A Kernel Density Based Approach for Large Scale Image Retrieval

Wei Tong
Department of Computer Science and Engineering
Michigan State University
East Lansing, MI, USA
tongwei@cse.msu.edu

Fengjie Li
Department of Computer Science and Engineering
Michigan State University
East Lansing, MI, USA
lifengji@cse.msu.edu

Tianbao Yang
Department of Computer Science and Engineering
Michigan State University
East Lansing, MI, USA
yangtian1@cse.msu.edu

Rong Jin
Department of Computer Science and Engineering
Michigan State University
East Lansing, MI, USA
rongjin@cse.msu.edu

Anil Jain
Department of Computer Science and Engineering
Michigan State University
East Lansing, MI, USA
jain@cse.msu.edu

ABSTRACT

Local image features, such as SIFT descriptors, have been shown to be effective for content-based image retrieval (CBIR). In order to achieve efficient image retrieval using local features, most existing approaches represent an image by a bag-of-words model in which every local feature is quantized into a visual word. Given the bag-of-words representation for images, a text search engine is then used to efficiently find the matched images for a given query. The main drawback with these approaches is that the two key steps, i.e., **key point quantization** and **image matching**, are separated, leading to sub-optimal performance in image retrieval. In this work, we present a statistical framework for large-scale image retrieval that unifies key point quantization and image matching by introducing kernel density function. The key ideas of the proposed framework are (a) each image is represented by a kernel density function from which the observed key points are sampled, and (b) the similarity of a gallery image to a query image is estimated as the likelihood of generating the key points in the query image by the kernel density function of the gallery image. We present efficient algorithms for kernel density estimation as well as for effective image matching. Experiments with large-scale image retrieval confirm that the proposed method is not only more effective but also more efficient than the state-of-the-art approaches in identifying visually similar images for given queries from large image databases.

Categories and Subject Descriptors

H.3.3 [**Information Storage and Retrieval**]: Information Search and Retrieval—*Retrieval models*

ICMR '11, April 17-20, Trento, Italy

General Terms

Theory

Keywords

Content-based Image Retrieval, Kernel Density Estimation, Keypoints Quantization

1. INTRODUCTION

Content-based image retrieval (CBIR) is a long standing challenging problem in computer vision and multimedia. Recent studies [18, 24, 15, 8, 21, 4] have shown that local image features (e.g. SIFT descriptor [10]), often referred to as key points, are effective for identifying images with similar visual content. The key idea is to represent each image by a set of "interesting" patches extracted from the image. By representing every image patch with a multi-dimensional vector, each image is represented by a bag of feature vectors, which is referred to as *bag-of-features* representation [1].

One of the major challenges faced in image retrieval using the bag-of-features representation is its efficiency. A naive implementation compares a query image to every image in the database, making it infeasible for large-scale image retrieval. Motivated by the success in text information retrieval [20], the *bag-of-words* representation has become popular for efficient large-scale image retrieval. This approach first quantizes image features to a vocabulary of "visual words", and represents each image by the counts of visual words or a histogram. Standard text retrieval techniques can then be applied to identify the images that share similar visual content as the query image. The quantization is typically achieved by grouping the key points into a specified number of clusters using a clustering algorithm. A number of studies have shown promising performance of the bag-of-words approach for image/object retrieval [18, 24, 15, 8, 22, 21, 4, 25, 3]

Despite its success, there are still drawbacks with most of the studies using the bag-of-words model. For instance, these approaches require clustering all the key points into a large number of clusters, which is computationally expensive when the number of key points is very large. Although recent progress on approximate nearest neighbor search [9, 2, 8, 23, 14] has made it feasible to group billions of key points into millions of clusters, the computational

cost of these approaches in key point quantization is still very high, as will be revealed in our empirical study.

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 *key point 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 key points into visual words is independent of the similarity measure used by the text search engine to find visually similar images. In this paper, we develop a statistical framework that unifies these two steps by the introduction of kernel density function. The key idea 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 $\mathcal{I}_i$ in the database to a given query image $\mathcal{Q}$ is computed by the query likelihood $p(\mathcal{Q}|\mathcal{I}_i)$, i.e., the likelihood of generating the observed bag of features in $\mathcal{Q}$ given the density function of $\mathcal{I}_i$. Thus, the key point 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., [26, 5, 7, 13, 6, 25]), 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 key points 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 address the two challenges by a specially designed kernel density function. We present two efficient algorithms, one for kernel density estimation and one for image search.

Besides providing a unified framework for key point quantization and image matching, the proposed framework also resolves the two shortcomings of the bag-of-words model for image retrieval: (a) by avoiding an explicit clustering of key points, the proposed framework is computationally more efficient. (b) by encoding the observed key points into a kernel density function, the proposed framework allows for partial matching between two similar but different key points.

We verify 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 key point quantization by a factor of 8 when compared to the hierarchical clustering methods, and by a factor of 30 when compared to 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: Section 2 presents the proposed framework for large-scale image retrieval and efficient computational algorithms for solving the related optimization problems. Section 3 presents our empirical study with large-scale image retrieval. Section 4 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 key points $\{\mathbf{x}_1^i, \ldots, \mathbf{x}_{n_i}^i\}$, where each key point $\mathbf{x}_i \in \mathbb{R}^d$ 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 $\mathbf{q}_i \in \mathbb{R}^d$. The objective of image retrieval is two folds:

1. efficiently identify the subset of images $\mathcal{R}$ from the gallery $\mathcal{G}$ that are likely to share similar visual content as that of the query image $\mathcal{Q}$, and
2. effectively rank the images in $\mathcal{R}$ according to their visual similarity to query $\mathcal{Q}$.

We emphasize the importance of the first goal. By efficiently identifying the subset of visually similar images for a given query without going through every image in the database, we are able to build a retrieval model that scale to databases with millions of images.

To facilitate the development of a statistical model for image retrieval, we assume that key points 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 [11], we need to efficiently compute (i) the density function $p(\mathbf{x}|\mathcal{I}_i)$ for every image $\mathcal{I}_i$ in gallery $\mathcal{G}$, and (ii) the query likelihood $p(\mathcal{Q}|\mathcal{I}_i)$, i.e., the probability of generating the key points in query $\mathcal{Q}$ given each image $\mathcal{I}_i$. Below we discuss the algorithms for the two problems.

2.1 Kernel Density Based Framework

Given the key points $\{\mathbf{x}_1, \ldots, \mathbf{x}_n\}$ 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.,

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

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 key points 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 [16]. 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, in order 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 key points $\mathbf{x}_1, \ldots, \mathbf{x}_n$ are randomly sampled from $p(\mathbf{x}|\mathcal{I})$, our first attempt is to estimate $\boldsymbol{\alpha}$ by a maximum likelihood estimation, i.e.,

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

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 hours 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

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

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. Its value is determined empirically, as shown in the experiments. The proposition 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 α to (3), we have $\alpha_j = 0$ for center $\mathbf{c}_j$ if* $\max_{1 \leq i \leq n} |\mathbf{c}_j - \mathbf{x}_i|_2 > \rho$

Proposition 1 follows directly from the fact that $\kappa(\mathbf{c}_j, \mathbf{x}_i) = 0, i = 1, \ldots, n$ if $\max_{1 \leq i \leq n} |\mathbf{c}_j - \mathbf{x}_i|_2 > \rho$. As implied by Proposition 1, α_j will be nonzero only if the center $\mathbf{c}_j$ is within a distance ρ of some key points. 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 conducting an efficient range search. 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

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

By maximizing the upper 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. In our implementation, we initialize $\alpha_j = 1/N, i = 1, \ldots, N$, and 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 key points $\{\mathbf{x}_i\}_{i=1}^{N}$ are far apart from each other, as shown by the following theorem.

THEOREM 1. *Let the kernel function be (4). Assume that all the key points $\mathbf{x}_1, \ldots, \mathbf{x}_n$ 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 key point could make contribution 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

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

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

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 statistical language model [11]. 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 [11]. We defer the discussion of how to compute α^g to the end of this subsection. Given the global set of

weights $\boldsymbol{\alpha}^g$, we introduce $\mathrm{KL}(\boldsymbol{\alpha}^g\|\boldsymbol{\alpha})$, the Kullback-Leibler divergence between $\boldsymbol{\alpha}^g$ and $\boldsymbol{\alpha}$, as a regularizer in (3), i.e.,

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

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)

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

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 high computational cost in image matching. We will discuss a method in the next subsection 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 key points observed in all the images of gallery $\mathcal{G}$, i.e.,

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

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 $\boldsymbol{\alpha}^i = (\alpha_1^i,\ldots,\alpha_N^i)$ the optimal solution that is obtained by maximizing the log-likelihood $\mathcal{L}(\mathcal{I}_i,\boldsymbol{\alpha}^i)$ of the key points observed in image $\mathcal{I}_i$. Given $\boldsymbol{\alpha}^i$ that maximizes $\mathcal{L}(\mathcal{I}_i,\boldsymbol{\alpha}^i)$, we have

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

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 $|\gamma|_2 = 1$.

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

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

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

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

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

which implies

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

Since $t \le \sqrt{N}$, 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.

2.3 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 key points $\{\mathbf{q}_i\}_{i=1}^m$ 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 $\boldsymbol{\alpha}^i = (\alpha_1^i,\ldots,\alpha_N^i)$ 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 can be 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 α_j^i is non-sparse, $\widehat{\alpha}_j^i$ is sparse. 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

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

where

$$s_Q = \sum_{j=1}^{m} \log\left(\frac{\lambda}{n_i+\lambda}\right) + \sum_{j=1}^{m} \log\left(\sum_{l=1}^{N} \alpha_l^g\kappa(\mathbf{x}_j,\mathbf{c}_l)\right) \tag{16}$$

Note that (i) the second term of s_Q is independent of the individual images for the same query, and (ii) $\log p(\mathcal{Q}|\mathcal{I}_i) \ge s_Q$ 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 relevant centers $\mathcal{C}_Q$ for a given query $\mathcal{Q}$.* Given a query image $\mathcal{Q}$ with key points $\mathbf{q}_1, \ldots, \mathbf{q}_m$, we first identify the subset of centers, denoted by $\mathcal{C}_Q$, that are within distance ρ of the key points 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 $\widehat{\alpha}_j^i$ for the centers in $\mathcal{C}_Q$, i.e.,

$$\mathcal{R}_Q = \left\{ \mathcal{I}_i \in \mathcal{G} : \sum_{\mathbf{c}_j \in \mathcal{C}_Q} \widehat{\alpha}_j^i > 0 \right\} \tag{17}$$

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 [11]: 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 $\widehat{\alpha}_j^i > 0$. Clearly, we have

$$\mathcal{R}_Q = \bigcup_{\mathbf{c}_j \in \mathcal{C}_Q} \mathcal{V}_j \tag{18}$$

2.4 Compare to the Bag-of-Words Model

To better understand the proposed kernel density based framework we compare it to the bag-of-words model for image retrieval. In fact, by viewing each random center $\mathbf{c}_i$ as a different visual word, and each $\boldsymbol{\alpha}$ as a histogram vector, we can see a direct correspondence between the bag-of-words model and the proposed framework. However, the kernel density based framework is advantageous in that:

First, it unifies key point quantization and image matching via the introduction of kernel density functions.

Second, the bag-of-words model requires clustering all the keypoints into a large number of clusters, while the proposed method only needs to randomly select a number of points from the date which is much more efficient.

Third, 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 which is in general significantly smaller than the number of key points, for example, one million centers v.s. on billion keypoints. This computational saving makes the proposed method more suitable for large image databases than the bag-of-words model.

Fourth, 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.

Lastly, 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; While 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.

Data set	# images	# features	Size of descriptors
5K	5,062	14,972,956	4.7G
5K+1M	1,002,805	823,297,045	252.7G
tattoo	101,745	10,843,145	3.4G

Table 1: Statistics of the datasets

3. EXPERIMENTS

3.1 Datasets

To evaluate the proposed method for large-scale image search, we conduct experiments on two benchmark data sets: (1) Oxford building dataset with $5,000$ images (**5K**) [18] and (2) Oxford building dataset plus 1 million Flickr images (**5K+1M**). In addition, we also test the proposed algorithm over a tattoo image dataset (**tattoo**) with about $100,000$ images. Table 1 shows the details of the three datasets.

Oxford building dataset (5K).

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. The Harris-Laplacian interesting point detector is used to detect key points for each image, and each key point is described by a 128-dimensional SIFT descriptor. On average, about $3,000$ key points are detected for each image.

Oxford building dataset plus 1 million Flickr images (5K+1M).

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 and represent key points from the crawled Flickr images.

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 developed system in distinguishing tattoo images from the other images. On average, about 100 Harris-Laplacian interesting points are detected for each image, and each key point is described by a 128-dimensional SIFT descriptor.

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

3.2 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. For all the experiments, we set $\rho = 0.6\bar{d}$, where $\bar{d}$ is the average distance of any two key points in the dataset that was estimated based on 1000 randomly sampled pairs. We set the parameter $\lambda = 1\bar{n}$ where $\bar{n}$ is the average number of key point in an image.

Two clustering based bag-of-words models are used as baselines. They are the hierarchical k-means (**HKM**) implemented in the FLANN library and the approximate k-means (**AKM**) [18] 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 [18] for approximate nearest neighbor search. A forest of 8 randomized k-d trees is used in all experiments. We initialize cluster centers by randomly selecting a number of key points 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.

For clustering based methods, a state-of-the-art text retrieval method, Okapi BM25 [19] 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. Overall, the Okapi BM25 method and the proposed method are efficient in finding matched images for a given query. For example, on tattoo image dataset, both the Okapi BM25 model and the proposed retrieval model take about 0.1 second to answer each query.

3.3 Evaluation Metrics

For the Oxford building dataset and the Oxford building plus Flickr dataset, we follow [18] and evaluate the retrieval performace 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 5 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.

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. Since for most query tattoo images, only one or two true matches exist in the database, another evaluation metric, termed Cumulative Matching Characteristics (CMC) score [12], is used 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 face recognition and forensic analysis.

Besides the evaluation of the retrieval results, we also compare the preprocessing time for key point quantization. For our two baselines, this is roughly equal to the time for clustering all key points in the dataset into a large number of clusters. For the proposed method, this is equal to the time for computing the weight vector α for all the images. We emphasize that the preprocessing time is important for a CBIR system when it comes to a large collection with millions of images.

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

	Proposed	HKM	AKM
5K	0.61	0.53	0.57
5K+1M	0.45	0.36	0.39

Table 2: MAP results of the three algorithms with one million cluster/random centers.

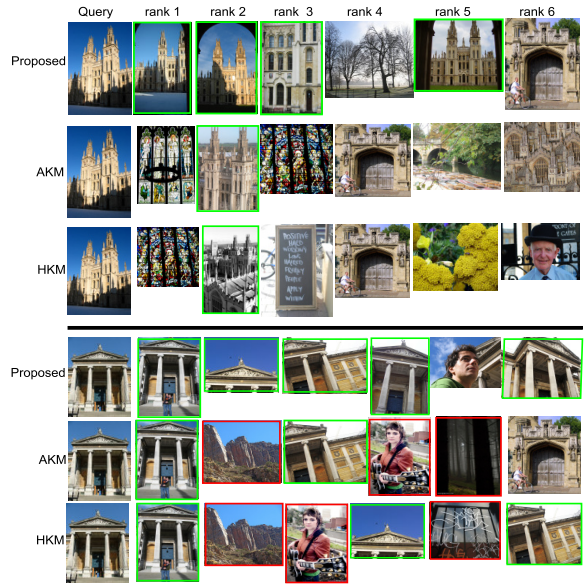

Figure 1: Examples of two queries and the retrieved images ranked from 1-6. The first three rows are based on the 5K dataset, the next three rows are based on the 5K+1M dataset. The correctly retrieved results are outlined in green and the images from the 1M Flickr collection are outlined in red.

3.4 Results on Oxford Building and Oxford Building + Flickr Datasets

Following the settings in [18], the MAP results of the three algorithms with one million cluster/random centers are listed in Table 2. In Figure 1, we show two examples of the queries and the retrieved images. Note that for the 5K+1M dataset, we follow the experimental protocol in [18] by only using the cluster/random centers that are the key points of the images in the 5K dataset. The results clearly show that the proposed method outperforms the two clustering based bag-of-words models.

The preprocessing times of the three algorithms are shown in the first two rows in Table 3. For both the datasets, the proposed method is significantly more efficient than the two clustering based methods. We emphasize that for the 5K+1M dataset, we split it into 82 subsets and each subset contains about 10,000,000 key points. These 82 subsets are processed separately on multiple machines, and are aggregated later to obtain the final result of key point quantization. The preprocessing time for 5K+1M dataset is estimated by the average processing time of each of the 82 subsets. Note that for the 5K+1M dataset, the preprocessing time of AKM is significantly smaller than HKM. This is because we use the same cluster centers that are generated from the 5K dataset to quantize the key points in the 5K+1M dataset. Hence, the processing time for the 5K+1M dataset only involves finding the nearest neighbor cluster center for each key point in the 5K+1M dataset. We find that the implementation of k-d tree based approximate nearest neigh-

	Proposed	HKM	AKM
5K	1.09h	11.4h	36.8h
5K+1M	95h	685h	262h
tattoo	1.02h	8.8h	31.1h

Table 3: Preprocessing time of the three methods with one million cluster/random centers.

	rank 1	rank 2	rank 3	rank 4	rank 5	rank 6	rank 7	rank 8	rank 9	rank 10
Proposed	0.6402	0.7497	0.7899	0.8011	0.8145	0.8168	0.819	0.8224	0.8246	0.8246
AKM	0.6201	0.7575	0.7866	0.7966	0.8011	0.8045	0.8067	0.8123	0.8145	0.8156
HKM	0.5899	0.7117	0.743	0.7587	0.7654	0.7687	0.771	0.771	0.7754	0.7754

Table 4: The CMC scores for tattoo image retrieval with one million cluster/random centers.

bor search employed in AKM is roughly 3 times faster than that of HKM, thereby, leading to a shorter processing time for AKM than for HKM for the 5K+1M dataset.

3.5 Results on Tattoo Image Dataset

We selected 995 images as queries, and manually identified the gallery images that have the same tattoo symbols as the query images. We randomly selected 100 images among the 995 images for training λ and selecting ρ and used the remaining images for test.

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 how sensitive the proposed algorithm is to the choice of parameters.

Table 4 shows the CMC curve for the first 10 retrieval ranking positions. The last row in Table 3 shows the preprocessing time of the three method based on 1 million cluster/random centers. Again, we observe (i) the proposed algorithm outperforms the two clustering based approaches, and (ii) the proposed methods is about 9 times faster than HKM, and 30 times faster than AKM.

Figure 2 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.

Figure 3 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 to 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.

Figure 4 shows the performance of the proposed method with different number of randomly selected centers. The λ and ρ 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

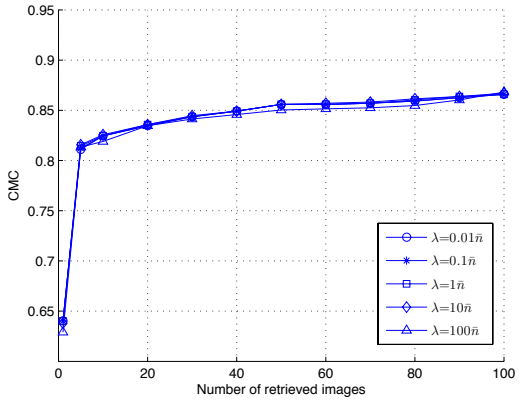

Figure 2: Results of the proposed method for tattoo image retrieval with different value of λ base on 1 million random centers with $\rho = 0.6\bar{d}$

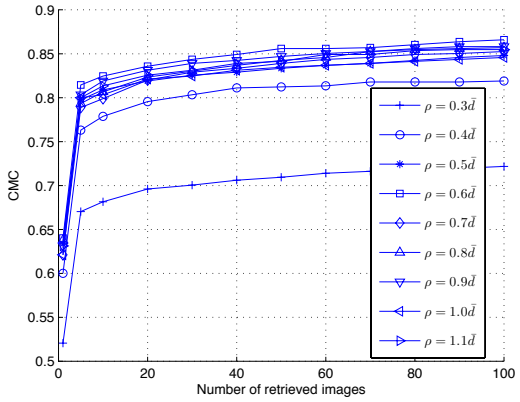

Figure 3: Results of the proposed method for tattoo image retrieval with different value of ρ base on 1 million random centers

results in a better discrimination between different SIFT keypoints and consequently leads 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 [18].

4. CONCLUSIONS AND FUTURE WORK

We have presented a statistical modeling approach for large-scale image retrieval. We developed efficient algorithms for (i) estimating the density function of key point distribution for each individual image, and (ii) identifying the subset of images in the gallery that is visually similar to a given query. Our empirical results on three large-scale image retrieval tasks show that the proposed method is both efficient and effective for identifying images that are visually similar to the query images. This study is limited to developing a statistical model for a bag of features. Several recent studies (e.g. [27, 17]) have shown that by incorporating the geometric relationship among the key points, one can further improve the accuracy of image retrieval. In our future work, we plan to develop statistical approaches that model both a bag of features and their geometric relationship.

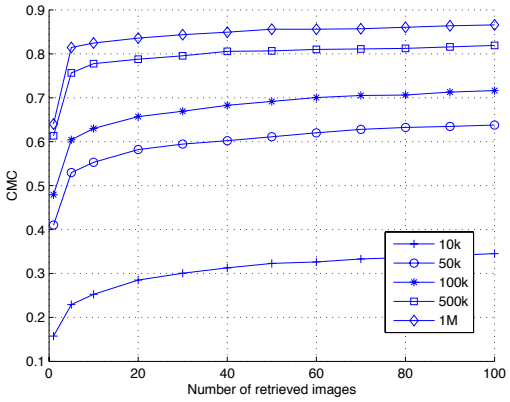

Figure 4: Results of the proposed method for tattoo image retrieval with different number of centers.

5. ACKNOWLEDGMENTS

This research was supported by US Army Research (ARO Award W911NF-08-010403), Office of Naval Research (ONR N00014-09-1-0663) and World Class University (WCU) program through the National Research Foundation of Korea funded by the Ministry of Education, Science and Technology (R31-10008).

6. REFERENCES

[1] G. Csurka and et al. Visual categorization with bags of keypoints. In *Workshop on Statistical Learning in Computer Vision, ECCV*, 2004.

[2] M. Datar and et al. Locality-sensitive hashing scheme based on p-stable distributions. In *SCG*, 2004.

[3] H. Jégou, M. Douze, and C. Schmid. Hamming embedding and weak geometric consistency for large scale image search. In *European Conference on Computer Vision*, 2008.

[4] Y. Ke and et al. Efficient near-duplicate detection and sub-image retrieval. In *Proceeding of ACM Multimedia*, pages 869–876, 2004.

[5] J. Kivinen and et al. Learning multiscale representations of natural scenes using dirichlet processes. In *ICCV*, 2007.

[6] R. I. Kondor and T. Jebara. A kernel between sets of vectors. In *ICML*, 2003.

[7] S. Lazebnik and et al. A sparse texture representation using affine-invariant regions. In *CVPR*, 2003.

[8] V. Lepetit and et al. Randomized trees for real-time keypoint recognition. In *CVPR*, 2005.

[9] T. Liu and et al. An investigation of practical approximate nearest neighbor algorithms. In *NIPS*, 2004.

[10] D. Lowe. Distinctive image features from scale-invariant keypoints. In *IJCV*, 2004.

[11] C. D. Manning and et al. *Introduction to Information Retrieval,*. Cambridge University Press, 2008.

[12] H. Moon and P. J. Phillips. Computational and performance aspects of pca-based face recognition algorithms. *Perception*, 30:303–321, 2001.

[13] P. J. Moreno and et al. A kullback-leibler divergence based kernel for svm classification in multimedia applications. In *NIPS*, 2003.

[14] M. Muja and D. G. Lowe. Fast approximate nearest neighbors with automatic algorithm configuration. In *International Conference on Computer Vision Theory and Applications*, 2009.

[15] D. Nister and H. Stewenius. Scalable recognition with a vocabulary tree. In *CVPR*, 2006.

[16] E. Parzen. On estimation of a probability density function and mode. *Ann. Math. Stat*, 1962.

[17] M. Perdoch and et al. Efficient representation of local geometry for large scale object retrieval. In *CVPR*, 2009.

[18] J. Philbin and et al. Object retrieval with large vocabularies and fast spatial matching. In *CVPR*, 2007.

[19] S. E. Robertson and et al. Okapi at trec-7. In *Proceedings of the Seventh Text REtrieval Conference*, 1998.

[20] G. Salton and M. J. McGill. *Introduction to Modern Information Retrieval*. McGraw-Hill, Inc., 1986.

[21] G. Shakhnarovich and et al. *Nearest-Neighbor Methods in Learning and Vision: Theory and Practice*. MIT Press, 2006.

[22] C. Silpa-Anan and R. Hartley. Localization using an imagemap. In *Proceedings of the 2004 Australasian Conference on Robotics & Automation*, 2004.

[23] C. Silpa-Anan and R. Hartley. Optimised kd-trees for fast image descriptor matching. In *CVPR*, 2008.

[24] J. Sivic and A. Zisserman. Video Google: A text retrieval approach to object matching in videos. In *ICCV*, 2003.

[25] J. C. van Gemert, J. M. Geusebroek, C. J. Veenman, and A. W. M. Smeulders. Kernel codebooks for scene categorization. In *European Conference on Computer Vision*, 2008.

[26] J. Winn and et al. Object categorization by learned universal visual dictionary. In *ICCV*, 2005.

[27] Z. Wu and et al. Bundling features for large scale partial-duplicate web image search. In *CVPR*, 2009.