Int J Multimed Info Retr (2012) 1:45–58
DOI 10.1007/s13735-012-0012-6

INVITED PAPER

Large-scale near-duplicate image retrieval by kernel density estimation

Wei Tong · Fengjie Li · Rong Jin · Anil Jain

Received: 1 December 2011 / Revised: 6 December 2011 / Accepted: 8 January 2012 / Published online: 3 April 2012

Abstract Bag-of-words model is one of the most widely used methods in the recent studies of multimedia data retrieval. The key idea of the bag-of-words model is to quantize the bag of local features, for example SIFT, to a histogram of visual words and then standard information retrieval technologies developed from text retrieval can be applied directly. Despite its success, one problem of the bag-of-words model is that the two key steps, i.e., *feature quantization* and *retrieval*, are separated. In other words, the step of generating bag-of-words representation is not optimized for the step of retrieval which often leads to a sub-optimal performance. In this paper we propose a statistical framework for large-scale near-duplication image retrieval which unifies the two steps by introducing kernel density function. The central idea of the proposed method is to represent each image by a kernel density function and the similarity between the query image and a database image is then estimated as the query likelihood. In order to make the proposed method applicable to large-scale data sets, we have developed efficient algorithms for both estimating the density function of each image and computing the query likelihood. Our empirical studies confirm that the proposed method is not only more effective but also more efficient than the bag-of-words model.

W. Tong (✉)
Language Technologies Institute,
Carnegie Mellon University, Pittsburgh, USA
e-mail: tongwei@cs.cmu.edu

F. Li · R. Jin · A. Jain
Department of Computer Science and Engineering,
Michigan State University, East Lansing, USA
e-mail: lifengji@cse.msu.edu

R. Jin
e-mail: rongjin@cse.msu.edu

A. Jain
e-mail: jain@cse.msu.edu

Keywords Content-based image retrieval · Near-duplicate image retrieval · Kernel density estimation · Bag-of-words model

1 Introduction

Content-based image retrieval (CBIR) is a long standing challenging problem in multimedia and computer vision. The earliest use of the term content-based image retrieval in the literature seems to have been by Hirata and Kato [12], to describe his experiments into automatic retrieval of images from a database by color and shape feature. The term has since been widely used to describe the process of retrieving desired images from a large collection on the basis of features that can be automatically extracted from the images themselves [5]. The main challenge of CBIR is the semantic gap, i.e., the gap between visual similarity and conceptual/perceptual relevance, which makes it a much harder problem than most researchers anticipated [35]. However, recent studies have shown that near-duplicate image retrieval [15] can be solved effectively using visual features. Unlike general content-based image retrieval that aims to identify images that are semantically relevant to a given query image, the objective of near-duplicate image retrieval is to identify images with high visual similarity (see Fig. 1), thereby avoiding the challenge of semantic gap.

A number of recent studies [15,19,33,38,42,44] have shown that local image features (e.g., SIFT descriptor [23]), often referred to as keypoints, are significantly more effective for near-duplicate image retrieval than global image features such as color [41,45,46], texture [2,14,21,25,47] and shape [28,32]. The main idea of the keypoint-based approach is to extract salient local patches from an image and represent each local patch by a multi-dimensional feature vector.

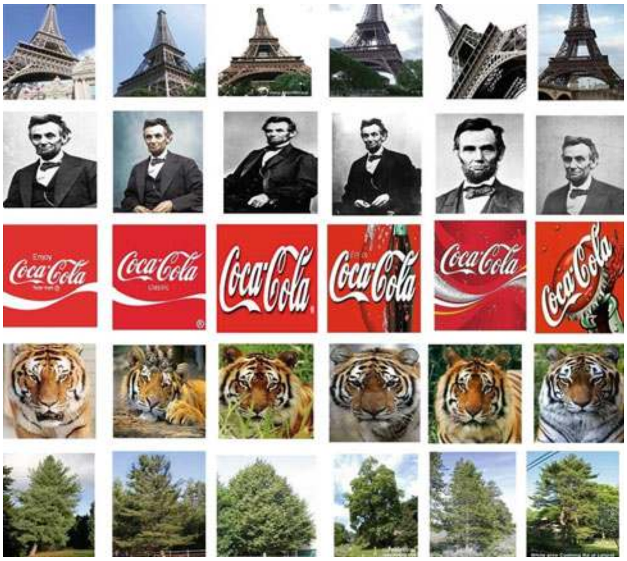

Fig. 1 Examples of near-duplicate image retrieval. The *first column* shows the query images and the subsequent columns are near-duplicate images

As a result, each image is represented by a collection of multi-dimensional vectors, which is often referred to as the *bag-of-features* representation [3].

One straightforward way to measure the distance between two images based on their bag-of-features representations is the optimal partial matching [1,24,50] which finds the best mapping between the keypoints in the two images that has the overall shortest distance. It has been shown [1,9,11,24] that despite its simplicity, the similarity based on the optimal partial matching performs well in comparison with the other similarity measures. The main shortcoming of the optimal partial matching is its high computational cost: given a query image, a linear scan is required to compute the similarity between the query and every image in the database, which does not scale well to a large image database. Several methods have been proposed to improve the computational efficiency of optimal partial matching [7,10,44]. Among them, the bag-of-words model [44] is probably the most popular one due to its empirical success. It takes advantage of the inverted index, which has been successfully used by web search engines to index billions of documents. The key idea is to quantize the continuous high-dimensional space of SIFT features to a vocabulary of visual words, which is typically achieved by a clustering algorithm. By treating each cluster center as a word in a codebook, this approach maps each image feature to its closest visual word and then represents each image by a histogram of visual words. A number of studies have shown promising performance of this approach for image retrieval [15,19,33,38,44] and object recognition [3,6,37,48,51,52]. Despite its success, the bag-of-words model suffers from the following drawbacks:

1. High computational cost in visual vocabulary construction. One of the key steps in constructing the bag-of-words model is to cluster a large number of keypoints into a relatively smaller number of visual words. For large-scale image retrieval, we often need to cluster billions of keypoints into millions of clusters. Although several efficient algorithms [4,8,19,22,31,38,43] have been developed for large-scale clustering problems, it is still expensive to generate a vocabulary with millions of visual words.
2. High computational cost in keypoint quantization. Given the constructed visual vocabulary, the next step in bag-of-words model is to map each keypoint in a database image to a visual word, which requires finding the nearest neighbor of every keypoint to the visual words. Since the computational cost for keypoint quantization is linear in the number of keypoints, it is expensive to quantize keypoints for a very large image database to a visual vocabulary. Even with the help of approximate nearest neighbor search algorithms, this step is still costly when good approximation is desired.
3. Inconsistent mapping of keypoints to visual words. The radius of clusters (i.e., the maximum distance between the keypoints in a cluster and its center) could vary significantly from cluster to cluster. As a result, for clusters with large radius, two keypoints can be mapped to the same visual word even if they differ significantly in visual features, leading to an inconsistent criterion for keypoint quantization and potentially poor performance in image matching.
4. Lack of a theoretic analysis. Most published studies on the bag-of-words model are motivated by efficiency considerations and are primarily focused on its empirical performance. Although [13] showed that the similarity between two bag-of-features representations can be interpreted as a matching algorithm between descriptors, it did not establish the relationship between the bag-of-words model and the optimal partial matching. Without a theoretical analysis, the success of the bag-of-words model may only be demonstrated based on empirical performance.
5. In this paper, we highlight another fundamental problem with the bag-of-words model for image retrieval that is usually overlooked by most researchers. In almost all the methods developed for large-scale image retrieval, the step of *keypoint quantization* is separated from the step of *image matching* that is usually implemented by a text search engine. In other words, the procedure used to quantize keypoints into visual words is independent of the similarity measure used by the text search engine

to find visually similar images which could result in the sub-optimal retrieval performance.

In this paper, we develop a statistical framework that not only overcomes the shortcomings of the bag-of-words model but also unifies the two steps mentioned earlier. The key idea of the proposed method is to view the bag of features extracted from each image as random samples from an underlying unknown distribution. We estimate, for each image, its underlying density function from the observed bag of features. The similarity of an image in the database to a given query image is then computed by the query likelihood, i.e., the likelihood of generating the observed bag of features with the given density function of an image. Thus, the keypoint quantization step is essentially related to the estimation of kernel density function, and the image matching step is essentially related to the estimation of query likelihood. Hence, the introduction of kernel density function allows us to link the two steps coherently.

We emphasize that although the idea of modeling a bag-of-features by a statistical model has been studied by many authors (e.g., [16–18,30,51]), there are two computational challenges that make them difficult to scale to image retrieval problems with large databases:

- How to efficiently compute the density function for each image? This is particularly important given the large size of image database and the large number of keypoints to be processed.
- How to efficiently identify the subset of images in the database that are visually similar to a given query? In particular, the retrieval model should explicitly avoid the linear scan of image database, which is a fundamental problem with many existing methods for image similarity measurements.

We have developed efficient algorithms which solve the two challenges. We verified both the efficiency and efficacy of the proposed framework by an empirical study with three large image databases. Our study shows that the proposed framework reduces the computational time for keypoint quantization by a factor of 8 when compared with the hierarchical clustering methods, and by a factor of 30 when compared with the flat clustering methods. For all the experiments, we observe that the proposed framework yields significantly higher retrieval accuracy than the state-of-the-art approaches for image retrieval.

The rest of the paper is organized as follows: Sect. 2 presents the proposed framework for large-scale near-duplicate image retrieval and efficient computational algorithms for solving the related optimization problems. In Sect. 3 we give a detailed analysis between the proposed method and the bag-of-words model. In Sect. 4 we presents our empirical study with large-scale near-duplicate image retrieval and Sect. 5 concludes this work.

2 Kernel density framework for image retrieval

Let $\mathcal{G} = \{\mathcal{I}_1, \ldots, \mathcal{I}_C\}$ be the collection of C images, and each image $\mathcal{I}_i$ be represented by a set of n_i keypoints $\{\mathbf{x}_1^i, \ldots, \mathbf{x}_{n_i}^i\}$, where each keypoint $\mathbf{x}_i \in \mathbb{R}^d$ is a d dimensional vector. Similarly, the query image $\mathcal{Q}$ is also represented by a bag of features, i.e., $\{\mathbf{q}_1, \ldots, \mathbf{q}_m\}$, where $\mathbf{q}_i \in \mathbb{R}^d$.

To facilitate the development of a statistical model for image retrieval, we assume that keypoints of an image $\mathcal{I}_i$ are randomly sampled from an unknown distribution $p(\mathbf{x}|\mathcal{I}_i)$. Following the framework of statistical language models for text retrieval [27], we need to efficiently compute (1) the density function $p(\mathbf{x}|\mathcal{I}_i)$ for every image $\mathcal{I}_i$ in gallery $\mathcal{G}$, and (2) the query likelihood $p(\mathcal{Q}|\mathcal{I}_i)$, i.e., the probability of generating the keypoints in query $\mathcal{Q}$ given each image $\mathcal{I}_i$. In the following, we discuss the details of the algorithms for the two problems.

2.1 Kernel density based framework

Given the keypoints $\{\mathbf{x}_1, \ldots, \mathbf{x}_n\}$ observed from image $\mathcal{I}$, we need to efficiently estimate its underlying density function $p(\mathbf{x}|\mathcal{I})$. The most straightforward approach is to estimate $p(\mathbf{x}|\mathcal{I})$ by a simple kernel density estimation, i.e.,

$$p(\mathbf{x}|\mathcal{I}) = \frac{1}{n}\sum_{i=1}^{n} \kappa(\mathbf{x}, \mathbf{x}_i) \quad (1)$$

where $\kappa(\cdot, \cdot) : \mathbb{R}^d \times \mathbb{R}^d \mapsto \mathbb{R}_+$ is the kernel density function that is normalized as $\int d\mathbf{z}\kappa(\mathbf{x}, \mathbf{z}) = 1$. Given the density function in (1), the similarity of $\mathcal{I}$ to the query image $\mathcal{Q}$ is estimated by the logarithm of the query likelihood $p(\mathcal{Q}|\mathcal{I})$, i.e.,

$$\log p(\mathcal{Q}|\mathcal{I}) = \sum_{i=1}^{m} \log p(\mathbf{q}_i|\mathcal{I}) = \sum_{i=1}^{m} \log \left(\frac{1}{n}\sum_{j=1}^{n} \kappa(\mathbf{x}_j, \mathbf{q}_i) \right)$$

Despite its simplicity, the major problem with the density function in (1) is its high computational cost when applied to image retrieval. This is because using the density function in (1), we have to compute the log-likelihood $p(\mathcal{Q}|\mathcal{I}_i)$ for every image in $\mathcal{G}$ before we can identify the subset of images that are visually similar to the query $\mathcal{Q}$, making it impossible for large scale image retrieval.

In order to make efficient image retrieval, we consider an alternative approach of estimating the density function for image $\mathcal{I}$. We assume that for any image $\mathcal{I}$ in the gallery $\mathcal{G}$, its density function $p(\mathbf{x}|\mathcal{I})$ is expressed as a weighted mixture models:

$$p(\mathbf{x}|\mathcal{I}) = \sum_{i=1}^{N} \alpha_i \kappa(\mathbf{x}, \mathbf{c}_i) \tag{2}$$

where $\mathbf{c}_i \in \mathbb{R}^d, i = 1, \ldots, N$ is a collection of N points (centers) that are randomly selected from all the keypoints observed in $\mathcal{G}$. The choice of randomly selected centers, although may seem to be naive at the first glance, is in fact strongly supported by the consistency results of kernel density estimation [34]. In particular, the kernel density function constructed by randomly selected centers is almost "optimal" when the number of centers is very large. The number of centers N is usually chosen to be very large, to cover the diverse visual content of images. $\boldsymbol{\alpha} = (\alpha_1, \ldots, \alpha_N)$ is a probability distribution used to combine different kernel functions. It is important to note that unlike (1), the weights $\boldsymbol{\alpha}$ in (2) are unknown and need to be determined for each image. As will be shown later, with an appropriate choice of kernel function $\kappa(\cdot, \cdot)$, the resulting weights $\boldsymbol{\alpha}$ will be sparse with most of the elements being zero. This is ensured by the fact that in a high-dimensional space, almost any two randomly selected data points are far away from each other. It is the sparsity of $\boldsymbol{\alpha}$ that makes it possible to efficiently identify images that are visually similar to the query without having to scan the entire image database.

2.2 Efficient kernel density estimation

In order to use the density function in (2), we need to efficiently estimate the combination weights $\boldsymbol{\alpha}$. By assuming keypoints $\mathbf{x}_1, \ldots, \mathbf{x}_n$ are randomly sampled from $p(\mathbf{x}|\mathcal{I})$, our first attempt is to estimate $\boldsymbol{\alpha}$ by a maximum likelihood estimation, i.e.,

$$\boldsymbol{\alpha} = \underset{\boldsymbol{\alpha} \in \Delta}{\arg\max}\, \mathcal{L}(\mathcal{I}, \boldsymbol{\alpha}) = \sum_{i=1}^{n} \log \left(\sum_{j=1}^{N} \alpha_j \kappa(\mathbf{x}_i, \mathbf{c}_j) \right) \tag{3}$$

where $\Delta = \{\boldsymbol{\alpha} \in [0, 1]^C : \sum_{i=1}^{C} \alpha_i = 1\}$ defines a simplex of probability distributions. It is easy to verify that the problem in (3) is convex and has a global optimal solution.

Although we can directly apply the standard optimization approaches to find the optimal solution $\boldsymbol{\alpha}$ for (3), it is in general computationally expensive because

- We have to solve (3) for every image. Even if the optimization algorithm is efficient and can solve the problem within one second, for a database with a million of images, it will take more than 277 h to complete the computation.
- The number of weights α to be determined is very large. To achieve the desired performance of image retrieval, we often need a very large number of centers, for example, one million. As a result, it requires solving an optimization problem with million variables even for a single optimization problem in (3).

In order to address the computational challenge, we choose the following local kernel function for this study:

$$\kappa(\mathbf{x}, \mathbf{c}) \propto I(|\mathbf{x} - \mathbf{c}|_2 \le \rho) \tag{4}$$

where $I(z)$ is an indicator function that outputs 1 if z is true and zero otherwise. The parameter $\rho > 0$ is a predefined constant that defines the locality of the kernel function and its value is determined empirically. The proposition shown below shows the sparsity of the solution $\boldsymbol{\alpha}$ for (3).

Proposition 1 *Given the local kernel function defined in (4), for the optimal solution $\boldsymbol{\alpha}$ to (3), we have $\alpha_j = 0$ for center $\mathbf{c}_j$ if $\max_{1 \le i \le n} |\mathbf{c}_j - \mathbf{x}_i|_2 > \rho$.*

Proposition 1 follows directly from the fact that $\kappa(\mathbf{c}_j, \mathbf{x}_i) = 0, i = 1, \ldots, n$ if $\max_{1 \le i \le n} |\mathbf{c}_j - \mathbf{x}_i|_2 > \rho$. As implied by Proposition 1, α_j will be nonzero only if the center $\mathbf{c}_j$ is within a distance ρ of some keypoints. By setting ρ to a small value, we will only have a small number of non-zero α_j. We can quickly identify the subset of centers with non-zero α_j by an efficient range search, for example using k-d tree [22]. In our study, this step reduces the number of variables from 1 million to about 1,000.

Although Proposition 1 allows us to reduce the number of variables dramatically, we still have to find a way to solve (3) efficiently. To this end, we resort to the bound optimization strategy that leads to a simple iterative algorithm for optimizing (3): we denote by $\boldsymbol{\alpha}'$ the current solution and by $\boldsymbol{\alpha}$ the updated solution for (3). It is straightforward to show that $\{\mathcal{L}(\mathcal{I}, \boldsymbol{\alpha}) - \mathcal{L}(\mathcal{I}, \boldsymbol{\alpha}')\}$ is bounded as follows:

$$\begin{aligned} \mathcal{L}(\mathcal{I}, \boldsymbol{\alpha}) - \mathcal{L}(\mathcal{I}, \boldsymbol{\alpha}') &= \sum_{i=1}^{n} \log \frac{\sum_{j=1}^{N} \alpha_j \kappa(\mathbf{x}_i, \mathbf{c}_j)}{\sum_{j=1}^{N} \alpha'_j \kappa(\mathbf{x}_i, \mathbf{c}_j)} \\ &\ge \sum_{i=1}^{n} \sum_{j=1}^{N} \frac{\alpha'_j \kappa(\mathbf{x}_i, \mathbf{c}_j)}{\sum_{l=1}^{N} \alpha'_j \kappa(\mathbf{x}_i, \mathbf{c}_l)} \log \frac{\alpha_j}{\alpha'_j} \end{aligned} \tag{5}$$

By maximizing the lower bound in (5), we have the following updating rule for α:

$$\alpha_j = \frac{1}{Z} \sum_{i=1}^{n} \frac{\alpha'_j \kappa(\mathbf{x}_i, \mathbf{c}_j)}{\sum_{l=1}^{N} \alpha'_l \kappa(\mathbf{x}_i, \mathbf{c}_l)} \tag{6}$$

where Z is the normalization factor ensuring $\sum_{j=1}^{N} \alpha_j = 1$. Note that $\boldsymbol{\alpha}$ obtained by iteratively running the updating equation in (6) is indeed globally optimal because the optimization problem in (3) is convex.

We can further simplify the computation of $\boldsymbol{\alpha}$ as following: we first initialize $\alpha_j = 1/N, i = 1, \ldots, N$, and then obtain the solution $\boldsymbol{\alpha}$ by only running the iteration once, i.e.,

$$\alpha_j = \frac{1}{n}\sum_{i=1}^{n}\frac{\kappa(\mathbf{x}_i,\mathbf{c}_j)}{\sum_{l=1}^{N}\kappa(\mathbf{x}_i,\mathbf{c}_l)} \tag{7}$$

We emphasize that although the solution in (7) is approximated in only one update, it is, however, the exact optimal solution when the keypoints $\{\mathbf{x}_i\}_{i=1}^{N}$ are far apart from each other, as shown by the following theorem:

Theorem 1 *Let the kernel function be* (4). *Assume that all the keypoints* $\mathbf{x}_1,\ldots,\mathbf{x}_n$ *are separated by at least* 2ρ. *The solution* α *in* (7) *optimizes the problem in* (3).

Proof When any two keypoints $\mathbf{x}_i$ and $\mathbf{x}_j$ are separated by at least 2ρ, we have $\kappa(\mathbf{x}_i,\mathbf{c}_k)\kappa(\mathbf{x}_j,\mathbf{c}_k) = 0$ for any center $\mathbf{c}_k$. This implies that no keypoint could contribute to the estimation of weight α_k simultaneously for two different centers in (6). As a result, the expression in (6) could be rewritten as

$$\begin{aligned}\alpha_j &= \frac{1}{Z}\sum_{i=1}^{n} I(|\mathbf{x}_i-\mathbf{c}_j|\le\rho)\frac{\alpha'_j}{\sum_{l=1}^{N}\alpha'_l\kappa(\mathbf{x}_i,\mathbf{c}_l)}\\ &= \frac{1}{Z}\sum_{i=1}^{n} I(|\mathbf{x}_i-\mathbf{c}_j|\le\rho)\frac{\alpha'_j}{\alpha'_j\kappa(\mathbf{x}_i,\mathbf{c}_j)}\\ &= \frac{1}{Z}\sum_{i=1}^{n} I(|\mathbf{x}_i-\mathbf{c}_j|\le\rho)\end{aligned}$$

As a result, the updating equation will give the fixed solution, which is the global optimal solution.

In Algorithm 1, we summarize the procedure of computing $\boldsymbol{\alpha}_i$ for each image $\mathcal{I}_i$ in the image collection. The key step of computing each $\boldsymbol{\alpha}_i$ is how to efficiently compute the value of kernel function in (4). In the algorithm, we resort to the k-d tree based range search to achieve the goal. More specifically, we first build a k-d tree for all the keypoints in the collection. For each center, we then search the keypoints which are within the distance ρ of that center using the k-d tree. For all the keypoints that are within the distance ρ of that center, their values to the kernel function (4) for this center is 1 and for other keypoints the value is 0. After we conduct the range search for every centers, we obtain the value of (4) for every pair of keypoints and centers for all the images in the collection which can be used directly for computing $\boldsymbol{\alpha}_i$ for each image.

Algorithm 1 Compute weight vector $\boldsymbol{\alpha}_i$ of each image $\mathcal{I}_i$ in the image collection

```
INPUT:
    – Image collection $\mathcal{G} = \{\mathcal{I}_1,\ldots,\mathcal{I}_C\}$ with each image $\mathcal{I}_i$ represented by a set of keypoints
    – Number of random centers $N$
    – Distance threshold $\rho$
Randomly select $N$ keypoints as the centers $\mathbf{z}_1,\ldots,\mathbf{z}_N$ from $X$ that consists of all the keypoints detected from images in $\mathcal{G}$
Construct a randomized k-d tree $T$ for all the keypoints in $X$
for $l = 1$ to $N$ do
    Using k-d tree $T$, search keypoints which are within the distance $\rho$ of the center $\mathbf{z}_l$.
    For all the returned keypoints, set their values of the kernel function (4) to be 1. For all other keypoints, set the value to be 0.
end for
for $i = 1$ to $C$ do
    Compute each element in the weight vector $\boldsymbol{\alpha}_i$ of image $\mathcal{I}_i$ using (7)
end for
```

2.3 Regularization

Although the sparse solution resulting from the local kernel is computationally efficient, the sparse solution may lead to a poor estimation of query-likelihood, as demonstrated in the study of statistical language model [27]. To address this challenge, we introduce $\boldsymbol{\alpha}^g = (\alpha_1^g,\ldots,\alpha_N^g)$, a global set of weights used for kernel density function. $\boldsymbol{\alpha}^g$ plays the same role as the background langauge model in statistical language models [27]. We defer the discussion of how to compute α^g to the end of this section. Given the global set of weights $\boldsymbol{\alpha}^g$, we introduce $\mathrm{KL}(\boldsymbol{\alpha}^g\|\boldsymbol{\alpha})$, the Kullback–Leibler divergence [30] between $\boldsymbol{\alpha}^g$ and $\boldsymbol{\alpha}$, as a regularizer in (3), i.e.,

$$\boldsymbol{\alpha} = \mathop{\arg\max}_{\boldsymbol{\alpha}\in\Delta} \mathcal{L}(\mathcal{I},\boldsymbol{\alpha}) - \lambda\mathrm{KL}(\boldsymbol{\alpha}^g\|\boldsymbol{\alpha}) \tag{8}$$

where $\lambda > 0$ is introduced to weight the importance of the regularizer. As indicated in (8), by introducing the KL divergence as the regularizer, we prefer the solution $\boldsymbol{\alpha}$ that is similar to $\boldsymbol{\alpha}^g$. Note that (8) is equivalent to the MAP estimation of $\boldsymbol{\alpha}$ by introducing a Dirichlet prior $\mathrm{Dir}(\boldsymbol{\alpha}) \propto \prod_{i=1}^{N}[\alpha_i]^{\beta_i}$, where $\beta_i = \lambda\alpha_i^g$. Similar to the bound optimization strategy used for solving (3), we have the following approximate solution for (8):

$$\alpha_j = \frac{1}{n+\lambda}\left(\lambda\alpha_j^g + \sum_{i=1}^{n}\frac{\kappa(\mathbf{x}_i,\mathbf{c}_j)}{\sum_{l=1}^{N}\kappa(\mathbf{x}_i,\mathbf{c}_j)}\right) \tag{9}$$

It is important to note that, according to (9), the solution for $\boldsymbol{\alpha}$ is no longer sparse if $\boldsymbol{\alpha}^g$ is not sparse, which could potentially lead to a high computational cost in image matching. We will discuss a method later that explicitly addresses this computational challenge.

The remaining question is how to estimate $\boldsymbol{\alpha}^g$, the global set of weights. To this end, we search for the weight $\boldsymbol{\alpha}^g$ that can explain all the keypoints observed in all the images of gallery $\mathcal{G}$, i.e.,

$$\boldsymbol{\alpha}^g = \mathop{\arg\max}_{\boldsymbol{\alpha}^g\in\Delta}\sum_{i=1}^{C}\mathcal{L}(\mathcal{I}_i,\boldsymbol{\alpha}^g) \tag{10}$$

Although we can employ the same bound optimization strategy to estimate $\boldsymbol{\alpha}^g$, we describe below a simple approach that directly utilizes the solution $\boldsymbol{\alpha}$ for individual images to construct $\boldsymbol{\alpha}^g$. We denote by $\boldsymbol{\alpha}^i = (\alpha^i_1, \ldots, \alpha^i_N)$ the optimal solution that is obtained by maximizing the log-likelihood $\mathcal{L}(\mathcal{I}_i, \boldsymbol{\alpha}^i)$ of the keypoints observed in image $\mathcal{I}_i$. Given $\boldsymbol{\alpha}^i$ that maximizes $\mathcal{L}(\mathcal{I}_i, \boldsymbol{\alpha}^i)$, we have

$$\begin{aligned}\mathcal{L}(\mathcal{I}_i, \boldsymbol{\alpha}^g) &\approx \mathcal{L}(\mathcal{I}_i, \boldsymbol{\alpha}^i) \\ &\quad + \frac{1}{2}(\boldsymbol{\alpha}^g - \boldsymbol{\alpha}^i)^\top \nabla^2 \mathcal{L}(I_i, \boldsymbol{\alpha}^i)(\boldsymbol{\alpha}^g - \boldsymbol{\alpha}^i)\end{aligned} \tag{11}$$

Hessian matrix $\nabla^2 \mathcal{L}(\mathcal{I}_i, \boldsymbol{\alpha})$ is computed as $\nabla^2 \mathcal{L}(\mathcal{I}_i, \boldsymbol{\alpha}) = -\sum_{k=1}^{n_i} \mathbf{u}_i^k [\mathbf{u}_i^k]^\top$, where $\mathbf{u}_i^k \in \mathbb{R}^N$ is a vector defined as $[\mathbf{u}_i^k]_j = \kappa(\mathbf{x}_k^i, \mathbf{c}_j)/(\sum_{l=1}^N \alpha_j \kappa(\mathbf{x}_k^i, \mathbf{c}_j))$. The lemma below allows us to bound the Hessian matrix $\nabla^2 \mathcal{L}(\mathcal{I}_i, \boldsymbol{\alpha}^i)$.

Lemma 1 $NI \succeq -\nabla^2 \mathcal{L}(\mathcal{I}_i, \boldsymbol{\alpha}^i)$.

Proof To bound the maximum eigenvalue $-\nabla^2 \mathcal{L}(\mathcal{I}_i, \boldsymbol{\alpha}^i)$, we consider the quantity $\gamma^\top \nabla^2 \mathcal{L}(\mathcal{I}_i, \boldsymbol{\alpha}^i) \gamma$ with $|\gamma|_2 = 1$.

$$\begin{aligned}\gamma^\top \nabla^2 \mathcal{L}(\mathcal{I}_i, \boldsymbol{\alpha}^i)\gamma &= \sum_{k=1}^{n_i} \frac{[\sum_{j=1}^N \gamma_j \kappa(\mathbf{x}_k^i, \mathbf{c}_j)]^2}{[\sum_{j=1}^N \alpha_j \kappa(\mathbf{x}_k^i, \mathbf{c}_j)]^2} \\ &\leq \left(\sum_{k=1}^{n_i} \frac{\sum_{j=1}^N |\gamma_j| \kappa(\mathbf{x}_k^i, \mathbf{c}_j)}{\sum_{j=1}^N \alpha_j \kappa(\mathbf{x}_k^i, \mathbf{c}_j)}\right)^2\end{aligned}$$

Define $\eta_j = |\gamma_j|/(\sum_{j=1}^N |\gamma_j|)$ and $\boldsymbol{\eta} = (\eta_1, \ldots, \eta_N)$. Define $t = \sum_{j=1}^N |\gamma_j|$. We have

$$\gamma^\top \nabla^2 \mathcal{L}(\mathcal{I}_i, \boldsymbol{\alpha}^i)\gamma \leq t^2 \left(\sum_{k=1}^{n_i} \frac{\sum_{j=1}^N \eta_j \kappa(\mathbf{x}_k^i, \mathbf{c}_j)}{\sum_{j=1}^N \alpha_j \kappa(\mathbf{x}_k^i, \mathbf{c}_j)}\right)$$

Since $\boldsymbol{\alpha}^i$ maximizes $\mathcal{L}(\mathcal{I}_i, \boldsymbol{\alpha})$, we have

$$(\boldsymbol{\eta} - \boldsymbol{\alpha}^i)^\top \nabla \mathcal{L}(\mathcal{I}_i, \boldsymbol{\alpha}) \leq 0,$$

which implies

$$\sum_{k=1}^{n_i} \frac{\sum_{j=1}^N \eta_j \kappa(\mathbf{x}_k^i, \mathbf{c}_j)}{\sum_{j=1}^N \alpha_j \kappa(\mathbf{x}_k^i, \mathbf{c}_j)} \leq 1$$

Since $t \leq \sqrt{N}$, we have $\nabla^2 \mathcal{L}(\mathcal{I}_i, \boldsymbol{\alpha}^i) \succeq -NI$.

Using the result in Lemma 1, the objective function in (10) can be approximated as

$$\sum_{i=1}^C \mathcal{L}(\mathcal{I}_i, \boldsymbol{\alpha}^g) \approx \sum_{i=1}^C \mathcal{L}(\mathcal{I}_i, \boldsymbol{\alpha}^i) - \frac{N}{2} \sum_{i=1}^C |\boldsymbol{\alpha}^i - \boldsymbol{\alpha}^g|_2^2 \tag{12}$$

The global weights $\boldsymbol{\alpha}^g$ maximizing (12) is $\boldsymbol{\alpha}^g = \frac{1}{C}\sum_{i=1}^C \boldsymbol{\alpha}^i$ which shows that $\boldsymbol{\alpha}^g$ can be computed as an average of $\{\boldsymbol{\alpha}^i\}_{i=1}^C$ that are optimized for individual images.

2.4 Efficient image search

Given the kernel density function $p(\mathbf{x}|\mathcal{I}_i)$ for each image in gallery $\mathcal{G}$ and a query $\mathcal{Q}$, the next question is how to efficiently identify the subset of images that are likely to be visually similar to the query $\mathcal{Q}$ and furthermore rank those images in the descending order of their similarity. Following the framework of statistical language models for text retrieval, we estimate the similarity by the likelihood of generating the keypoints $\{\mathbf{q}_i\}_{i=1}^m$ observed in the query $\mathcal{Q}$, i.e.,

$$\log p(\mathcal{Q}|\mathcal{I}_i) = \sum_{k=1}^m \log\left(\sum_{j=1}^N \alpha_j^i \kappa(\mathbf{q}_k, \mathbf{c}_j)\right) \tag{13}$$

where $\boldsymbol{\alpha}^i = (\alpha^i_1, \ldots, \alpha^i_N)$ are the weights for constructing the kernel density function for image $\mathcal{I}_i$. Clearly, a naive implementation will require a linear scan of all the images in the database before the subset of similar ones is found. To achieve the efficient image retrieval, we need to exploit the sparse structure of α in (9). We define

$$\widehat{\alpha}_j^i = \frac{1}{n_i}\sum_{k=1}^{n_i} \frac{\kappa(\mathbf{x}_k^i, \mathbf{c}_j)}{\sum_{l=1}^N \kappa(\mathbf{x}_k^i, \mathbf{c}_l)} \tag{14}$$

We then write α_j^i as

$$\alpha_j^i = \frac{\lambda}{n_i + \lambda}\alpha_j^g + \frac{n_i}{n_i + \lambda}\widehat{\alpha}_j^i \tag{15}$$

Note that although $\widehat{\alpha}_j^i$ is sparse, α_j^i is not. Our goal is to effectively explore the sparsity of $\widehat{\alpha}_j^i$ for efficient image retrieval. Using the expression in (15), we have $\log p(\mathcal{Q}|\mathcal{I}_i)$ expressed as

$$\begin{aligned}\log p(\mathcal{Q}|\mathcal{I}_i) &= \sum_{j=1}^m \log\left(\sum_{l=1}^N \left(\frac{\lambda}{n_i + \lambda}\alpha_l^g + \frac{n_i}{n_i + \lambda}\widehat{\alpha}_l^i\right)\kappa(\mathbf{x}_j, \mathbf{c}_l)\right) \\ &= \sum_{j=1}^m \log\left(1 + \frac{n_i}{\lambda}\frac{\sum_{l=1}^N \widehat{\alpha}_l^i \kappa(\mathbf{x}_j, \mathbf{c}_l)}{\sum_{l=1}^N \alpha_l^g \kappa(\mathbf{x}_j, \mathbf{c}_l)}\right) + s_Q\end{aligned} \tag{16}$$

where

$$s_Q = \sum_{j=1}^m \log\left(\frac{\lambda}{n_i + \lambda}\right) + \sum_{j=1}^m \log\left(\sum_{l=1}^N \alpha_l^g \kappa(\mathbf{x}_j, \mathbf{c}_l)\right) \tag{17}$$

Note that (1) the second term of s_Q is independent of the individual images for the same query, and (2) $\log p(\mathcal{Q}|\mathcal{I}_i) \geq s_Q$ for any image $\mathcal{I}_i$. Given the above facts, our goal is to efficiently find the subset of images whose query log-likelihood is *strictly* larger than s_Q, i.e., $\log p(\mathcal{Q}|\mathcal{I}_i) > s_Q$. To this end, we consider the following procedure:

- *Finding the relevant centers $\mathcal{C}_Q$ for a given query $\mathcal{Q}$* Given a query image $\mathcal{Q}$ with keypoints $\mathbf{q}_1, \ldots, \mathbf{q}_m$, we first identify the subset of centers, denoted by $\mathcal{C}_Q$, that are within distance ρ of the keypoints in $\mathcal{Q}$, i.e., $\mathcal{C}_Q = \{\mathbf{c}_j : \exists \mathbf{q}_k \in \mathcal{Q} \text{ s. t. } |\mathbf{q}_k - \mathbf{c}_j|_2 \leq \rho\}$.
- *Finding the candidates of similar images using the relevant centers* Given the relevant centers in $\mathcal{C}_Q$, we find

the subset of images that have at least one non-zero $\widehat{\alpha}^i_j$ for the centers in $\mathcal{C}_Q$, i.e.,

$$\mathcal{R}_Q = \left\{ \mathcal{I}_i \in \mathcal{G} : \sum_{\mathbf{c}_j \in \mathcal{C}_Q} \widehat{\alpha}^i_j > 0 \right\} \tag{18}$$

Theorem 2 shows that all the images with query log-likelihood larger than s_Q belong to $\mathcal{R}_Q$.

Theorem 2 *Let $\mathcal{S}_Q$ denote the set of images with query log-likelihood larger than s_Q, i.e., $\mathcal{S}_Q = \{\mathcal{I}_i \in \mathcal{G} : \log p(\mathcal{Q}|\mathcal{I}_i) > s_Q\}$. We have $\mathcal{S}_Q = \mathcal{R}_Q$.*

It is easy to verify the above theorem. In order to efficiently construct $\mathcal{R}_Q$ (or $\mathcal{S}_Q$) for a given query $\mathcal{Q}$, we exploit the technique of invert indexing [27]: we preprocess the images to obtain a list for each $\mathbf{c}_j$, denoted $\mathcal{V}_j$, that includes all the images $\mathcal{I}_i$ with $\widehat{\alpha}^i_j > 0$. Clearly, we have

$$\mathcal{R}_Q = \bigcup_{\mathbf{c}_j \in \mathcal{C}_Q} \mathcal{V}_j \tag{19}$$

Algorithm 2 summarizes the procedure of efficient image retrieval.

Algorithm 2 Efficient image retrieval algorithm

```
INPUT:
  – A query image $Q$ with keypoints $\mathbf{q}_1, \ldots, \mathbf{q}_m$
  – Inverted indices $\mathcal{V}_j, j = 1, \ldots, N$
  – Number of images to be retrieved, $k$
OUTPUT:
  – $k$ images sorted descendingly by their similarity to the query image
Construct $\mathcal{C}_Q$ of the query $Q$ as
  $\mathcal{C}_Q = \{\mathbf{c}_j : \exists \mathbf{q}_k \in \mathcal{Q} \text{ s. t. } |\mathbf{q}_k - \mathbf{c}_j|_2 \le \rho\}$
Construct candidate image set $\mathcal{R}_Q$ of query $Q$ as
  $\mathcal{R}_Q = \bigcup_{\mathbf{c}_j \in \mathcal{C}_Q} \mathcal{V}_j$
for every image $\mathcal{I}_i$ in $\mathcal{R}_Q$ do
  Compute the likelihood $\log p(\mathcal{Q}|\mathcal{I}_i)$ using (16)
end for
Sort images in $\mathcal{R}_Q$ based on their likelihood $\log p(\mathcal{Q}|\mathcal{I}_i)$
Return the first $k$ images in $\mathcal{R}_Q$
```

3 Comparing with the bag-of-words model

To better understand the proposed method in (2), we compare it with the bag-of-words model. More specifically, we can view each random center $\mathbf{c}_i$ as a different visual word and each $\boldsymbol{\alpha}$ as a histogram vector. One computational advantage of the proposed method is that, while the bag-of-words model requires clustering all the keypoints into a large number of clusters, the proposed method only needs to randomly select a number of keypoints from the database which is computationally efficient. Although recent progress on approximate nearest neighbor search [4,19,22,31,43] has made it feasible to group billions of keypoints into millions of clusters, the computational cost is still very high. We will see this clearly later in our empirical study.

Second, in the bag-of-words model, we need to map each keypoint to the closest visual word(s). Since the computational cost of this procedure is linear in the number of keypoints, it is time consuming when the number of keypoints is very large; the proposed method, however, only needs to conduct a range search for every randomly selected centers and the number of those centers is in general significantly smaller than the number of keypoints, for example, one million centers versus on billion keypoints. This computational saving makes the proposed method more suitable for large image databases than the bag-of-words model.

Third, in the bag-of-words model, the radius of clusters (i.e., the maximum distance between the keypoints in a cluster and its center) could vary significantly from cluster to cluster. As a result, for cluster with large radius, two keypoints can be mapped to the same visual word even if they differ significantly in visual features, leading to an inconsistent criterion for keypoints' quantization and potentially suboptimal performance in retrieval; on the contrary, the proposed method uses a range search for each center which ensures that only "similar" keypoints, which are within the distance of r to the center, will contribute to the corresponding element in the weight $\boldsymbol{\alpha}$ of that center.

Fourth, a keypoint is ignored by the proposed method if its distances to all the centers are larger than the threshold. The underlying rationale is that if a keypoint is far away from all centers, it is very likely to be an outlier and therefore should be ignored, whereas in the bag-of-words model every keypoint must be mapped to a cluster center even if the keypoint is far away from all the cluster centers. We will see this advantage of the proposed method clearly demonstrated in the experiments.

We also noticed that a recently developed random seeding keypoints quantization method [20] for generating the bag-of-words representation utilizes the same randomly sampling and range search strategy as the proposed method. In this random seeding method, a large set of keypoints are first randomly sampled from the whole collection of the keypoints and those keypoints are called seeds. In the next, a range search is performed around each seed to find out which keypoints are within certain range of the seed. If a keypoint is found within the range of a seed, then the keypoint is quantized by that seed. With this simple strategy, the bag-of-words model can be constructed efficiently than using clustering.

It is also clear that the random seeding method has the same advantages of the proposed method over the clustering-based bag-of-words methods mentioned earlier. However, one of the major differences between the proposed method and the random seeding is that in the random seeding method, the keypoints quantization and the image retrieval are still two separated components, while in the proposed method the two steps are unified by the introduction of density function. The second very important advantage of the proposed method over the random seeding method is that in the retrieval step the random seeding method uses the ad-hoc term weighting methods, for example, TF-IDF, while the weighting scheme of the proposed method is integrated into the estimation of the model of each image which is actually decided by a maximum likelihood estimation. It has been proved in the text retrieval that the integrated term weighting scheme is in general superior than the ad-hoc methods [27] and our empirical study in the Sect. 4 also clearly demonstrate that the proposed method outperforms the random seeding method.

4 Experiments

4.1 Datasets

To evaluate the proposed method for large-scale image search, we conduct experiments on three benchmark data sets: (1) tattoo image dataset (**Tattoo**) with about 100,000 images. (2) Oxford building dataset with 5,000 images (**Oxford5K**) [38] and (3) Oxford building dataset plus one million Flickr images (**Oxford5K+Flickr1M**). Table 1 shows the details of the three datasets.

4.1.1 Tattoo image dataset (Tattoo)

Tattoos have been commonly used in forensics and law enforcement agencies to assist in human identification. The tattoo image database used in our study consist of 101, 745 images, among which 61,745 are tattoo images and the remaining 40,000 images are randomly selected from the ESP dataset.[1] The purpose of adding images from the ESP dataset is to verify the capacity of the algorithms in distinguishing tattoo images from the other images. On average, about 100 Harris–Laplacian interesting points are detected for each image, and each keypoint is described by a 128-dimensional SIFT descriptor.

4.1.2 Oxford building dataset (Oxford5K)

The Oxford building dataset consists of 5,062 images. Although it is a small data set, we use it for evaluating the proposed algorithm for image retrieval mainly because it is one of the widely used benchmark datasets. When detecting keypoints for each image, we use both Harris–Laplacian and Hessian–Affine interesting point detectors and each keypoint is described by a 128-dimensional SIFT descriptor. Since the algorithms perform similarly with keypoints detected by the two methods, we only report the results based on the Harris–Laplacian detector. On average, about 3,000 keypoints are detected for each image.

4.1.3 Oxford building dataset plus one million Flickr images (Oxford5K+Flickr1M)

In this dataset, we first crawled Flickr.com to find about one million images of medium resolution and then added them into the Oxford building dataset. The same procedure is applied to extract keypoints from the crawled Flickr images.

4.2 Implementation and baselines

For the implementation of the proposed method, the kernel function (4) is used. The centers for the kernel are randomly selected from the datasets. We employ the FLANN library[2] to perform the efficient range search.

Two clustering-based bag-of-words models are used as baselines. They are hierarchical k-means (**HKM**) implemented in the FLANN library and the approximate k-means (**AKM**) [38] in which the exact nearest neighbor search is replaced by k-d tree based approximate NN search. For HKM the branching factor is set to be 10 based on our experience. For AKM we use the implementation supplied by [38] for approximate nearest neighbor search. A forest of eight randomized k-d trees is used in all experiments. We initialize cluster centers by randomly selecting a number of keypoints in the dataset. The number of iterations for k-means is set to be 10 because we observed that the cluster centers of k-means remains almost unchanged after 10 iterations.

The third baseline used in the empirical study is the random seeding method (**RS**) [20] that we mentioned in the Sect. 3 in which a large set of keypoints are first randomly sampled as seeds and a range search with fixed radius over each seed is then conducted to quantize the keypoints. The bag-of-words model is finally generated using the range search results over the seeds. There are two parameters in the random seeding method: one is the number of seeds and the other is the radius for the range search. Since the random seeding method and the proposed method share the same procedures of the randomly sampling and range search, we use the same parameters as the proposed method which yield the bast performance.

[1] http://www.gwap.com/gwap/gamesPreview/espgame/.

[2] http://www.cs.ubc.ca/~mariusm/index.php/FLANN/FLANN.

Table 1 Statistics of the datasets

Data set	# images	# features	Descriptor size (GB)
Tattoo	101,745	10,843,145	3.4
Oxford5K	5,062	14,972,956	4.7
Oxford5K + Flickr1M	1,002, 805	823,297,045	252.7

For all of the three baseline methods, a state-of-the-art text retrieval method, Okapi BM25 [39] is used to compute the similarity between a query image and images in the gallery given their bag-of-words representations. The inverted indices for both Okapi BM25 and the proposed retrieval model are stored in memory to make the retrieval procedure efficient.

4.3 Evaluation

In order to examine the efficiency of the proposed method, we measure the time spent on preprocessing as well as retrieval stage of the retrieval systems. For the proposed method, the preprocessing stage consists of three steps, i.e., randomly selecting a number of centers, identifying keypoints within the predefined range of the selected centers and computing weights $\boldsymbol{\alpha}$ of every image; for the baseline method **RS**, it is almost the same as the proposed method except without the computation of $\boldsymbol{\alpha}$. For the clustering-based baseline **HKM** and **AKM**, it consists of two steps, constructing visual vocabulary by clustering and mapping keypoints to visual words. In terms of retrieval time, we report the averaged retrieving time of one query for the four methods. We emphasize that besides the retrieval time, the preprocessing time is also very important for an image retrieval system when it comes to a large collection of images and the image collection is updated frequently. Take http://Flickr.com as an example, which is one of the most popular online photo sharing web sites; there are about 900,000 new images uploaded every day [26]. These images must be preprocessed in time to be used for retrieval, which requires the preprocessing of an algorithm be very efficient.

To evaluate the retrieval accuracy of the proposed method, we use two different metrics for the datasets. For tattoo image dataset, the retrieval accuracy is evaluated based on whether a system could retrieve images that share the tattoo symbol as in the query image. We adapt the evaluation metric termed Cumulative Matching Characteristics (CMC) score [29] in this study. For a given rank position k, its CMC score is computed as the percentage of queries whose matched images are found in the first k retrieved images. The CMC score is similar to recall, a common metric used in Information Retrieval. We use CMC score on the tattoo database because it is the most widely used evaluation metric in forensic analysis.

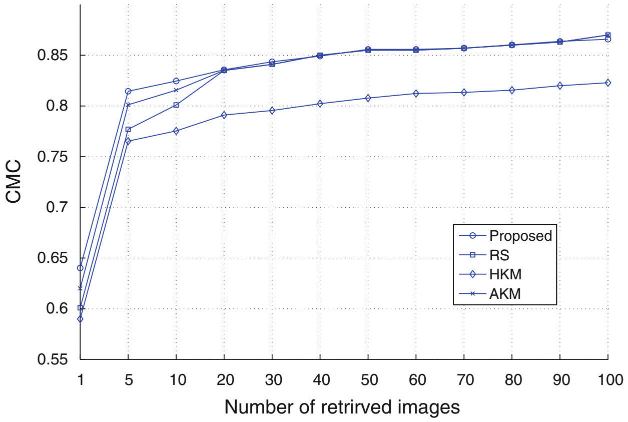

Fig. 2 The CMC scores for tattoo image retrieval with one million cluster/random centers

For the Oxford building dataset and the Oxford building plus Flickr dataset, we follow [38] and evaluate the retrieval performance by Average Precision (AP) which is computed as the area under the precision–recall curve. In particular, an average precision score is computed for each of the five queries from a landmark specified in the Oxford building dataset, and these results are averaged to obtain the mean Average Precision (mAP) for each landmark.

4.4 Results on the tattoo image dataset

We select 995 images as queries and manually identify the gallery images that have the same tattoo symbols as the query images. We randomly select 100 images among the 995 query images and use them to train the optimal values for both λ and ρ. The learned parameter λ and ρ are used for the consequential experiments. The remaining images are used for testing.

We first show the retrieval results of both the proposed method and the baseline methods with the parameters tuned to achieve the best performance and then show the sensitive of the proposed algorithm to the choice of parameters. Figure 2 gives the retrieval performance of the four methods in CMC curves for the first 100 retrieved images. It is clear that the proposed algorithms outperform the baseline methods, especially when the number of retrieved images is small.

The efficiency of the four methods are listed in Table 2. For the preprocessing time, the proposed method is almost

Table 2 The preprocessing and retrieval time of the four methods on tattoo dataset

	Preprocessing time (h)	Retrieval time (s)
Proposed	1.0	0.02
RS	1.0	0.01
HKM	8.8	0.01
AKM	31.1	0.01

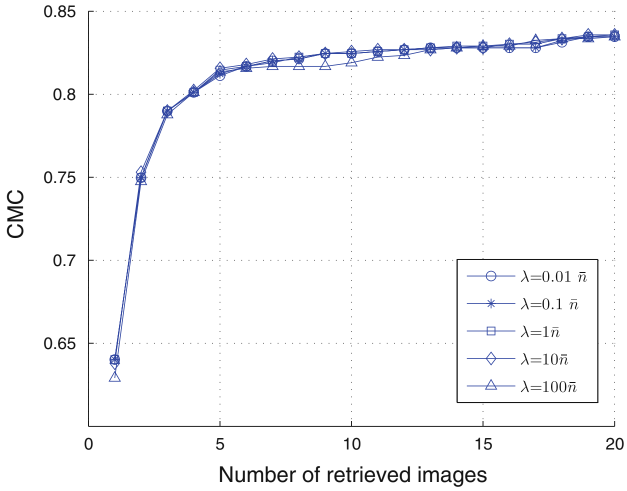

Fig. 3 Results of the proposed method for tattoo image retrieval with different value of λ base on one million random centers with $\rho = 0.6\bar{d}$

the same as random seeding method. Note that, in preprocessing the only difference between the proposed method and the random seeding method is that the proposed method needs to compute $\boldsymbol{\alpha}$ while the random seeding method does not. From this result, we observe the computation of $\boldsymbol{\alpha}$ is very efficient and in general its computational cost can be ignored. This result demonstrates that the proposed method is as efficient as the random seeding method in preprocessing. Comparing with the clustering-based methods, we can clearly observe that both the proposed method and the random seeding method are significantly more efficient which is about 8 times faster than the hierarchical k-means clustering more than 30 times faster than the approximate k-means method.

For the retrieval time, the proposed method is a little bit slower than the three baseline methods. After carefully checking the implementation, we found that the difference in retrieval time is because the logarithm function used by the proposed method in (16) takes a significantly longer time to be computed than the simple addition and multiplication used by baseline methods which is BM25 model. In a real retrieval system, however, this disadvantage can be overcome by some engineering tricks. For example, a logarithm look up table can be built in advance and computing the logarithm of a value can be simplified as checking the lookup table. In fact, this trick is commonly used in the implementation of automatic speech recognition systems.

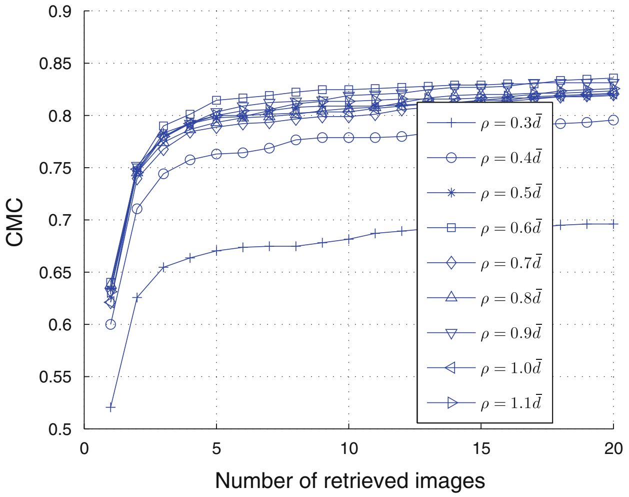

Fig. 4 Results of the proposed method for tattoo image retrieval with different value of ρ base on one million random centers

4.5 Parameter λ

Figure 3 shows the CMC curves of the proposed method with λ varied from $0.01\bar{n}$ to $100\bar{n}$, where $\bar{n}$ is the average number of keypoints in an image. In this experiment, we set the number of random centers to be one million, and ρ to be 0.6 $\bar{d}$, where $\bar{d}$ is the average distance between any two keypoints which is estimated from 1,000 randomly sampled keypoints from the collection. This result shows the performance of the proposed method is overall not sensitive to the choice of λ.

4.6 Parameter ρ

Figure 4 shows the CMC curves of the proposed method with ρ varied from $0.3\bar{d}$ to $1.1\bar{d}$. In this experiment, we again fixed the number of centers to be one million. From the figure we observe that with the exception of the smallest radius ρ (i.e., $r = 0.3\bar{d}$), the retrieval system achieves similar performance for different values of ρ. This indicates that the proposed algorithm is in general insensitive to the choice of ρ as long as ρ is large enough compared with the average inter-points distance between keypoints. This result can be understood by the fact that in a high-dimensional space, most data points are far from each other and as a result, unless we dramatically change the radius ρ, we do not expect the points within a distance ρ of the centers to change significantly.

4.7 Number of random centers

Figure 5 shows the performance of the proposed method with different number of randomly selected centers. The λ and ρ

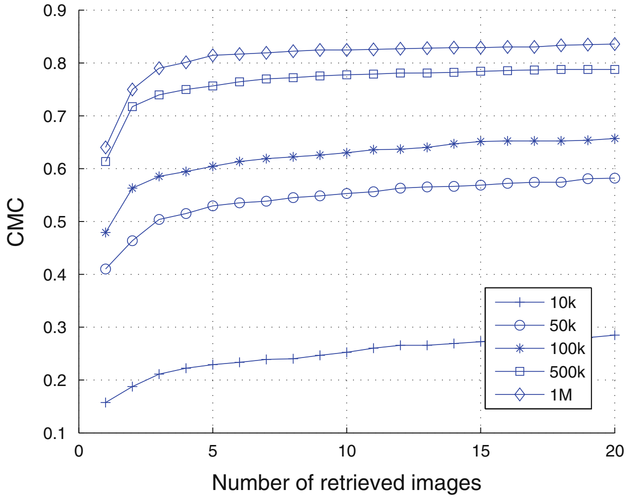

Fig. 5 Results of the proposed method for tattoo image retrieval with different number of centers

Table 3 mAP results of the proposed method and baseline methods for Oxford5K building data set and Oxford5K + Flickr1M data set

	Proposed	RS	HKM	AKM
Oxford5K	0.61	0.57	0.53	0.57
Oxford5K + Flickr1M	0.45	0.43	0.36	0.39

are selected to maximize the performance for the given number of centers. We clearly observe a significant increase in the retrieval accuracy when the number of centers is increased from 10K to 1M. This is not surprising because a large number of random centers usually result in a better discrimination between different SIFT keypoints and consequently lead to an improvement in the detection of similar images. A similar observation is also found when we run our retrieval system using the bag-of-words model approach which is consistent with the observation in [38].

4.8 Results on Oxford building and Oxford building + Flickr datasets

Based on the observation from the experiments of tattoo image retrieval and the similar observation in [38], we use one million cluster/random centers in this experiment. The parameters of the proposed methods are set as the following based on our experiments done with tattoo images. We set $\rho = 0.6\bar{d}$, where $\bar{d}$ is the average inter-points distance that was estimated based on 1,000 randomly sampled pairs. We set the parameter $\lambda = 10\bar{n}$ where $\bar{n}$ is the average number of keypoint in an image.

The mAP results of the proposed method and baseline methods are listed in Table 3. Note that for the Oxford5K + Flickr1M dataset, we follow the experimental protocol in [38] by only using the cluster/random centers that are obtained from the images in the Oxford5K dataset. The results clearly show that the proposed method outperforms baselines.

As expected the performance of the proposed method drops slightly when the 1M Flickr images are added to the Oxford5K dataset. In contrast, the two clustering-based bag-of-words based methods suffer from a significant loss in the performance when we include one million images into the Oxford5K data set. We believe this difference in the performance is due to the fact that the visual content of the one million Flickr images is significantly different from that of the Oxford 5K images, i.e., the keypoints extracted from the Flickr images are generally far away from those in the Oxford5K images. As discussed earlier, the proposed method is robust to the outlying keypoints which makes it less sensitive to the inclusion of the Flickr1M dataset than the clustering-based bag-of-words model. To verify this, we measure the distance between keypoints and centers for both Oxford5k data set and Oxford5k+Flickr1M data set. We find that for the Oxford building images, there are ∼8 % keypoints that are separated from any of the centers by a distance larger than $\rho = 0.6\bar{d}$. This percentage is increased to ∼24 % for the Flickr images, indicating that a large portion of keypoints from the Flickr images are significantly different from the keypoints from the Oxford building images.

In Fig. 6, we show two examples of the queries and the retrieved images. From the retrieval results of the first query, we can clearly observe that the proposed method retrieves images with different patterns to the images retrieved by the three baseline methods. This is mainly because of the different term weighting scheme used in the proposed method as we emphasized in the early sections. For example, because the random seeding method employs different weighting scheme to the proposed method, even if it uses the same sampling and range search procedure as the proposed method, it retrieves quite different images from the proposed method. On the other hand, because the random seeding method uses the same retrieval model, e.g., the same term weighting scheme, as the other two clustering-based methods, even if it uses quite different strategy than the clustering-based methods in generating the bag-of-words representation, they still retrieve images with similar patterns.

The preprocessing and retrieval times of the two algorithms are shown in Table 4. For preprocessing, we split the Oxford5K + Flickr1M dataset into 82 subsets and each subset contains about 10,000,000 keypoints. These 82 subsets are processed separately on multiple machines and are aggregated later to obtain the final result of keypoint quantization. The preprocessing time for Oxford5K + Flickr1M dataset is estimated by the average processing time of each of the 82 subsets. Note that for the Oxford5K + Flickr1M dataset, the preprocessing time of AKM is significantly shorter than HKM. This is because we use the same cluster centers that

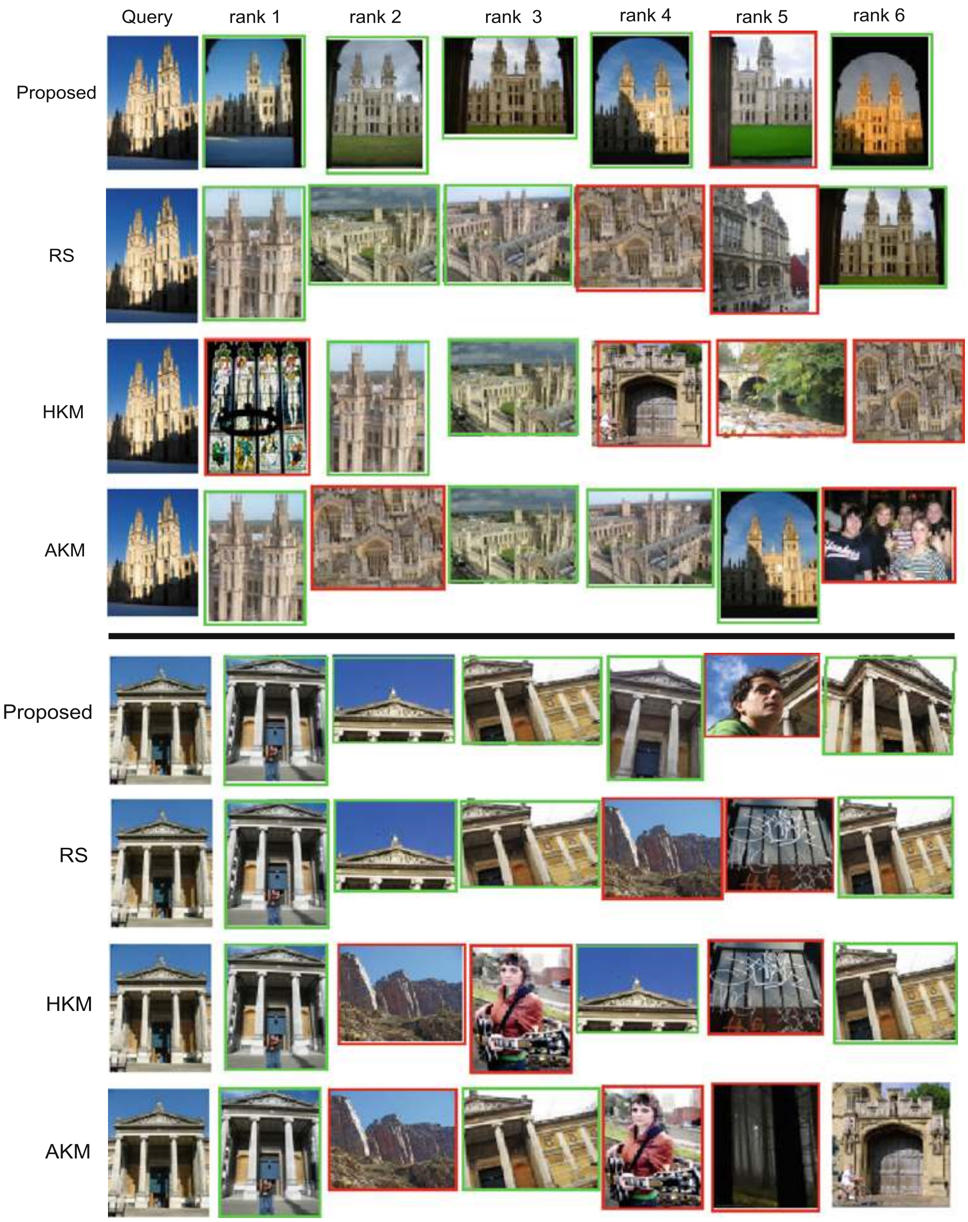

Fig. 6 Examples of two queries (*the first column*) and the first six retrieved images. The *first four rows* give the retrieved results for the Oxford5K building database, and the *next four rows* give the retrieved results for Oxford5K + Flickr1M database. The correctly retrieved results are outlined in *green* and irrelevant images are marked in *red* (color figure online)

Table 4 Preprocessing and retrieval times of the proposed method and the baseline methods with one million cluster/random centers

	Oxford5K		Oxford5K + Flickr1M	
	Preprocess (h)	Retrieval (s)	Preprocess (h)	Retrieval (s)
Proposed	1.1	0.12	95	1.3
RS	1.1	0.07	95	0.68
HKM	11.4	0.08	685	0.84
AKM	36.8	0.08	262	0.89

are generated from the Oxford5K dataset to quantize the keypoints in Oxford5K + Flickr1M dataset. Hence, the processing time for the Oxford5K + Flickr1M dataset only involves finding the nearest neighbor cluster center for each keypoint in the Oxford5K + Flickr1M dataset. We find that the implementation of k-d tree based approximate nearest neighbor search employed in AKM is roughly three times faster than that of HKM, thereby leading to a shorter processing time for AKM than for HKM for the Oxford5K + Flickr1M dataset. From the table, it shows clearly that, for both of the datasets, the proposed method is significantly more efficient, for example, ten times faster, in preprocessing time than the clustering-based methods. Combining the results of preprocessing time from the previous experiment on the tattoo image dataset, we can draw the conclusion that the proposed method is significant more efficient than the clustering-based methods in terms of preprocessing time, which makes it more applicable to large-scale image retrieval.

For the retrieval time, the proposed method is a little bit slower than the baseline methods which is similar to what we have observed on the tattoo dataset. As we discussed earlier, this is mainly due to the slow computation of the logarithm function used by the proposed method in (16) which can be easily overcome by some engineering tricks, for example, a pre-built logarithm look up table.

5 Conclusion and future work

In this paper, we presented a statistical modeling approach for large-scale near-duplicate image retrieval. The key idea of the proposed method is to view the bag of features extracted from each image as random samples from an underlying unknown distribution. More specifically, for each image, we first estimate its underlying density function from the observed bag of features. The similarity between the given query and a data base image is then computed by the query likelihood, i.e., the likelihood of generating the observed bag of features of the query image with the given density function of an database image.

There are two major challenges when applying such idea onto large scale datasets: how to efficiently estimate the density function of keypoint distribution for each image and how to quickly identify the subset of images in the gallery that is visually similar to a given query. We have developed algorithms in this paper which successfully solve these two challenges. Comparing with the widely used clustering-based bag-of-words model, the new method proposed has a couple of advantages.

First, in the bag-of-words model, the step of keypoints quantization and step of image matching are totally separated while in the proposed method these two steps are naturally unified with the introduction of density function of each image. With the unification of the two steps, the whole process can then be optimized which leads to improved retrieval performance.

Second, because of the separation of the keypoint quantization and image marching in the bag-of-words model, it often employs ad-hoc term weighting scheme in the retrieval step, such as TF-IDF. However, in the proposed method the term weighting is embedded into the estimation of query likelihood. In the field of text retrieval, this embedded term weighting scheme, such as statistical language model, has been shown to be more effective than the ad-hoc schemes. Our empirical studies of the proposed method also clearly demonstrate this advantage.

Third, the proposed method is much more efficient in preprocessing the data than the clustering-based bag-of-words model. This is because the proposed method first avoids the step of clustering and simply randomly selects a number of keypoints as centers, which is very efficient. Then the proposed method only conducts the range search over the sampled centers while the clustering-based bag-of-words model needs to do the nearest neighbor search over all the keypoints. Since the number of randomly sampled centers is much smaller than the number of overall keypoints, for example, one millon randomly sampled centers versus one billion keypoints, the proposed method is much more computational efficient.

Fourth, in the clustering-based bag-of-words model, the radius of clusters could vary significantly from cluster to cluster which leads to an inconsistent criterion for keypoints quantization and potentially suboptimal performance in retrieval; on the contrary, the proposed method uses a range search for each center which ensures that only “similar” keypoints will contribute to the corresponding element in the weight $\boldsymbol{\alpha}$ of that center.

Finally, the proposed method is more robust to the outlier keypoints than the clustering-based bag-of-words model. This is because if a keypoint is far away from all centers it is very likely to be an outlier and such keypoints are ignored by the proposed method, whereas in the clustering-based bag-of-words model, even if a keypoint is far away from all the cluster centers, it has to be mapped to a cluster center.

In the future research, we would like to enrich the method developed in this paper along the direction of incorporating the geometric relationship between keypoints. Several recent studies [36,40,49,53–56] have shown that by incorporating the geometric relationship among the keypoints, one can further improve the accuracy of image retrieval. For example, in [40,49], rigid spatial information is embedded by partitioning and quantizing the image space; in [55,56], geometry-preserving visual phrases are introduced to model the co-occurrences of visual words, either in the entire images or in local neighborhoods. It is particularly interesting to notice that the idea of visual phrases in those studies is a very close analogy to the widely used n-gram model in the filed of text retrieval which could gives us a good starting point to develop statistical methods to incorporate geometry information into image retrieval.

References

1. Boughorbel S, Tarel JP, Fleuret F (2004) Non-mercer kernels for svm object recognition. In: BMVC

2. Carson C, Belongie S, Greenspan H, Malik J (1997) Region-based image querying. In: Proceedings of IEEE workshop on content-based access of image and video libraries, pp 42–49
3. Csurka G, Dance C, Fan L, Willamowski J, Bray C (2004) Visual categorization with bags of keypoints. In: Workshop on statistical learning in computer vision
4. Datar M, Immorlica N, Indyk P, Mirrokni VS (2004) Locality-sensitive hashing scheme based on p-stable distributions. In: Proceedings of the twentieth annual symposium on computational geometry
5. Eakins J, Graham M (1999) Content-based image retrieval. Tech. Rep. JTAP-039, Institute for Image Data Research, University of Northumbria Newcastle
6. Fei-Fei L, Perona P (2005) A bayesian hierarchical model for learning natural scene categories. In: CVPR
7. Felzenszwalb PF, Huttenlocher DP (2003) Pictorial structures for object recognition. In: IJCV
8. Gionis A, Indyk P, Motwani R (1999) Similarity search in high dimensions via hashing. In: VLDB
9. Grauman K, Darrell T (2006) Approximate correspondences in high dimensions. In: NIPSnewpage
10. Grauman K, Darrell T (2007) Pyramid match hashing: Sub-linear time indexing over partial correspondences. In: CVPR
11. Grauman K, Darrell T, Perona P (2007) The pyramid match kernel: efficient learning with sets of features. J Mach Learn Res 8:725–760
12. Hirata K, Kato T (1992) Query by visual example—content based image retrieval. In: Third international conference on extending database technology
13. Jegou H, Douze M, Schmid C (2008) Hamming embedding and weak geometric consistency for large scale image search. In: ECCV
14. Kaplan LM, Murenz IR, Namuduri KR (1998) Fast texture database retrieval using extended fractal features. In: Storage and retrieval for image and video databases VI, pp 162–173
15. Ke Y, Sukthankar R, Huston L (2004) Efficient near-duplicate detection and sub-image retrieval. In: ACM Multimedia
16. Kivinen J, Sudderth E, Jordan M (2007) Learning multiscale representations of natural scenes using dirichlet processes. In: ICCV
17. Kondor RI, Jebara T (2003) A kernel between sets of vectors. In: ICML
18. Lazebnik S et al (2003) A sparse texture representation using affine-invariant regions. In: CVPR
19. Lepetit V, Lagger P, Fua P (2005) Randomized trees for real-time keypoint recognition. In: CVPR
20. Li F, Tong W, Jin R, Jain A (2009) An efficient key point quantization algorithm for large scale image retrieval. In: ACM multimedia international conference workshop on large-scale multimedia retrieval and mining
21. Liu F, Picard RW (1996) Periodicity, directionality and randomness: wold features for image modelling and retrieval. IEEE Trans Pattern Anal Mach Intell 18(7):722–733. doi:10.1109/34.506794. http://dx.doi.org/10.1109/34.506794
22. Liu T, Moore A, Gray A, Yang K (2004) An investigation of practical approximate nearest neighbor algorithms. In: NIPS
23. Lowe D (2004) Distinctive image features from scale-invariant keypoints. Int J Comput Vis 60(2):91–110. doi:10.1023/B:VISI.0000029664.99615.94. http://dx.doi.org/10.1023/B:VISI.0000029664.99615.94
24. Lyu S (2005) Mercer kernels for object recognition with local features. In: CVPR
25. Ma WY, Manjunath BS (1998) A texture thesaurus for browsing large aerial photographs. J Am Soc Inf Sci 49(7):633–648
26. Mallapragada PK, Jin R, Jain AK (2010) Online visual vocabulary pruning using pairwise constraints. In: CVPR
27. Manning CD, Raghavan P, Schntze H (2008) Introduction to information retrieval. Cambridge University Press, Cambridge
28. Mehrotra R, Gary JE (1995) Similar-shape retrieval in shape data management. Computer 28(9):57–62. doi:10.1109/2.410154. http://dx.doi.org/10.1109/2.410154
29. Moon H, Phillips PJ (2001) Computational and performance aspects of PCA-based face-recognition algorithms. Perception 30(3):303–321
30. Moreno PJ et al (2003) A Kullback-Leibler divergence based kernel for svm classification in multimedia applications. In: NIPS
31. Muja M, Lowe DG (2009) Fast approximate nearest neighbors with automatic algorithm configuration. In: International conference on computer vision theory and applications
32. Niblack CW, Barber R, Equitz W, Flickner MD, Glasman EH, Petkovic D, Yanker P, Faloutsos C, Taubin G (1993) The qbic project: querying images by color, texture and shape. Tech. Rep. RJ-9203, IBM Research
33. Nister D, Stewenius H (2006) Scalable recognition with a vocabulary tree. In: CVPR
34. Parzen E (1962) On estimation of a probability density function and mode. Ann Math Stat 33(3):1065–1076
35. Pavlidis T (2008) Limitations of cbir. In: ICPR
36. Perdoch M et al (2009) Efficient representation of local geometry for large scale object retrieval. In: CVPR
37. Perronnin F, Dance C, Csurka G, Bressian M (2006) Adopted vocabularies for generic visual categorization. In: ECCV
38. Philbin J, Chum O, Isard M, Sivic J, Zisserman A (2007) Object retrieval with large vocabularies and fast spatial matching. In: CVPR
39. Robertson SE, Walker S, Hancock-Beaulieu M (1998) Okapi at trec-7. In: Proceedings of the seventh text retrieval conference
40. Schmid C (2006) Beyond bags of features: spatial pyramid matching for recognizing natural scene categories. In: CVPR
41. Sebe N, Lew MS (2001) Color-based retrieval. Pattern Recognit Lett 22(2):223–230
42. Shakhnarovich G, Darrell T, Indyk P (2006) Nearest-neighbor methods in learning and vision: theory and practice. MIT Press, Cambridge
43. Silpa-Anan C, Hartley R (2008) Optimised kd-trees for fast image descriptor matching. In: CVPR
44. Sivic J, Zisserman A (2003) Video Google: A text retrieval approach to object matching in videos. In: ICCV
45. Stricker M, Orengo M (1995) Similarity of color images. In: Proceedings of SPIE, vol 2420, pp 381–392
46. Swain MJ, Ballard DH (1991) Color indexing. Int J Comput Vis 7:11–32
47. Tamura H, Mori S, Yamawaki T (1978) Textural features corresponding to visual perception. IEEE Trans Syst Man Cybern
48. Tirilly P, Claveau V, Gros P (2008) Language modeling for bag-of-visual words image categorization. In: CIVR
49. Viitaniemi V, Laaksonen J (2009) Spatial extensions to bag of visual words. In: Proceeding of the ACM international conference on image and video retrieval
50. Wallraven C, Caputo B, Graf A (2003) Recognition with local features: the kernel recipe. In: CVPR
51. Winn J, Criminisi A, Minka T (2005) Object categorization by learned universal visual dictionary. In: ICCV
52. Wu L, Li M, Li Z, Ma W, Yu N (2007) Visual language modeling for image classification. In: CIVR
53. Wu Z et al (2009) Bundling features for large scale partial-duplicate web image search. In: CVPR
54. Yuan J, Wu Y, Yang M (2007) Discovery of collocation patterns: from visual words to visual phrases. In: CVPR
55. Zhang Y, Chen T (2009) Effcient kernels for identifying unbounded-order spatial features. In CVPR
56. Zhang Y, Jia Z, Chen T (2011) Image retrieval with geometry-preserving visual phrases. In: CVPR