Spatially-Constrained Similarity Measure for Large-Scale Object Retrieval

Xiaohui Shen, *Member, IEEE*, Zhe Lin, *Member, IEEE*,
Jonathan Brandt, *Member, IEEE*, and Ying Wu, *Senior Member, IEEE*

Abstract—One fundamental problem in object retrieval with the bag-of-words model is its lack of spatial information. Although various approaches are proposed to incorporate spatial constraints into the model, most of them are either too strict or too loose so that they are only effective in limited cases. In this paper, a new spatially-constrained similarity measure (SCSM) is proposed 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. During the retrieval process, object localization in the database images can also be simultaneously achieved using SCSM without post-processing. Furthermore, based on the retrieval and localization results of SCSM, 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 data sets show that SCSM significantly outperforms other spatial models including RANSAC-based spatial verification, while k-NN re-ranking outperforms most state-of-the-art approaches using query expansion. We also adapted SCSM for mobile product image search with an iterative algorithm to simultaneously extract the product instance from the mobile query image, identify the instance, and retrieve visually similar product images. Experiments on two product image search data sets show that our approach can robustly localize and extract the product in the query image, and hence drastically improve the retrieval accuracy over baseline methods.

Index Terms—Object retrieval, bag-of-words, spatially-constrained similarity measure, k-NN re-ranking, product image search

1 Introduction

The standard bag-of-words model [1] has been widely used in most state-of-the-art image and visual object retrieval approaches. While this model works generally well, it suffers from one fundamental problem: the loss of spatial information when representing the images as histograms of quantized features. A natural way to complement this approach is adding a verification step using the spatial information of the features after initial search [2]. However, since the verification needs to be performed for each retrieved image using RANSAC, it is computationally expensive and can only be performed on a limited number of database images. Various methods therefore have been proposed to incorporate the spatial information in the initial search step, yet adding appropriate spatial information without losing retrieval efficiency, and accommodating object appearance change due to rotation, scaling, viewpoint change and deformation, is still a challenging problem.

In this paper, we address this issue by proposing a novel spatially-constrained similarity measure (SCSM). In SCSM, only the matched feature pairs with spatial consistency (i.e., roughly coincident feature locations under some similarity transformation) are considered. Based on that, a voting-based approach, which is inspired by generalized Hough transform [3], [4], is further proposed and incorporated to inverted-file based search process to efficiently calculate the similarity with low extra memory and search time. Our method can simultaneously localize the object in each retrieved image in the initial search step, which is rarely done by previous retrieval methods. To the best of our knowledge, only [5] and [6] try to localize the object by sub-image search, which is relatively slow when the database is large, while RANSAC-based spatial verification can only localize the objects on a very limited number of retrieved images as a post-step after search due to its computational complexity. Moreover, by adopting Gaussian weighting and down-scaling in the voting process, our approach can also work for some non-rigid objects such as faces or human bodies as shown in Fig. 1, which can hardly be achieved by previous strict RANSAC-based spatial verification methods.

Meanwhile, since we have already localized the object in the retrieved images using SCSM, we can further use such information to refine our results. We observe that, a database image is similar to the query object 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 based on our localization results. After the initial search, localized objects in the top-k retrieved images are further used as queries to perform search. A database image will have different ranks

- *X. Shen, Z. Lin and J. Brandt are with Adobe Research, 345 Park Ave, San Jose, CA 95110. E-mail: {xshen, zlin, jbrandt}@adobe.com.*
- *Y. Wu is with the Department of Electrical Engineering and Computer Science, Northwestern University, 2145 Sheridan Road, M322, Evanston, IL 60208. E-mail: yingwu@eecs.northwestern.edu.*

Manuscript received 21 Dec. 2012; revised 1 July 2013; accepted 12 Nov. 2013. Date of publication 4 Dec. 2013; date of current version 12 May 2014.
Recommended for acceptance by C. Lampert.
For information on obtaining reprints of this article, please send e-mail to: reprints@ieee.org, and reference the Digital Object Identifier below.
Digital Object Identifier no. 10.1109/TPAMI.2013.237

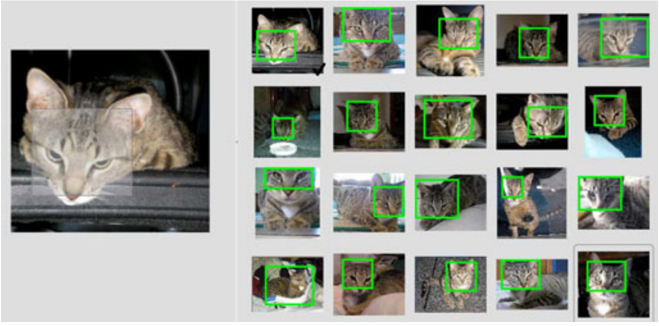


Fig. 1. An example search result of our approach on a real-world database. Our method can also retrieve and localize some non-rigid objects.

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. Meanwhile our scoring strategy in re-ranking ensures that a database image is re-ranked high only when it is close to the majority of the nearest neighbors. As a result, it can successfully retrieve objects with large variations, while avoiding degradation when there are irrelevant objects in the k-nearest neighbors. Experimental results show it significantly outperforms other methods using query expansion.

We further applied SCSM in the scenario of mobile product image search. In this scenario, the objects in the database images are mostly well aligned and captured in studio environments with controlled lighting and clean background, while mobile query images are usually taken under very different lighting conditions with cluttered background. Due to background clutter in the query image, the results using the standard bag-of-words model may be largely affected by the features extracted from the background of the query image. We use SCSM to automatically localize and extract the product in the query image and remove the distraction of the background. In our approach, each retrieved database image predicts a location and an outline shape (or mask) for the query object. The center location and the support region of the query object can then be inferred by a weighted object mask voting and aggregation while removing the outliers. Based on that, the query object is automatically segmented and filled with clean background, which is used to refine the search results in the next round. Since better search results yield better query object extraction, and vice versa, the above two procedures are performed in an iterative and interleaved way, hence forming a closed-loop adaptation between query object extraction and object retrieval.

We collected two data sets for mobile product image search. Experimental results on these two data sets show that our automatic query extraction using SCSM yields even better segmentation results than manual segmentation with a bounding rectangle as initialization, while our iterative retrieval process significantly outperforms previous retrieval methods.

To summarize, the contributions of this paper are three-fold:

1) We propose a spatially-constrained similarity measure with a voting-based approach to evaluate SCSM, which can simultaneously retrieve and localize an the object of interest in the database images. Our SCSM significantly outperforms the bag-of-words model, and existing methods with spatial constraints in terms of search accuracy.
2) We propose 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 data sets.
3) We apply SCSM in mobile product image search to automatically extract the products in mobile query images, and significantly improves the retrieval performance.

This paper is an extension of our conference papers [7], [8]. Compared with [7] and [8], this paper provides a more comprehensive and systematic report of our work. The content is re-organized to make it more self-contained, and more details in algorithm description and implementation are provided. More extensive experiments are also conducted to validate the proposed method. The rest of the paper is organized as follows: Section 2 briefly introduces the methods that are closed related to our approach. Our spatially-constrained similarity measure is defined in Section 3. The voting-based approach to calculate SCSM, and the integration of this method with inverted files is also subsequently introduced. Section 4 describes the k-NN re-ranking method based on the localization results using SCSM. Section 5 introduce the application of SCSM in mobile product image search. Experimental results are presented and discussed in Section 6, and the conclusions are drawn in Section 7.

2 Related Work

Since our approach follows the pipeline of the bag-of-words model using local features, visual vocabularies and inverted files, we briefly introduce the methods designed to handle the drawbacks of the standard bag-of-words model, including the incorporation of spatial information, better feature quantization and query expansion. Meanwhile, the background of mobile product image search will also be introduced.

2.1 Improvements on the Bag-of-Words Model

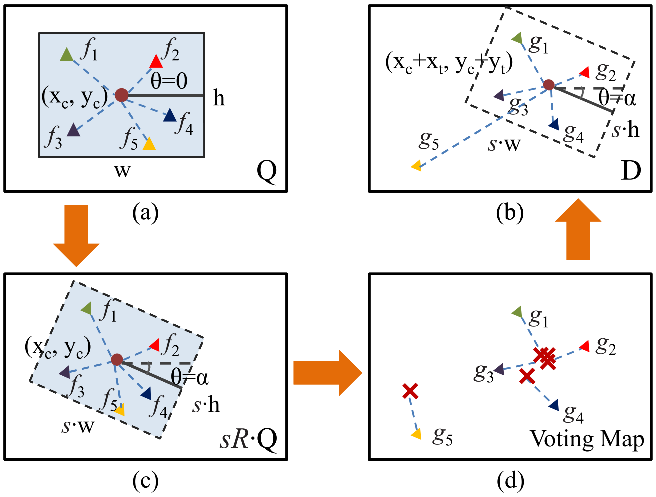
In [2], spatial information is used in a post-verification step after initial search using RANSAC. However it comes with a high computational cost, and can consequently only verify a limited number of top-ranked images. Therefore, various approaches are proposed to encode relatively weak spatial constraints 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 [5], [9], [10], [11]. They are used to check the matching order consistency as in bundled features [9], to project the features to different bins to form an ordered spatial bag-of-features model [11], or to search the object in local sub-regions[5]. Visual phrases are also proposed [10], by calculating the location offset of two matched features. Other ways of encoding spatial information include local affine frames for each feature [12], angle and scale parameters [13] and feature spatial distances [14]. However, these spatial constraints are either too restrictive so that only translation can be handled [5], [9], [10], or too loose to capture enough information [11], [13].

To alleviate the information loss in feature quantization, soft assignment is adopted in [15], while contextual weighting on the vocabulary is introduced in [16]. The probabilistic relationships between the visual words is learned in [17]. Feature metrics are also learned either to increase the feature discriminative power [18], [19] or to reduce the descriptor dimensionality [20].

Another way to compensate the deficiency in feature matching is to automatically expand the query [21], [22]. 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 [23], the re-ranking is still performed only on the top-ranked images. In [24], 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 [25] where pair-wise feature distances between images are updated using k-nearest neighbors. However constructing such pairwise data structure is computationally too expensive with large data set. Different from these methods, we propose a spatially constrained similarity measure and a k-NN reranking method that can efficiently retrieve objects with large appearance variations while robust to the distraction of irrelevant images.

2.2 Mobile Product Image Search

While general image search has been well-studied, research efforts devoted to mobile product image search are still limited. Some search engines for product images have recently been developed [26], [27], [28]. However, in these works, the query images are very similar to the database images (i.e., captured in the same settings). Google Goggles[1] and Amazon Flow[2] are well-known commercial mobile product image search engines, but are working robustly only for a few near-planar, textured object categories such as logos/trademarks, books/CD covers, landmarks, artworks, text, etc. In [29], a new database for mobile visual search is proposed, in which the objects are still limited to planar categories such as books and CD covers. Retrieving more general object categories (either severely non-planar, non-rigid, or less-textured objects) from mobile phones is still an open research question, and a search engine specifically designed for mobile product image search for more challenging object categories is highly demanded. The effort that tries to reduce the imaging quality difference of mobile query images and database images is also limited. In [30], [31], the query object is segmented after a user identifies the object with a bounding box. However, such simple labeling does not necessarily guarantees accurate segmentation results, as demonstrated in our experiments, while more extensive and careful labeling would largely increase users' burden. Different from their method, we propose an automatic query object extraction method that achieves better performance in terms of both segmentation and retrieval results.



(a) (b) (c) (d)

Fig. 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 The Spatially-Constrained Similarity Measure

In this section, we first introduce the spatially-constrained similarity measure, and then propose the voting-map based method to calculate the similarity and localize the objects. The integration of this method with inverted files is subsequently discussed.

3.1 Formulation

Consider that the object in the query image is bounded by a rectangle $\mathbf{B} = \{x_c, y_c, w, h, \theta\}$, as shown in Fig. 2a, 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 spatial transformation $\mathbf{T}(\cdot)$ in a database image. Here we consider that $\mathbf{T}(\cdot)$ is composed of three parameters $\mathbf{T}(\cdot) = \{R(\alpha), s, \mathbf{t}\}$, where $R(\alpha)$ is the rotation, s is the scale change, and $\mathbf{t} = (x_t, y_t)$ is the translation. Accordingly, the transformed object rectangle in the database image would be $\mathbf{B}' = \mathbf{T}(\mathbf{B}) = \{x_c + x_t, y_c + y_t, s \cdot w, s \cdot h, \theta = \alpha\}$,[3] as shown in Fig. 2b.

By the above definition, our task becomes (1) evaluate the similarity between the query object and a database image by finding a (transformed) sub-rectangle in the database image

[1] www.google.com/mobile/goggles/.
[2] http://flow.a9.com.

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

which matches best to the query object; and (2) sort the database images based on the similarity.

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

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

where k denotes the kth 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. Therefore only the feature pairs that are consistent with the transformation will contribute to the similarity score. Figs. 2a, 2b 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 Eq. (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.

In Eq. (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 [32], [33] 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. We found that such normalization can well penalize those invalid correspondences generated by repeated visual words, which is also an advantage of our similarity measure compared with measures used in the bag-of-words model and other approaches.

Given the above definition, for each database image, the goal is to find the transformation with the highest similarity, i.e.:

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

Once the best transformation $\mathbf{T}^*$ is estimated, the object in the database image can be localized. Meanwhile, $S^*(Q, D) = S(Q, D|\mathbf{T}^*)$ is the similarity between Q and D. All the database images are then ranked according to $S^*(Q, D)$.

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. 2c shows an example of such 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. 2d 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 Eq. (3), the accumulated score at each location is exactly the similarity measure $S(Q, D|\mathbf{T})$ in Eq. (1). To choose the best translation $\mathbf{t}^*$, we only need to simply select the mode in the voting map.

Each rotation and scale change hypothesis could generate a voting map. Therefore there are $n_R * n_s$ 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.

In practice, when the objects are mostly upright, we can switch off rotation. For further speed up, we maintain a voting 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, after voting, each map is convolved by a 5×5 Gaussian kernel $\exp(-d/\sigma^2)$, where d is the distance of the grid to the estimated object center. This allows the grids around the estimated center to also receive certain votes. We found that a low-resolution voting map with Gaussian smoothing can largely reduce the computational memory and time cost, while maintaining retrieval accuracies as demonstrated in the experiments.

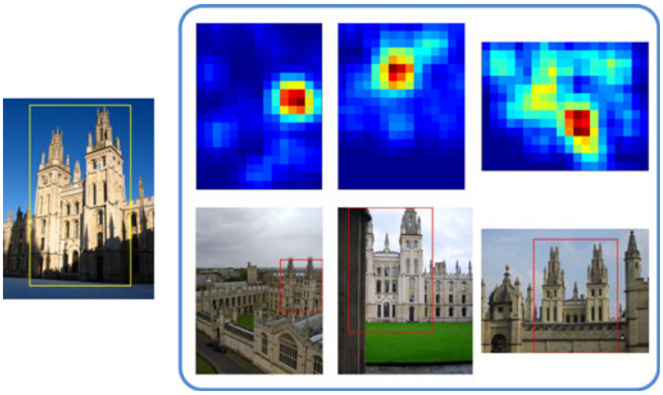


Fig. 3. Example of voting maps and localized objects.

We follow the general retrieval framework and incorporate the voting process into the search process that is based on inverted files. For each word k in the query, we retrieve the IDs of database images and the locations of k in these images through the inverted files. Object center locations and scores are then determined by Eq. (3), and votes are cast on corresponding voting maps. The entire search process is summarized in Algorithm 1.

Algorithm 1: Object ranking and localization with spatially constrained similarity measure

```
$Q$ =Query object, $D_n$ = the $n$-th database image.
$S_n^*$ = the similarity between $Q$ and $D_n$, $T_n^*$ = the
best transformation for $D_n$
for $i$= 1 to $n_R$ do
    for $j$= 1 to $n_s$ do
        Calculate the feature locations in $Q$ after
        rotation $R(\alpha_i)$ and scale change $s_j$;
        Traverse the inverted files, for each $(f, g)$
        do
            find $D$ containing $g$, allocate a voting
            map for $D$ if visited for the first time;
            estimate $L(c_D)$, vote using Eqn. 3;
        end
        After all the matched pairs have voted
        for $n$=1 to Size of database do
            select the highest score $S_n$ from all the
            voting map for $D_n$, with
            transformation $T_n$;
            if $S_n > S_n^*$ then
                $S_n^* \leftarrow S_n$;
                $T_n^* \leftarrow T_n$;
            end
        end
    end
end
Rank all the database images according to $S_n^*$.
```

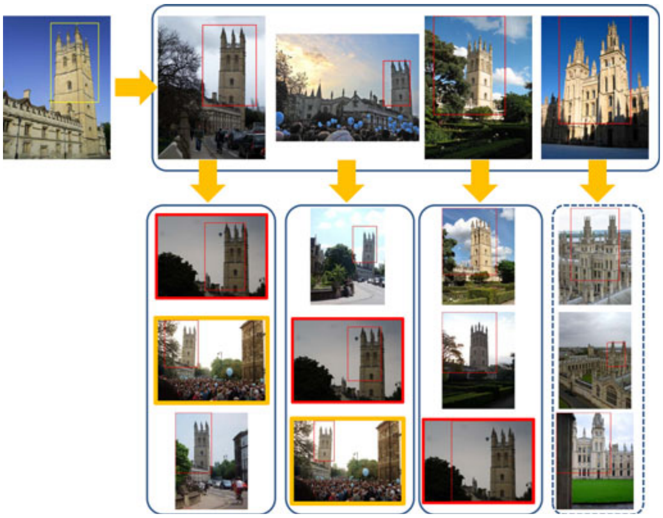


Fig. 4. Example of k-NN re-ranking. The 4th nearest neighbor is an irrelevant image. However, its nearest neighbors in the dashes box will not receive high scores from other images. On the contrary, the images with red and orange boxes are close to the majority of the query's nearest neighbors and will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

Compared with spatial verification and other spatial models, the advantages of the proposed SCSM can be summarized as follows: 1) Though still based on the bag-of-words model, the proposed similarity measure takes both the spatial transformation and the burstiness of visual words into account, which is more sophisticated than RANSAC and shows better retrieval performance. 2) By calculating the similarity measure using Generalized Hough voting with Gaussian smoothing, our method can still maintain high retrieval accuracy when only storing low-dimension voting maps, which generates very low additional memory and time cost. On the contrary, the parametric affine model in RANSAC is sensitive to outliers, and its performance is largely degraded in low-resolution, as demonstrated in the experiments. Our method even works for some non-rigid objects (as the low-resolution voting maps and Gaussian smoothing relaxes the spatial constraints, see Fig. 1, for example). 3) We can localize the objects in retrieved images directly in the initial search step without additional cost. Other weak spatial models employed in initial search [9], [10], [11] can hardly do so. RANSAC-based spatial verification is also able to localize the objects, but it can only be performed as a post-processing step on limited number of retrieved images, and takes considerable amount of computational time.

Fig. 3 shows an example of generated voting maps and corresponding localized objects. The approach can localize the object even if there is dramatic scale and view point change, or severe occlusion.

4 k-NN Re-Ranking

Since we have localized the object in each retrieved database image using SCSM, 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 ith 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.

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. 4 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 $R(N_i, D)$. When $R(N_i, D)$ is smaller, the database image D is closer to neighbor N_i, and N_i should have higher contribution to the score of D.Therefore according to the rank, we assign a score $1/R(N_i, D)$ to each database image. For the simplicity of presentation, we regard query itself as its 0th nearest neighbor, i.e., $Q = N_0$. The final scores of the database images are then collaboratively determined as:

$$\overline{S}(Q, D) = \sum_{i=0}^{k} \frac{w_i}{R(N_i, D)}, \tag{4}$$

where w_i is the weight, which indicates the importance of N_i. Naturally if N_i and the original query Q are closer, N_i could be regarded with higher importance. Although N_i appears in the nearest neighbor set of Q, it does not necessarily guarantee that Q can be ranked high when using N_i as a query. Therefore we consider N_i and Q are close only when $R(Q, N_i)$ and $R(N_i, Q)$ are both high. Accordingly we set the weight to be $w_i = 1/(R(Q, N_i) + R(N_i, Q) + 1)) = 1/(i + R(N_i, Q) + 1)$. $R(Q, N_i)$ can be easily obtained when the search with N_i as the query is completed, and no additional computation is needed. The final scores of database images are then determined by:

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

Images are then re-ranked based on $\overline{S}(Q, D)$. 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, with our scoring strategy, our re-ranking method is robust to falsely retrieved images in $\mathcal{N}_q$. Consider Fig. 4 as an example, the fourth image in the nearest neighbor set N_4 is an irrelevant image. However, when using N_4 as a query, since there are many other images similar to N_4 (e.g., images in the dashed box), the original query image is not ranked high. Therefore the weight corresponding to N_4 in Eq. (5) is low, and its contribution is limited. Furthermore, 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. The images in the dashed box only receive scores from N_4, their final scores will be relatively small. 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 that our method is not sensitive to the selection of nearest neighbor number k, and even achieves better performance with larger k when many outliers exist. The robustness against outliers is a key advantage of our re-ranking method compared with query expansion, in which careful and time-consuming spatial verification needs to be performed, and yet irrelevant features might be introduced and largely degrade the performance.

5 Application in Mobile Product Image Search

As mentioned in Section 1, the spatially-constrained similarity measure can not only be used to generate the voting maps and localize the objects in the database images, but also be adopted to localize the query object if it is not specified. In this section, we present a simultaneous query object extraction and retrieval algorithm using SCSM in the scenario of mobile product image search.

Mobile product image search aims at identifying a product, or retrieving similar products from a database based on a photo captured from a mobile phone camera. The database images are usually high-resolution images with products well aligned and captured in controlled environment. As a result, the background are mostly clean, and the background color can be easily identified by finding the peak of the color histogram built upon the entire image. The mask of the object in each database image can be accordingly obtained by comparing with the background color.

Meanwhile, Mobile query images are usually taken under different lighting conditions with various background. The search results therefore may be significantly distracted by the background and many irrelevant product images would be retrieved. Unlike previous methods that need user input [30], [31], we use SCSM to automatically localize and extract the query object and reduce the influence of background clutter. In our approach, the query object location and its support map is estimated by aggregating votes from the top-retrieved database images with their object masks. The estimated object support map is then used to generate a trimap for GrabCut [34], by which the query object is segmented.

Since we have obtained the object masks of database images, we use the masks in the top-retrieved images to vote on the query image using SCSM and the same voting process as in Section 3, as illustrated in Fig. 5. We switch off rotation here and uniformly quantize the scale change to four bins in the range of 1/2 and 2. After voting and selecting the best mode, each D accordingly has a prediction of the object location in the query image, which can be characterized by a vector $[x_c, y_c, s]^T$, where, (x_c, y_c) is the location of the object center in the query, and s is the relative scale factor between the query object compared with the object in D. Based on that, a transformed object mask of D is voted at the estimated query location (x_c, y_c) with scale factor s, see Fig. 5c for example.

However, not all the top retrieved images can correctly localize the query object, especially when irrelevant objects are retrieved. Therefore, the outliers need to be excluded. Although sophisticated outlier removal methods such as spatial verification using RANSAC can be adopted here, the computational cost of these methods is typically high, and RANSAC does not handle non-planar, non-rigid, and less

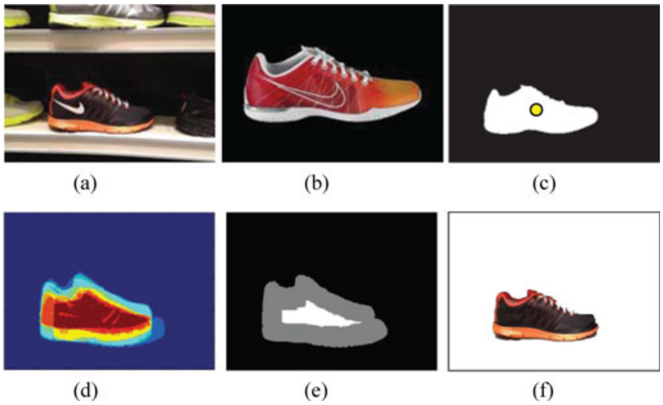


Fig. 5. Illustration of query object localization and extraction. (a) Query image Q, (b) database image D, (c) voted mask of D on the object support map of Q, (d) query object support map by aggregating the voted masks of the top retrieved database images, (e) generated trimap based on the support map, (f) segmentation result using GrabCut with the trimap in (e).

textured objects very well. Therefore, we only use their location predictions $[x_c, y_c, s]^T$ to effectively remove the outliers.

Consider that top N retrieved images are used to localize the query object, we get N location predictions $[x_c^i, y_c^i, s^i]^T (i = 1 \dots N)$. Let $[\overline{x}_c, \overline{y}_c, \overline{s}]^T$ be the median values of all the predictions. For each $[x_c^i, y_c^i, s^i]^T$, if the squared distance

$$D = \left(x_c^i - \overline{x}_c\right)^2 + \left(y_c^i - \overline{y}_c\right)^2 + \lambda(s^i - \overline{s})^2 > \tau \tag{6}$$

the corresponding database images will be removed from localization. In Eqn. (6) τ is a predefined threshold, and λ is a weight parameter.

We iterate this outlier removal and vote aggregation process multiple times to refine the object location, in which the median values $[\overline{x}_c, \overline{y}_c, \overline{s}]^T$ are updated after removing some outliers in each iteration. Once the outliers are removed, each inlier database image accumulates a mask at the estimated location with a weight. The weight can be determined as square root of the inverse of the rank, to assign more confidence on votes from higher ranked images. This process generates a soft map indicating the query object support region (Fig. 5d).

This algorithm is very simple, but can very effectively localize the object in the query image. See the first row of Fig. 6 for an example, even when irrelevant objects are retrieved, the location map can still accurately localize the object, while providing higher aggregated voting scores on the object support region. In more challenging cases where appearance variation in the same object category is quite large, irrelevant objects can dominate the top retrieval results. However, even in such challenging cases, our estimated object support map can still provide good predictions for the object location due to matches in the object boundary features, as shown in the second row in Fig. 6.

Once the object support map is generated, we use it to generate a trimap for GrabCut [34]. We first normalize the support map to a gray-scale image and perform dilation on the map. The pixels below a threshold (< 50) are set as background. Erosion is also performed, and the pixels above a high threshold (> 200) are set as foreground. All the other regions are labeled as uncertain. See Fig. 5e for an example, the black regions represent the background, and the white and gray regions indicate the foreground and uncertain areas, respectively. In more challenging retrieval tasks (e.g., retrieving objects of the same semantic category but with large appearance changes, see Fig. 5), since shape information is not obvious in the estimated support map, to avoid false foreground labeling, only background and uncertain regions are labeled. Such a trimap is used as an input for GrabCut, and the final segmentation result is obtained as shown in Fig. 5f. Experimental results show that the overall segmentation results using our trimap are better than GrabCut with bounding boxes as user input.

We then extract the query object and fill the query image with a clean background to generate a new query, which is then used to perform search using Eq. (1) in the next round. By reducing the background influence, the retrieval performance is dramatically improved. Therefore we can further use the refined search results to update the query object localization and segmentation. By performing query object extraction and object search in an iterative way, the results of localization, segmentation and retrieval are simultaneously boosted. In practice, we observe that 1 or 2 iterations besides the initial search suffice for achieving very good segmentation results. All our results are reported with only 1 additional iteration. The entire procedure is summarized in Algorithm 2.

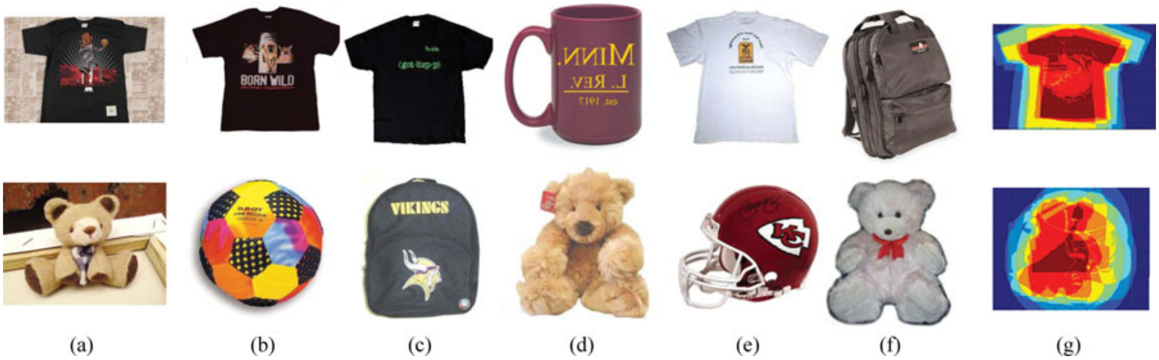


Fig. 6. Our query object localization method is robust to retrieved irrelevant objects. (a) Query images, (b)-(f) top 5 retrieved images, (g) voted object support maps.

Algorithm 2: Mobile product image search by query object extraction.

```
$Q$ = query image, $D_i$ is the $i$-th retrieved
database image.
for iteration = 1 to $k$ do
    Top $N$ retrieved images $\{D_i\}$ ← Search with
    $Q$ as query;
    if iteration = $k$ then
        stop, return retrieved results;
    end
    else
        Prediction $[x_c^i, y_c^i, s^i]^T$ in $Q$ ← Voting with
        $D_i$;
        Remove the outliers using Eqn.6;
        Object support map ← Mask aggregation
        using $[x_c^i, y_c^i, s^i]^T$;
        Trimap for Grabcut ← Thresholding on
        object support map;
        New query $Q'$ ← Grabcut and fill with a
        clean background;
        $Q \leftarrow Q'$;
    end
end
```

6 Experiments

We have implemented our own retrieval system with SIFT descriptors [3] with the gravity constraint [12] and fast approximate k-means clustering [35].We first evaluate our approach (i.e., SCSM and k-NN re-ranking) in the scenario of general, targeted object retrieval on four public data sets, and then evaluate the application of SCSM to mobile product search.

6.1 General Object Retrieval

6.1.1 Data Sets and Implementation Details

We evaluate our approach on four public data sets: *Oxford building*,[4] *Paris*,[5] *INRIA Holidays*,[6] and *University of Kentucky*.[7] 100,000 and 1M Flickr images downloaded with random tags are also added to *Oxford* as distractors to form the *Oxford105k* and *Oxford 1M* data set. 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 data sets. 1M vocabularies are trained for *Oxford* and *Paris*, A 200k vocabulary is trained for *INRIA* as in [13]. 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 data sets except *INRIA Holidays*, as there are only one or two relevant images for most queries in this data set.

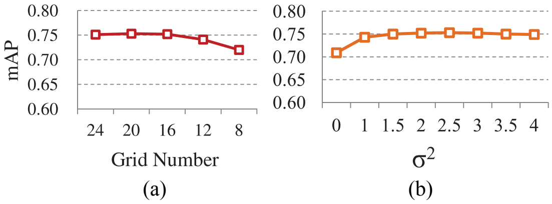


Fig. 7. mAP on *Oxford5k* with different parameters. (a) Voting map size, (b) Gaussian weight. It shows that SCSM is not sensitive to wide range of grid numbers and Gaussian weights.

In evaluation, as in most of previous methods, the retrieval accuracy on the first three data sets and their extensions is measured with the mean average precision (mAP), while the performance measure on the *Kentucky* data set is the top-4 score, i.e., the average number of relevant images in the query's top 4 retrieved images as in [36].

6.1.2 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 size of voting map (i.e., the grid number of the voting map), and σ^2 in the Gaussian smoothing $\exp(-d/\sigma^2)$.

The mAP on *Oxford5k* with different map sizes is shown in Fig. 7a. 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 and used in our experiments, which allows us to encode the feature location in a 1-byte integer, i.e., quantize the X- and Y- coordinates of the features to $0, 1, \ldots, 15$ and further encode them as a 8-bit integer $l_f = y_f * 16 + x_f$. Without downsizing and encoding, each of the original X- and Y- coordinates needs to be represented using a 2-byte integer. Therefore the storage of the encoded feature locations is only $1/4$ of the original size, which has largely reduced the memory cost.

The performance with different σ^2 in voting is shown in Fig. 7b. $\sigma^2 = 0$ means there is no Gaussian smoothing. The results show that enabling Gaussian smoothing is noticeably better than voting on one grid. It is easy to understand as such a Gaussian smoothing allows object deformation and also reduces quantization errors. However, once the Gaussian smoothing 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 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 data sets. 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 0.752 to 0.729 and 0.685 respectively). This indicates SCSM is more scalable to larger databases.

We also compared SCSM with spatial verification [2] after the initial search on *Oxford5k* in terms of retrieval performance. We implemented their algorithm and tested using the same experimental setting as in SCSM. The results

[4] http://www.robots.ox.ac.uk/~vgg/data/oxbuildings.
[5] http://www.robots.ox.ac.uk/~vgg/data/parisbuildings.
[6] http://lear.inrialpes.fr/~jegou/data.php.
[7] http://www.vis.uky.edu/~stewe/ukbench.

TABLE 1
The Performance of Our Method on Public Data Sets

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 2
Comparisons with RANSAC on *Oxford5k*

Methods	BoW	BoW+RANSAC	SCSM
Mean average precision	0.649	0.657	0.752
Search time per query	0.084s	4.3s	0.089s

TABLE 3
Comparisons of SCSM with Other Spatial Models, Including: Geometric-Preserving Visual Phrases[10], Local Bag-of-Features[5], Spatial Bag-of-Features[11], and Hamming Embedding[13]

Datasets	SCSM	[10]	[5]	[11]	[13]
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	-	-

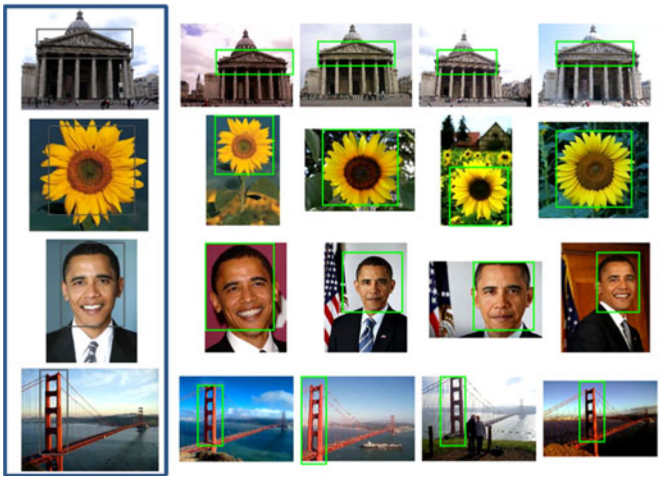


Fig. 8. Examples of object localization by SCSM.The images in the first column are the queries, while the localized results on the top-4 ranked images are presented.

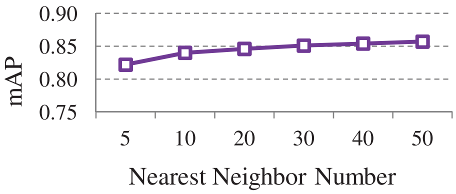


Fig. 9. mAP on *Oxford5k* using different NN numbers in re-ranking. Our method gets better when k becomes larger, indicating it is robust to outliers.

regarding the performance as well as the search time are listed in Table 2. The performance improvement of RANSAC over bag-of-words is not as large as reported in [2] (which shows up to 5 percent increase). The main reason is that we have quantized and encoded the feature locations in a 1-byte integer for memory efficiency. As a result, the estimation of the parametric affine model in RANSAC, which is quite sensitive to outliers, is less accurate, and false matches may be easily introduced. It also explains that in our experiment the best performance of spatial verification is achieved when verifying top-200 retrieved images, and the accuracy would decrease when more images are verified. On the contrary, our voting process with Gaussian smoothing is less affected by the quantization error of feature locations, and SCSM achieves more than 10 percent mAP increase over bag-of-words. Meanwhile, our method is much faster than RANSAC.

We further compared our approach to other methods with spatial models, as listed in Table 3. Our approach outperforms all those methods on all the data sets. Some examples of object retrieval and localization are provided in Fig. 8.

6.1.3 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. 9 shows the performance on *Oxford5k* when we change k (only single iteration is used). Even with only five 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-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 databases. The queries in *Kentucky* has only three 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 data set.

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 4 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* data set, while there are only three other relevant images for each query. Nevertheless, our method performs reasonably well even under such an unfavorable condition.

TABLE 4
Comparisons with Other State-of-the-Art Methods, Including: Philbin et al. CVPR07 [2], Hello Neighbor [24], Burstiness of Visual Elements [32], Query Expansion Revisited [22], Learning a Fine Vocabulary [17], Efficient Representation of Local Geometry [12], Discovery of Co-Occurrence [33], and Learned Descriptors [19]

Datasets	SCSM	SCSM+Re-ranking	[2]	[24]	[32]	[22]	[17]	[12]	[33]	[19]
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	-	-	-	-	-	-	-
Paris	0.741	**0.911**	-	0.803	-	0.805	0.824	-	-	0.689
INRIA	0.762	-	-	-	**0.848**	-	0.758	0.736	-	-
Kentucky	3.52	3.56	3.45	**3.67**	3.64	-	-	-	-	-

6.2 Mobile Product Image Search

For the application of SCSM in mobile product image search, we evaluated our method on two product image data sets, and compared it with the baseline bag-of-words model, our standard SCSM using original query images, as well as the query extraction method by GrabCut with a bounding rectangle as manual initialization (similar as in [30]) in terms of both segmentation and retrieval accuracy.

6.2.1 Data Sets

Since there are no well-known public benchmarks for product image search except some data sets with near-planar objects such as books/CD covers and artworks, we collected two data sets for product image search. The first one is a real-world sports product image (SPI) data set, with 10 categories (hats, shirts, trousers, shoes, socks, gloves, balls, bags, neckerchief and bands) and 43,953 catalog images. The objects in the database images are all well aligned, with clean background. See Fig. 10 for some examples. We also collected 67 query images captured with a mobile phone in local stores under various backgrounds, illumination and viewpoints. The objects in the query images are all shoes, and each has one exact same instance in the database, while there are totally 5,925 catalog images in the shoe category. The task hence is to retrieve the same product from the database images. cumulative match characteristic curve (CMC) is used for performance evaluation, since it is equivalent to a 1:1 identification problem.

The second data set is an object category search (OCS) data set. Given a single query object, objects with the same semantic category need to be retrieved from the database. We collected 868 product images from Caltech 256 [37], in which the objects are positioned at the image center, with clean background. We also collected 60 query images for six categories from internet (each category has 10 queries). The query images contain background clutter, and the objects have large appearance differences, which makes it a very challenging task for object retrieval. See Fig. 11 for some examples. The number of relevant database images for the six categories ranges from 18 to 53. Average precision at rank k , i.e., the percentage of relevant images in the top-k retrieved images, is used to evaluate the performance on this data set.

6.2.2 Results

We trained a vocabulary with 10,580 visual words, which is used throughout all the experimental evaluations. Top 10 retrieved database images are used for query object localization.

The results of all the methods on the SPI data set are shown in Fig. 12a, in which the x-axis indicates the number of retrieved images k, and the y-axis indicates the probabilities that the correct object appears in the top k retrieved images. The CMC curve only shows the results for top 15, as images with low ranks are far less important in most applications. It shows that the standard bag-of-words model cannot retrieve the correct object well for mobile product images. SCSM removes some falsely matched features by more precise spatial matching, and improves the performance. However, it is still severely affected by the features extracted from the background. By automatically extracting the query object, our approach further improves the performance, and even outperforms the retrieve approach with manually initialized query object segmentation. Using query object extraction with SCSM, 40 percent

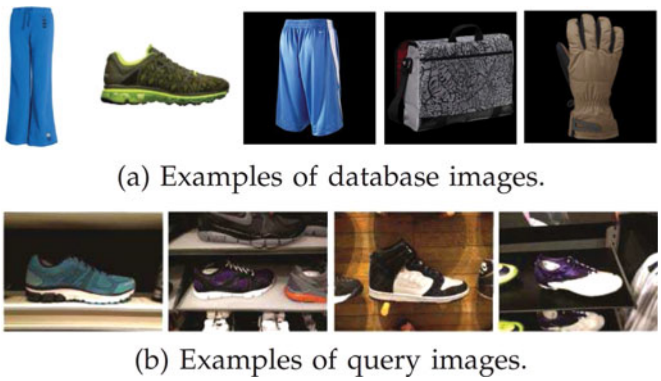

(a) Examples of database images.
(b) Examples of query images.

Fig. 10. Example images in the SPI data set.

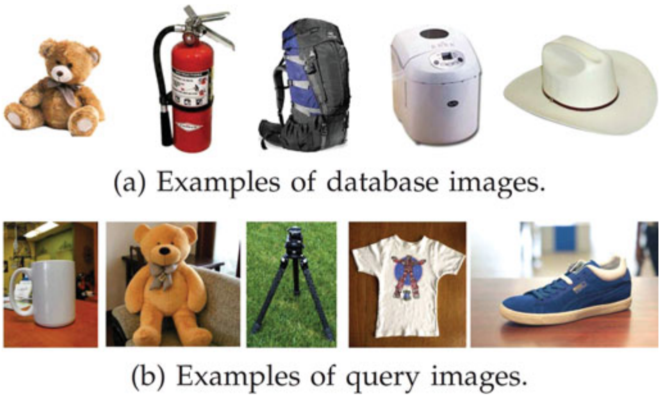

(a) Examples of database images.
(b) Examples of query images.

Fig. 11. Example images in the OCS data set.

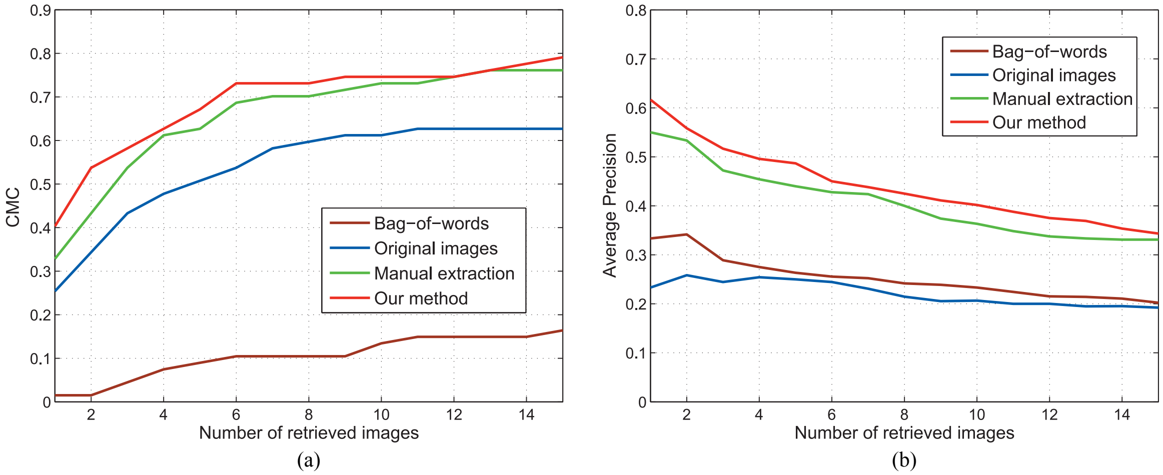


Fig. 12. Performance evaluation on the two collected mobile product image data set. “Original Images” refers to SCSM using the original query image as a whole. (a) CMC curve on the sports product image data set. Our method has significantly better performance than other methods. (b) Average precision at rank k on the object category search data set. Our method consistently yields better precision than other methods.

of the query images rank the correct catalog object at top 1, while the percentages for manually initialized segmentation and using original images are 32.8 and 25.3 percent respectively.

Fig. 12b shows the average precision at rank k, i.e., the average percentage of relevant objects appearing in the top-k retrieved images, on the OCS data set for all the four methods. In this data set, the appearance variation is very large within one category, while SCSM is mainly targeted for instance retrieval instead of object category retrieval. As a result, we can see that the performance of SCSM using original query images is slightly worse than the bag-of-words model. The average precision at rank k for these two methods remains 20 to 30 percent, which indicates that the retrieval task for this data set is quite difficult.

By using our simultaneous query object extraction and retrieval method, the average precision is dramatically improved, as shown in Fig. 12b. Similar to the SPI data set, our method still produces better retrieval performance than manually initialized query object extraction, which demonstrates the effectiveness of our method on this challenging task.

Fig. 13 shows some examples of our query object extraction. We can see our object support maps accurately indicate the object regions. As a result, we can accurately extract the query object, and in many cases achieve more accurate performance than manually initialized segmentation.

6.3 Discussions on Complexity and Scalability

Compared to the bag-of-words model, the additional memory storage of our method are the feature locations in the inverted files. As mentioned in Section 6.12, we encoded each feature location in 1-byte. Suppose we have 500 features in an image, the additional memory for a 1 million data set is 500 MB, which is much smaller than the size of the inverted file.

Another memory cost is the allocation of voting maps during object localization. Since we need to cast votes on multiple scales and rotation angles, a traditional 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, but the memory cost would be relatively high. 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, and some calculations are repeatedly operated during this process. For example, the score in Eq. (3) is always the same for each feature pair yet needs to be calculated every time when traversing the inverted files. To reduce such redundancy, we store all the information needed for voting in the first iteration, including: the locations of the matched features in the database images, the

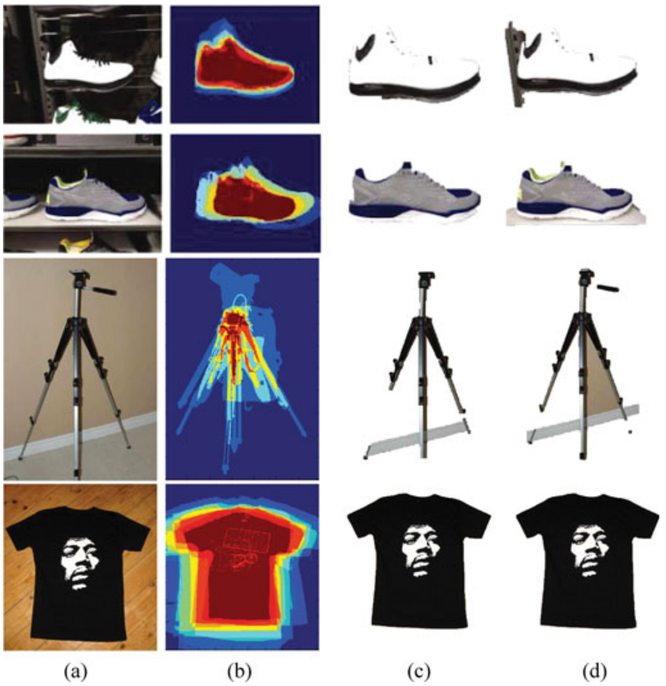


Fig. 13. Examples of query object extraction. (a) Original query images, (b) object support maps, indicating the object regions, (c) automatic object cut using the support maps in (b), (d) GrabCut with manual initialization.

offsets from the features to the center in the query image, and the voting scores. To vote at the next scales and rotation angles, we can directly pull out the stored information instead of repeatedly traversing the inverted files and calculating the similarity scores. The best estimated locations and the similarity scores are then updated accordingly at each scale and rotation angle. Since the vocabulary size is large, the matched features between the query image and a database image are quite sparse, the memory cost of those stored information is far less than that of multiple voting maps.

A voting map is a 16×16 matrix with floating values, which therefore needs 1,024 byte memory storage. Since we only allocate one maps for an image, and the additional memory cost for each image is 1K. Furthermore, 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 data set. Suppose there are 100k database images having common visual words with the query, the needed memory for the voting maps would be around 100M.

In the search process, additional time is needed when we generate the voting maps and do the Gaussian smoothing. With 3G CPU, the average search time for the bag-of-words model is 0.084 s in *Oxford5k*, while SCSM takes 0.089 s in average. k-NN re-ranking with single iteration needs k additional search. However, since the search process of each neighbor is highly independent, it can be easily processed in parallel. With a 8-core PC, the search time of k-NN re-ranking using 30 neighbors is less than four times of the computational cost of SCSM.

In mobile product image search, the most time-consuming step of our method is the GrabCut segmentation. Excluding Grabcut, with 3G CPU, the search procedure for one iteration step takes 0.380 s on average on the SPI database with 45k images, and the whole process for each query can be performed within 3 seconds without code optimization.

7 Conclusions

Unlike previous image retrieval methods that focus on image ranking, we achieves simultaneous object retrieval and localization in the initial search step by employing a new spatially-constrained similarity measure, with a voting-based method. Our SCSM significantly outperforms other spatial models in object retrieval in terms of retrieval accuracy. 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 data sets demonstrates our method achieves the state-of-the-art performance.

Moreover, we apply SCSM in mobile product image search to automatically extract the product from the query image with the help of top-retrieved database images. The influence of the background clutter in the query image is therefore largely avoided, and the retrieval accuracy is significantly improved. Experiments show that our method achieves more than 200 percent improvement over the baseline bag-of-words model, and even outperforms the method with manually initialized query object extraction.

Our SCSM can be integrated in a retrieval system with other components such as soft quantization[15] and learned vocabulary[17] to better serve object and image retrieval. Meanwhile, the localized objects in retrieved images can be adopted in other vision tasks such as image tagging and object detection, which merits further study.

Acknowledgments

This work was partially supported by Adobe Systems, Incorporated, and in part by US 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] J. Sivic and A. Zisserman, "Video Google: A Text Retrieval Approach to Object Matching in Videos," *Proc. IEEE Int'l Conf. Computer Vision (ICCV)*, 2003.

[2] J. Philbin, O. Chum, M. Isard, J. Sivic, and A. Zisserman, "Object Retrieval With Large Vocabularies and Fast Spatial Matching," *Proc. IEEE Conf. Computer Vision and Pattern Recognition (CVPR)*, 2007.

[3] D.G. Lowe, "Distinctive Image Features from Scale-Invariant Keypoints," *Int'l J. Computer Vision*, vol. 60, no. 2, pp. 91-110, 2004.

[4] B. Leibe, A. Leonardis, and B. Schiele, "Combined Object Categorization and Segmentation with an Implicit Shape Model," *Proc. ECCV Workshop Statistical Learning in Computer Vision*, 2004.

[5] Z. Lin and J. Brandt, "A Local Bag-of-Features Model for Large-Scale Object Retrieval," *Proc. 11th European Conf. Computer Vision (ECCV)*, 2010.

[6] C.H. Lampert, "Detecting Objects in Large Image Collections and Videos by Efficient Subimage Retrieval," *Proc. IEEE Int'l Conf. Computer Vision (ICCV)*, 2009.

[7] X. Shen, Z. Lin, J. Brandt, S. Avidan, and Y. Wu, "Object Retrieval and Localization with Spatially-Constrained Similarity Measure and K-NN Reranking," *Proc. IEEE Conf. Computer Vision and Pattern Recognition (CVPR)*, 2012.

[8] X. Shen, Z. Lin, J. Brandt, and Y. Wu, "Mobile Product Image Search by Automatic Query Object Extraction," *Proc. 12th European Conf. Computer Vision (ECCV)*, 2012.

[9] Z. Wu, Q. Ke, M. Isard, and J. Sun, "Bundling Features for Large Scale Partial-Duplicate Web Image Search," *Proc. IEEE Conf. Computer Vision and Pattern Recognition (CVPR)*, 2009.

[10] Y. Zhang, Z. Jia, and T. Chen, "Image Retrieval with Geometry-Preserving Visual Phrases," *Proc. IEEE Conf. Computer Vision and Pattern Recognition (CVPR)*, 2011.

[11] Y. Cao, C. Wang, Z. Li, L. Zhang, and L. Zhang, "Spatial-Bag-of-Features," *Proc. IEEE Conf. Computer Vision and Pattern Recognition (CVPR)*, 2010.

[12] M. Perd'och, O. Chum, and J. Matas, "Efficient Representation of Local Geometry for Large Scale Object Retrieval," *Proc. IEEE Conf. Computer Vision and Pattern Recognition (CVPR)*, 2009.

[13] H. Jégou, M. Douze, and C. Schmid, "Hamming Embedding and Weak Geometric Consistency for Large Scale Image Search," *Proc. 10th European Conf. Computer Vision (ECCV)*, 2008.

[14] O. Chum, M. Perd'och, and J. Matas, "Geometric Min-Hashing: Finding a (Thick) Needle in a Haystack," *Proc. IEEE Conf. Computer Vision and Pattern Recognition (CVPR)*, 2009.

[15] J. Philbin, O. Chum, M. Isard, J. Sivic, and A. Zisserman, "Lost in Quantization: Improving Particular Object Retrieval in Large Scale Image Databases," *Proc. IEEE Conf. Computer Vision and Pattern Recognition (CVPR)*, 2008.

[16] X. Wang, M. Yang, T. Cour, S. Zhu, K. Yu, and T.X. Han, "Contextual Weighting for Vocabulary Tree Based Image Retrieval," *Proc. IEEE Int'l Conf. Computer Vision (ICCV)*, 2011.

[17] A. Mikulík, M. Perd'och, O. Chum, and J. Matas, "Learning a Fine Vocabulary," *Proc. 11th European Conf. Computer Vision (ECCV)*, 2010.

[18] H. Jégou, H. Harzallah, and C. Schmid, "A Contextual Dissimilarity Measure for Accurate and Efficient Image Search," *Proc. IEEE Conf. Computer Vision and Pattern Recognition (CVPR)*, 2007.

[19] J. Philbin, M. Isard, J. Sivic, and A. Zisserman, "Descriptor Learning for Efficient Retrieval," *Proc. 11th European Conf. Computer Vision Conf. Computer Vision (ECCV)*, 2010.

[20] H. Jégou, M. Douze, C. Schmid, and P. Pérez, "Aggregating Local Descriptors into a Compact Image Representation," *Proc. IEEE Conf. Computer Vision and Pattern Recognition (CVPR)*, 2010.

[21] O. Chum, J. Philbin, J. Sivic, M. Isard, and A. Zisserman, "Total Recall: Automatic Query Expansion with a Generative Feature Model for Object Retrieval," *Proc. IEEE Int'l Conf. Computer Vision (ICCV)*, 2007.

[22] O. Chum, A. Mikulík, M. Perd'och, and J. Matas, "Total Recall II: Query Expansion Revisited," *Proc. IEEE Conf. Computer Vision and Pattern Recognition (CVPR)*, 2011.

[23] G. Tolias and Y. Avrithis, "Speeded-Up, Relaxed Spatial Matching," *Proc. IEEE Int'l Conf. Computer Vision (ICCV)*, 2011.

[24] D. Qin, S. Gammeter, L. Bossard, T. Quack, and L. VanGool, "Hello Neighbor: Accurate Object Retrieval with K-Reciprocal Nearest Neighbors," *Proc. IEEE Conf. Computer Vision and Pattern Recognition (CVPR)*, 2011.

[25] D.C.G. Pedronette and R. da S. Torres, "Exploiting Contextual Spaces for Image Re-Ranking and Rank Aggregation," *Proc. ACM First Int'l Conf. Multimedia Retrieval (ICMR)*, 2011.

[26] Y. Jing and S. Baluja, "Pagerank for Product Image Search," *Proc. 17th Int'l Conf. World Wide Web (WWW)*, 2008.

[27] X. Lin, B. Gokturk, B. Sumengen, and D. Vu, "Visual Search Engine for Product Images," *Proc. SPIE*, 2008.

[28] B. Girod, V. Chandrasekhar, D. Chen, N.-M. Cheung, R. Grzeszczuk, Y. Reznik, G. Takacs, S. Tsai, and R. Vedantham, "Mobile Visual Search," *IEEE Signal Processing Magazine*, vol. 28, no. 4, July 2011.

[29] V. Chandrasekhar, D. Chen, S. Tsai, N.-M. Cheung, H. Chen, G. Takacs, Y. Reznik, R. Vedantham, R. Grzeszczuk, J. Bach, and B. Girod, "The Stanford Mobile Visual Search Dataset," *Proc. ACM Multimedia Systems Conf.*, 2011.

[30] J. He, T.-H. Lin, J. Feng, and S.-F. Chang, "Mobile Product Search with Bag of Hash Bits," *Proc. 19th ACM Int'l Conf. Multimedia (MM)*, 2011.

[31] J. He, X. Liu, T. Cheng, J. Feng, T.-H. Lin, H. Chung, and S.-F. Chang, "Mobile Product Search with Bag of Hash Bits and Boundary Reranking," *Proc. IEEE Conf. Computer Vision and Pattern Recognition (CVPR)*, 2012.

[32] H. Jégou, M. Douze, and C. Schmid, "On the Burstiness of Visual Elements," *Proc. IEEE Conf. Computer Vision and Pattern Recognition (CVPR)*, 2009.

[33] O. Chum and J. Matas, "Unsupervised Discovery of Co-Occurrence in Sparse High Dimensional Data," *Proc. IEEE Conf. Computer Vision and Pattern Recognition (CVPR)*, 2010.

[34] C. Rother, V. Kolmogorov, and A. Blake, "Grabcut: Interactive Foreground Extraction Using Iterated Graph Cuts," *Proc. ACM SIGGRAPH*, 2004.

[35] M. Muja and D.G. Lowe, "Fast Approximate Nearest Neighbors with Automatic Algorithm Configuration," *Proc. VISAPP Int'l Conf. Computer Vision Theory and Applications*, 2009.

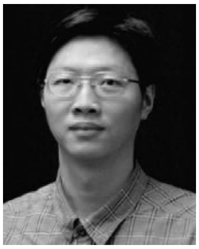
[36] D. Nistér and H. Stewénius, "Scalable Recognition with a Vocabulary Tree," *Proc. IEEE Conf. Computer Vision and Pattern Recognition (CVPR)*, 2006.

[37] G. Griffin, A. Holub, and P. Perona, "Caltech-256 Object Category Dataset," Technical Report 7694, California Inst. of Technology, http://authors.library.caltech.edu/7694, 2007.

Xiaohui Shen received the BS and MS degrees from the Automation Department of Tsinghua University, China, in 2005 and 2008 respectively, and the PhD degree from the EECS Department of Northwestern University in 2013. He is currently a research scientist at Adobe Research, San Jose, California. His research interests include image/video processing and computer vision. He is a member of the IEEE.

Zhe Lin received the BEng degree in automatic control from the University of Science and Technology of China in 2002, the MS degree in electrical engineering from the Korea Advanced Institute of Science and Technology in 2004, and the PhD degree in electrical and computer engineering from the University of Maryland, College Park, in 2009. He has been a research intern at Microsoft Live Labs Research. He is currently a senior research scientist at Adobe Research, San Jose, California. His research interests include object detection and recognition, image classification, content-based image and video retrieval, human motion tracking, and activity analysis. He is a member of the IEEE.

Jonathan Brandt received the BS degree in electrical engineering from the University of Illinois, Urbana-Champaign, and the MS and PhD degrees in computer science from the University of California, Davis. He is currently a computer vision researcher at Adobe Research, where he manages the Vision Technology Group. Prior to Adobe, he was a visiting associate professor at the Japan Advanced Institute of Science and Technology and a Member of the Technical Staff at Silicon Graphics. His research interests include image retrieval, scene categorization, object detection and recognition, and machine learning. He is a member of the IEEE.



Ying Wu received the BS degree in automatic control from the Huazhong University of Science and Technology, China, in 1994, the MS degree from the Automation Department, Tsinghua University, China, in 1997, and the PhD degree in electrical and computer engineering from the University of Illinois at Urbana-Champaign (UIUC) in 2001. He is currently a full professor in the EECS Department of Northwestern University. His research interests include computer vision/graphics, image/video processing, multimedia, machine learning, human motion, human-computer intelligent interaction, and virtual environments, etc. He received the US National Science Foundation (NSF) CAREER award in 2003 and the Robert T. Chien Award at UIUC in 2001. He is a senior member of the IEEE.