by using some regular clustering methods such as k-means clustering and hierarchical k-means clustering, each visual word is associated with at least one training sample. Using the training samples remained for a visual word, we can construct a median vector needed for generating Hamming embedding code. However, the vocabulary construction manner of PKM is very different from those regular methods. To construct a large vocabulary, we need only construct a series of subvocabularies using a small training set, and combine these subvocabularies by Cartesian product. This means that no a single original training sample is

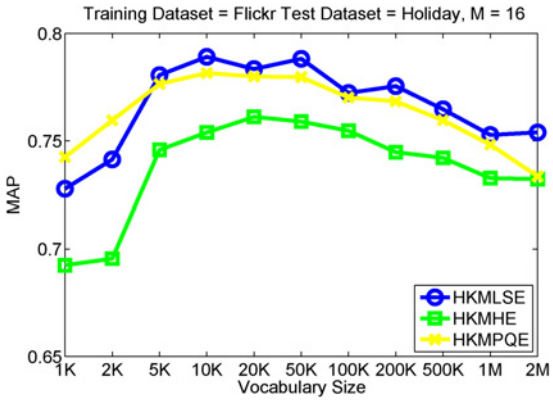

Fig. 9. Evaluation on the effectiveness of the LSE embedding scheme.

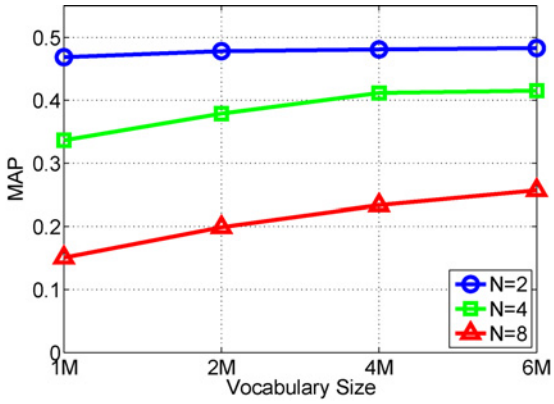

Fig. 10. Effect of subspace number on image search performance.

TABLE II
RESULTS FOR DIFFERENT IMAGE SEARCH SCHEMES

Method	Effectiveness (MAP)	Efficiency (s)
HKM+HE	0.761	4.775
HKM+PQE	0.779	10.536
HKM+LSE	0.783	4.254
PKM+LSE	0.751	3.909

associated with a visual word. Therefore, using PKM scheme cannot build the median vector for a visual word. In our experiments, we adopt hierarchical k-means clustering method to construct visual vocabulary. The experimental results are illustrated in Fig. 9. Both HKMLSE and HKMPQE clearly outperform HKMHE scheme, and the proposed HKMLSE is better than HKMPQE scheme. This means that the proposed LSE embedding scheme has better discriminative capability and can more effectively prune false matches. In addition, Table II lists the search precision of HKMHE, HKMPQE, HKMLSE and PKMLSE schemes by using $K = 20\,000$ (which is the general setting for existing work in [28]). As shown in Table II, the HKMLSE scheme outperforms the other state-of-the-art embedding schemes with the same HKM scheme. This means that the proposed LSE embedding method performs better than PQE and HE embedding methods for the general setting. Note that the search precision of PKMLSE with $K = 20\,000$ is inferior to other schemes. The main reason is that the marginalization of PKM degrades the search precision due to the relatively small size of visual vocabulary. Fortunately, the effect of marginalization can be greatly alleviated by using a large size of visual vocabulary.

E. Effect of Marginalization

Since features are correlated, directly decomposing feature space will definitely lead to information loss. In the subsection, we will discuss the effect of marginalization on image search performance. As shown in Fig. 10, the search performance obtained by using the PKM scheme decreases when N increases from 2 to 8. That is, marginalization reduces the discriminative power of the original features. From the view of visual vocabulary construction, this means that the quantization error of PKM is large and discriminative capability of visual words from PKM is not optimal. In Fig. 11, we compare the PKM scheme with two generally used FKM and HKM schemes. Since setting N to 2 can greatly reduce the computational cost when constructing a large-scale vocabulary (e.g., K = 1M), we only set N to 2 in our experiments. As shown, the image search performance by using the PKM scheme is inferior to both FKM and HKM schemes for any fixed K. That is, the marginalization of the PKM scheme indeed enlarges the quantization error compared to other schemes.

However, it cannot affect the final conclusions in the paper. The essential of BoW method is to get a certain number of points (i.e., visual words) in the original feature space, and to divide the space into a series of regions (i.e., cells) according to these points. The local descriptors in database are then assigned to their nearest regions. While the optimal visual vocabulary construction scheme will lead to more discriminative power of the original feature, the nonoptimal ones can easily obtain a comparable or better performance by using a larger visual vocabulary. That is, the size of visual vocabulary plays a more important role than the construction methods. Therefore, while marginalization of PKM makes lots of visual words useless, there are still enough visual words to provide discriminative power when the size of vocabulary is large enough. For example, when size of visual vocabulary increases from 200K to 1M, the search performance is remarkably improved from 0.426 to 0.468. Fortunately, the PKM scheme can easily generate a very large visual vocabulary and can map local descriptors to visual words with high efficiency. More important, the embedding code can greatly alleviate the effect of marginalization. As shown in Fig. 8, when visual vocabulary size is large enough (here, larger than 100K), the PKM scheme achieves comparable or better performance to the HKM scheme with the same embedding method (i.e., LSE).

F. Complexity Evaluation

In this subsection, we will analyze the complexity of BoW-based embedding methods from two aspects, i.e., space complexity and time complexity. Since the on-line query stage is more important for users, we only discuss the complexity in the searching process.

For space complexity, the main memory usage lies in two steps: index structure (here, inverted table) loading and query preprocessing. Since we use the same index structure for all the

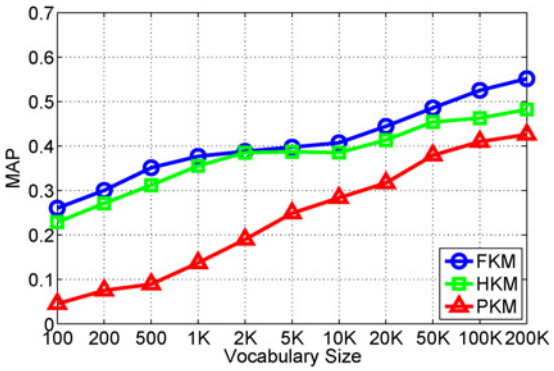

Fig. 11. Comparison of different visual vocabulary construction methods.

methods, the main difference lies in the query preprocessing step. For FKM and HKM schemes, the memory usage is only related to the size of visual vocabulary and descriptor dimension, i.e., $O(K \times D)$. However, HKM scheme requires more space, as the intermediate nodes need to be stored to speed-up word assignment. In contrast, the space complexity of PKM scheme is only $O(K^{1/N} \times D)$. Clearly, the space complexity of PKM scheme is far lower than two generally used schemes.

For time complexity, the computational cost in query stage includes word assignment, embedding code generating and score voting process. However, there is no efficiency comparison for the process of generating embedding code. The main reason is that the time cost for generating embedding code can be negligible compared to other two steps. As analyzed in Section IV-C, the proposed PKM scheme can theoretically achieve higher efficiency than the existing schemes (e.g., FKM, HKM). Here, we experimentally validate the conclusion in the real-world image search system. In our experiments, Flickr1M and Holiday datasets are treated as training and testing datasets respectively, and l and N are set to 2. The experimental results are illustrated in Table III. As expected, these results are consistent with our conclusion. More interestingly, compared to HKM, there is a larger gain beyond the theoretical value (twice times for $l = N = 2$) for PKM scheme. This is possibly because the memory access of PKM is far less than HKM scheme.

As discussed in Section IV-C, the computational cost in voting process phase is related to the number of matched descriptors in database, which heavily depends on the visual vocabulary construction methods. In addition, matching efficiency of different embedding methods also plays an important role in score voting. Since our main purpose is to evaluate the effect of different embedding methods on voting efficiency, we employ the same visual vocabulary construction method, i.e., the HKM scheme. The experimental results are shown in Table IV. The matching efficiency of HE and LSE schemes are comparable at almost all the visual vocabularies sizes. This is because both of them employ the same Hamming distance. In contrast, the PQE method has higher matching efficiency when visual vocabulary size is relative small, whereas HE and LSE schemes perform better when vocabulary size is large. For small visual vocabularies, the number of items in the matched cells are very large, using ADC matching method in PQE scheme will save lots of time. Nevertheless, when vocabulary size is large, the amount of items needed to refine will dramatically decrease. In this satiation, Hamming distance is more efficient than ADC matching, as the cost of constructing a lookup table for ADC is no long trivial.

TABLE V
RESULTS FOR FOUR DIFFERENT IMAGE SEARCH SCHEMES

Methods	MAP
BoW	0.306
HE	0.497
PQE	0.517
PKMLSE	0.530

For the whole search efficiency, it is difficult to give a fair comparison since the different implementations will greatly affect the final results, especially on a large-scale database (e.g., 1M images). To give a rough comparison, we still make some experiments for different schemes on the Holidays dataset by using the general setting (i.e., $K = 20\,000$). As shown in Table II, the PKMLSE scheme remarkably outperforms HKM-based schemes in average search time. This means that PKM vocabulary construction method indeed gains higher search efficiency. In fact, when the size of visual vocabulary is large scale, the search efficiency of PKM based scheme will be more outstanding. Note that all the results are obtained by using a computer with 3.33 GHZ Intel Xeon(R) CPU, 18 GB and MATLAB implementation.

In brief, the proposed PKMLSE method can achieve outstanding search efficiency, and the majority of complexity reduction comes from the PKM scheme.

G. Large Scale Image Search

In this section, the proposed method is further compared with several state-of-the-art methods, i.e., Hamming embedding method [14] and product quantization based embedding method [15], for large-scale image search. To our best knowledge, these two state-of-the-art approaches remarkably outperform other existing image search methods. The detailed analysis for previous methods can be found in [14]. To give a complete comparison, one more frequently used state-of-the-art baseline, i.e, original BoW scheme, is also provided for comparison. Since the comparison is focused on jointly optimizing BoW and embedding methods, post-processing steps, such as weighting, geometrical verification and reranking, are not taken into account for all the methods. We evaluate these systems on a large scale database, i.e., Flickr1M + Holidays dataset, which contains over 1M images. The SIFT descriptors are extracted and also available online [32].

Table V lists the experimental results of four methods. For BoW, Hamming embedding and PQE methods, the results are directly copied from [14] and [15]. These results are obtained on the same testing dataset with the optimal parameter settings. For the proposed method, we set the parameters following the proposed observations. Note that the parameter setting is not specially tuned to be optimal since an empirical setting leads to a good result. In our experiments, the visual vocabulary is

TABLE III
TIME COST (SECONDS) ON WORD ASSIGNMENT FOR THREE DIFFERENT IMAGE SEARCH SCHEMES

Method	K										
	1K	2K	5K	10K	20K	50K	100K	200K	500K	1000K	2000K
FKM	86.5	166.4	391.0	760.8	1491.8	3701.3	7377.6	14702.4	36728.1	73469.0	146844.5
HKM	418.6	420.8	432.9	445.7	467.3	510.4	538.7	577.6	688.4	790.8	1115.9
PKM	9.9	12.39	15.89	17.8	27.58	36.0	48.4	62.5	93.0	127.3	192.7

TABLE IV
TIME COST (SECONDS) ON SCORE VOTING FOR THREE DIFFERENT IMAGE SEARCH SCHEMES

Method	K										
	1K	2K	5K	10K	20K	50K	100K	200K	500K	1000K	2000K
HE	1639.5	1010.3	537.4	420.5	350.8	312.0	308.3	305.8	300.0	298.6	297.8
PQE	569.5	475.2	418.7	406.2	398.4	395.2	393.3	393.3	393.1	392.3	392.3
LSE	2178.7	1259.8	610.1	454.8	361.1	319.0	300.4	295.3	291.6	288.5	287.3

obtained by the proposed PKM method with two subspaces (N = 2) and 2M visual words (K = 2M). And M is set to 16. For the embedding code generation, we uniformly divide each original 128-dimensional vector into 64 segments and generate a 64-b embedding code. Another important parameter is the number of reserved matches after using the embedding codes to prune false matches. In our experiments, it is empirically set to 5, namely, only 5 most possible matches are reserved for each query descriptor. As shown in Table V, the MAP of the proposed method is 0.530, which is much better than 0.306, 0.497 and 0.517 from the state-of-the-art methods. These results again validate that the proposed observations are indeed useful for designing a better image search scheme.

VI. CONCLUSION

In this paper, we systematically investigated the underlying working mechanism of the BoW model and embedding methods, and summarized several observations based on the experiments on descriptor matching. Following these observations, we proposed a new image search scheme, which is jointly optimized in terms of effectiveness and efficiency. Our comprehensive experiments demonstrated that these observations were beneficial to designing new image search schemes. In addition, we also tested the proposed scheme on a very large image dataset in order to demonstrate its scalability. The experimental results showed that the proposed approach outperformed the state-of-the-art methods.

However, while the proposed image search scheme achieved outstanding performance, the issue of memory usage existing in previous work was still not addressed. The inverted table, whose size is closely related to the number of local descriptors and the length of embedding code, cannot be fitted into memory. In the future, we will develop some more compact image representation to address this problem.

REFERENCES

[1] D. Nister and H. Stewenius, "Scalable recognition with a vocabulary tree," in *Proc. IEEE Conf. Comput. Vision Pattern Recognit.*, vol. 5. Oct. 2006, pp. 2161–2168.

[2] M. Broilo and F. De Natale, "A stochastic approach to image retrieval using relevance feedback and particle swarm optimization," *IEEE Trans. Multimedia*, vol. 12, no. 4, pp. 267–277, Jun. 2010.

[3] Z. Lu and H. Ip, "Spatial Markov kernels for image categorization and annotation," *IEEE Trans. Syst., Man, Cybern., B*, vol. 41, no. 4, pp. 976–989, Aug. 2011.

[4] K. Huang, D. Tao, Y. Yuan, X. Li, and T. Tan, "Biologically inspired features for scene classification in video surveillance," *IEEE Trans. Syst., Man, Cybern. B*, vol. 41, no. 1, pp. 307–313, Feb. 2011.

[5] S. Wei, Y. Zhao, Z. Zhu, and N. Liu, "Multimodal fusion for video search reranking," *IEEE Trans. Knowledge Data Eng.*, vol. 22, no. 8, pp. 1191–1199, Aug. 2010.

[6] A. Smeaton and T. Ianeva, "TRECVID-2006 search task," in *Proc. TRECVID*, Nov. 2006, pp.1–40.

[7] E. Nowak, F. Jurie, and B. Triggs, "Sampling strategies for bag-of-features image classification," in *Proc. Eur. Conf. Comput. Vision*, vol. 3954. 2006, pp. 490–503.

[8] S. Wei, Y. Zhao, C. Zhu, C. Xu, and Z. Zhu, "Frame fusion for video copy detection," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 21, no. 1, pp. 15–28, Jan. 2011.

[9] J. Sivic and A. Zisserman, "Video google: A text retrieval approach to object matching in videos," in *Proc. IEEE Int. Conf. Comput. Vision*, vol. 2. Oct. 2003, pp. 1470–1477.

[10] F. Moosmann, W. Triggs, and F. Jurie, "Randomized clustering forests for building fast and discriminative visual vocabularies," in *Proc. Neural Inform. Process. Syst. Conf.*, Nov. 2006, pp.1–7.

[11] J. Philbin, O. Chum, M. Isard, J. Sivic, and A. Zisserman, "Object retrieval with large vocabularies and fast spatial matching," in *Proc. IEEE Conf. Comput. Vision Pattern Recognit.*, vol. 3613. Jun. 2007, pp. 1575–1589.

[12] F. Jurie and B. Triggs, "Creating efficient codebooks for visual recognition," in *Proc. IEEE Int. Conf. Comput. Vision*, vol. 1. Oct. 2005, pp. 604–610.

[13] Y. Hu, X. Cheng, L.-T. Chia, X. Xie, D. Rajan, and A.-H. Tan, "Coherent phrase model for efficient image near-duplicate retrieval," *IEEE Trans. Multimedia*, vol. 11, no. 8, pp. 1434–1445, Dec. 2009.

[14] H. Jegou, M. Douze, and C. Schmid, "Hamming embedding and weak geometric consistency for large scale image search," in *Proc. Eur. Conf. Comput. Vision*, vol. 5302. 2008, pp. 304–317.

[15] H. Jegou, M. Douze, and C. Schmid, "Product quantization for nearest neighbor search," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 33, no. 1, pp. 117–128, Jan. 2011.

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

[17] T. Tuytelaars and C. Schmid, "Vector quantizing feature space with a regular lattice," in *Proc. IEEE Int. Conf. Comput. Vision*, Oct. 2007, pp. 1–8.

[18] Y. Mu, J. Sun, T. Han, L. Cheong, and S. Yan, "Randomized locality sensitive vocabularies for bag-of-features model," in *Proc. Eur. Conf. Comput. Vision*, vol. 6313. 2010, pp. 748–761.

[19] M. Muja and D. Lowe, "Fast approximate nearest neighbors with automatic algorithm configuration," in *Proc. Int. Conf. Comput. Vision Theory Appl.*, 2009, pp. 331–340.
[20] H. Jegou, M. Douze, and C. Schmid, "Improving bag-of-features for large scale image search," *Int. J. Comput. Vision*, vol. 87, nos. 1–2, pp. 316–336, 2010.
[21] V. Athitsos, J. Alon, S. Sclaroff, and G. Kollios, "BoostMap: An embedding method for efficient nearest neighbor retrieval," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 30, no. 1, pp. 89–104, Jan. 2008.
[22] G. R. Hjaltason and H. Samet, "Properties of embedding methods for similarity searching in metric spaces," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 25, no. 5, pp. 530–549, May 2003.
[23] S. Zhang, Q. Huang, G. Hua, S. Jiang, W. Gao, and Q. Tian, "Building contextual visual vocabulary for large-scale image applications," in *Proc. ACM Int. Conf. Multimedia*, 2010, pp. 501–510.
[24] K. Grauman and T. Darrell, "The pyramid match kernel: Discriminative classification with sets of image features," in *Proc. IEEE Int. Conf. Comput. Vision*, vol. 2. Oct. 2005, pp. 1458–1465.
[25] S. Lazebnik, C. Schmid, and J. Ponce, "Beyond bags of features: Spatial pyramid matching for recognizing natural scene categories," in *Proc. IEEE Conf. Comput. Vision Pattern Recognit.*, vol. 2. 2006, pp. 2169–2178.
[26] J. Yang, K. Yu, Y. Gong, and T. Huang, "Linear spatial pyramid matching using sparse coding for image classification," in *Proc. IEEE Conf. Comput. Vision Pattern Recognit.*, Jun. 2009, pp. 1794–1801.
[27] J. Wang, J. Yang, K. Yu, F. Lv, T. Huang, and Y. Gong, "Locality-constrained linear coding for image classification," in *Proc. IEEE Conf. Comput. Vision Pattern Recognit.*, Jun. 2010, pp. 3360–3367.
[28] M. Jain, H. Jegou, and P. Gros, "Asymmetric hamming embedding," in *Proc. ACM Int. Conf. Multimedia*, Nov. 2011, pp. 1441–1444.
[29] H. Jegou, M. Douze, and C. Schmid, "On the burstiness of visual elements," in *Proc. IEEE Conf. Comput. Vision Pattern Recognit.*, Jun. 2009, pp. 1169–1176.
[30] A. Gersho and R. Gray, *Vector Quantization and Signal Compression*. Dordrecht, The Netherlands: Kulwer Academic, 1992.
[31] G. Motta, F. Rizzo, and J. Storer, "Partitioned vector quantization: Application to lossless compression of hyperspectral images," in *Proc. IEEE Int. Conf. Acoust. Speech Signal Process.*, vol. 1. Jul. 2003, pp. 241–244.
[32] H. Jegou, "INRIA Holidays dataset," Oct. 2008 [Online]. Available: http://lear.inrialpes.fr/people/jegou/data.php#holidays
[33] S.K. Wei, X.X Wu, and D. Xu, "Partitioned k-means clustering for fast construction of unbiased visual vocabulary," The Era of Interactive Media, Aug. 2012, pp. 483–493.

Shikui Wei received the M.E. degree and Ph.D. degree in signal and information processing from Beijing Jiaotong University (BJTU), Beijing, China, in 2005 and 2010, respectively.

From 2010 to 2011, he worked as a research fellow in the School of Computer Engineering, Nanyang Technological University, Singapore. He is currently an Associate Professor with the Institute of Information Science, Beijing Jiaotong University, Beijing, China. His research interests include computer vision, image/video analysis and retrieval, and copy detection.

Dong Xu is currently an Associate Professor with the School of Computer Engineering, Nanyang Technological University, Singapore.

Xuelong Li (M'02–SM'07–F'12) is currently a Full Professor with the Center for OPTical IMagery Analysis and Learning, State Key Laboratory of Transient Optics and Photonics, Xi'an Institute of Optics and Precision Mechanics, Chinese Academy of Sciences, Xi'an, Shaanxi, China.

Yao Zhao (M'06-SM'12) received the B.S. degree from Fuzhou University, Fuzhou, China, in 1989 and the M.E. degree from Southeast University, Nanjing, China, in 1992, both from the Radio Engineering Department, and the Ph.D. degree from the Institute of Information Science, Beijing Jiaotong University (BJTU), Beijing, China, in 1996.

He became an Associate Professor at BJTU in 1998 and became a Professor in 2001. From 2001 to 2002, he worked as a Senior Research Fellow in the Information and Communication Theory Group, Faculty of Information Technology and Systems, Delft University of Technology, Delft, The Netherlands. He is now the Director of the Institute of Information Science, BJTU. His research interests include image/video coding, digital watermarking and forensics, and video analysis and understanding. He is now also leading several national research projects from the 973 Program, 863 Program, and National Science Foundation of China.

Dr. Zhao serves on the editorial boards of several international journals, including as an area editor of *Signal Processing: Image Communication* (Elsevier), and as an associate editor of *Circuits, System & Signal Processing* (Springer). Dr. Zhao was named a Distinguished Young Scholar by the National Science Foundation of China in 2010.