# Fast Object Retrieval Using Direct Spatial Matching

Zhiyuan Zhong, Jianke Zhu, *Member, IEEE*, and Steven C. H. Hoi, *Senior Member, IEEE*

***Abstract*—The conventional bag-of-visual-words (BoW) model is popular for the large-scale object retrieval system but suffers from the critical drawback of ignoring spatial information. RANSAC-based methods attempt to remedy this drawback, but often require traversing all the feature matches for each hypothesis, leading to the heavy computational cost which limits the number of gallery images to be verified for each online query. We propose an efficient direct spatial matching (DSM) approach to directly estimate the scale variation using region sizes, in which all feature matches voted for estimating geometric transformation. DSM is much faster than RANSAC-based methods and exhaustive enumeration approaches. A logarithmic term frequency-inverse document frequency (log tf-idf) weighting scheme is introduced to boost the performance of the base system. We have conducted extensive experimental evaluations on four benchmark datasets for object retrieval. The proposed DSM method, together with a carefully-tailored reranking scheme, achieves the state-of-the-art results on the Oxford buildings and Paris datasets, which demonstrates the efficacy and scalability of our novel DSM technique for large scale object retrieval systems.**

***Index Terms*—Images reranking, log tf-idf, object retrieval, spatial matching.**

## I. Introduction

THE goal of an object retrieval system is to retrieve the items containing the target object from a large image corpus. A typical object retrieval engine is based on the techniques of matching local features such as SIFT [1] and its various extensions [2], [3].

Although achieving the promising retrieval performance, the bag-of-visual-words (BoW) [4] model suffers the drawback of ignoring spatial information, which is crucial to find object locations in images. Generally, the spatial matching is important to boost the retrieval performance and conduct query expansion on the geometrically verified gallery images.

To this end, Philbin *et al.* [5] proposed a fast spatial matching (FSM) approach based on RANSAC [6], which is currently adopted by many systems [5], [7], [8], [2], [3]. Although FSM can efficiently generate the hypothesis from single pair of correspondence, it remains very time-consuming as it requires to traversing all feature matches to obtain the inliers, where only the top ranked images are considered.

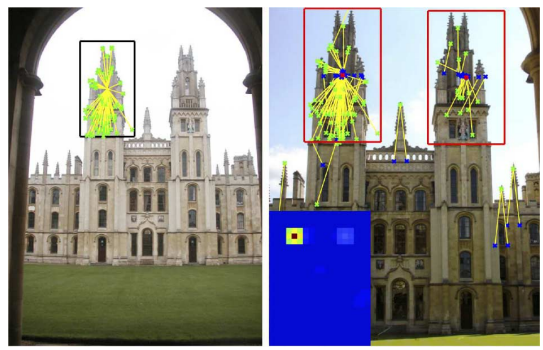


Fig. 1. Locating the near-duplicate structures using direct spatial matching by performing non-maximal suppression on voting map (bottom left).

Recently, Shen *et al.* [9] presented a spatially-constrained similarity measure (SCSM) method that employs Hough voting scheme [10] to simultaneously compute the ranking score and locate the object. It estimates the scale variations by exhaustively enumerating a certain number of scales, which is linear with respect to the computational cost and storage requirement. Furthermore, SCSM may fail to correctly estimate the object whose scale change is out of the scope.

We address the above limitations by taking advantage of a generalized Hough voting scheme [10] with the star model. Inspired by [9], we try to locate the object by searching the peaks in the voting map, as shown in Fig. 1. Moreover, our method only traverses each match once in the inverted file, which is much faster than RANSAC-based methods. Furthermore, it is quite easy to locate duplicate structures by finding multiple peaks using the non-maximal suppression, and Fig. 1 shows a typical example. Instead of exhaustively searching over the enumerated scale variations like SCSM [9], we directly calculate the scale ratio for each matched feature points through the parameters of ellipse shape, which is not limited to certain ranges. To effectively pre-filter the gallery images, we implement a base system using log tf-idf, which performs much better than the standard weighting scheme. Additionally, we suggest a modified $k$-nearest neighbor reranking method [9] and combine it with average query expansion [7] to further boost the object retrieval performance.

In summary, the main contributions of this paper are: (1) we propose a very efficient direct spatial matching (DSM) method for geometric verification, which estimates the local scale variations from ellipse region parameters so as to compute the ranking score and locate the object simultaneously; (2) a log tf-idf weighting scheme is presented to tackle the burstiness

Manuscript received August 26, 2014; revised February 01, 2015 and June 04, 2015; accepted June 05, 2015. Date of publication June 16, 2015; date of current version July 15, 2015. This work was supported by the National Natural Science Foundation of China under Grant 61103105 and Grant 91120302. The associate editor coordinating the review of this manuscript and approving it for publication was Prof. Enrico Magli.

Z. Zhong and J. Zhu are with the College of Computer Science, Zhejiang University, Hangzhou 310027, China (e-mail: zyzhong@zju.edu.cn; jkzhu@zju.edu.cn).

S. C. H. Hoi is with the School of Information System, Singapore Management University, Singapore 178902 (e-mail: chhoi@smu.edu.sg).

This paper has supplementary downloadable multimedia material available at http://ieeexplore.ieee.org provided by the authors. This includes additional tables and figures that are not included within the paper itself. This material is 588 KB in size.

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

Digital Object Identifier 10.1109/TMM.2015.2446201

issue in visual words; (3) we suggest an effective combination of average query expansion and the modified $k$-nearest neighbor reranking; and (4) we have conducted extensive experimental evaluation on a set of benchmark testbeds. We achieve a new state-of-the-art performance on Oxford5K, Oxford105K and Paris dataset. Besides significant performance gain, our main contribution is that the proposed DSM method is much faster than the existing spatial reranking method. To demonstrate the efficacy of our presented method, we make the full source code of our implementation publicly available at https://github.com/jkzhu/dsm.

## II. Related Work

The conventional BoW model [4], [5] is easy to implement and yields promising results in practice. However, it has two limitations: (1) the discriminative power of SIFT descriptors is reduced after quantization and (2) spatial of SIFT points are discarded in the standard tf-idf ranking scheme.

To address the first problem, Jegou *et al.* [11] proposed the VLAD method to quantify the image via the aggregated residuals. Arandjelovic and Zisserman [12] exploited intra normalization to deal with the problem of visual words burstiness [13]. Tolias *et al.* [14] proposed a selective match kernel framework, where Hemming Embedding, BoW and VLAD can be modeled with different kernels. Arandjelovic and Zisserman [2] found that the proper normalization on SIFT descriptor can greatly enhance the retrieval performance.

To tackle the second issue, various methods attempt to impose geometric constraints to rerank the retrieval results, which can be roughly categorized into the following two groups.

The first group performs post verification based on the estimation of geometric transformation. Philbin *et al.* [5] presented a fast spatial matching method (FSM) by RANSAC. Tolias and Avrithis [15] employed a pyramid matching scheme to speed up spatial verification, which is able to rerank one order of magnitude more images than FSM.

The second group of research aims to embed the geometric information into the ranking procedure. Wu *et al.* [16] bundled features into group and encode local spatial configurations in MSER regions. Zhang *et al.* [17] considered the coordinate offset between two matches and code them as geometry-preserving visual phrases. Zhou *et al.* [18] captured the spatial relations of horizontal and vertical directions using binary code. Liu *et al.* [19] presented a coordinate system with the position and orientation of feature points to divide the plane into several parts. However, most of these methods only take into account of the translation invariance while neglecting the scale and rotation invariance. In [20], Stewenius *et al.* proposed a novel scoring method by unifying candidate extraction and geometric verification. Perdoch *et al.* [3] encoded the local affine parameters as integers to obtain efficient geometry representations. Jegou *et al.* [21] included scale and angle in inverted files while ignoring translation. They managed to remove some false matches by embedding the scale and orientation of the extracted features into the inverted file as weak geometry consistency. Additionally, Shen *et al.* [9] employed the offset between the feature points and center of the query rectangle to vote its center in gallery images, where the query time is proportional to the product of selected enum scale and orientation.

Query expansion is able to greatly improve the performance for an object retrieval system [7], [22], [9]. Average query expansion (AQE) intended to find the average BoW representation of the spatial-verified images, which is a good compromise between the accuracy and computational cost [7], [22]. Arandjelovic *et al.* [2] presented discriminative query expansion (DQE) approach by training a linear classifier to identify the results. Shen *et al.* [9] introduced a novel $k$-nearest neighbor ($k$-NN) reranking method, in which the top $k$ spatial-verified images are employed as new queries to calculate the final ranking result. Recently, Xie *et al.* [23] employed link analysis to iteratively conduct reranking.

## III. Experimental Testbed and Base System

In this section, we first describe the experimental dataset and evaluation methodology, and then present the implementation of our base system for object retrieval.

### A. Testbed and Evaluation

We employ four publicly available datasets as testbed: Oxford 5K, Oxford 105K,[1] Paris 6 K,[2] and Holidays.[3] For Holidays dataset, we resize the image into the maximum width or height with 1024 pixels. In our implementation, we extract SIFT features with gravity vector constraints [3]. Although several specific descriptor learning methods [24] improve the performance of baseline system, we exploit the standard fast approximate $k$-means clustering algorithm based on FLANN [25] to build the visual codebook. RootSIFT [2] is employed to boost the retrieval performance. To facilitate fair comparison, we adopt the same setting to build the vocabulary for each dataset. Specifically, 1M visual words are extracted from Oxford5K dataset, which is used in the evaluation for both Oxford5K and Oxford105K dataset. Similarly, we build the vocabulary of 1M visual words for Paris 6 K dataset and 200 K visual words for Holidays dataset, respectively. Since there is no specific input rectangle in Holidays dataset like others, the whole image is directly treated as query. As Holidays dataset includes the rotated images, the gravity vector assumption does not hold. Therefore, we manually rotate the images as in [3]. In our experiments, mean Average Precision (mAP) is employed as the performance metric.

### B. Base System

As discussed in [13], the burstiness of visual words commonly occurs in the image with duplicate structures, which tends to corrupt the visual similarity measure. Most of previous approaches employ standard tf-idf with $L_2$ distance to build the based system, where the words of burstiness may be weighted

[1][Online]. Available: http://www.robots.ox.ac.uk/~vgg/data/oxbuildings

[2][Online]. Available: http://www.robots.ox.ac.uk/~vgg/data/parisbuildings

[3][Online]. Available: http://lear.inrialpes.fr/~jegou/data.php

TABLE I
PERFORMANCE COMPARISON OF VARIOUS BASE SYSTEMS WITHOUT SPATIAL RERANKING. MAP IS EMPLOYED AS THE PERFORMANCE METRIC

| Datasets | tf-idf | log tf-idf | square root | visualindex |
|---|---|---|---|---|
| Oxford5K | 0.771 | **0.814** | 0.803 | 0.750 |
| Oxford105K | 0.640 | **0.755** | 0.726 | 0.612 |
| Paris | 0.751 | **0.782** | 0.770 | - |
| Holidays | 0.708 | 0.723 | **0.728** | - |

too much by directly multiplying the term frequency. To address this problem, we introduce the logarithm tf-idf weighting scheme [26] to the object retrieval task

$$w_{i,j} = \begin{cases} (1 + \log \mathrm{tf}_{i,j}) \log \frac{N}{n_i}, & \text{if } \mathrm{tf}_{i,j} > 0 \\ 0, & \text{otherwise} \end{cases} \quad (1)$$

where $w_{i,j}$ is the weight of word $i$ in images $j$, and $\mathrm{tf}_{i,j}$ is the integer-based term frequency. $N$ is the total number of images, and $n_i$ is the number of images containing word $i$. Note that our down-weighted function is different from the log function in [13]. The proposed log tf-idf is motivated from the sub-linear term frequency scaling in text retrieval [26], which calculate log term frequency before normalization.

Since cosine similarity yields better result in our empirical study, we employ it as distance measure. Both vectors are very sparse, the similarity score can be computed very efficiently.

To reduce the computational cost for the online object retrieval, the term frequency of each word in gallery images is stored in an inverted file. Instead of using four bytes integer, we simply employ unsigned char with single byte to represent a term frequency, as it rarely exceeds 255 even for a large vocabulary. Thus, we can precompute the logarithm value range from 1 to 255, which saves around 1/3 computational time to calculate log term frequency for each visual word. The proposed method can consistently boost the performance of base system without requiring extra computational cost. In our experiments, ranking 105 K images only costs 6 milliseconds by taking advantage of the inverted file structure.

Table I shows the experimental results on the base systems. In [13], square root is used to handle burstiness. Visualindex is provided by Dr. Andrea Vedaldi on Github.[4] It can be seen that our proposed logarithm tf-idf approach outperforms the standard weighting scheme on all datasets. Moreover, the improvement is more significant especially with large dataset, i.e., nearly 18% performance gain achieved on Oxford105K.

## IV. DIRECT SPATIAL MATCHING

To boost the object retrieval performance, the spatial matching is an important technique to verify the relevant images and perform spatial reranking.

### A. Problem Formulation

Spatial matching aims to find the geometric transformation between the query and gallery images. Generally, affine transformation is employed to verify the geometric consistency in object retrieval.

[4][Online]. Available: https://github.com/vedaldi/visualindex>

Let $\mathbf{x} = [\, x \quad y \,]^\top$ denote the location of feature point, where $x$ and $y$ are the image coordinates. To avoid the computational intensive pairwise feature matching, the correspondences are implicitly built by associating the different feature points with the same visual word. Given a pair of matched feature points $\{\mathbf{x}_q, \mathbf{x}_g\}$, the main goal of spatial matching is to estimate the mapping between the point $\mathbf{x}_q$ in query window and $\mathbf{x}_g$ in gallery image $\mathbf{x}_g = \mathbf{M}\mathbf{x}_q + \mathbf{t}$, where $\mathbf{M} \in \mathbb{R}^{2\times 2}$ is the scale and rotation matrix for the matched feature points, and $\mathbf{t}$ is the translation vector.

### B. Fast Spatial Matching

In contrast to the conventional approach, Philbin *et al.* [5] presented a fast spatial matching (FSM) method to estimate the affine transformation from single pair of feature correspondence. It greatly reduces the number of possible hypotheses to be taken into consideration, which is essentially fast in practice. The key of FSM is to fix the orientation ambiguity of affine covariant points by gravity vector assumption [5], [3]. Specifically, a point $\mathbf{u}$ on the ellipse shape of SIFT feature in [3] satisfies the following equation: $(\mathbf{u} - \mathbf{u}_0)^\top \mathbf{E} (\mathbf{u} - \mathbf{u}_0) = 1$, where $\mathbf{u}_0$ is the center of the ellipse and $\mathbf{E} \in \mathbb{R}^{2\times 2}$ is a positive definite matrix. Moreover, the points on ellipse can be mapped onto a unit circle by matrix $\mathbf{A} \in \mathbb{R}^{2\times 2}$ where $\mathbf{E} = \mathbf{A}^\top \mathbf{A}$, and the scale of ellipse is $\frac{1}{\sqrt{\det \mathbf{A}}}$. If $\mathbf{A}$ is a lower triangle matrix, the corresponding affine transformation has one eigenvector equal to $[\, 0 \quad 1 \,]^\top$.

For a pair of matched feature points $\{\mathbf{x}_q, \mathbf{x}_g\}$, $\mathbf{A}_q$ and $\mathbf{A}_g$ are the lower triangle decomposition matrix of their corresponding ellipse shapes, which can be directly computed from the symmetric matrix $\mathbf{E}_q$ and $\mathbf{E}_g$. Let $\mathbf{R}_q$ and $\mathbf{R}_g$ denote the orientation transformation matrix for the pair of matched points, $\mathbf{M}_s$ denotes the scale and rotation matrix which can be estimated with a single pair of matched points, then $\mathbf{M}_s$ can be written as $\mathbf{M}_s = \mathbf{A}_g^{-1} \mathbf{R}_g^\top \mathbf{R}_q \mathbf{A}_q$. According to gravity vector assumption, the up-right orientation can still be preserved without rotation matrix. The gravity vector assumption [5], [3] sets $\mathbf{R}_g^\top \mathbf{R}_q = \mathbf{I}$, then $\mathbf{M}_s$ can be directly calculated by $\mathbf{M}_s = \mathbf{A}_g^{-1} \mathbf{A}_q$, which is a lower triangle matrix. Thus, the translation vector can be computed as $\mathbf{t} = \mathbf{x}_g - \mathbf{M}_s \mathbf{x}_q$.

Since FSM requires to traverse all the matched points in order to calculate the inliers for each hypotheses generated by single pair of feature correspondence, its time complexity is $\mathcal{O}(n^2)$. $n$ is the total number of feature correspondences.

### C. Direct Spatial Matching

To address the above limitations of FSM, we try to tackle the spatial matching problem through generalized Hough voting scheme [10]. Motivated by spatially-constrained similarity measure (SCSM) method [9], we employ a star model to vote the object center along with scale and orientation using the feature correspondences. As shown in Fig. 1, we intend to locate the object in gallery image by estimating its center of rectangle along with the scale and orientation changes.

Let $\mathbf{c}_q$ denote the center of rectangle in the query image. Similarly, $\mathbf{c}_g$ is the center of located object in the gallery image. Therefore, the affine transformation between the centers of two rectangles can be represented as below $\mathbf{c}_g = \mathbf{M}_s \mathbf{c}_q + \mathbf{t}$. As $\mathbf{M}_s$

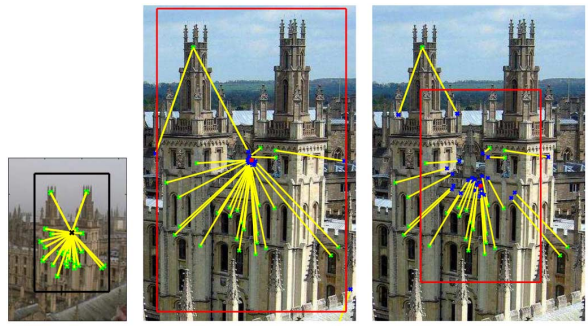
is known, we can directly estimate the object center from single pair of match by $\mathbf{c}_g = \mathbf{x}_g + \mathbf{M}_s(\mathbf{c}_q - \mathbf{x}_q)$.

We rasterize images into an $N \times N$ grid, where $N$ is empirically set to 24 in our implementation. According to our empirical study, the voting cell size for DSM only has the slight impact on performance. More specifically, we have evaluated the several voting cell sizes on Oxford5K dataset: 16, 24, 32 and 48. In our empirical study, their mAPs are quite close: 0.856, 0.851, 0.840 and 0.843, respectively. Furthermore, a voting map is employed to account for all the feature correspondences, in which each pair of co-occurred words in inverted file votes its corresponding node with the value as in [9]: $\frac{\mathrm{idf}^2(i)}{\mathrm{tf}_{i,j} \times \mathrm{tf}_{i,q}}$, where idf is the inverse document frequency, $\mathrm{tf}_{i,j}$ and $\mathrm{tf}_{i,q}$ is standard term frequency for the $j$-th gallery image and query image, respectively. We just enumerate $\mathrm{tf}_{i,j} \times \mathrm{tf}_{i,q}$ possible feature matches if the term frequency is unequal to one. To deal with deformations along with encoding loss and viewpoint changes, we smooth the neighborhood voting cells with Gaussian filter with size of $3 \times 3$. Thus, we vote on a $3 \times 3$ window around the estimated center grid for each matched pair. Moreover, Gaussian weighted function is defined as: $w = \exp(-d/\sigma^2)$, where $d$ is the distance between the estimated center and the voting cell. In this paper, we empirically set $\sigma^2 = 2.5$. Therefore, the voting score for each grid is the initial value multiplied by $w$.

Since $\mathbf{M}$ is a lower triangle matrix, it accounts for the affine transformation with 5 DoF of anisotropic scaling and vertical shear. If only the diagonal elements in $\mathbf{M}$ are used, we can obtain the anisotropic scale affine transformation with 4 DoF. Furthermore, an isotropic scale affine model only takes consideration of the scale changes and translation, which has 3 DoF. As discussed in [5], the number of DoF in affine transformation model has little impact on the object retrieval performance. We draw the similar conclusion in our empirical study on the Oxford5K and Paris dataset.

In the following, we only deal with the isotropic scale change $s$ to estimate the center of object in the gallery image $\mathbf{c}_g = \mathbf{x}_g + s(\mathbf{c}_q - \mathbf{x}_q)$. SCSM [9] estimates the isotropic scale change by enumerating several scales, i.e, 8 scales range from 0.5 to 2, which needs to calculate the voting map for each scale. The computational cost and storage requirement are linear with the total number of enumerated scales. Thus, SCSM method can only handle the limited scale variations. It may fail to correctly estimate the object with the scale changes out of the predefined range, as illustrated in Fig. 2.

Instead of exhaustively searching multiple voting maps, we directly compute $s$ by the ratio between the scale of query point and the one in the gallery image $s = \sqrt{\frac{\det \mathbf{A}_q}{\det \mathbf{A}_g}}$. The object center can be accurately located by mapping the peak in voting map onto the image coordinate. We name our proposed method as direct spatial matching (DSM). Note that only one voting map is needed to estimate the object location in our presented method. Since the scale variations are not bounded within a certain range like SCSM, it can handle large scale changes as illustrated in Fig. 2. In this case, the scale change is around 3.1, which exceed the predefined maximal scale change (typical choice is 2) of SCSM [9]. Fig. 2(b) shows that our DSM method estimates the correct scale change; Fig. 2(c) indicates that the enum scale selection method obtains the scale change 2. As a result, DSM achieves 24 inliers, while the fix scale method get just 15 inliers with the same 31 potential matched points.



(a) (b) (c)

Fig. 2. Examples of geometric transformation estimation using direct spatial matching and multiple scale enumeration SCSM method. (a) Query image. (b) Direct spatial matching. (c) SCSM.

TABLE II
PERFORMANCE EVALUATION ON SPATIAL MATCHING USING VARIOUS SCALE ESTIMATION METHOD

| Dataset | Hesaff [3]+SCSM [9] | | Lowe [1]+DSM | | Hesaff [3]+DSM | |
|---|---|---|---|---|---|---|
| | tf-idf | SCSM | tf-idf | DSM | tf-idf | DSM |
| Oxford5K | 0.649 | 0.752 | 0.685 | 0.746 | 0.771 | **0.850** |
| Paris | 0.630 | 0.741 | 0.681 | 0.725 | 0.751 | **0.814** |
| Holidays | 0.462 | 0.762 | 0.678 | 0.720 | 0.708 | **0.771** |

TABLE III
COMPUTATIONAL TIME OF SPATIAL MATCHING METHODS IN SECONDS

| Dataset | DSM | SCSM[9] | FSM[3] | HPM[27] |
|---|---|---|---|---|
| Oxford5K | **0.012** | 0.089 | 0.238 | 0.210 |
| Oxford105K | **0.090** | - | 0.247 | - |

To estimate the scale change between two windows in the query and gallery images, we introduce a simple voting method based on histogram, which votes to select the most frequent scale change. The scale is first enlarged by a factor of 10, which is further rounded to the nearest integer. Thus, the precision for the scale is 0.1. Similarly, we can deal with orientation changes by building a histogram for the rotations.

### D. Evaluation on Spatial Matching

As the proposed DSM approach relies on the scale estimation of the specific feature detector, we first investigate the retrieval performance of three different SIFT implementations: (1) the baseline algorithm using Lowe's SIFT binary [1]; (2) SIFT detector implementation in [9]; (3) Hessian affine regions with gravity vector assumption [3]. Table II shows the experimental evaluation. SCSM [9] refers to the exhaustive numeration method; Lowe [1] + DSM employs the scale information in SIFT detector; Hesaff [3] + DSM uses the SIFT feature with gravity vector assumption. It can be seen that our proposed DSM method with Hessian affine region detector outperforms Lowe's SIFT detector at a very large margin. DSM not only yields better results than SCSM but also requires much less memory for storing the voting map. As shown in Table III, DSM is around 8 times faster than SCSM by directly estimating the scale changes. The FSM method achieves 0.83 mAP in visualindex implementation.

TABLE IV
PERFORMANCE COMPARISONS OF SPATIAL MATCHING METHODS

| Dataset | DSM | SPAUG[2] | SCSM[9] | FSM[3] | HPM[27] |
|---|---|---|---|---|---|
| Oxford5K | **0.850** | 0.838 | 0.752 | 0.786 | 0.789 |
| Oxford105K | **0.836** | 0.767 | 0.729 | 0.723 | 0.730 |
| Paris | **0.814** | - | 0.741 | - | 0.725 |
| Holidays | 0.771 | - | 0.762 | 0.765 | **0.79** |

In contrast to FSM with RANSAC, the proposed method can directly traverse the relevant items in the inverted file. Generally, it is inefficient to vote the images with few co-occurrence word especially for the large dataset. In our implementation, we first pre-filter the relevant images using the effective log tf-idf method described in Section III. DSM turns out to be a post verification step for object retrieval. Our empirical study on the Oxford105K dataset shows that the proposed scheme performs two times faster than the method without pre-filtering. Also, there is no noticeable performance drop.

We conduct the performance comparisons on the methods without query expansion, as shown in Table IV. It can be observed that our proposed DSM method yields the best results on the Oxford5K, Oxford105K and Paris datasets. Especially, DSM outperforms about 7% against the state-of-the-art result [2] on Oxford105K, which takes advantage of FSM with the spatial database-side feature augmentation (SPAUG). Moreover, there is only slight performance drop for the proposed DSM approach on the Oxford dataset with extra 100 K distractive images, which demonstrates the robustness and scalability of our proposed DSM approach.

We compare the computational time of various spatial matching methods, as shown in Table III. The evaluations on DSM were conducted on a PC with 3.4 GHz CPU. We employ a single-threaded implementation, and calculate the average query time over all 55 queries of Oxford5K. DSM reranks about 2,800 images in 0.012s on Oxford5k, and about 18,500 images in 0.09s on Oxford105K. In our implementation the number of images to be reranked can be adjusted according to the tf-idf ranking scores, which is more robust than the reranking scheme using the fixed length of short list. HPM [27] reranks 1,000 images on Oxford5k in 0.21s. It takes 6 milliseconds to calculate the ranking list using log tf-idf score on Oxford105K. It can be seen that DSM is the most efficient approach. Given $n$ pairs of matches, the time complexity of FSM algorithm is $\mathcal{O}(n^2)$. The proposed DSM method does not need to verify the inliers for each hypothesis generated by feature matching, while only takes $\mathcal{O}(n)$ time to locate the object center and estimate the scale parameter. Typically, FSM with early aborting scheme and HPM deal with 1,000 images for the spatial verification and reranking. DSM is faster than FSM and HPM, which thus can process much more images.

More importantly, our proposed DSM approach can naturally handle the case of multiple models, which is usually difficult for RANSAC-based methods like FSM. We employ the standard non-maximal suppression method to find out the object center for each model, and term frequency is treated as the confidence measure. Moreover, the scale variation for each model is estimated by calculating the mean scale ratio of the inlier votes, as shown in Fig. 1.

TABLE V
PERFORMANCE EVALUATION ON QUERY EXPANSION STRATEGIES

| Dataset | DSM | AQE | DQE[2] | $k$-NN | AQE+$k$-NN | $k$-NN$^2$ |
|---|---|---|---|---|---|---|
| Oxford5K | 0.850 | 0.928 | 0.929 | 0.932 | **0.951** | 0.950 |
| Oxford105K | 0.836 | 0.883 | 0.891 | 0.915 | 0.924 | **0.932** |
| Paris | 0.814 | 0.891 | 0.910 | 0.896 | 0.912 | **0.915** |

## V. QUERY EXPANSION

In this section, we study the query expansion to further enhance the object retrieval performance.

### A. $k$-NN Reranking With AQE

Among various query expansion techniques, we mainly investigate two representative methods: average query expansion (AQE) [7] and $k$ nearest neighbor ($k$-NN) reranking method [9].

Generally, AQE is a good choice to rerank the retrieval results [7] due to its simplicity and high efficiency. Specifically, AQE employs the top $m$ spatial-verified results to form a new query, which is the average BoW representation of the original query and the top $m$ results. In our implementation, $m$ is usually less than 50. Moreover, the voting value without idf weighting can be viewed as the confidence to determine the inlier match. Practically, we view the image as relevant if such value is greater than or equal to 4.

Given a query $q$ and a set of gallery images $G$, $R(q, G)$ denotes the rank of images in $G$ for the query $q$. $k$-NN re-ranking issues the query with the top $k$ spatially verified images and calculates the new ranking score [9] as follows:

$$F(q,G) = \frac{1}{(c+1)R(q,G)} + \sum_{i=1}^{k} \frac{1}{(i+c+1)R(N_i,G)} \quad (2)$$

where $N_i$ is the $i$-th result in the original rank list. Note that Shen *et al.* [9] just consider the case of $c = 0$. However, this will lead to the fact that the contribution of $R(N_1, D)$ accounts for half weight of $R(q, D)$. If $i = 1$, then the denominators are 1 for $R(q, G)$ and 1/2 for $R(N_1, G)$, respectively. $k$-NN may not be able to decrease the ranking of the negative results. Therefore, we may find a proper value of $c$, which can assign the reasonable weight to the ranking score. In our empirical study, $k$-NN reranking with $c = 0$ always obtains the worst result, and the overall retrieval performance is improved when $c$ is greater than 2.

As described in [7], the better retrieval results can be obtained by recursively conducting reranking on the previous results. Therefore, we perform $k$-NN twice in order to achieve better performance than [9]. However, it is very time consuming to estimate the object location using the spatial matching. Recursive reranking using $k$-NN will involve the heavy computational cost. On the other hand, AQE method does not require the post spatial verification and takes consideration of the top ranked images only. This makes it extremely fast in practice. Thus, we propose to perform $k$-NN reranking on the AQE results to obtain the better retrieval performance with less computational efforts.

### B. Evaluations on Query Expansion

Tables V and VI show the object retrieval performance and search time with various reranking methods on the experimental

TABLE VI
RANKING TIME OF QUERY EXPANSION STRATEGIES IN SECONDS

| Dataset | DSM | AQE | $k$-NN | AQE+$k$-NN | $k$-NN+$k$-NN |
|---|---|---|---|---|---|
| Oxford5K | **0.012** | 0.020 | 0.296 | 0.312 | 0.602 |
| Oxford105K | **0.090** | 0.131 | 2.071 | 2.130 | 4.101 |
| Paris | **0.018** | 0.027 | 1.184 | 1.208 | 2.302 |

TABLE VII
PERFORMANCE COMPARISONS WITH THE STATE-OF-THE-ART METHODS

| Dataset | Ours | [2] | [9] | [3] | [24] | [28] |
|---|---|---|---|---|---|---|
| Oxford5K | **0.950** | 0.929 | 0.884 | 0.900 | 0.849 | 0.850 |
| Oxford105K | **0.932** | 0.891 | 0.864 | 0.856 | - | 0.816 |
| Paris | **0.915** | 0.910 | 0.911 | - | 0.824 | 0.855 |

testbed. Since most of queries in Holidays dataset only have one or two relevant images, its query expansion result is not included. DSM denotes reranking using direct spatial matching. '$\text{AQE}+k\text{-NN}$' means $k$-NN after AQE. '$k\text{-NN}^2$' denotes recursively conducting $k$-NN reranking twice. It can be observed that $k$-NN reranking performs better than AQE while requiring much more computational cost on spatial matching. Additionally, both recursively conducting $k$-NN twice and combing AQE with $k$-NN perform better than other reference methods. Furthermore, $k$-NN after AQE achieves similar results with around half computational time compared to the recursive $k$-NN.

We compare our proposed DSM approach with query expansion against the previous methods. As shown in Table VII, it can be seen that our method achieves a new state-of-the-art performance on three popular datasets for object retrieval including Oxford5K, Oxford105K and Paris. Our proposed approach outperforms SCSM [9] on Oxford5K and Oxford105K dataset at a large margin, which indicates that our presented scale estimation scheme is more effective than the exhaustive search. Note that the evaluation result of our method on Oxford105K is even better than the previous best record on Oxford5K [2] without extra 100 K distractive images, which demonstrates the effectiveness and scalability of our proposed spatial matching and query expansion methods.

## VI. CONCLUSION AND FUTURE WORK

This paper investigated the efficient spatial matching method and reranking scheme for object retrieval tasks. Instead of exhaustively enumerating the most possible scale changes in the conventional Hough voting scheme, we directly estimated the affine transformation by taking advantage of the ellipse region using the gravity vector assumption. The proposed direct spatial matching approach is much faster than the previous methods while retaining the high retrieval accuracy. Moreover, it can naturally handle the multiple model fitting by finding peaks in voting map. Furthermore, we introduced a log tf-idf weighting scheme to build the base system, which outperforms the standard scheme without extra cost. Finally, we obtained the state-of-the-art object retrieval results by combining average query expansion with $k$-NN reranking.

## REFERENCES


[1] D. Lowe, "Distinctive image features from scale-invariant key-points," *Int. J. Comput. Vis.*, vol. 60, no. 2, pp. 91–110, 2004.

[2] R. Arandjelović and A. Zisserman, "Three things everyone should know to improve object retrieval," in *Proc. IEEE Conf. Comput. Vis. Pattern Recog.*, Jun. 2012, pp. 2911–2918.

[3] M. Perdoch, O. Chum, and J. Matas, "Efficient representation of local geometry for large scale object retrieval," in *Proc. IEEE Conf. Comput. Vis. Pattern Recog.*, Jun. 2009, pp. 9–16.

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

[5] J. Philbin, O. Chum, M. Isard, J. Sivic, and A. Zisserman, "Object retrieval with large vocabularies and fast spatial matching," in *Proc. IEEE Conf. Comput. Vis. Pattern Recog.*, Jun. 2007, pp. 1–8.

[6] M. A. Fischler and R. C. Bolles, "Random sample consensus," *Commun. ACM*, vol. 24, no. 6, pp. 381–395, 1981.

[7] O. Chum, J. Philbin, J. Sivic, M. Isard, and A. Zisserman, "Total recall: Automatic query expansion with a generative feature model for object retrieval," in *Proc. IEEE 11th Int. Conf. Comput. Vis.*, Oct. 2007, pp. 1–8.

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

[9] X. Shen, Z. Lin, J. Brandt, S. Avidan, and Y. Wu, "Object retrieval and localization with spatially-constrained similarity measure and k-NN re-ranking," in *Proc. IEEE Conf. Comput. Vis. Pattern Recog.*, Jun. 2012, pp. 3013–3020.

[10] B. Leibe, A. Leonardis, and B. Schiele, "Combined object categorization and segmentation with an implicit shape model," in *Proc. ECCV Workshop Statist. Learning Comput. Vis.*, 2004, pp. 17–32.

[11] H. Jégou, M. Douze, C. Schmid, and P. Pérez, "Aggregating local descriptors into a compact image representation," in *Proc. IEEE Conf. Comput. Vis. Pattern Recog.*, Jun. 2010, pp. 3304–3311.

[12] R. Arandjelović and A. Zisserman, "All about VLAD," in *Proc. IEEE Conf. Comput. Vis. Pattern Recog.*, Jun. 2013, pp. 1578–1585.

[13] H. Jégou, M. Douze, and C. Schmid, "On the burstiness of visual elements," in *Proc. IEEE Conf. Comput. Vis. Pattern Recog.*, Jun. 2009, pp. 1169–1176.

[14] G. Tolias, Y. Avrithis, and H. Jégou, "To aggregate or not to aggregate: Selective match kernels for image search," in *Proc. IEEE Int. Conf. Comput. Vis.*, Dec. 2013, pp. 1401–1408.

[15] G. Tolias and Y. S. Avrithis, "Speeded-up, relaxed spatial matching," in *Proc. IEEE Int. Conf. Comput. Vis.*, Nov. 2011, pp. 1653–1660.

[16] Z. Wu, Q. Ke, M. Isard, and J. Sun, "Bundling features for large scale partial-duplicate web image search," in *Proc. IEEE Conf. Comput. Vis. Pattern Recog.*, Jun. 2009, pp. 25–32.

[17] Y. Zhang, Z. Jia, and T. Chen, "Image retrieval with geometry-preserving visual phrases," in *Proc. IEEE Conf. Comput. Vis. Pattern Recog.*, Jun. 2011, pp. 809–816.

[18] W. Zhou, H. Li, Y. Lu, and Q. Tian, "Large scale image search with geometric coding," in *Proc. ACM Int. Conf. Multimedia*, 2011, pp. 1349–1352.

[19] Z. Liu, H. Li, W. Zhou, and Q. Tian, "Embedding spatial context information into inverted file for large-scale image retrieval," in *Proc. ACM Int. Conf. Multimedia*, 2012, pp. 199–208.

[20] H. Stewénius, S. H. Gunderson, and J. Pilet, "Size matters: Exhaustive geometric verification for image retrieval," in *Proc. Eur. Conf. Comput. Vis.*, 2012, pp. 674–687.

[21] H. Jégou, M. Douze, and C. Schmid, "Improving bag-of-features for large scale image search," *Int. J. Comput. Vis.*, vol. 87, no. 3, pp. 316–336, 2010.

[22] A. Mikulik and M. Perdoch, "Total recall II: Query expansion revisited," in *Proc. IEEE Conf. Comput. Vis. Pattern Recog.*, Jun. 2011, pp. 889–896.

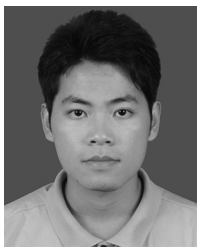
[23] L. Xie, Q. Tian, W. Zhou, and B. Zhang, "Fast and accurate near-duplicate image search with affinity propagation on the ImageWeb," *Comput. Vis. Image Understanding*, vol. 124, pp. 31–41, 2014.

[24] A. Mikulik, M. Perdoch, O. Chum, and J. Matas, "Learning a fine vocabulary," in *Eur. Conf. Comput. Vis.*, 2010, pp. 1–14.

[25] M. Muja and D. G. Lowe, "Scalable nearest neighbor algorithms for high dimensional data," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 36, no. 11, pp. 2227–2240, Nov. 2014.

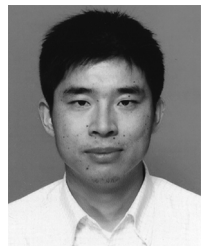
[26] B. Ribeiro-Neto and R. Baeza-Yates, *Modern Information Retrieval*. Reading, MA, USA: Addison-Wesley, 1999.

[27] Y. Avrithis and G. Tolias, "Hough pyramid matching: Speeded-up geometry re-ranking for large scale image retrieval," *Int. J. Comput. Vis.*, vol. 107, no. 1, pp. 1–19, Mar. 2013.
[28] D. Qin, C. Wengert, and L. J. V. Gool, "Query adaptive similarity for large scale object retrieval," in *Proc. IEEE Conf. Comput. Vis. Pattern Recog.*, Jun. 2013, pp. 1610–1617.



**Zhiyuan Zhong** received the B.S. degree in computer science from the South China University of Technology, Guangzhou, China, and is currently working toward the M.S. degree in computer science and technology at Zhejiang University, Hangzhou, China.

His current research interests include machine learning, computer vision, and multimedia information retrieval.



**Jianke Zhu** (S'06–M'09) received the Ph.D. degree in computer science and engineering from the Chinese University of Hong Kong, Hong Kong.

He was previously a Postdoctoral Researcher with the BIWI Computer Vision Laboratory, ETH Zürich, Zurich, Switzerland. He is currently an Associate Professor with the College of Computer Science, Zhejiang University, Hangzhou, China. His current research interests include computer vision and multimedia information retrieval.

**Steven C. H. Hoi** (S'04–M'06–SM'14) received the B.S. degree in computer science from Tsinghua University, Beijing, China, and the M.S. and Ph.D. degrees in computer science and engineering from the Chinese University of Hong Kong, Hong Kong.

He is currently an Associate Professor with the School of Information System, Singapore Management University, Singapore. His current research interests include machine learning, multimedia information retrieval, web search, and data mining.

Dr. Hoi is a member of the ACM.