Randomized Spatial Context for Object Search

Yuning Jiang, Jingjing Meng, *Member, IEEE*, Junsong Yuan, *Senior Member, IEEE*, and Jiebo Luo, *Fellow, IEEE*

***Abstract*— Searching visual objects in large image or video data sets is a challenging problem, because it requires efficient matching and accurate localization of query objects that often occupy a small part of an image. Although spatial context has been shown to help produce more reliable detection than methods that match local features individually, how to extract appropriate spatial context remains an open problem. Instead of using fixed-scale spatial context, we propose a randomized approach to deriving spatial context, in the form of spatial random partition. The effect of spatial context is achieved by averaging the matching scores over multiple random patches. Our approach offers three benefits: 1) the aggregation of the matching scores over multiple random patches provides robust local matching; 2) the matched objects can be directly identified on the pixelwise confidence map, which results in efficient object localization; and 3) our algorithm lends itself to easy parallelization and also allows a flexible tradeoff between accuracy and speed through adjusting the number of partition times. Both theoretical studies and experimental comparisons with the state-of-the-art methods validate the advantages of our approach.**

***Index Terms*— Object search, spatial context, random partition.**

I. Introduction

THE matching of local visual features plays a critical role in the state-of-the-art systems for visual object search and detection. The fundamental problem is to measure the similarity between an object (query) and a sub-region of an image. Sub-regions with the highest similarity scores are identified as the detection or search results. One category of methods represents each image as a collection of local features, and assume that they are independent from each other. Thus the matching score of the whole or subimage can be calculated as the summation of the matching scores of its individual features. Such a Naive-Bayes assumption, e.g., Naive-Bayes Nearest Neighbor classifier [1], [2], [23], [39], has led to successes in visual object recognition, detection and search.

However, as local features are in fact not spatially independent, rather than matching local features individually, some methods propose to consider the spatial context for matching. For example, a group of co-located visual features can be bundled together and matched as a whole. The benefits of introducing such a feature group for visual matching have been proven to generate more reliable and discriminative results than matching individual features, thus leading to a higher precision in visual matching and search [7], [12], [19], [20], [25], [28], [34], [40], [42].

Despite previous successes in employing spatial context for more discriminative visual feature matching, e.g. visual phrases [41], [43], [44] or bundled features [13], [36], one problem remains unsolved: how to select the appropriate spatial context when matching local features?

In general, there are two ways to select the spatial context. The first category of methods relies on image segments or regions to determine the spatial context [29], [30], [35], [37], [42], where local features located in the same image region or segment are bundled together and matched as a whole. Although such spatial context is reasonable, this approach is highly dependent on the quality of image segmentation or region detection results, which require a time consuming pre-process to obtain and are usually unreliable.

The second category of methods selects the spatial context at a relatively fixed scale. The most common way is to bundle each local point with its k spatial nearest neighbors, namely k-NN group [32], [41]. However, as reported in [42], unstable local features may be detected when images are resized or stretched, resulting in varying numbers of detected local features at different scales. Hence for each local point, its k-NN group may be totally different from that at a different scale, as shown in Fig. 2(a). Therefore, spatial context provided by the k-NN group is not scale invariant. Furthermore, it is difficult to determine an appropriate k. Using a larger k reveals more contextual information while running a higher risk of introducing noise from the background. Moreover, if the user wants to change the value of k, he will need to re-calculate the spatial threshold and re-index the feature groups all over.

Grid-based local feature bundling is an alternative to the k-NN group for the fixed-scale spatial context [13]. An image is partitioned into fixed-size grids and all features within each grid are bundled together and matched as a whole. However, similar to the k-NN selection, the grid-based spatial context is also not invariant to scale and it is difficult to choose a proper grid size without knowing the size of the target object. In addition, as shown in Fig. 2(b), local points near the edges of the grids may be separated from their nearest neighbors,

Manuscript received May 22, 2014; revised October 20, 2014; accepted January 6, 2015. Date of publication February 24, 2015; date of current version March 23, 2015. This work is supported in part by Nanyang Assistant Professorship M58040015.040 and Singapore MoE Tier-1 Grant M4011272.040. The associate editor coordinating the review of this manuscript and approving it for publication was Prof. Amit K. Roy Chowdhury.

Y. Jiang, J. Meng, and J. Yuan are with the Department of Electrical and Electronics Engineering, Nanyang Technological University, Singapore 639798 (e-mail: jyn@megvii.com; jingjing.meng1@gmail.com; jsyuan@ntu.edu.sg).

J. Luo is with the Department of Computer Science, University of Rochester, Rochester, NY 14627 USA (e-mail: jluo@cs.rochester.edu).

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

Digital Object Identifier 10.1109/TIP.2015.2405337

therefore the grids may not accurately describe spatial context.

We believe that an ideal spatial context selection for object search task should satisfy the following requirements: 1) it can support robust object matching despite scale variations, rotation and partial occlusions; 2) it can support fast object localization in the cluttered backgrounds; and 3) it can be efficiently extracted and indexed.

To address the three requirements, we propose a new spatial context selection approach based on random partition of images. After extracting local invariant features, we randomly partition the image multiple times to form a pool of overlapping image patches. Each patch bundles the local features inside it and is characterized by a histogram of visual words. Essentially, for each individual point, we generate a collection of random image patches in varying sizes and rectangular shapes as its spatial context. Instead of matching an individual feature point, we match its randomized spatial context, i.e., all local features in a random image patch. Fig. 2(c) explains the randomized spatial context. For each spatial context (i.e., each image patch), we independently calculate the similarity score between it and the query object as its voting weight. The final confidence score of each pixel in the image is calculated as the expectation of the voting weights of all patches that contain this pixel, and we record the pixel-wise confidence scores on a confidence map. Finally, the matched regions can be identified on the confidence map as the detected objects.

Our random partition approach provides several benefits. First of all, compared with the state-of-the-art systems for object search, our approach results in better matching and thus better retrieval performance thanks to the randomized spatial context. Moreover, it is robust to the scale variations and partial occlusions of the objects. Second, our spatial random partition-based patch voting scheme indirectly solves the object localization problem, as the object can be segmented out directly from the confidence map. This largely reduces the computational cost compared with the subimage search methods for object localization [6], [17], [18]. Third, our approach allows the user to make a trade-off between effectiveness and efficiency through adjusting the number of partition times on-line without re-indexing the database; this is important for a practical search system. In addition, the design of the algorithm makes it ready for parallelization and thus well suited for large scale applications.

To evaluate our spatial random partition approach, we conduct visual object search first on a movie database, and then on a benchmark logo database with a million-level image database from Flickr as distractors. In Fig. 1, we provide some sample results of visual object search. Although in some cases it is challenging even for human observers to find and locate the small query objects in the cluttered scenes, our algorithm performs very well.

The remainder of the paper is organized as follows: Section II introduces the background and related work on object search in recent years. In Section III, we present our random partition-based object search algorithm to account for multi-scale spatial context. In Section IV, we provide theoretical validation of our algorithm, and describe its parallel implementation. The experimental results are shown in Section V, followed by the conclusions in Section VI.

Fig. 1. Visual object search is a more challenging problem than whole-image retrieval, since the target objects in the database can vary greatly in location, scale, orientation and appearance.

II. Related Work

The developments of invariant local visual features and fast image indexing and search algorithms have led to great successes in image search and retrieval. Given an image query, the state-of-the-art image retrieval systems [3], [4], [8]–[11], [21], [24], [38], [45], [46] have been able to retrieve and rank similar or near-duplicate images within million-level databases. Despite rapid progress in whole-image retrieval, visual object search, whose goal is to find and accurately locate the target object in image collections, remains a challenging problem. This is due to the fact that the target objects usually appear in the cluttered backgrounds, occupying a very small and unknown portion of an image, and can differ significantly from the query because of the changes in scale, view point and color, as well as partial occlusions. In this respect, visual object search can be viewed as two tasks combined: object matching and object localization.

For object matching, the bag-of-visual-words (BoVW) scheme [5], [16], [26], [27], [31], [33] has been widely adopted although there is the obvious drawback of quantizing high-dimensional descriptors into visual words. In general, there are two ways to address the quantization error incurred by BoVW scheme. One is to match individual descriptors in the feature space directly, e.g. the Naive-Bayes Nearest Neighbor (NBNN) classifier proposed in [1] and [2]. The method in [23] uses the NBNN-classifier and calculates the mutual information score between each local feature and the query object independently. However, the NBNN-based algorithms are all under the Naive-Bayes assumption that each feature point is independent from the others, therefore they can fail when the assumption is violated. Besides, searching nearest neighbors in the feature space is costly both in memory and time.

Another way to mitigate the quantization error is to consider spatial context instead of an individual point, which is also used in other image-related applications. By bundling co-occurring visual words within a constrained spatial distance into a visual phrase [41], [43], [44] or feature group [42] as the

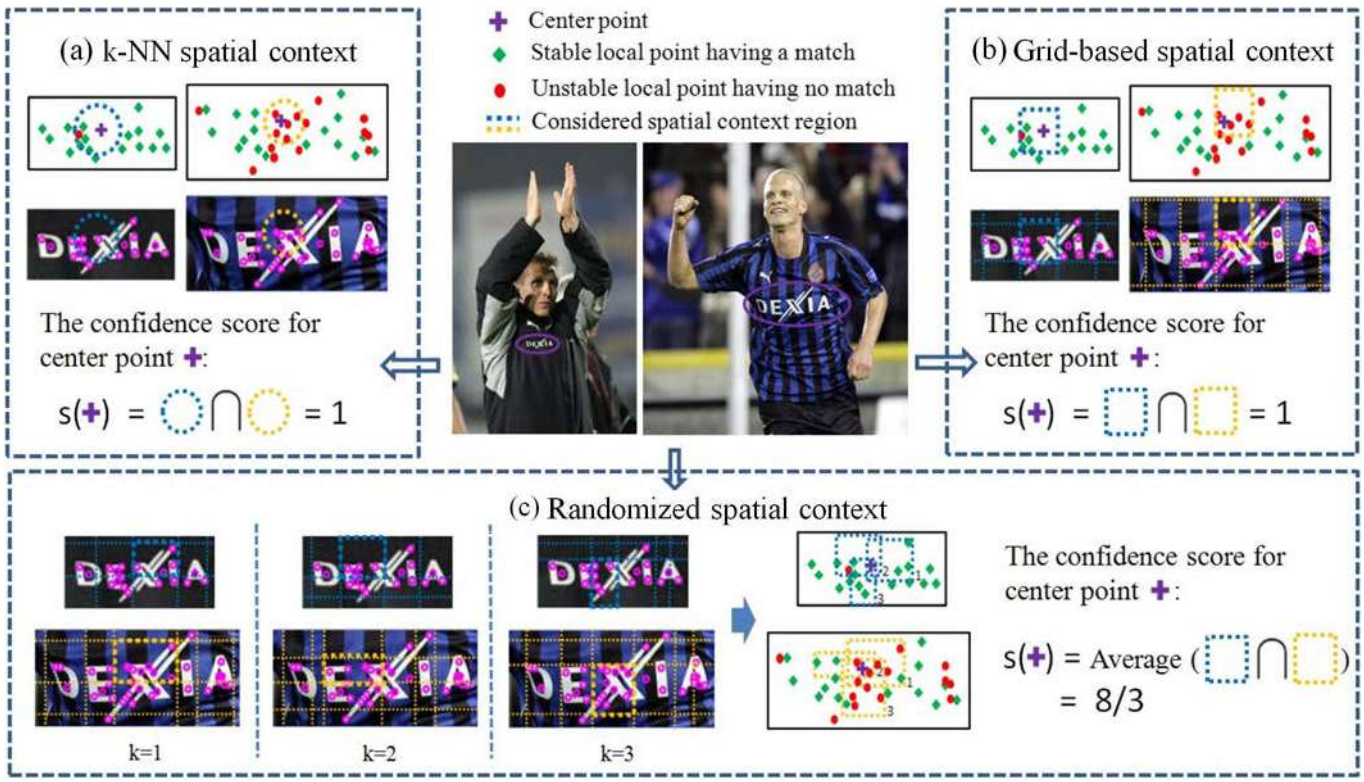


Fig. 2. Comparison between different ways to choose the spatial context. The similarity between two spatial context regions are calculated as the number of matched points (including the center point) in them, denoted by ∩.

basic unit for object matching, the spatial context information is incorporated to enhance the discriminative power of visual words. In [32], each local feature is combined with its k spatial nearest neighbors to generate a feature group. And in [13], each image is partitioned into non-overlapping grid cells which bundle the local features into grid features. However, unlike the whole-image retrieval problem, our target object may appear at all possible scales. Therefore such feature groups are not scale invariant and not capable of handling the various objects without a *priori* knowledge. Also it is not a trivial problem to select the optimal k or grid size. Moreover, it is not convenient if the user wants to change the scale of feature group because he would need to re-index the whole database. As an earlier version of this paper, [14] proposes the Randomized Visual Phrases (RVPs) to consider spatial context in varying shapes and sizes, and thereby provides a robust partial matching.

For object localization, in most previous work the relevant images are retrieved firstly and then the object location is determined as the bounding box of the matched regions in the post-processing step through a geometric verification, such as RANSAC [26] or neighboring feature consistency [32]. Since geometric verification methods are usually computationally expensive, they are applied only to the top images in the initial ranking list. Alternatively, efficient subimage retrieval (ESR) [17] and efficient subwindow search (ESS) [18] are proposed to find the subimage with maximum similarity to the query. In addition, spatial random partition is proposed in [40] to discover and locate visual common objects.

III. Multi-Scale Spatial Context via Random Partition

Given a database $\mathcal{D} = \{\mathcal{I}_i\}$ of I images, our objective is to retrieve all the images $\{\mathcal{I}_g\}$ that contain the object, and identify the object's locations $\{\mathcal{L}_g\}$, where $\mathcal{L}_g \subset \mathcal{I}_g$ is a subimage of $\mathcal{I}_g$. An overview of our proposed algorithm is presented in Alg. 1 and Fig. 3.

A. Image Description

We first represent each image $\mathcal{I}_i \in \mathcal{D}$ as a collection of local interest points, denoted by $\{f_{i,j}\}$. Follow the BoVW scheme, each local descriptor f is quantized to a visual word using a vocabulary of V words, represented as $w = (x, y, v)$, where (x, y) is the location and $v \in \{1, \ldots, V\}$ is the corresponding index of the visual word. Using a stop list analogy, the most frequent visual words that occur in almost all images are discarded. All feature points are indexed by an inverted file so that only words that appear in the queries will be checked.

B. Spatial Random Partition

We randomly partition each image $\mathcal{I}_i$ into $M \times N$ non-overlapping rectangular patches and perform such partition K rounds independently. This results in a pool of $M \times N \times K$ image patches for each $\mathcal{I}_i$, denoted as: $\mathcal{P}_i = \{P_i\}$. Note that for a given partition $k \in \{1, 2, \ldots, K\}$ the $M \times N$ patches are non-overlapping, while the patches from different partition rounds may overlap. Since in the k_{th} partition, each

Algorithm 1 Spatial Random Partition for Object Search

Input:

an image database $\mathcal{D} = \{\mathcal{I}_i\}$

the query object Q_+ (sometimes the negative query Q_- is also given to model the backgrounds),

Output:

subimages $\{\mathcal{L}_g\}$, which contain the retrieved object.

1: **Partition:** $\forall \mathcal{I}_i \in \mathcal{D}$, partition it into $M \times N$ patches for K times randomly, and obtain a pool of patches $\mathcal{P}_i = \{P_i\}$ containing $M \times N \times K$ patches (Sec. III-B).

2: **Matching:** $\forall P_i \in \mathcal{P}_i$, match it against the query object Q_+ (or both Q_+ and Q_-), and assign it a weight proportion to its similarity to the query object Q_+ (Sec. III-C).

3: **Voting:** $\forall P_i \in \mathcal{P}_i$, distribute its voting weight to each pixel it contains, and a pixel-wise confidence map is generated for each image $\mathcal{I}_i$ (Sec. III-C).

4: **Localization:** $\forall \mathcal{I}_i \in \mathcal{D}$, segment out the dominant region $\mathcal{L}_i$ from its confidence map as the object location (Sec. III-D).

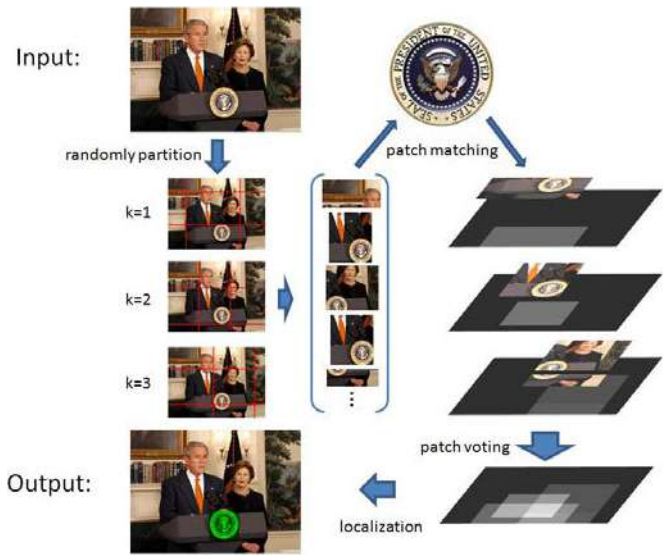


Fig. 3. Illustration of object search via spatial random partition ($M \times N \times K = 3 \times 3 \times 3$). The input includes a query object and an image containing the object, while the output is the segmentation of the object (highlighted in green).

pixel t falls in a unique patch $P_t^{(k)}$, in total there are K patches containing the pixel t after K rounds of partitions, denoted as:

$$\Omega_t^K = \{P_t^{(k)}\} = \{P_i \mid t \in P_i\}, \quad k = 1, \ldots, K. \tag{1}$$

Then each patch P is composed of a set of visual words, denoted as $P : \{w | w \in P\}$, and is further characterized as a V-dimensional histogram h_P recording the word frequency of P.

Given each pixel $t \in \mathcal{I}_i$, we consider the collection of all possible patches containing t, denoted by $\Omega_t = \{P_t\}$. Then after K rounds of partitions, we essentially sample the collection K times and obtain a subset $\Omega_t^K = \{P_t^{(k)}\}_{k=1}^K \subset \Omega_t$. The sizes and aspect ratios of the patches in the subset Ω_t^K are random since these patches result from K independent random partitions. Therefore, for the pixel t, its spatial context at different scales has been taken into consideration by matching the random patch set Ω_t^K against the query object. To simplify the problem, we assume the probability that each patch will be sampled in the k_{th} partition is the same, which means $p(P_t^{(k)}) = \frac{1}{|\Omega_t^K|} = \frac{1}{K}$ is a constant.

TABLE I
SEVERAL VECTOR DISTANCES FOR PATCH MATCHING

symbol	similarity function
$Bin(h_Q, h_P)$	$\sum_v \min(h_Q^v h_P^v, 1)$
$HI(h_Q, h_P)$	$\sum_v \min(h_Q^v, h_P^v)$
$NHI(h_Q, h_P)$	$\sum_v \min(h_Q^v, h_P^v) / \sum_v \max(h_Q^v, h_P^v)$
$dot(h_Q, h_P)$	$\sum_v h_Q^v h_P^v$
$\rho_{bhatt}(h_Q, h_P)$	$\frac{1}{\sqrt{\|h_Q\|_1 \|h_R\|_1}} \sum_k \sqrt{h_Q^v h_P^v}$

C. Patch Matching and Voting

Given a pixel t, its confidence score $s(t)$ is calculated as the expectation of similarity scores of its spatial context, i.e., the patch P_t, and the query object Q_+, denoted as:

$$\begin{aligned} s(t) = E(s(P_t)) &= \sum_{P_t \in \Omega_t} p(P_t) s(P_t) \\ &\approx \sum_{P_t^{(k)} \in \Omega_t^K} p(P_t^{(k)}) s(P_t^{(k)}) = \frac{1}{K} \sum_{k=1}^K s(P_t^{(k)}), \end{aligned} \tag{2}$$

where the expectation is estimated using the subset Ω_t^K instead of the complete collection Ω_t. Now our problem becomes how to define the similarity score $s(P)$ for each patch P. And as mentioned in [23], the input types of a practical search system could be 1) only positive query Q_+, i.e., the target which user wants to search; 2) both positive query Q_+ and negative query Q_-, i.e., the noise which user wants to avoid. Considering these two kinds of cases, here we provide two ways to address the patch matching problem, respectively.

1) Normal Patch Matching: First let us consider the case that only positive query Q_+ is available, which is represented as the word-frequency histogram h_{Q+} as well. In this case we can adopt any vector distance listed in Tab. I as the matching kernel, and match each patch against the query just like a whole image. Here we use the normalized histogram intersection $NHI(\cdot)$ as an example:

$$s(t) = \frac{1}{K} \sum_{k=1}^K s(P_t^{(k)}) = \frac{1}{K} \sum_{k=1}^K NHI(h_{P_t^{(k)}}, h_{Q_+}). \tag{3}$$

In addition, some other vector distances can be chosen instead of $NHI(\cdot)$, resulting in reduced computational cost, as shown in Tab. I. The comparison between all these distances will be discussed in later experiments.

2) Discriminative Patch Matching: Then we consider the case in which both positive queries Q_+ and negative queries Q_- are given. This case is similar to the discriminative grid matching [13], and we calculate the pixel-wise mutual information score $MI(Q_+, P)$ as the similarity score $s(P)$ as follows:

$$\begin{aligned} s(P) &= MI(Q_+, P) = \log \frac{p(P|Q_+)}{p(P)} \\ &= \log \frac{p(P|Q_+)}{p(Q_+)p(P|Q_+) + p(P|Q_-)p(Q_-)} \\ &= \log \frac{1}{p(Q_+) + \frac{p(P|Q_-)}{p(P|Q_+)} p(Q_-)}. \end{aligned} \tag{4}$$

We estimate the likelihood $p(P|Q)$ in Eqn. 4 using the normalized histogram intersection:

$$p(P|Q) = NHI(h_P, h_Q) = \frac{|h_P \cap h_Q|}{|h_P \cup h_Q|} \in [0, 1]. \tag{5}$$

Note that according to Eqn. 4, we need to estimate the prior probability $p(Q_+)$ or $p(Q_-)$, which is a constant for all pixels and patches. In the paper we assume the prior of positive and negative class are equal, as in [23] and [39]. However this assumption leads to a bias in results since in fact the negative class is much larger than the positive class. We will address the bias when localizing the object.

D. Object Localization

After assigning each pixel $t \in \mathcal{I}_i$ a confidence score, we obtain a pixel-wise confidence map for each image $\mathcal{I}_i$. Object localization then becomes an easy task since we just need to identify the dominant region $\mathcal{L}_i$ from $\mathcal{I}_i$ as the object location:

$$\mathcal{L}_i = \{t | s(t) > thres, \forall t \in \mathcal{I}_i\}. \tag{6}$$

In an ideal case if the confidence map is generated by discriminative patch matching, $thres = 0$ should be used as the threshold, which indicates that the mutual information score between a pixel and the query is zero. However, due to the invalid assumption made in Eqn. 4 (i.e., $p(Q_+)$ equals to $p(Q_-)$), the threshold has a bias from 0. Therefore we set the threshold $thres$ adaptively, which is in proportion to the average confidence score of the whole image $\mathcal{I}_i$:

$$thres_i = \frac{\alpha}{|\mathcal{I}_i|} \sum_{t \in \mathcal{I}_i} s(t), \tag{7}$$

where $|\mathcal{I}_i|$ is the number of the non-zero pixels in $\mathcal{I}_i$ and α is the parameter. Then all the pixels whose confidences are higher than the threshold will be directly segmented out and finally compose the detected regions. The score of a detected region is calculated as the sum of all the scores of the pixels it contains, and its location is returned as a detected target, regardless of the size and shape. And by adjusting the coefficient α, we can modify the bias caused by the assumption to some extent and obtain more accurate localization results.

Moreover, in practice we set the coefficient $\alpha > 1$ to degrade the influence of the noisy points in the image background. From Eqn. 7 it is obvious to see that the threshold cannot be higher than the average confidence score when $\alpha \leq 1$. In the condition, given any a confidence map there must be some relatively salient regions containing higher scores than the threshold, even if the regions are just caused by the isolated points (see the 4_{th} row in Fig. 4). Therefore, with the objective to filter the isolated points, we experimentally use a larger α to heighten the threshold. By doing so, the thresholding strategy favors the matched points to co-locate in a local region since the co-located points will reinforce each other and finally generate a salient enough region to be segmented out; otherwise, if the matched points are distributed sparsely in the map, there may be no dominant region above the same threshold (see Fig. 5). Such a property is important for searching small object such as a logo, because the positive matched feature points are usually co-located in a small local region, while the noisy points are usually distributed sparsely in the background. Thus this thresholding strategy can effectively help to reduce the false alarm detections.

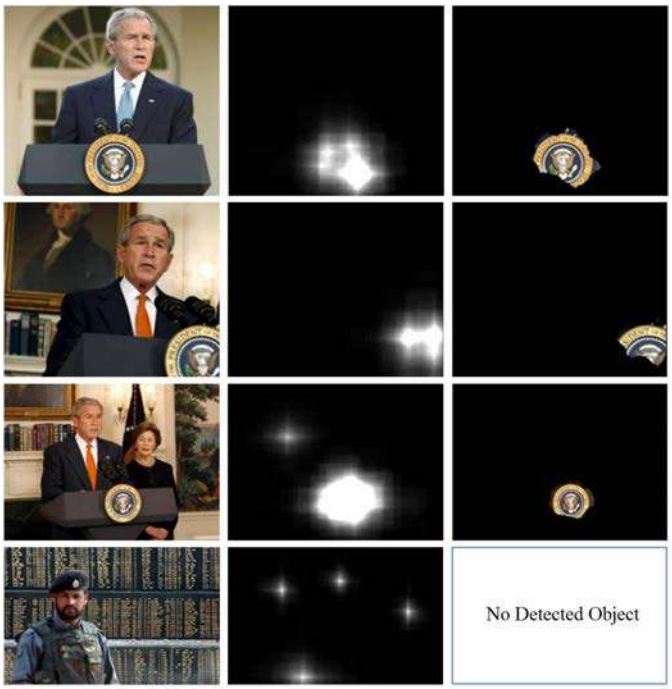


Fig. 4. Examples for voting and localization. The query logo is the same as in Fig. 3. The 1_{st} column shows the original images. The 2_{nd} column shows the confidence maps after 200 random partitions. The 3_{rd} column shows the segmentation results with the coefficient $\alpha = 5.0$. By comparing the last two rows with the first two rows, we can see that our algorithm is robust to the noisy points in the background (3_{rd} row), and can reduce the false alarm detections as well (4_{th} row).

IV. Algorithm Analysis

A. Asymptotic Property

The asymptotic property is given below as the theoretical justification of our algorithm.

Proposition 1 (Asymptotic Property): We consider two pixels $i, j \in \mathcal{I}$, where $i \in \mathcal{G} \subset \mathcal{I}$ is located inside the groundtruth region while $j \notin \mathcal{G}$ is located outside. Suppose $\boldsymbol{S}_i^K$ and $\boldsymbol{S}_j^K$ are the total votes (or scores)

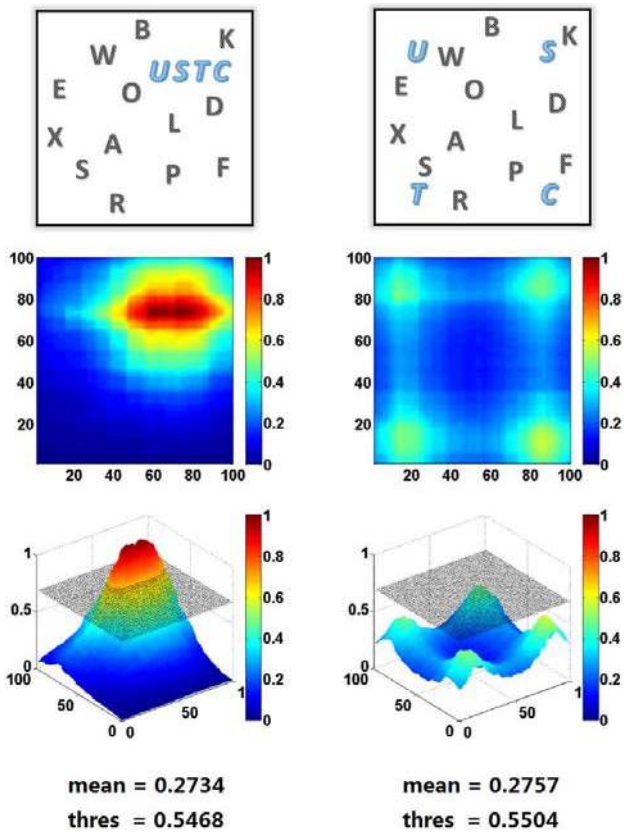


Fig. 5. The simulated experiment for voting and localization. The target object is the *USTC* word (denoted in blue) in the left-top image while the right-top image contains the same letters but not co-located. Their voting maps after 200 rounds are shown in the second row, from which we can see that their average confidence scores are almost the same. That is, the thresholds of the two maps are also very close multiplied by the coefficient ($\alpha = 2$, denoted by the surface in the dash). However, the right image will not be retrieved since it cannot generate such dominant regions above the threshold with these sparsely distributed points.

for i and j, respectively, considering K times random partitions. Both $\mathbf{S}_i^K$ and $\mathbf{S}_j^K$ are discrete random variables, and we have:

$$lim_{K\to\infty}(\mathbf{S}_i^K - \mathbf{S}_j^K) > 0 \tag{8}$$

The above theorem states that when we have enough rounds of partitions for each image, the groundtruth region $\mathcal{G}$ must receive more votes, so that it can be easily discovered and located. The explanation of Proposition 1 is given in the supplementary material because of space limit.

B. Parallel Implementation

One of the most challenging problems for visual object search is the efficiency and scalability, especially for the web-scale databases. On the other hand, nowadays the computational capability of PC has been improved significantly with the advances in hardware. Thanks to the development of multi-core CPU and programmable GPU, we can now divide one computation task into several independent threads and execute them in parallel. However, not all algorithms could be parallel implemented such as some interactive algorithms, in which the computational tasks are highly interrelated.

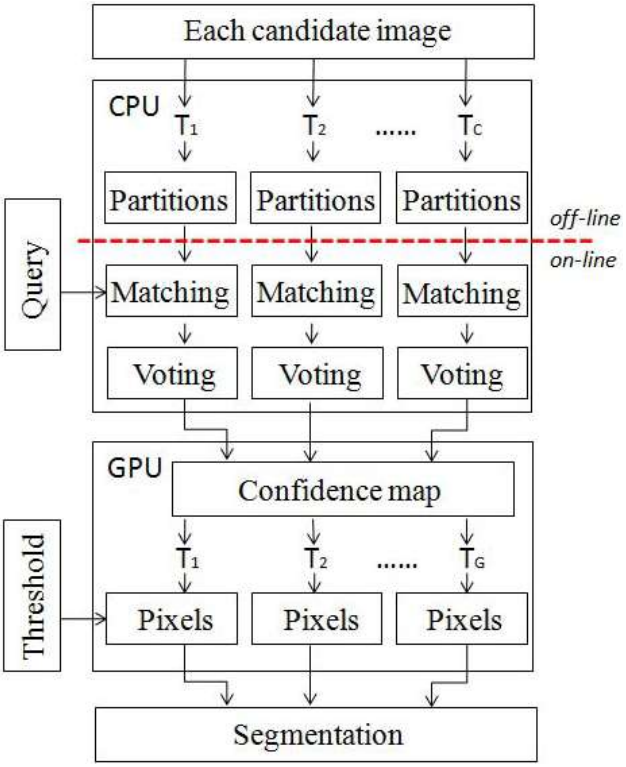


Fig. 6. An overview of the parallel implementation of our algorithm.

Therefore, whether it can be easily parallelized has become an important criterion to evaluate the feasibility of an object search algorithm, although it used to be ignored in previous work. In this section we briefly describe the parallel implementation of our random partition algorithm.

Fig. 6 shows the parallel implementation of our algorithm. There are two parts that can be parallelized on CPU and GPU, respectively. The first part is for the image partition, patch matching and voting. Compared with the subimage search methods [17], [18] which employ the iterative branch-and-bound search, our algorithm guarantees the independence of each round of partition, hence the patches from different partition rounds can be processed simultaneously. In later experiments we implement the parallelization in $C = 16$ threads on CPU, denoted as $\{T_c\}_{c=1}^{C}$ in Fig. 6. So the time complexity of our algorithm is $O(KMN/C)$. The second parallelized part is for the pixel-level object segmentation. After generating a confidence map, in which each pixel has an independent confidence score, we just need to check whether the confidence score of each pixel is larger than the threshold or not. GPU is exactly designed for this job: huge amount of repeated but simple computation. We configure the thread hierarchy on GPU as 64 thread blocks with 64 threads in each block in our experiment, hence the total number of GPU threads is $G = 64 \times 64 = 4096$.

V. Experiments

In this section, our random partition approach is compared with several previous object retrieval algorithms in terms of both speed and performance. We compare our approach with

three categories of methods: the first is between the fixed-scale spatial context methods, i.e., the k-NN group [32] and the grid feature [13] (Sec. V-B); the second is the individual point matching method under the Naive-Bayes assumption, i.e., the DIP algorithm [23] (Sec. V-C); the third is the state-of-the-art subimage search methods, i.e., ESR [17] and ESS [18] (Sec. V-E). All these algorithms are implemented in C++ and performed on a Dell workstation with 2.67 GHz Intel CPU and 16 GB of RAM. The algorithms are implemented without parallelization unless emphasized. Three challenging databases are used as the testbeds:

Groundhog Day Database: The database consists of 5640 keyframes extracted from the entire movie *Groundhog Day* [32], from which 6 visual objects are chosen as queries. As in [32], local interest points are extracted by the Harris-Affine detector and the MSER detector respectively, and described by 128D SIFT descriptors [22]. To reduce noise and reject unstable local features, we follow the local feature refinement method in [42]: all the keyframes are stretched vertically and horizontally, and local interest points are extracted from the stretched keyframes. Those local features that survive image stretching are supposed to be affine invariant and hence are kept as refined features. All the refined features, more than 5 million, are clustered into a vocabulary of 20K visual words using the Hierarchical K-Means (HKM) method [26].

Belgalogo Database: Belgalogo is a very challenging logo database containing $10,000$ images covering various aspects of life and current affairs. As in [15], all images are re-sized with a maximum value of height and width equal to 800 pixels, while preserving the original aspect ratio. Since the database is larger and the image backgrounds are more cluttered, more than 24 million SIFTs are extracted from the database and clustered into a large vocabulary of 1M visual words to ensure the discriminative power of visual words. A total of 6 external logos from Google are selected as the query objects. Meanwhile, to test our discriminative random partition approach (DRP), we randomly pick out two images containing no logos from the database as negative queries.

Belgalogo + Flickr Database: To further verify the scalability and effectiveness of our approach, we build a 1M image database by adding crawled Flickr images to the Belgalogo database as distractors. In total about 2 billion SIFTs (2,000 points per image on average) are extracted. We randomly pick 1% points from the feature pool to generate a vocabulary of 1M visual words. All points are indexed by an inverted file costing about 12G RAM.

For all the databases above, a stop list is made to remove the top 10 percent most frequent visual words. In this way, the most frequent but meaningless visual words that occur in almost all images are suppressed. To evaluate the retrieval performance, in most cases we adopt the Average Precision (AP) and mean Average Precision (mAP) as the measures. Given a ranking list including R retrieved results, the AP is calculated as the area under the Precision/Recall curve:

$$AP = \frac{\sum_{r=1}^{R} Prec(r) \times rel(r)}{\#Ground\ Truth}, \tag{9}$$

Fig. 7. Image examples from the three databases. (a) Groundhog Day database consisting of 5640 keyframes; (b) Belgalogo database, a benchmark database for logo retrieval; (c) Flickr database, containing nearly 1M images which are added as the distractors for Belgalogo.

TABLE II
mAP FOR DIFFERENT VECTOR DISTANCES WITH $\alpha = 3.0$

	Bin	HI	NHI	Dot	ρ_{bhatt}
mAP	0.435	0.444	0.449	0.397	0.406

TABLE III
mAP FOR DIFFERENT SEGMENT COEFFICIENT α USING $Bin(\cdot)$

α	1.0	2.0	3.0	4.0	5.0
mAP	0.403	0.422	0.435	0.434	0.420

where $Prec(r)$ is the precision at cut-off r in the list, and $rel(r)$ is an indicator function equaling 1 if the r^{th} result contains the target objects (i.e., ground truth), 0 otherwise; then the mAP is the mean average precision over all queries. Since some previous work published their results in different measures, we will follow their measures when comparing with them.

A. Sensitivity of Parameters

In this section, the sensitivity of several parameters of the random partition approach is firstly tested on the Groundhog Day database.

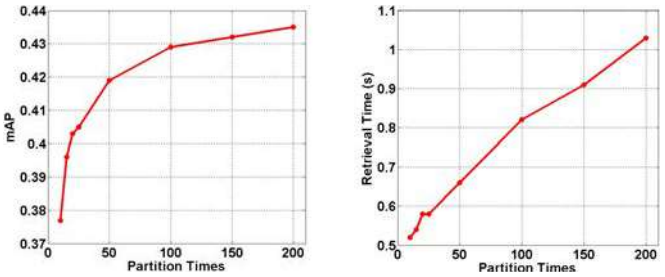
At first we test vector matching kernel and segment coefficient α. The normal random partition (NRP) approach is implemented with the partition parameters $K \times M \times N = 200 \times 16 \times 8$, where $M \times N$ is set according to the aspect ratio of the keyframes empirically. The results are evaluated by mAP over 6 query objects. All the vector matching kernels in Tab. I are tested, and the results are showed in Tab. II. $NHI(\cdot)$ performs sightly better than the others although it is slower. Also, we test the impact of the segment coefficient α, as shown in Tab. III, from which we can see that α has marginal influence on the retrieval performance.

Next, we study how the partition parameters affect the retrieval performance in both accuracy and efficiency. We first

Fig. 8. The influence of the number of partition times. The 1_{st} row lists three pairs of queries (denoted by yellow box on the left) and an example image containing the object (denoted by blue box on the right). The output includes a confidence map on the left and a segmentation result on the right. The 2_{nd}, 3_{rd}, 4_{th} row are associated with the number of partition times $K = 25$, $K = 50$, $K = 100$, respectively. As the number of partition times increases, the confidence map becomes more salient and the object is located more accurately.

TABLE IV
mAP FOR DIFFERENT PARTITION PARAMETERS $M \times N$

	Query Size	8×4	16×8	24×12	32×16
Black Clock	$65p \times 60p$	0.387	0.456	**0.470**	0.426
Digital Clock	$165p \times 100p$	**0.423**	0.412	0.409	0.405
Frames Sign	$297p \times 67p$	0.426	0.486	0.499	**0.508**
Microphone	$63p \times 77p$	0.186	**0.238**	0.229	0.225
Phil Sign	$75p \times 50p$	0.743	**0.767**	0.757	0.765
Red Clock	$60p \times 60p$	0.204	**0.249**	0.229	0.221
Avg.		0.395	**0.435**	0.432	0.425



(a) (b)

Fig. 9. Performance of different number of partition times, from 10 to 200: a) the mAP curve as the number of partition times increases; b) the time cost for different number of partition times, including patch matching, confidence map generation and object segmentation (no parallel implementation).

fix $K = 200$ and test different $M \times N$, from 8×4 to 32×16, and compare their performance in Tab. IV. It shows that the highest AP scores of the query objects Microphone, Phil Sign and Red Clock are achieved at $M \times N = 16 \times 8$. Given the size of the queries, we can infer that the best matching accuracy is more likely to be achieved when the average size of the random patches is close to the target object size. However, we also note that there is an exception case, namely the Frames Sign, where the query object is of a relative large size but the AP decreases with the average size of the random patches increases. It is because the size of the Frames Signs in the video varies quite a lot, and most of them are much smaller than the query one. From this experiment we can see that although the random partition approach could handle the scale invariant to some extent, it essentially implies the assumption on the target object size when partitioning the images.

Then we fix $M \times N = 16 \times 8$ and vary the number of partition times K from 10 to 200, and record their mAP and average time cost, as shown in Fig. 9. It shows that as the number of partition times increases, the retrieval results improve in accuracy while cost more time. And the retrieval accuracy tends to convergence when the number of partition times is large enough. Therefore the approach based on random partition allows the user to easily make a trade-off between accuracy and speed since he can adjust the partition time on-line without re-indexing the database. Increasing the number of partition times leads to a more salient confidence map and better object localization, as showed in Fig. 8.

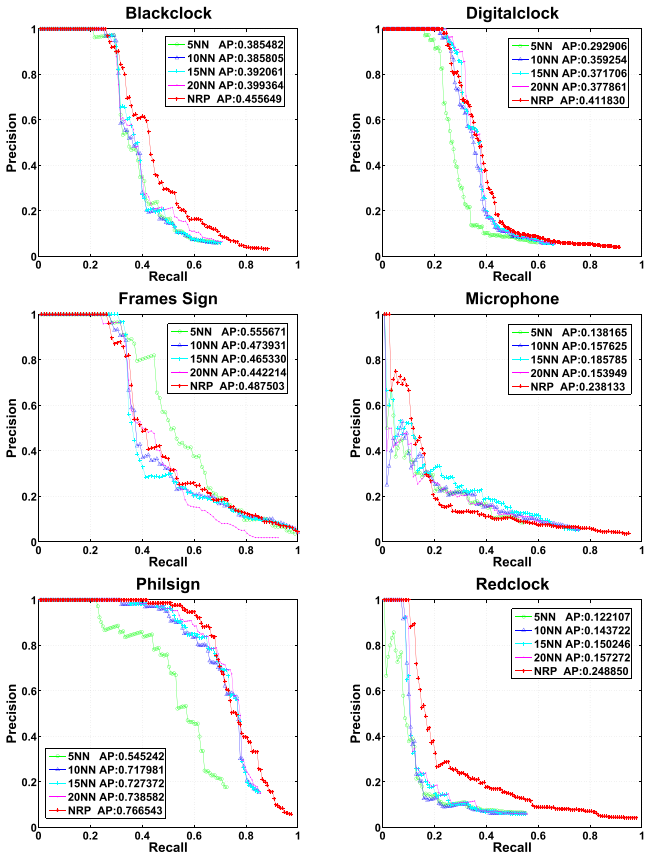
B. Comparison With Fixed-Scale Spatial Context Methods

First, we compare our NRP approach with the spatial k-Nearest Neighbor (k-NN) method [32]. Here we set $k = 5, 10, 15, 20$ to test the retrieval performance when considering spatial context at different scales. $Bin(\cdot)$ is selected as the matching kernel. As in [32], random patches or k-NN regions are rejected if they have less than two visual words matched with the query, which means no spatial support. We fix partition parameters $K \times M \times N = 200 \times 16 \times 8$ and $\alpha = 3.0$ for all queries in this database. The experimental results are shown in Fig. 10, from which we can see that: 1) the optimal scale of spatial context differs for different query objects. As k increases, the retrieval performance improves for most queries while it drops for the Frames Sign. The reason is that the Frames Sign objects in groundtruth keyframes are much smaller than the query

TABLE V

INTERACTIVE SEARCH RESULTS FOR DIP [23] AND DRP. SINCE BASE AND KIA ARE NOT OPTED IN [23], HERE WE ONLY COMPARE THE RESULTS ON THE OTHER 4 LOGOS. TO MAKE A FAIR COMPARISON, WE COMPARE THE PRECISIONS AT THE SPECIFIC RECALL LEVEL GIVEN IN [23]

		Dexia	Ferrari	Mercedes	President	Average
1_{st} round	*recall*	*0.096*	*0.013*	*0.145*	*0.357*	
DIP [23]	precision	0.810	0.010	0.917	0.050	0.359
DRP	precision	0.667	1.000	0.917	1.000	0.896
2_{nd} round	*recall*	*0.060*	*0.039*	*0.184*	*1.000*	
DIP [23]	precision	0.100	0.750	1.000	0.826	0.669
DRP	precision	1.000	1.000	1.000	1.000	1.000



	5-NN	10-NN	15-NN	20-NN	NRP
Black Clock	0.385	0.386	0.392	0.399	0.456
Digital Clock	0.293	0.359	0.372	0.378	0.412
Frames Sign	0.556	0.474	0.465	0.442	0.486
Microphone	0.138	0.158	0.186	0.154	0.238
Phil Sign	0.545	0.718	0.727	0.739	0.767
Red Clock	0.122	0.144	0.150	0.157	0.249
mAP	0.340	0.373	0.382	0.378	0.435

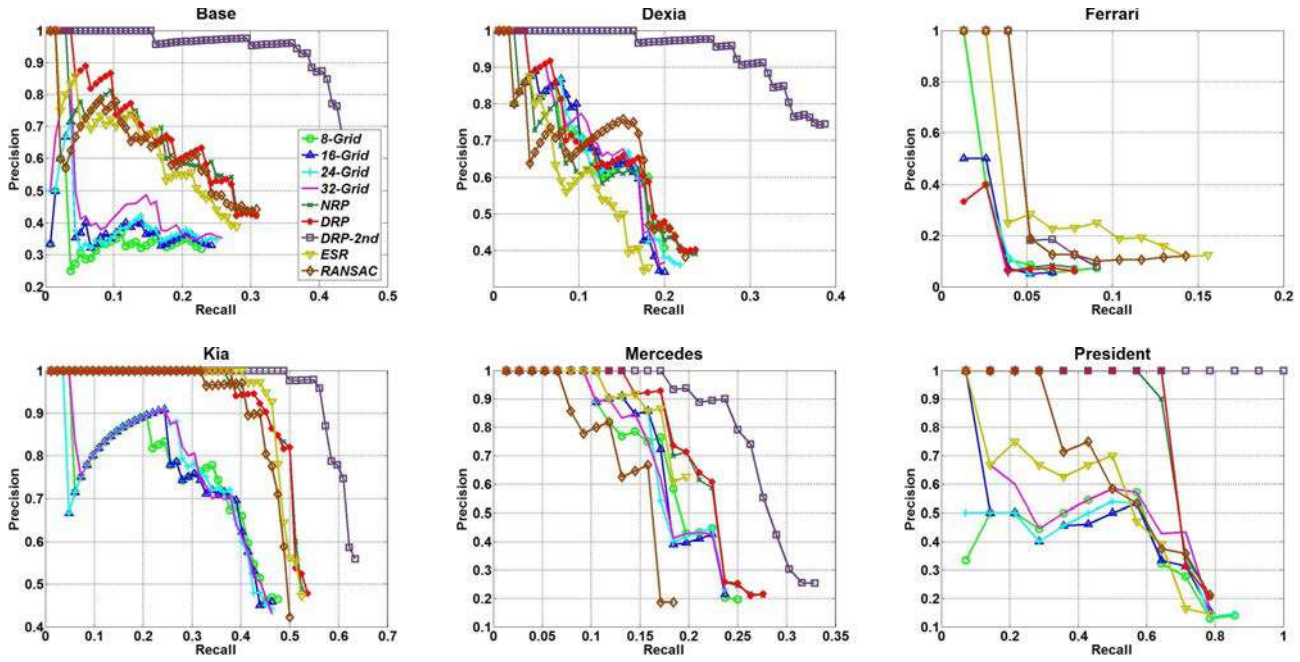
Fig. 10. Precision/Recall curves and AP scores for the six query objects in the Groundhog Day database. Each plot contains 5 curves, referring to the 5-NN, 10-NN, 15-NN, 20-NN and NRP approach respectively. In the bottom table, the red number in each row is the best result for the given query object while the blue one is the second best.

so that it is easier to introduce the noise with a larger context scale; 2) although the optimal scale is unknown, our NRP approach is stable and robust to the scale variations of the objects, therefore achieves a better performance over the k-NN methods.

Further, our discriminative random partition (DRP) approach is compared with the discriminative grid-based algorithm [13] on the Belgalogo database. The partition parameters are set to $K \times M \times N = 200 \times 16 \times 16$ for this database and the segment coefficient $\alpha = 5.0$ is fixed for all queries. Similar to the k-NN methods, 4 different grid sizes, from 8×8 to 32×32, are tested. Normalized histogram intersection $NHI(\cdot)$ is chosen as the similarity function. The top 100 retrieval results are used for evaluation. The comparison results are given in the 2_{nd} to 5_{th} columns and 9_{th} column of Fig. 11, which show that the mAP of DRP is improved by more than 40% over that of the grid-based approach using the same local features and matching kernel. It validates that the random spatial context is superior to fixed-scale spatial context bundled by grids.

C. Comparison With Naive-Bayes Point Matching Methods

In this section, we employ the interactive search strategy and make a comparison between DRP and [23], in which an interactive object search algorithm based on discriminative individual point (DIP) matching is proposed. After the 1_{st} round DRP search, the top $R = 5$ returned results are verified manually. Denoting by $\{\mathcal{L}_r\}$ the collection that contains R verified segments, and representing each segment as a word-frequency histogram $h_{\mathcal{L}_r}$, a new query $\tilde{Q}_+$ is constructed by averaging the word-frequency histograms of $\{\mathcal{L}_r\}$: $h_{\tilde{Q}_+} = \frac{1}{R}\sum_r h_{\mathcal{L}_r}$. Similarly, we can construct an new negative query and repeat the DRP search in the 2_{nd} round. Since the published DIP results are reported in Precision/Recall scores, here we compare with their precisions given the same recall, as shown in Tab. V. From this experimental result, we can see that our DRP approach outperforms the DIP approach in both the 1_{st} and 2_{nd} rounds except for Dexia in the first round. Because in [23] the local descriptors are matched in the high-dimensional feature space independently (i.e., under the Naive-Bayes assumption), DIP could avoid quantization error completely but considers no spatial context. Therefore, the experiment indicates that considering spatial context is a better way to mitigate the quantization error from BoVW and enhance the discriminative power of local features. Since the low recall level limits our observation, we also evaluate the performance of interactive search by AP and P/R curve, as shown in the 10_{th} column of Fig. 11. It shows that the mAP of DRP in 2_{nd} round (DRP-2_{nd}) has a 52% improvement over that in 1_{st} round, and hence highlights the effectiveness of our straightforward interactive strategy.



	8-Grid	16-Grid	24-Grid	32-Grid	ESR [17]	RANSAC [26]	NRP	DRP	DRP-2_{nd}
Base	0.079	0.093	0.099	0.116	0.179	0.194	0.208	0.215	0.440
Dexia	0.144	0.143	0.151	0.145	0.117	0.151	0.153	0.165	0.366
Ferrari	0.023	0.015	0.011	0.010	0.052	0.051	0.013	0.013	0.046
Kia	0.365	0.355	0.358	0.364	0.497	0.473	0.506	0.506	0.612
Mercedes	0.185	0.184	0.183	0.181	0.180	0.139	0.215	0.216	0.275
President	0.346	0.368	0.353	0.424	0.446	0.537	0.675	0.680	1.000
mAP	0.190	0.193	0.192	0.207	0.245	0.258	0.295	0.299	0.457

Fig. 11. Precision/Recall curves and AP scores of grid-based approach with different grid sizes (8×8, 16×16, 24×24 and 32×32), ESR [17], RANSAC [26], NRP, DRP and DRP-2_{nd} for the 6 query logos on the BelgaLogos database.

D. Comparison With RANSAC Methods

As one of the most popular geometric verification algorithms, RANSAC has been usually adopted as the post-processing step in the state-of-the-art image retrieval system [4], [26]. In this section, we compare our random partition approaches with the RANSAC-based system on the Belgalogo database.

As done in [4] and [26], firstly all the images in the database are fast ranked by their HI scores with the help of the inverted file. Then for the top 100 images in the initial ranking list, we employ the RANSAC algorithm to estimate an affine transformation model with 6 degrees of freedom between the query and each of the retrieved images. Finally the number of the inliers according to the affine transformation model is regarded as the new similarity score (the position error tolerance is set to 3 pixels), by which the initial retrieved images are re-ranked as the final result.

The performance of RANSAC is shown in the 7_{th} column in Fig. 11. From the mAP over all 6 queries we can see that our random partition approaches have an about 14% improvement over RANSAC on the database. Moreover, after carefully studying the performance of RANSAC on different query logos, we find some interesting results: for most queries, e.g., the Base, Dexia, Kia and President logos, RANSAC performs comparably to or sightly worse than NRP; but the Ferrari logo and the Mercedes logo are two extremes. For the Ferrari logo, RANSAC has a much better performance than NRP, and in fact it gets the highest AP among all algorithms; but for the Mercedes logo, its performance is even much worse than the grid-based algorithms. The reason is that the Ferrari logos appearing in the database are usually of a much larger size, while the Mercedes logos are usually tiny (see Fig. 12). When the target object is large, the object search problem is close to the traditional whole-image retrieval problem, where RANSAC has proven successful but NRP may fail due to its assumption on the object size; however, when the target object is relatively small, there may be no enough matched points to estimate an exact transformation model by RANSAC. On the contrary, NRP is less strict since it only requires the matched points are distributed in a local region. Therefore, compared to RANSAC, the proposed random partition approaches are more competent in searching small objects.

We also compare the time cost of NRP and RANSAC. The NRP algorithm is implemented parallelized as proposed in Section IV-B. All algorithms are re-run for 3 times to calculate the average retrieval time, and the results are shown in Tab. VI. Because only the top 100 images in the initial

Fig. 12. There are two examples for the Ferrari and Mercedes logos, respectively. For the Ferrari logo (left), RANSAC works well since it has enough matched points to estimate the transformation model and does not constrain on the size of the objects; however, for the much smaller Mercedes logo (right), there are not enough matched points to estimate an accurate transformation model by RANSAC. On the contrary, the random partition method is less strict since it only assumes the target object appears in a compact local region. That means, when the target object is too larger than its assumption on object size, the random partition method may fail to accurately segment an entire object out. Instead, it tends to over-segment an entire object into a set of smaller regions. Therefore, compared to RANSAC, the proposed approaches are more competent for the small object search job.

TABLE VI
RETRIEVAL TIME OF NRP, RANSAC AND ESR ON THE BELGALOGO DATABASE

	ESR [17]	RANSAC [26]	NRP	NRP (parallel)
Time (s)	2.97	1.17	2.84	0.44

list are processed, the total time cost of the RANSAC-based system is reduced sharply, and it is in fact not a fair comparison. Even though, we can see that the NRP algorithm has a significant advantage in efficiency with parallel implementation.

E. Comparison With Subimage Search Methods

Subimage search algorithms employing the branch-and-bound scheme are the state-of-the-art for object search, e.g., the efficient subimage retrieval (ESR) algorithm [17] and the efficient subwindow search (ESS) [18] algorithm. The advantage of this category of algorithms is that it can find the global optimal subimage very quickly and return this subimage as the object's location. In this section we compare our random partition approach with ESR on the Belgalogo database and with ESS on the Belgalogo+Flickr database in both accuracy and speed.

The implement details of ESR and ESS are as follows: for both ESR and ESS, we relax the size and shape constraints on the candidate subimages, to ensure that the returned subimage is global optimal; $NHI(\cdot)$ is adopted as the quality function f, and for a set of regions $\mathcal{R}$, the region-level quality bound $\hat{f}$ is defined as: $\hat{f} = \frac{|\overline{h}_{\mathcal{R}} \cap h_Q|}{|\underline{h}_{\mathcal{R}} \cup h_Q|} \geq f$, where $\overline{h}_{\mathcal{R}}$ and $\underline{h}_{\mathcal{R}}$ are the histograms of the union and intersection of all regions in $\mathcal{R}$; for ESR, given a set of images $\mathcal{I}$, the image-level quality bound $\tilde{f}$ is defined as: $\tilde{f} = \frac{|\overline{h}_{\mathcal{I}} \cap h_Q|}{|\underline{h}_{\mathcal{I}} \cup h_Q|}$; the inverted files are used to quickly calculate the visual word histograms.

First we compare our NRP approach with ESR on the Belgalogo database. We set the partition parameters

Fig. 13. Examples of the search results by ESR and our approach. The images in the first column are retrieved by ESR, in which the red bounding boxes are returned as the object location; the second column are the confidence maps generated by our NRP approach, and the third column are the segmentation results (highlighted in green). Note that each row stands for a specific case (from top to bottom): multiple target objects, noisy background and discrete matched points (false alarm by ESR).

$K \times M \times N = 200 \times 16 \times 16$ and $\alpha = 5.0$, and choose $NHI(\cdot)$ as the matching kernel as well. The retrieval performance is given in the 6_{th} and 8_{th} columns of Fig. 11. We can see that the NRP approach leads to a better retrieval performance compared with the state-of-the-art ESR algorithm, although ESR could return the top 100 optimal subimages with highest NHI scores as detections. The reason is that ESR only searches for the subimage of the most similar word-frequency histogram with the query, but does not require these matched visual words fall in a spatial neighborhood region. In other words, as long as an image has several matched visual words, even if these words may be distributed very dispersedly, it is likely to be retrieved by ESR. On the contrary, the NRP approach bundles the local features by random patches. It favors matched points that are distributed compactly, otherwise the confidence map will not produce a salient enough region. Therefore, compared with our NRP approach, ESR leads to more false alarms, especially when the background is noisy. Moreover, our approach could more easily handle the case in which one image contains multiple target objects. Fig. 13 gives a comparison between ESR and NRP by several examples. In addition, by comparing the performances of NRP and DRP, shown in the 8_{th} and 9_{th} columns of Fig. 11 respectively, we see that negative queries will help to improve the retrieval accuracy.

Next, the NRP algorithm is compared with ESR in retrieval speed (see Tab. VI). As we can see, without parallel implementation NRP is comparable with ESR in speed; and the parallel implementation for NRP achieves about 7 times speedup.

Finally to verify the scalability of our algorithm, we further perform the NRP approach on the Belgalogo+Flickr database consisting of 1M images. Both $HI(\cdot)$ and $NHI(\cdot)$ are tested in NRP approach with parallel implementation. Since ESR is

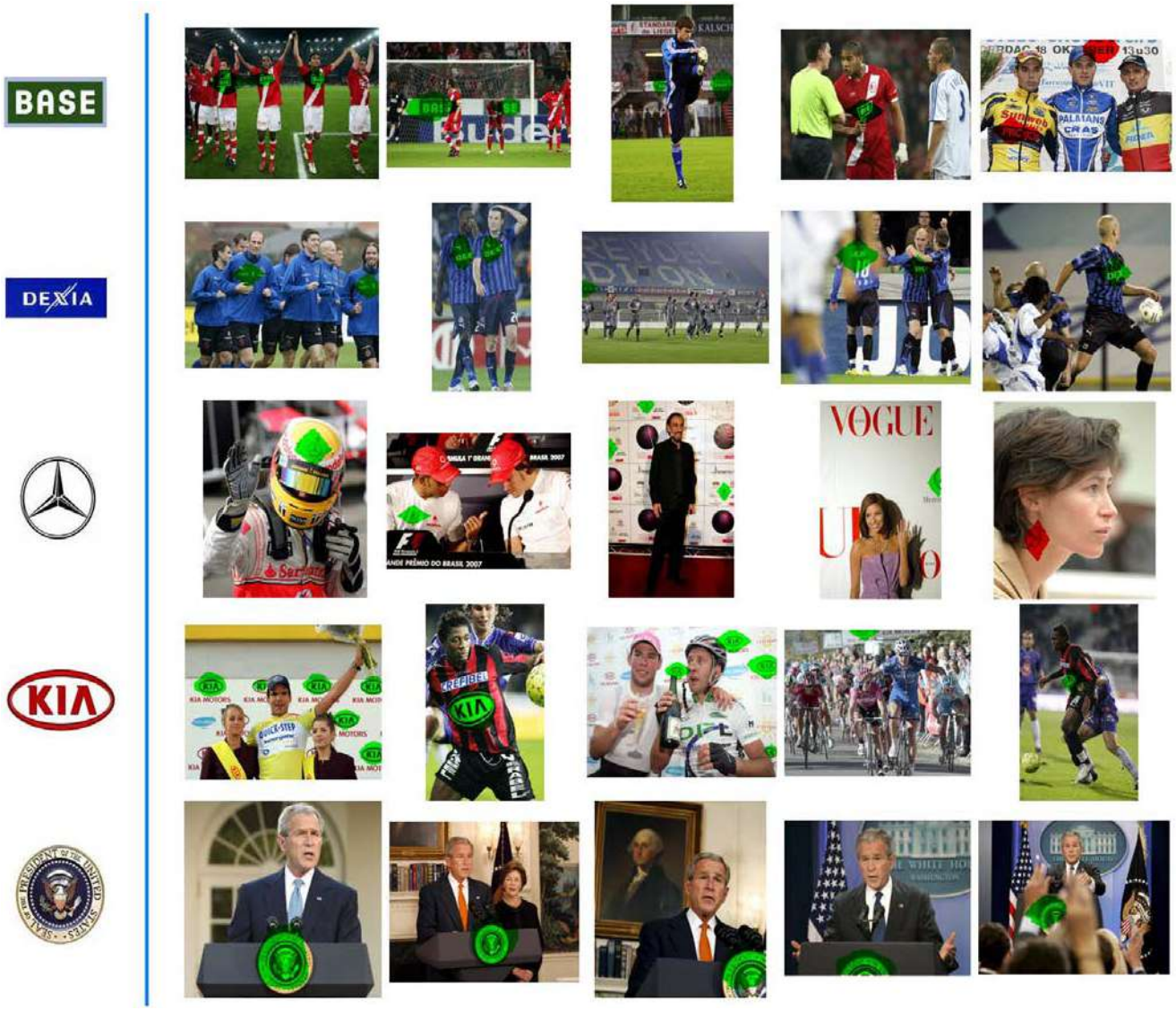


Fig. 14. Examples of our search results on the BelgaLogos database for 5 logos: Base, Dexia, Mercedes, Kia and President (from top to bottom). Queries from Google are in the first column. The selected search results are in the right columns. The correct detections are denoted in green while the wrong detections are in red. We can see that our random partition approach is able to produce satisfactory results even for challenging images, such as non-rigid deformation (row 1, column 5) and bad partial occlusion (row 3, column 5). Moreover, it can handle the multiple objects case (row 4, column 2).

essentially an extension of ESS to improve efficiency and we have compared NRP with ESR on the Belgalogo database, here we compare our NRP approach with ESS on this 1M database. The speed of the algorithms is evaluated by the average processing time per retrieved image. Tab.VII shows the comparison results between ESS and NRP on this 1M database, in which our NRP algorithm beats ESS in both accuracy and speed. This experimental results shows that: 1) employing either $HI(\cdot)$ or $NHI(\cdot)$ as the matching kernel, our NRP approach produces a more than 120% improvement of mAP over ESS. It highlights the effectiveness of our approach; 2) compared to the results on the Belgalogo database consisting of only 10K images, the retrieval performances of both NRP and ESS/ESRbecome worse. However, the mAP of ESS/ESR decreases much more sharply than that of NRP. It verifies the analysis we made above that compared with our approach, ESR is not robust to a cluttered database and leads to more false alarms; 3) $HI(\cdot)$ kernel is much faster (about 4 times) than $NHI(\cdot)$ but has a lower mAP. With the parallel implementation our NRP approach adopting $HI(\cdot)$ kernel could process more than 500 images in one second,

TABLE VII
COMPARISON ON THE BELGALOGO+FLICKR DATABASE

	ESS [18]	NRP(HI)	NRP(NHI)
Base	0.050	0.165	0.189
Dexia	0.029	0.105	0.118
Ferrari	0.017	0.020	0.023
Kia	0.244	0.406	0.418
Mercedes	0.032	0.115	0.148
President	0.165	0.386	0.543
mAP	0.090	0.200	0.240
Time cost per retrieved image (ms)	25.4	1.8	7.8

Fig. 15. Besides the logo queries, more general objects are tested by our search system. Here we give several examples for 5 general query objects: service cap, football, car, helmet and woman face. Similarly to Fig. 14, the queries are denoted in the yellow bounding boxes shown in the left column, and the selected results are shown in the right.

therefore it has a great potential in large-scale applications such as online detection.

VI. Conclusions

In this paper, we propose a scalable visual object search system based on spatial random partition. Our main contribution is the introduction of randomized spatial context for robust sub-region matching. We validate its advantages on three challenging databases in comparison with the state-of-the-art systems for object retrieval. It is shown that compared with systems using only individual local features or fixed-scale spatial context, our randomized approach achieves better search results in terms of accuracy and efficiency. It can also handle object variations in scale, shape and orientation, as well as cluttered backgrounds and occlusions. We also describe the parallel implementation of our system and demonstrate its performance on the one million image database. Moreover, we can use discriminative patch matching and interactive search to further improve the results.

Although we have only used quantized SIFT descriptors to match the random patches, other regional features, e.g., color histogram, can also be incorporated into the similarity score for patch matching. Furthermore, we believe that as a novel way to select suitable spatial context, random partition can be applied to other image-related applications as well.

References

[1] R. Behmo, P. Marcombes, A. Dalalyan, and V. Prinet, "Towards optimal naive Bayes nearest neighbor," in *Proc. Eur. Conf. Comput. Vis.*, 2010, pp. 171–184.

[2] O. Boiman, E. Shechtman, and M. Irani, "In defense of nearest-neighbor based image classification," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jun. 2008, pp. 1–8.

[3] O. Chum, M. Perdoch, and J. Matas, "Geometric min-hashing: Finding a (thick) needle in a haystack," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jun. 2009, pp. 17–24.

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

[5] J. Deng, A. C. Berg, and L. Fei-Fei, "Hierarchical semantic indexing for large scale image retrieval," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jun. 2011, pp. 785–792.

[6] R. Girshick, J. Donahue, T. Darrell, and J. Malik. (2014). "Rich feature hierarchies for accurate object detection and semantic segmentation." [Online]. Available: http://arxiv.org/abs/1311.2524

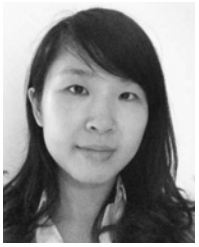
[7] J. He, S.-F. Chang, and L. Xie, "Fast kernel learning for spatial pyramid matching," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jun. 2008, pp. 1–7.

[8] J. He, R. Radhakrishnan, S.-F. Chang, and C. Bauer, "Compact hashing with joint optimization of search accuracy and time," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jun. 2011, pp. 753–760.

[9] H. Jegou, M. Douze, and C. Schmid, "Hamming embedding and weak geometric consistency for large scale image search," in *Proc. Eur. Conf. Comput. Vis.*, 2008, pp. 304–317.

[10] H. Jegou, M. Douze, and C. Schmid, "Packing bag-of-features," in *Proc. IEEE 12th Int. Conf. Comput. Vis. (ICCV)*, Sep./Oct. 2011, pp. 2357–2364.
[11] H. Jegou, M. Douze, C. Schmid, and P. Perez, "Aggregating local descriptors into a compact image representation," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jun. 2010, pp. 3304–3311.
[12] H. Jégou, H. Harzallah, and C. Schmid, "A contextual dissimilarity measure for accurate and efficient image search," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jun. 2007, pp. 1–8.
[13] Y. Jiang, J. Meng, and J. Yuan, "Grid-based local feature bundling for efficient object search and localization," in *Proc. IEEE Conf. Image Process.*, Sep. 2011, pp. 113–116.
[14] Y. Jiang, J. Meng, and J. Yuan, "Randomized visual phrases for object search," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jun. 2012, pp. 3100–3107.
[15] A. Joly and O. Buisson, "Logo retrieval with a contrario visual query expansion," in *Proc. 17th ACM Multimedia*, 2009, pp. 581–584.
[16] Y.-H. Kuo, H.-T. Lin, W.-H. Cheng, Y.-H. Yang, and W. H. Hsu, "Unsupervised auxiliary visual words discovery for large-scale image object retrieval," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jun. 2011, pp. 905–912.
[17] C. H. Lampert, "Detecting objects in large image collections and videos by efficient subimage retrieval," in *Proc. IEEE 12th Int. Conf. Comput. Vis.*, Sep./Oct. 2009, pp. 987–994.
[18] C. H. Lampert, M. B. Blaschko, and T. Hofmann, "Beyond sliding windows: Object localization by efficient subwindow search," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jun. 2008, pp. 1–8.
[19] S. Lazebnik, C. Schmid, and J. Ponce, "Beyond bags of features: Spatial pyramid matching for recognizing natural scene categories," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jun. 2006, pp. 2169–2178.
[20] T. Li, T. Mei, I.-S. Kweon, and X.-S. Hua, "Contextual bag-of-words for visual categorization," *IEEE Trans. Circuits Syst. Video Technol.*, vol. 21, no. 4, pp. 381–392, Apr. 2011.
[21] J. He, R. Radhakrishnan, S.-F. Chang, and C. Bauer, "Compact hashing with joint optimization of search accuracy and time," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jun. 2011, pp. 753–760.
[22] D. G. Lowe, "Distinctive image features from scale-invariant keypoints," *Int. J. Comput. Vis.*, vol. 60, no. 2, pp. 91–110, Nov. 2004.
[23] J. Meng, J. Yuan, Y. Jiang, N. Narasimhan, V. Vasudevan, and Y. Wu, "Interactive visual object search through mutual information maximization," in *Proc. Int. Conf. ACM Multimedia*, 2010, pp. 1147–1150.
[24] O. Chum, J. Philbin, and A. Zisserman, "Near duplicate image detection: Min-hash and tf-idf weighting," in *Proc. BMVC*, 2008, pp. 812–815.
[25] M. Perd'och, O. Chum, and J. Matas, "Efficient representation of local geometry for large scale object retrieval," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jun. 2009, pp. 9–16.
[26] J. Philbin, O. Chum, M. Isard, J. Sivic, and A. Zisserman, "Object retrieval with large vocabularies and fast spatial matching," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jun. 2007, pp. 1–8.
[27] J. Philbin, O. Chum, M. Isard, J. Sivic, and A. Zisserman, "Lost in quantization: Improving particular object retrieval in large scale image databases," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jun. 2008, pp. 1–8.
[28] D. Qin, S. Gammeter, L. Bossard, T. Quack, and L. Van Gool, "Hello neighbor: Accurate object retrieval with k-reciprocal nearest neighbors," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jun. 2011, pp. 777–784.
[29] R. Rahmani, S. A. Goldman, H. Zhang, S. R. Cholleti, and J. E. Fritts, "Localized content-based image retrieval," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 30, no. 11, pp. 1902–1912, Nov. 2008.
[30] B. C. Russell, A. A. Efros, J. Sivic, W. T. Freeman, and A. Zisserman, "Using multiple segmentations to discover objects and their extent in image collections," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jun. 2006, pp. 1605–1614.
[31] J. Sivic and A. Zisserman, "Video Google: A text retrieval approach to object matching in videos," in *Proc. IEEE Int. Conf. Comput. Vis.*, Oct. 2003, pp. 1470–1477.
[32] J. Sivic and A. Zisserman, "Efficient visual search of videos cast as text retrieval," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 31, no. 4, pp. 591–606, Apr. 2009.
[33] X. Wang, K. Liu, and X. Tang, "Query-specific visual semantic spaces for web image re-ranking," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jun. 2011, pp. 857–864.
[34] X. Wang, M. Yang, T. Cour, S. Zhu, K. Yu, and T. X. Han, "Contextual weighting for vocabulary tree based image retrieval," in *Proc. IEEE Int. Conf. Comput. Vis. (ICCV)*, Barcelona, Spain, Nov. 2011, pp. 209–216.
[35] J. Winn and N. Jojic, "LOCUS: Learning object classes with unsupervised segmentation," in *Proc. IEEE Int. Conf. Comput. Vis.*, Oct. 2005, pp. 756–763.
[36] Z. Wu, Q. Ke, M. Isard, and J. Sun, "Bundling features for large scale partial-duplicate web image search," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jun. 2009, pp. 25–32.
[37] L. Xie, Q. Tian, M. Wang, and B. Zhang, "Spatial pooling of heterogeneous features for image classification," *IEEE Trans. Image Process.*, vol. 23, no. 5, pp. 1994–2008, May 2014.
[38] L. Xie, Q. Tian, W. Zhou, and B. Zhang, "Fast and accurate near-duplicate image search with affinity propagation on the ImageWeb," *Comput. Vis. Image Understand.*, vol. 124, pp. 31–41, Jul. 2014.
[39] J. Yuan, Z. Liu, and Y. Wu, "Discriminative subvolume search for efficient action detection," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jun. 2009, pp. 2442–2449.
[40] J. Yuan and Y. Wu, "Spatial random partition for common visual pattern discovery," in *Proc. IEEE 11th Int. Conf. Comput. Vis.*, Oct. 2007, pp. 1–8.
[41] J. Yuan, Y. Wu, and M. Yang, "Discovery of collocation patterns: From visual words to visual phrases," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jun. 2007, pp. 1–8.
[42] S. Zhang, Q. Huang, G. Hua, S. Jiang, W. Gao, and Q. Tian, "Building contextual visual vocabulary for large-scale image applications," in *Proc. Int. Conf. ACM Multimedia*, 2010, pp. 501–510.
[43] S. Zhang, Q. Tian, G. Hua, Q. Huang, and S. Li, "Descriptive visual words and visual phrases for image applications," in *Proc. 17th Int. Conf. ACM Multimedia*, 2009, pp. 75–84.
[44] Y. Zhang, Z. Jia, and T. Chen, "Image retrieval with geometry-preserving visual phrases," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jun. 2011, pp. 809–816.
[45] Y.-T. Zheng *et al.*, "Tour the world: Building a web-scale landmark recognition engine," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit.*, Jun. 2009, pp. 1085–1092.
[46] W. Zhou, Y. Lu, H. Li, Y. Song, and Q. Tian, "Spatial coding for large scale partial-duplicate web image search," in *Proc. Int. Conf. ACM Multimedia*, 2010, pp. 511–520.

Yuning Jiang received the degree from the University of Science and Technology of China, Hefei, China, in 2010. He is currently a full-time Researcher with the School of Electrical and Electronics Engineering, Nanyang Technological University, Singapore. His current research interests cover on visual object search, image retrieval, natural scene understanding, and face recognition/verification.



Jingjing Meng (M'09) received the B.E. degree in electronics and information engineering from the Huazhong University of Science and Technology, China, in 2003, and the M.S. degree in computer science from Vanderbilt University, Nashville, TN, USA, in 2006. She is currently pursuing the Ph.D. degree with the School of Electrical and Electronics Engineering, Nanyang Technological University (NTU), Singapore. She was a Senior Research Staff Engineer with the Motorola Applied Research Center, Schaumburg, IL, USA, from 2007 to 2010. She is a Research Associate with NTU. Her current research interests include computer vision, human–computer interaction, and image and video analysis.

Junsong Yuan (M'08–SM'14) received the Ph.D. degree from Northwestern University, USA, and the M.Eng. degree from the National University of Singapore. Before that, he graduated from Special Class for the Gifted Young with the Huazhong University of Science and Technology, China. He is currently a Nanyang Assistant Professor and the Program Director of Video Analytics with the School of Electrical and Electronic Engineering, Nanyang Technological University, Singapore. His research interests include computer vision, video analytics, large-scale visual search and mining, and human—computer interaction.

He received the Nanyang Assistant Professorship from Nanyang Technological University, the Outstanding EECS Ph.D. Thesis Award from Northwestern University, the Doctoral Spotlight Award from the IEEE Conference on Computer Vision and Pattern Recognition (CVPR'09), and the National Outstanding Student from the Ministry of Education, China. He co-chairs workshops at SIGGRAPH Asia'14, CVPR'12'13'15, and ICCV'13. He serves as the Program Co-Chair of the IEEE Visual Communications and Image Processing (VCIP'15), the Organizing Co-Chair of the Asian Conference on Computer Vision (ACCV'14), and the Area Chair of the IEEE Winter Conference on Computer Vision (WACV'14), the IEEE Conference on Multimedia Expo (ICME'14'15), and ACCV'14. He also serves as a Guest Editor of the *International Journal of Computer Vision*, and an Associate Editor of *The Visual Computer* journal and the *Journal of Multimedia*. He gave tutorials at ACCV'14, ICIP'13, FG'13, ICME'12, SIGGRAPH VRCAI'12, and PCM'12.

Jiebo Luo (S'93–M'96–SM'99–F'09) joined the University of Rochester in Fall 2011, after over 15 years at Kodak Research Laboratories, where he was a Senior Principal Scientist leading research and advanced development. He is a fellow of the International Society for Optics and Photonics, and the International Association for Pattern Recognition. He has been involved in numerous technical conferences, and served as the Program Co-Chair of ACM Multimedia 2010 and the IEEE CVPR 2012. He is the Editor-in-Chief of the *Journal of Multimedia*, and has served on the Editorial Boards of the IEEE TRANSACTIONS ON PATTERN ANALYSIS AND MACHINE INTELLIGENCE, the IEEE TRANSACTIONS ON MULTIMEDIA, the IEEE TRANSACTIONS ON CIRCUITS AND SYSTEMS FOR VIDEO TECHNOLOGY, *Pattern Recognition*, *Machine Vision and Applications*, and the *Journal of Electronic Imaging*.