Object retrieval and localization with spatially-constrained similarity measure and k-NN re-ranking

Xiaohui Shen[1] Zhe Lin[2] Jonathan Brandt[2] Shai Avidan[3] Ying Wu[1]

[1]Northwestern University
2145 Sheridan Road
Evanston, IL 60208
{xsh835, yingwu}@eecs.northwestern.edu

[2]Adobe Systems Inc.
345 Park Ave
San Jose, CA 95110
{zlin, jbrandt}@adobe.com

[3]Tel-Aviv University
Tel-Aviv 69978
Israel
avidan@eng.tau.ac.il

Abstract

One fundamental problem in object retrieval with the bag-of-visual words (BoW) model is its lack of spatial information. Although various approaches are proposed to incorporate spatial constraints into the BoW model, most of them are either too strict or too loose so that they are only effective in limited cases. We propose a new spatially-constrained similarity measure (SCSM) to handle object rotation, scaling, view point change and appearance deformation. The similarity measure can be efficiently calculated by a voting-based method using inverted files. Object retrieval and localization are then simultaneously achieved without post-processing. Furthermore, we introduce a novel and robust re-ranking method with the k-nearest neighbors of the query for automatically refining the initial search results. Extensive performance evaluations on six public datasets show that SCSM significantly outperforms other spatial models, while k-NN re-ranking outperforms most state-of-the-art approaches using query expansion.

1. Introduction

Most state-of-the-art image and visual object retrieval approaches adopt the standard bag-of-words model initially introduced in [23]. While this model works generally well, it suffers from two main problems: 1) the loss of spatial information when representing the images as histograms of quantized features; and 2) the deficiency of feature's discriminative power, either because of the degradation caused by feature quantization, or due to feature's intrinsic limitation to tolerate large variation of object appearance.

In this paper, we address both of these issues by proposing a novel spatially-constrained similarity measure (SCSM), a voting-based approach to efficiently compute the measure, and a re-ranking method with the query's k-nearest neighbors. In SCSM, only the matched visual word pairs with spatial consistency (i.e., roughly coincident feature locations under some similarity transformation) are considered. In other words, the similarity measure is designed to handle object rotation, translation and scaling, and performs well with moderate object deformation.

Figure 1. An example search result of our approach on a real-world database with thousands of personal images with mixtures of buildings, people, pets, animals, faces, flowers, party, sports, etc. Left is the query image with the rectangle. The top 20 retrieval and localization results are shown on the right.

Based on that, a voting-based approach is further proposed to efficiently calculate the similarity with low extra memory and search time, which is inspired by the generalized Hough transform method[13, 11]. Our method can simultaneously localize the object with high accuracy in each retrieved image in the initial search step, which is rarely done by previous retrieval methods. To the best of our knowledge, only [12] and [10] try to localize the object by sub-image search, which is relatively slow when the database is large. Moreover, our approach can robustly retrieve and localize non-rigid objects such as faces or human bodies while previous RANSAC-based localization method (as post-processing) cannot due to non-rigidity of object categories. See Fig.1 for an example.

Meanwhile, since we have already accurately localized

the object in the retrieved images, we can further use such information to refine our results. We observe that, a database image is similar to the query image if it is also similar to the nearest neighbors of the query. An image that contains the query object may not be visually close to the query due to feature variations caused by view point change, occlusion or deformation. However, some of query's neighbors, which can be considered as variations of the query object, may share the same features with that image.

Therefore, we propose a re-ranking method with the k-nearest neighbors (k-NN) of the query. After the initial search, localized objects in the top-k retrieved images are also used as queries to perform search. A database image will have different ranks when using those neighbors as queries. Accordingly a new score of each database image is collaboratively determined by those ranks, and re-ranking is performed using the new scores. Unlike previous query expansion and re-ranking methods, our method is rank-order based, which discards the features and their distances when measuring the score. Therefore, it can successfully retrieve the objects with large variations, while avoiding degradation when there are irrelevant objects in the k-nearest neighbors. Experimental results show it achieves higher and more robust performance than query expansion.

The contributions of this paper are three-fold:

1. A spatially-constrained similarity measure, which significantly outperforms the bag-of-words model, and existing methods with spatial constraints.
2. A voting-based approach to to evaluate the similarity measure that simultaneously, and very efficiently, retrieves and localizes the object in the database images.
3. A re-ranking method with the k-nearest neighbors of the query. Using SCSM and k-NN reranking, we meet or exceed state-of-the-art retrieval performance on standard datasets.

2. Related Work

In this section, we briefly introduce the methods designed to handle the above mentioned two problems of the bag-of-words model, i.e., incorporation of spatial information, and query expansion.

In [19], spatial information is used in a post-verification step after initial search using RANSAC. However it comes with high computational cost, and can consequently only verify a limited number of top-ranked images. Therefore, various approaches are proposed to encode relatively weak spatial constraints in the initial search step without sacrificing much retrieval efficiency. Feature locations are probably the most frequently used spatial information as they can be easily integrated into the inverted file representation[26, 12, 27, 1]. They are used to check the matching order consistency as in bundled features [26], to project the features to different bins to form an ordered spatial bag-of-features model[1], or to search the object in local sub-regions[12]. Visual phrases are also proposed[27], by calculating the location offset of two matched features. Other ways of encoding spatial information include local affine frames for each feature[18], angle and scale parameters[6] and feature spatial distances[4]. However, these spatial constraints are either too restrictive so that only translation can be handled[26, 12, 27], or too loose to capture enough information[1, 6].

To alleviate the information loss in feature quantization, soft assignment on visual words is adopted in [20], while contextual weighting on the vocabulary is introduced in [25]. The probabilistic relationships between the visual words is learned in [14]. Feature metrics are also learned either to increase the feature discriminative power[9, 21] or to reduce the descriptor dimensionality[8].

Another way to compensate the deficiency in feature matching is to automatically expand the query[5, 3]. It tends to improve the retrieval performance especially when the appearance of the object has large variation. However, the performance of query expansion tends to be degraded by false positive search results. Therefore it requires accurate spatial verification which needs high computational cost. Though a faster method is proposed recently[24], the re-ranking is still performed only on the top-ranked images. In [22], a close set (i.e. the images likely containing the same object) of database images is pre-constructed before searching. A similar idea was proposed in [17] where pair-wise feature distances between images are updated using k-nearest neighbors. However constructing such pair-wise data structure is computationally too expensive with large dataset. Different from these methods, we propose a spatially constrained similarity measure and a k-NN re-ranking method without sacrificing much efficiency.

3. Object similarity ranking and localization

3.1. Spatially constrained similarity measure

Given a query image with a specified object, the spatial information of the object can be represented by a rectangle $\mathbf{B} = \{x_c, y_c, w, h, \theta\}$, as shown in Fig.2(a), where (x_c, y_c) is the coordinate of the rectangle center, w and h are the width and height of the rectangle respectively, and θ is the rotated angle of the rectangle ($\theta = 0$ for the query rectangle). We would like to find the same object with certain similarity transformation $\mathbf{T}$ in a database image. $\mathbf{T}$ can be decomposed into three parameters $\mathbf{T} = \{R(\alpha), s, \mathbf{t}\}$, where α is the rotated angle of the object and $R(\alpha) = \begin{bmatrix} \cos\alpha & -\sin\alpha \\ \sin\alpha & \cos\alpha \end{bmatrix}$, s is the scale change, and $\mathbf{t} = (x_\mathbf{t}, y_\mathbf{t})$ is the translation. Accordingly, the transformed object rectangle in the database image would be

$\mathbf{B}' = \mathbf{T}(\mathbf{B}) = \{x_c + x_\mathbf{t}, y_c + y_\mathbf{t}, s \cdot w, s \cdot h, \theta = \alpha\}$ [1], as shown in Fig.2(b).

By the above definition, our task becomes (1) evaluating the similarity between the query object and a database image by finding a (transformed) sub-rectangle in the database image which matches best to the query object; and (2) sorting the database images based on the similarity.

To achieve this, we first define our spatially constrained similarity measure (SCSM). Denote the object rectangle in the query by Q, and the features extracted from Q by $\{f_1, f_2, \cdots, f_m\}$. Similarly, denote the database image by D, and the features in D by $\{g_1, g_2, \cdots, g_n\}$. Given a transformation $\mathbf{T}$, the similarity between Q and D is defined as:

$$S(Q, D|\mathbf{T}) = \sum_{k=1}^{N} \sum_{\substack{(f_i, g_j) \\ f_i \in Q, g_j \in D \\ w(f_i)=w(g_j)=k \\ ||\mathbf{T}(L(f_i))-L(g_j)||<\varepsilon}} \frac{\mathrm{idf}^2(k)}{\mathrm{tf}_Q(k) \cdot \mathrm{tf}_D(k)} \qquad (1)$$

where k denotes the k-th visual word in the vocabulary, and N is the vocabulary size. $w(f_i) = w(g_j) = k$ means f_i and g_j are both assigned to visual word k. $L(f) = (x_f, y_f)$ is the 2D image location of f, and $\mathbf{T}(L(f))$ is its location in D after the transformation. The spatial constraint $||\mathbf{T}(L(f_i)) - L(g_j)|| < \varepsilon$ means that after transformation, the locations of two matched features should be sufficiently close.

In Eqn.1, $\mathrm{idf}(k)$ is the inverse document frequency of visual word k, and $\mathrm{tf}_Q(k)$ is the term frequency (i.e. number of occurrence) of visual word k in Q. Similarly, $\mathrm{tf}_D(k)$ is the term frequency of visual word k in D. This is a normalization term to penalize those visual words repeatedly appearing in the same image. When repeated patterns (*e.g.* building facades, windows and water waves) exist in an image, many features tend to be assigned to the same visual word. Such "burstiness" of visual words violates the assumption in the bag-of-words model that visual words are emitted independently in the image, and therefore could corrupt the similarity measure. This phenomenon is also investigated in [7, 2]. For example, if m features in Q and n features in D are quantized to visual word k respectively, there will be $m \cdot n$ matched pairs between two images, some of which may also satisfy our spatial constraint, as they tend to appear in a local neighborhood. However, if features are directly matched without quantization, there should be at most $\min(m, n)$ matched pairs. In other words, most of these $m \cdot n$ pairs are invalid correspondences and would largely bias our similarity measure if no normalization is applied.

For each database image, the goal is to find the transformation with the highest similarity, i.e.:

$$\mathbf{T}^* = \{R(\alpha^*), s^*, \mathbf{t}^*\} = \arg\max_{\mathbf{T}} S(Q, D|\mathbf{T}) \qquad (2)$$

As a result, $S^*(Q, D) = S(Q, D|\mathbf{T}^*)$ can serve as the similarity between Q and D. All the database images are then ranked according to $S^*(Q, D)$.

Fig.2(a) and (b) illustrates our similarity measure where $w(f_i) = w(g_i)$, but only $\{(f_i, g_i)(i = 1, 2, 3)\}$ are spatially consistent with the transformation. (f_5, g_5) is considered as a false match. As for (f_4, g_4), it depends on the selection of tolerance parameter ε in Eqn.1. If we allow relatively large object deformation and set ε higher, (f_4, g_4) is considered as inliners, otherwise it is also excluded.

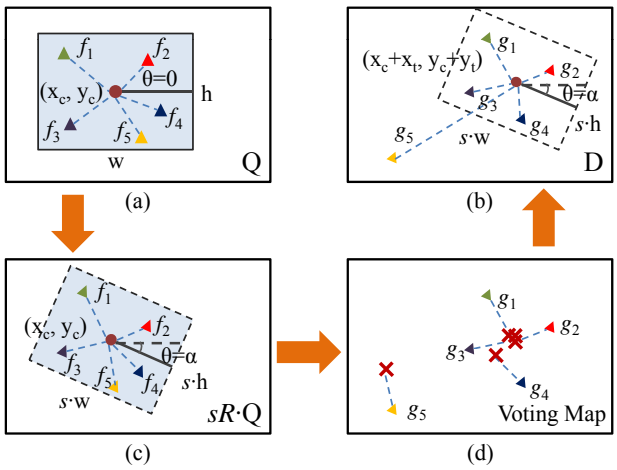

Figure 2. Illustration on SCSM. (a) Query image Q with specified object in the blue rectangle. (b) A database image D containing the same object with a certain transformation. (c) The object in Q is transformed to a different scale and rotation angle. (d) The voting map is generated according to the relative positions of the matched features with respect to the rectangle center. The transformation with the highest voting score are chosen as the best.

3.2. Optimization of the similarity measure

In order to evaluate $S^*(Q, D)$ we need to find the transformation T^* that maximizes the similarity score. In lieu of a practical method to search for the true optimum, we propose an approximation based on discretizing the transformation space, which is decomposed into rotation, scaling and translation. We first quantize the rotation angle space to n_R values between $0 \sim 2\pi$ (Typically $n_R = 4$ or 8). Similarly, the scale space is also discretized to n_s values (typically $n_s = 8$) in a range from $1/2$ to 2, which generally covers most cases. These discretizations yield a set of possible transformation hypotheses (up to translation). The query object is then transformed based on each hypothesis, while keeping the location of the rectangle center the same (i.e., no translation). Fig.2(c) shows an example of such

[1] We keep the aspect ratio of the object fixed but our similarity measure can handle a large range of object deformation and viewpoint changes.

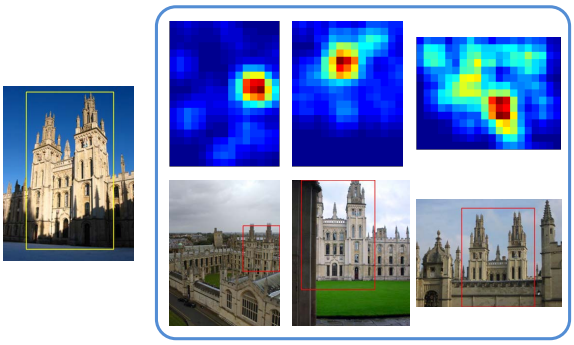

Figure 3. Example of voting maps and localized objects.

transformation hypothesis. To perform the transformation, we only need to re-calculate the relative locations of all the query features with respect to the center.

After the query rectangle is transformed to a particular quantized rotation angle and scale, we then use a voting scheme to find the best translation. Consider a matched pair (f, g) between Q and D. Denote by $V(f)$ the relative location vector from the rotated and scaled location of f to the rectangle center c_Q. (f, g) can determine a translation based on their locations, and this translation enforces the possible location of the rectangle center in D to be $L(c_D) = L(g) - V(f)$. Therefore, given a matched pair, we can find the location of rectangle center in D, and vote a score for that location. If $w(f) = w(g) = k$, the voting score for the pair (f, g) is defined as:

$$Score(k) = \frac{\mathrm{idf}^2(k)}{\mathrm{tf}_Q(k) \cdot \mathrm{tf}_D(k)} \tag{3}$$

Apparently if some matched feature pairs are spatially consistent, the center locations they are voting should be similar. See Fig.2(d) for an example.

The cumulative votes of matched features (f, g) generate a voting map, in which each location represents a possible new object center associated with a certain translation $\mathbf{t}$. When we cast votes using Eqn.3, the accumulated score at each location is exactly the similarity measure $S(Q, D|\mathbf{T})$ in Eqn.1. We choose the best translation $\mathbf{t}^*$ by simply selecting the mode in the voting map.

Remember before voting, we have transformed our query to n_R rotation angles and n_s scales. Therefore there are $n_R \cdot n_s$ voting maps in total. The best transformation $\mathbf{T}^*$ is achieved by finding the location with the highest score in all voting maps. Meanwhile the best score naturally serves as the similarity between the query and the database image, which is subsequently used for ranking. This scheme allows us to *simultaneously* achieve object retrieval and localization without sub-window search or post-processing, which is rarely done in previous work.

In practice, when the objects are mostly upright, we can switch off rotation. When generating the voting map, we can maintain a map with much smaller size compared to the images, by quantizing the map to $n_x \times n_y$ grids. To avoid quantization errors and allow object deformation, instead of voting on one grid, we vote on a 5×5 window around the estimated center grid for each matched pair. The voting score of each grid is the initial $Score(k)$ in Eqn.3 multiplied by a Gaussian weight $\exp(-d/\sigma^2)$, where d is the distance of the grid to the center. This has the effect of spatially smoothing the votes and is equivalent to generating a single vote and smoothing with a Gaussian filter afterwards.

Fig.3 shows an example of generated voting maps and corresponding localized objects. Given the query object in the left, the voting maps generated for three database images are shown in the first row. Each voting map has a single peak as most feature pairs in the same object cast their votes on the same location. The approach robustly localizes the object even if there is dramatic scale and view point change, or severe occlusion.

3.3. Similarity evaluation using inverted files

To calculate our spatially-constrained similarity measure and determine the best transformation, the locations (X- and Y-coordinates) of the features are stored in the inverted files.

When calculating the voting map, we follow the general retrieval framework, i.e., for each word k in the query, retrieve the image IDs and locations of k in these images through the inverted files. Object center locations and scores are then determined by Eqn.3, and votes are cast on corresponding voting maps.

There are two ways to consider rotation and scale change in the search process. One way is to allocate $n_R \cdot n_s$ voting maps at each search round. When traversing the inverted files, we vote on all those maps. Therefore we only have to traverse the inverted files once. Another way is to sequentially generate voting maps for each quantized rotation and scale value. Therefore only one voting map is maintained for each database image. However, we need to retrieve $n_R \cdot n_s$ times. To make a trade-off between search time and memory, in practice we perform search for each quantized rotation step, and generate n_s voting maps with different scales in each search process. In that case, we maintain n_s voting maps for each image, and perform search n_R times.

4. k-NN re-ranking

Since we have localized the object in each retrieved database image, we can further use the top-k retrieved object to refine our retrieval results.

Given a query image, the rank of a database image according to S^* is denoted by $R(Q, D)$. Let N_i be the query's i-th retrieved image. Obviously $R(Q, N_i) = i$. Accordingly $\mathcal{N}_q = \{N_i\}_{\{i=1,\ldots,k\}}$ are the query's k-nearest neighbors, as shown in Fig.4.

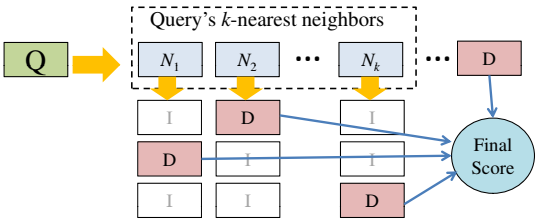

Figure 4. Illustration of k-NN re-ranking. The final rank of a database image is determined by its ranks in the retrieval results of the query and query's k-NN.

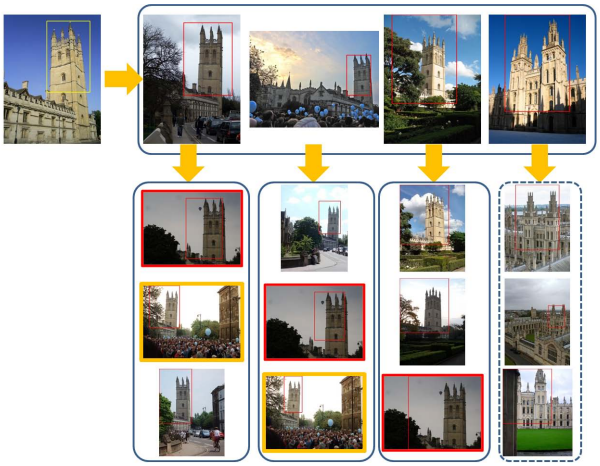

Figure 5. Example of k-NN re-ranking. The 4-th nearest neighbor is an irrelevant image. However, its nearest neighbors in the dashes box will not receive high scores from other images. On the contrary, the images with red and orange boxes are close to a majority of the query's nearest neighbors and will have high ranks.

In most cases, the majority of these k-nearest neighbors contain the same object as in the query image, while there are also some false alarms. See Fig.5 for example. As the features are variant to view point change, occlusion or object deformation, some images with the same object are not visually close to the query, and are ranked very low. However, they may be visually similar to certain images in $\mathcal{N}_q$.

To utilize such information, we also use each localized object in $\mathcal{N}_q$ as a query and perform search. The rank of a database image D when using N_i as the query is $R(N_i, D)$, as shown in Fig.4. According to the rank, we assign a score $1/R(N_i, D)$ to each database image. The final scores of the database images are then collaboratively determined as:

$$\bar{S}(Q, D) = \frac{w_0}{R(Q, D)} + \sum_{i=1}^{k} \frac{w_i}{R(N_i, D)} \tag{4}$$

where w_i is the weight, which is determined by the rank of N_i in the initial search. We set $w_0 = 1$ and $w_i = 1/(R(Q, N_i) + 1) = 1/(i + 1)$. Query itself can be regarded as the 0-th nearest neighbor, and Eqn.4 is accordingly rewritten as:

$$\bar{S}(Q, D) = \sum_{i=0}^{k} \frac{w_i}{R(N_i, D)} = \sum_{i=0}^{k} \frac{1}{(i+1)R(N_i, D)} \tag{5}$$

We also consider the rank of the query in each of its nearest neighbors' retrieval results, i.e., $R(N_i, Q)$. Here, the rank is a unidirectional measure. Query Q and its nearest neighbor N_i are close only if $R(Q, N_i)$ and $R(N_i, Q)$ are both high. Hence we modify the weight w_i to be $w_i = 1/(R(Q, N_i) + R(N_i, Q) + 1)) = 1/(i + R(N_i, Q) + 1)$, and the final scores of database images are determined by:

$$\bar{S}(Q, D) = \sum_{i=0}^{k} \frac{1}{(i + R(N_i, Q) + 1)R(N_i, D)} \tag{6}$$

Images are then re-ranked based on $\bar{S}(Q, D)$.

After re-ranking, we can further use the new top-k retrieved images to perform re-ranking iteratively. In most cases, the first iteration brings significant performance improvement.

The proposed k-NN re-ranking approach takes advantage of the localized objects in the retrieved images by SCSM, as we can ignore those irrelevant features outside the objects. Furthermore, as a rank-based approach, our re-ranking method is robust to false retrieval results in $\mathcal{N}_q$. Unlike query expansion[5, 3], in our method, the score is inversely related to the ranking, and the feature information of all the k-NN images is intentionally discarded. A database image will not be re-ranked very highly unless it is close to the query and the majority of those k-NN images. Consider Fig.5 as an example, the irrelevant image in $\mathcal{N}_q$ assigns scores to its top-retrieved results. However, the weight corresponding to this outlier is relatively small as the rank itself in the query's retrieval list is not high. Furthermore, the images in the dashed box will not receive scores from other images in $\mathcal{N}_q$ and accordingly their scores for re-ranking is still low. On the contrary, a relevant image such as the one with red bounding box or orange box is close to several images in $\mathcal{N}_q$ and will have a high score. Experimental results indicate our method is not sensitive to the selection of nearest neighbor number k. Even if k is large and there are many outliers in $\mathcal{N}_q$, the retrieval accuracy is still very high. Since our method is robust to outliers, no spatial verification is needed. Also, re-ranking can be efficiently performed on the entire database.

5. Experiments

5.1. Datasets and implementation details

We have implemented our own retrieval system with SIFT descriptors[13] and fast approximate k-means clustering[15]. We evaluate our approach on four public

datasets: *Oxford building*[2], *Paris*[3], *INRIA Holidays*[4], and *University of Kentucky*[5]. 100,000 and 1M Flickr images downloaded with random tags are also added to *Oxford* as distractors to form the *Oxford105k* and *Oxford 1M* dataset. In *Oxford* and *Paris*, each query has a specified object rectangle, while no such rectangles are specified in *INRIA* and *Kentucky*. So we use the entire frames as our query rectangles for these two datasets. 1M vocabularies are trained for *Oxford* and *Paris*, A 200k vocabulary is trained for *INRIA* as in [6]. The vocabulary size for *Kentucky* is set to 500k as there are only 7M features.

In the implementation of the voting-based method, we switch off rotation in *Oxford* and *Paris* as most of these query objects are upright. k-NN re-ranking is performed on all the datasets except *INRIA Holidays*, as there are only one or two relevant images for most queries in this dataset.

In evaluation, as in most of previous methods, the retrieval accuracy on the first three datasets and their extensions is measured with the mean average precision (mAP), while the performance measure on the *Kentucky* dataset is the top-4 score, i.e., the average number of relevant images in the query's top 4 retrieved images as in [16].

5.2. Results of SCSM

Parameters: We first evaluate the performance of our approach given different settings of parameters. There are two main parameters in our method: the grid size (the number of grid cells) of the voting map[6], and σ^2 in the Gaussian weights $\exp(-d/\sigma^2)$.

The mAP on *Oxford5k* with different map sizes is shown in Fig.6(a). As we can see, when the grid number is larger than 16, the mAP remains flat. Therefore a 16×16 voting map is already large enough, which allows us to encode the feature location in a 1-byte integer. The performance with different σ^2 in voting is shown in Fig.6(b). $\sigma^2 = 0$ means there is no Gaussian voting, i,e, each matched pair only vote on one grid corresponding to its estimated object center. The results show that voting on a window with Gaussian weighting is noticeably better than voting on one grid. It is easy to understand as such a Gaussian voting allows object deformation and also reduces quantization errors. However, once the Gaussian voting is adopted, the mAP does not change much with different values of σ, which indicates that our method is not sensitive to this parameter. When $\sigma^2 = 2.5$, our method achieves the highest mAP. This parameter is fixed at 2.5 in all subsequent experiments.

Comparisons: We compared SCSM with the baseline bag-of-words model. The results are shown in Table 1. We can see SCSM significantly outperforms the bag-of-words model on all the datasets. Furthermore, in *Oxford105k* and *Oxford 1M*, when distractors are added, the mAP of the baseline method decreases from 0.649 to 0.568 and 0.535 respectively, while our method is only slightly affected (from 0.753 to 0.729 and 0.685 respectively). This indicates SCSM is more scalable to larger databases. We also compared our approach to other methods with spatial models, as listed in Table 2. Our approach outperforms all those methods on all the datasets. Some examples of object retrieval and localization are provided in Fig.7.

[2]http://www.robots.ox.ac.uk/~vgg/data/oxbuildings.
[3]http://www.robots.ox.ac.uk/~vgg/data/parisbuildings.
[4]http://lear.inrialpes.fr/~jegou/data.php.
[5]http://www.vis.uky.edu/~stewe/ukbench.
[6]The grid spacing is then determined by the maximum of image size divided by the grid size.

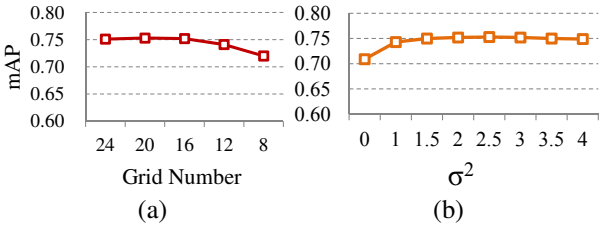

Figure 6. mAP on *Oxford5k* with different parameters. (a) Voting map size, (b) Gaussian weight. It shows that SCSM is not sensitive to wide range of grid numbers and Gaussian weights.

Datasets	BoW	SCSM	SCSM+Re-ranking
Oxford5k	0.649	0.752	0.884
Oxford105k	0.568	0.729	0.864
Oxford 1M	0.535	0.685	0.841
Paris	0.630	0.741	0.911
INRIA	0.462	0.762	-
Kentucky	3.35	3.52	3.56

Table 1. The performance of our method on public datasets.

Datasets	SCSM	[27]	[12]	[1]	[6]
Oxford5k	**0.752**	0.713	0.647	0.651	0.547
Oxford105k	**0.729**	0.604	-	-	-
Oxford 1M	**0.685**	0.532	-	0.550	-
Paris	**0.741**	-	-	0.632	-
INRIA	**0.762**	-	-	-	0.751
Kentucky	**3.52**	3.26	3.29	-	-

Table 2. Comparisons of SCSM with other spatial models.

5.3. Results of k-NN re-ranking

Parameters: There is only one parameter in our k-NN re-ranking method, the number of nearest neighbors k. Fig.8 shows the performance on *Oxford5k* when we change k (only single iteration is used). Even with only 5 nearest neighbors, the mAP is already improved to 0.822. When the k-NN set $\mathcal{N}_q$ becomes larger, the mAP keeps increasing. Although, there are many irrelevant images in $\mathcal{N}_q$ when k is large (some of the queries only have less than 10 relevant images). Our approach can still achieve very high accuracy in that case, which demonstrates the robustness of this rank-

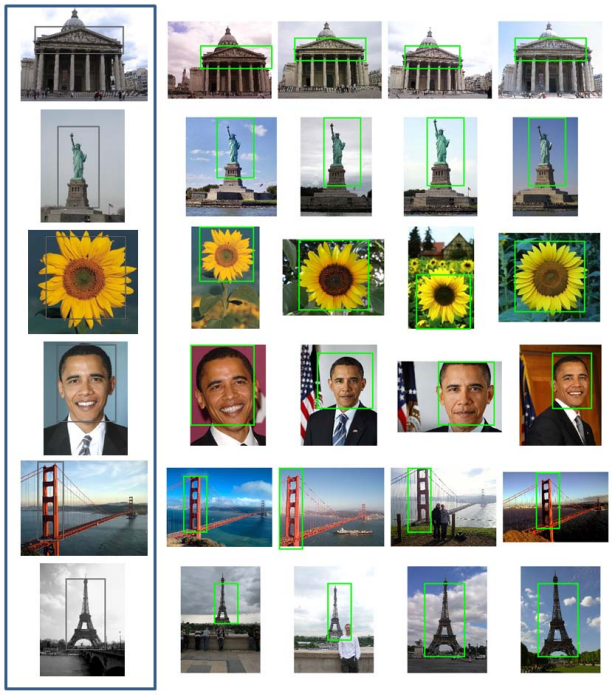

Figure 7. Examples of object localization by SCSM.The images in the first column are the queries, while the localized results on the top-4 ranked images are presented.

based method to outliers. When we use two iterations, i.e. performing re-ranking again with the newly retrieved top-k images, the mAP is further improved to 0.884 when k is 30.

Similar phenomena are observed on *Oxford105k*, *Oxford1M* and *Paris*, in which the numbers of relevant images are similar with those in *Oxford5k*. We use the same setting ($k = 30$ with two iterations) in all these datasets. The queries in *Kentucky* has only 3 other relevant images. As a result, we observed that $k = 1, 2, 3$ yield similar performance on $Kentucky$. Considering computational efficiency, we choose $k = 1$ with one iteration for this dataset.

Comparisons: The performance of k-NN re-ranking is shown in Table 1. It further significantly improves the retrieval performance. The mAP of re-ranking on *Oxford105k* and *Oxford 1M* achieves 0.864 and 0.841 respectively, indicating that our method is very robust to distractors.

Table 3 shows the comparisons of our method with other state-of-the-art approaches. Most of these methods use query expansion. Some of them employ additional techniques such as post-verification and soft assignment (which are not used but could be further incorporated in our method). The results of our approach are among the best on *Oxford5k* and *Oxford105k*, and significantly better than previously best-achieved results on *Paris* (from 0.824 to 0.911). The assumption of SCSM is frequently violated in the *Kentucky* dataset, while there are only 3 other relevant

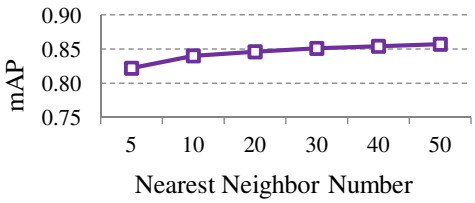

Figure 8. mAP on *Oxford5k* using different numbers of nearest neighbors in re-ranking. Our method gets better when k becomes larger, indicating it is robust to outliers.

images for each query. Nevertheless, our method performs reasonably well even under such an unfavorable condition.

5.4. Complexity and Scalability

Compared to the bag-of-words model, the additional memory cost of our method includes the feature locations in the inverted files, and the voting maps in object localization. Since we use 16×16 voting maps, the X- and Y-coordinates of the features can be quantized to $0, 1, ..., 15$ and further encoded as a 8-bit integer $l_f = 16 \cdot y_f + x_f$. Therefore we only need 1 more byte storage for each feature. Suppose we have 500 features in an image, the additional memory for a 1 million dataset is 500 MB which is much smaller than the size of the inverted file.

A voting map is a 16×16 matrix with floating values, which therefore needs 1024 byte memory storage. We allocate $n_s = 8$ maps for an image, and the additional memory cost for each image is 8K. However, the voting maps are only assigned for those database images having common visual words with the query. The number of these relevant images is much smaller than the size of the dataset.

In the search process, additional time is needed when we generate the voting maps. Given a visual word k, suppose it has m occurrences in the query and n occurrences in a database image, there would be $m \cdot n$ possible matched pairs. Therefore we need to vote $m \cdot n$ times, while in the bag-of-words model the calculation is only carried once. However, when the vocabulary is large, m and n are 1 in most cases. Meanwhile, when m and n is large, the voting score in Eqn. 3 is very small. Therefore in practice we do not perform voting when $m \cdot n$ is larger than 10, which speeds up the search process. With 3G Duo CPU, the average search time for the bag-of-words model is 0.084s in *Oxford5k*, while SCSM takes 0.089s in average. k-NN re-ranking with single iteration needs k additional search, but it can be processed in parallel when the database is large.

6. Conclusions

Unlike previous image retrieval methods that focus on image ranking, we achieves simultaneous object retrieval and localization by employing a new spatially-constrained similarity measure (SCSM), with a voting-based method.

Datasets	SCSM	SCSM+Re-ranking	[19]	[22]	[7]	[3]	[14]	[18]	[2]	[21]
Oxford5k	0.752	0.884	0.647	0.814	0.685	0.827	0.849	**0.901**	-	0.707
Oxford105k	0.729	**0.864**	0.541	0.767	0.628	0.767	0.795	0.856	**0.864**	0.615
Oxford 1M	0.685	**0.841**	0.465	-	-	-	-	-	-	0.689
Paris	0.741	**0.911**	-	0.803	-	0.805	0.824	-	-	-
INRIA	0.762	-	-	-	**0.848**	-	0.758	0.736	-	-
Kentucky	3.52	3.56	3.45	**3.67**	3.64	-	-	-	-	-

Table 3. Comparisons with other state-of-the-art methods.

Our SCSM significantly outperforms other spatial models in object retrieval. Meanwhile, the objects are accurately localized in relevant images. Based on the retrieved images and localized objects, a k-NN re-ranking method is further proposed to improve the retrieval performance. Extensive evaluation on several datasets demonstrates our method achieves the state-of-the-art performance. Our method can be integrated in a retrieval system with other components such as soft assignment[20] in feature quantization, and learned vocabulary[14] to better serve object and image retrieval. Meanwhile, the localized objects in retrieved images can be adopted for other vision tasks such as image tagging and object detection, which merits further study.

Acknowledgements

This work is partially supported by Adobe Systems, Inc., and in part by National Science Foundation grant IIS-0347877, IIS-0916607, US Army Research Laboratory and the US Army Research Office under grant ARO W911NF-08-1-0504, and DARPA Award FA 8650-11-1-7149.

References

[1] Y. Cao, C. Wang, Z. Li, L. Zhang, and L. Zhang. Spatial-bag-of-features. In *CVPR*, 2010.

[2] O. Chum and J. Matas. Unsupervised discovery of co-occurrence in sparse high dimensional data. In *CVPR*, 2010.

[3] O. Chum, A. Mikulík, M. Perd'och, and J. Matas. Total recall II: Query expansion revisited. In *CVPR*, 2011.

[4] O. Chum, M. Perd'och, and J. Matas. Geometric min-hashing: Finding a (thick) needle in a haystack. In *CVPR*, 2009.

[5] 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 *ICCV*, 2007.

[6] H. Jégou, M. Douze, and C. Schmid. Hamming embedding and weak geometric consistency for large scale image search. In *ECCV*, 2008.

[7] H. Jégou, M. Douze, and C. Schmid. On the burstiness of visual elements. In *CVPR*, 2009.

[8] H. Jégou, M. Douze, C. Schmid, and P. Pérez. Aggregating local descriptors into a compact image representation. In *CVPR*, 2010.

[9] H. Jégou, H. Harzallah, and C. Schmid. A contextual dissimilarity measure for accurate and efficient image search. In *CVPR*, 2007.

[10] C. H. Lampert. Detecting objecs in large image collections and videos by efficient subimage retrieval. In *ICCV*, 2009.

[11] B. Leibe, A. Leonardis, and B. Schiele. Combined object categorization and segmentation with an implicit shape model. In *ECCV Workshop on Statistical Learning in Computer Vision*, 2004.

[12] Z. Lin and J. Brandt. A local bag-of-features model for large-scale object retrieval. In *ECCV*, 2010.

[13] D. G. Lowe. Distinctive image features from scale-invariant keypoints. *IJCV*, 60(2):91–110, 2004.

[14] A. Mikulík, M. Perd'och, O. Chum, and J. Matas. Learning a fine vocabulary. In *ECCV*, 2010.

[15] M. Muja and D. G. Lowe. Fast approximate nearest neighbors with automatic algorithm configuration. In *VISAPP*, 2009.

[16] D. Nistér and H. Stewénius. Scalable recognition with a vocabulary tree. In *CVPR*, 2006.

[17] D. C. G. Pedronette and R. da S. Torres. Exploiting contextual spaces for image re-ranking and rank aggregation. In *ICMR*, 2011.

[18] M. Perd'och, O. Chum, and J. Matas. Efficient representation of local geometry for large scale object retrieval. In *CVPR*, 2009.

[19] J. Philbin, O. Chum, M. Isard, J. Sivic, and A. Zisserman. Object retrieval with large vocabularies and fast spatial matching. In *CVPR*, 2007.

[20] J. Philbin, O. Chum, M. Isard, J. Sivic, and A. Zisserman. Lost in quantization: Improving particular object retrieval in large scale image databases. In *CVPR*, 2008.

[21] J. Philbin, M. Isard, J. Sivic, and A. Zisserman. Descriptor learning for efficient retrieval. In *ECCV*, 2010.

[22] D. Qin, S. Gammeter, L. Bossard, T. Quack, and L. VanGool. Hello neighbor: accurate object retrieval with k-reciprocal nearest neighbors. In *CVPR*, 2011.

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

[24] G. Tolias and Y. Avrithis. Speeded-up, relaxed spatial matching. In *ICCV*, 2011.

[25] X. Wang, M. Yang, T. Cour, S. Zhu, K. Yu, and T. X. Han. Contextual weighting for vocabulary tree based image retrieval. In *ICCV*, 2011.

[26] Z. Wu, Q. Ke, M. Isard, and J. Sun. Bundling features for large scale partial-duplicate web image search. In *CVPR*, 2009.

[27] Y. Zhang, Z. Jia, and T. Chen. Image retrieval with geometry-preserving visual phrases. In *CVPR*, 2011.