# Randomized visual phrases for object search

Jiang, Yuning; Meng, Jingjing; Yuan, Junsong

2012

Jiang, Y., Meng, J., Yuan, J. (2012). Randomized Visual Phrases for Object Search. IEEE Conference on Computer Vision and Pattern Recognition (CVPR), pp.3100-3107.

https://hdl.handle.net/10356/100684

https://doi.org/10.1109/CVPR.2012.6248042

# Randomized Visual Phrases for Object Search

Yuning Jiang Jingjing Meng Junsong Yuan *
School of EEE, Nanyang Technological University, Singapore
{ynjiang, jingjing.meng, jsyuan}@ntu.edu.sg

## Abstract

*Accurate matching of local features plays an essential role in visual object search. Instead of matching individual features separately, using the spatial context, e.g., bundling a group of co-located features into a visual phrase, has shown to enable more discriminative matching. Despite previous work, it remains a challenging problem to extract appropriate spatial context for matching. We propose a randomized approach to deriving visual phrase, in the form of spatial random partition. By averaging the matching scores over multiple randomized visual phrases, our approach offers three benefits: 1) the aggregation of the matching scores over a collection of visual phrases of varying sizes and shapes provides robust local matching; 2) object localization is achieved by simple thresholding on the voting map, which is more efficient than subimage search; 3) our algorithm lends itself to easy parallelization and also allows a flexible trade-off between accuracy and speed by 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.*

## 1. Introduction

Despite rapid progress in the whole-image retrieval techniques [3, 4, 5, 15, 11, 23], visual object search, whose goal is to accurately locate the target object in image collections, remains a challenging problem. This is due to the fact that the target objects, e.g., logos, usually occupy only a small portion of an image with cluttered background, and can differ significantly from the query in scale, orientation, viewpoint and color. These all lead to difficulties in object matching, and thereby raise the need for highly discriminative visual features.

Using spatial context is one of the most effective ways to enhance the discriminative power of individual local features, in which a group of co-located visual features can be bundled together to form a visual phrase and matched as a whole. The benefits of using visual phrase have been proven to boost local feature matching [5, 10, 14, 18, 24, 20, 22]. However, it remains a challenging problem to select the appropriate spatial context to compose the visual phrase.

Currently, there are mainly two ways to select the spatial context to compose the visual phrase. The first category of methods relies on image segmentation or region detection to generate the spatial context for matching [16, 19, 20]. However, this is highly dependent on the accuracy of the image segmentation. The second category of methods selects the visual phrase at a relatively fixed scale, e.g., bundling each local feature with its $k$ spatial nearest neighbors [17, 22] or with a fixed-size image grid [6], or extract geometry-preserving visual phrases that can capture long-range spatial layouts of the words [24]. However, as reported in [23], these unstable feature points result in a varying number of detected local features at different scales. Hence for each local point, its $k$-NN phrase may be totally different from that at a different scale, as shown in Fig. 1(a). Similarly, the visual phrase provided by the fixed-size image grid is also not scale invariant, as shown in Fig. 1(b). Furthermore, it is difficult to determine an appropriate $k$ or the grid size without a priori knowledge.

We believe that an ideal visual phrase selection for object search task should satisfy the following requirements: 1) it should be able to handle scale variations of the objects, and be robust to detect objects appearing in the cluttered backgrounds; and 2) it should not rely on the image segmentation or region detection, thus it can be efficiently extracted and indexed to support fast search.

To address these requirements, we propose a novel visual phrase selection approach based on random partition of images [21]. After extracting local invariant features, we randomly partition the image for multiple times to form a pool of overlapping image patches. Each patch bundles the local features inside it and is characterized by a group of visual words, i.e., a visual phrase. Essentially, for each local feature, we generate a number of randomized visual phrases (RVP) in varying sizes and shapes as its spatial contexts (see Fig. 1(c)). For each RVP, we independently calculate the similarity score between it and the query object, and treat it as the voting weight of the corresponding patch. The final confidence score of each pixel is calculated as the ex-

*This work is supported in part by the Nanyang Assistant Professorship (SUG M58040015) to Dr. Junsong Yuan.

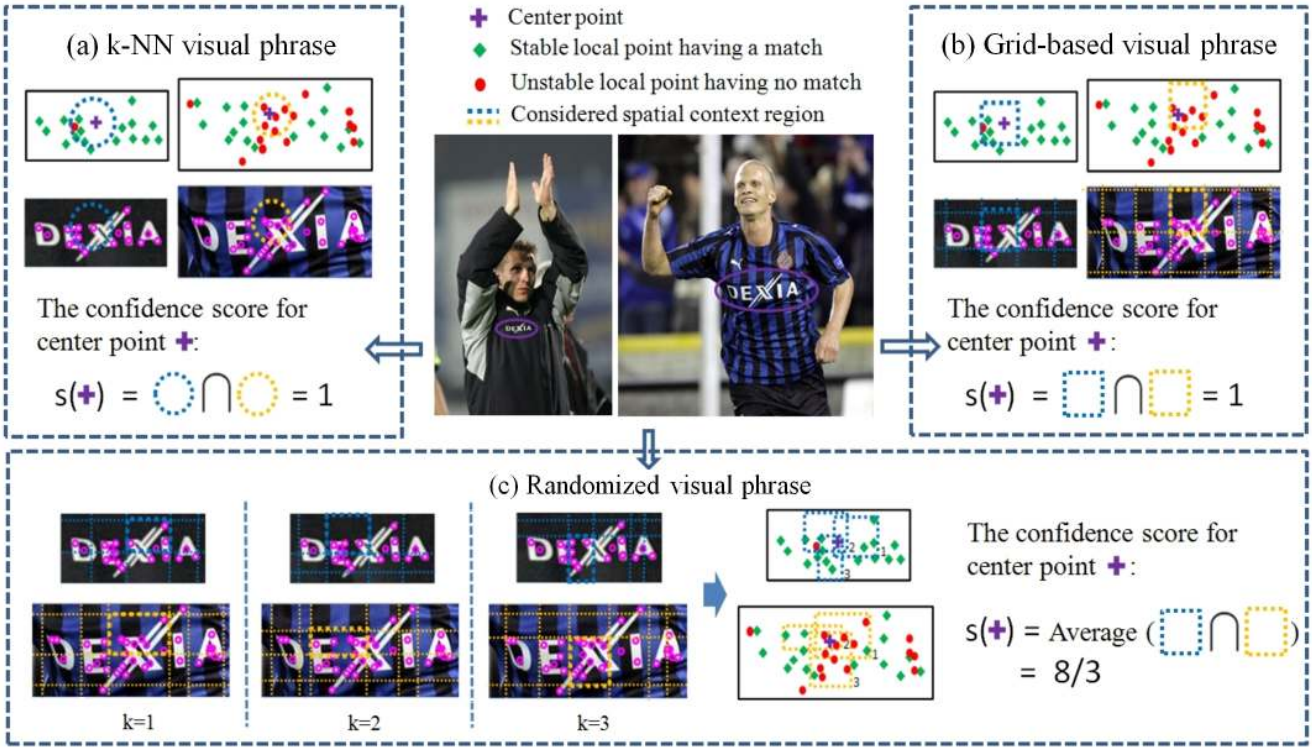



Figure 1: Comparison among different ways to compose the visual phrase. The similarity between two visual phrase regions are calculated as the number of matched points (including the center point) of them, denoted by ∩.

pectation of the voting weights of all patches that contain this pixel. By establishing the pixel-wise voting map, the matched object can finally be identified.

Our randomized visual phrase approach provides several benefits. First it is robust to the cluttered backgrounds as well as the variations 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 voting map. This largely reduces the computational cost compared with the subimage search methods for object localization [8, 9]. Third, our approach allows the user to make a trade-off between effectiveness and efficiency by adjusting the number of partition times on-line without re-indexing the database. This is important for a practical retrieval system. In addition, the design of the algorithm makes it ready for parallelization and thus suitable for large-scale image search. To evaluate our approach, we conduct visual object search on a benchmark movie database, and a challenging logo database with one million images from Flickr as distractors. The experimental results highlight both the effectiveness and the efficiency of the proposed algorithm.

## 2. Related Work

Visual object search can be viewed as two combined tasks: object matching and object localization. For object matching, to avoid the quantization error incurred by the bag-of-visual-words (BoVW) scheme [21, 3, 4, 5, 17, 7], the Naive-Bayes Nearest Neighbor (NBNN) classifiers are adopted in [1, 2, 13] by assuming that each feature point is independent from the others. However, NBNN may fail when the Naive-Bayes assumption is violated. Another way to mitigate the quantization error is to consider spatial context instead of using individual point. By bundling co-occurring visual words within a constrained spatial distance into a visual phrase [22, 24] or feature group [23] as the basic unit for object matching, the spatial context information is incorporated to enhance the discriminative power of visual words. In [17], each local feature is combined with its $k$ spatial nearest neighbors to generate a $k$-NN visual phrase. And in [6], each image is partitioned into non-overlapping grid cells which bundle the local features into grid features. However, such fixed-scale visual phrases or feature groups are not capable of handling large variations, and thereby can not provide robust object 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 [15]. Alternatively, efficient subimage retrieval (ESR) [8] and efficient subwindow search (ESS) [9] are proposed to find the subimage with maximum similarity to the query. In addition, spatial random partition is proposed in [21] to discover and locate visual common objects.

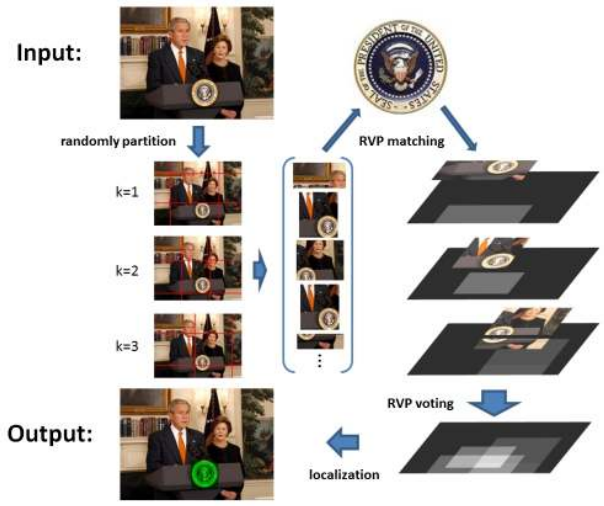



Figure 2: Illustration of object search via spatial random partition ($M \times N \times K = 3 \times 3 \times 3$). The input includes a query object and an image containing the object, while the output is the segmentation of the object (highlighted in green).

## 3. Randomized Visual Phrase 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 query object $Q$, and identify the object's locations $\{\mathcal{L}_g\}$, where $\mathcal{L}_g \subset \mathcal{I}_g$ is a segmentation or sub-region of $\mathcal{I}_g$. An overview of our proposed algorithm is illustrated in Fig. 2.

### 3.1. 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.

### 3.2. Spatial Random Partition

We randomly partition each image $\mathcal{I}_i$ into $M \times N$ non-overlapping rectangular patches and perform such partition $K$ times independently. This results in a pool of $M \times N \times K$ image patches for each $\mathcal{I}_i$, denoted as: $\mathcal{P}_i = \{P_{i,m,n,k}\}$. 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 pixel $t$ falls into a single patch $P_{t,k}$, in total there are $K$ patches containing $t$ after $K$ rounds of partitions, formulated as:

$$\{P_{t,k}\} = \{P_{i,m,n,k} | t \in P_{i,m,n,k}\} \quad k = 1, \ldots, K. \quad (1)$$

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

Given a pixel $t$ we consider the collection of all visual phrases (i.e., patches) containing it, denoted by $\mathbf{\Omega}_t = \{P_t\}$. Then after $K$ times of partitions, we essentially sampled the collection $K$ times and obtained a subset $\mathbf{\Omega}_{t,K} = \{P_{t,k}\}_{k=1}^{K} \subset \mathbf{\Omega}_t$. The sizes and shapes of the visual phrases in the subset $\mathbf{\Omega}_{t,K}$ are random since these visual phrases result from $K$ independent random partitions. Hence for pixel $t$, its spatial context at different scales has been taken into consideration by matching the randomized visual phrase (RVP) set $\mathbf{\Omega}_{t,K}$ against the query. To simplify the problem, we assume the probability that each RVP will be sampled in the $k_{th}$ partition is the same, which means $p(P_{t,k}) = \frac{1}{|\mathbf{\Omega}_{t,K}|} = \frac{1}{K}$ is a constant.

### 3.3. RVP Matching and Voting

Given each pixel $t$, its confidence score $s(t)$ is defined as the expectation of similarity scores of all visual phrases that contain it, denoted as:

$$\begin{aligned} s(t) &= E(s(P_t)) = \sum_{P_t \in \mathbf{\Omega}_t} p(P_t) s(P_t) \\ &\approx \sum_{P_{t,k} \in \mathbf{\Omega}_{t,K}} p(P_{t,k}) s(P_{t,k}) = \frac{1}{K} \sum_{k=1}^{K} s(P_{t,k}), \end{aligned} \quad (2)$$

where the expectation is calculated approximately on the subset $\mathbf{\Omega}_{t,K}$ instead of $\mathbf{\Omega}_t$. Now our problem becomes how to define the similarity score $s(P_{t,k})$ for each RVP. In fact we can adopt any an vector distance listed in Tab. 1 as the matching kernel, and match each RVP against the query just like a whole image. Here we use the normalized histogram intersection $NHI(\cdot)$ as an example:

$$s(t) = \frac{1}{K} \sum_{k=1}^{K} s(P_{t,k}) = \frac{1}{K} \sum_{k=1}^{K} NHI(h_{P_{t,k}}, h_Q). \quad (3)$$

From Eq. 3 we can see that because of the independence of each round of partition, the RVP from different partition rounds can be processed in parallel.

The correctness of our spatial random partition and voting strategy is based on the following theorem that justifies the asymptotic property of our algorithm.

**Theorem 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 $\mathcal{G}$ while $j \notin \mathcal{G}$ is located outside. Suppose $s^K(i)$ and $s^K(j)$ are the confidence*

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

Table 1: Several vector distances for visual phrase matching.

*scores for $i$ and $j$, respectively, considering K times of random partitions and voting, we have:*

$$lim_{K\to\infty}(s^K(i) - s^K(j)) > 0. \quad (4)$$

The above theorem states that after enough rounds of partitions for each image, the pixels in the groundtruth region $\mathcal{G}$ will receive more votes than the pixels in the background, so that the groundtruth can be easily discovered and located. The proof of Theorem 1 is given in the supplementary material because of space limit.

### 3.4. Object Localization

After assigning each pixel $t \in \mathcal{I}_i$ a confidence score, we obtain a voting map for each image $\mathcal{I}_i$. Object localization then becomes an easy task since we do not need to search in the large collection of subimages of $\mathcal{I}_i$. Instead, we just need to segment out the dominant region $\mathcal{L}_i$ from $\mathcal{I}_i$ as the object location:

$$\mathcal{L}_i = \{t | s(t) > thres, \forall t \in \mathcal{I}_i\}. \quad (5)$$

In our paper the threshold $thres$ is set adaptively in proportion to the average confidence score of each image $\mathcal{I}_i$:

$$thres_i = \frac{\alpha}{|\mathcal{I}_i|} \sum_{t\in\mathcal{I}_i} s(t), \quad (6)$$

where $|\mathcal{I}_i|$ is the number of pixels in $\mathcal{I}_i$ and $\alpha$ is the parameter coefficient. This adaptive localization strategy indicates that the matched points in an image should be distributed densely so that the dominant region will be salient enough to be segmented out; otherwise, if the matched points are distributed sparsely in the image, the threshold is the same but there may be no dominant region segmented out. The property of our algorithm is important for object search task, since an object, especially a small object, always has the dense matched points within a compact structure, while the noisy points in the background are usually distributed sparely. Thus our algorithm can reduce the number of false alarms and be robust to background clutter. Fig. 3 illustrates the object localization process.

Figure 3: Examples for object localization. The query logo is the same as in Fig. 2. The $1_{st}$ column are the original images. The $2_{nd}$ column are the voting maps after 200 random partitions. The $3_{rd}$ column are the segmentation results with the coefficient $\alpha = 5.0$. By comparing the $3_{rd}$ row with the first two rows, we can see that this localization strategy is robust to the noisy points which are sparsely distributed in the background.

## 4. Experiments

In this section, our randomized visual phrase approach is compared with previous object retrieval algorithms in terms of both speed and performance. We compare our approach with two categories of methods: the first is the fixed-scale visual phrase approaches, i.e., the $k$-NN phrase [17] and the grid feature [6]; and the second is the state-of-the-art subimage search algorithms, i.e., ESR [8] and ESS [9]. All these algorithms are implemented in C++ and performed on a 16-thread 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 testbed:

**Groundhog Day database** The database consists of 5640 keyframes extracted from the entire movie *Groundhog Day* [17], from which 6 visual objects are chosen as queries. As in [17], local interest points are extracted by the Harris-Affine detector and the MSER detector respectively, and described by 128-dimensional SIFT descriptors [12]. To reduce noise and reject unstable local features, we follow the local feature refinement method in [23]: all the keyframes are stretched vertically and horizontally, and local interest points are extracted from the stretched keyframes. Local features that have survived image stretching are supposed to be affine invariant and hence are kept as refined features. All the refined features, over 5 million, are clustered into a vocabulary of 20K visual words using the Hierarchical K-Means (HKM) method [15].

**Belgalogo** Belgalogo is a very challenging logo database

containing $10,000$ images covering various aspects of life and current affairs. As in [6], 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 and then 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 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 feature 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, we adopt the Average Precision (AP) and mean Average Precision (mAP) as the measures.

### 4.1. Sensitivity of Parameters

First of all, the sensitivity of parameters in the random partition method are tested on the Groundhog Day database. We first test vector matching kernels and segment coefficient $\alpha$. The random partition 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. 1 are tested, and the results are showed in Tab. 2. $NHI(\cdot)$ performs sightly better than the others although it is slower. Also, we test the impact of the segment coefficient $\alpha$, as shown in Tab. 3, from which we can see that $\alpha$ 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 fix $K = 200$ and test different $M \times N$, from $8 \times 4$ to $32 \times 16$, and compare their performance in Tab. 4; then we fix $M \times N = 16 \times 8$ and increase the number of partition times $K$ from 10 to 200, and record their mAP and average time cost, as shown in Fig. 5. 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 online without re-indexing the database. Increasing the number of partition times leads to a more salient voting map and better object localization, as showed in Fig. 4.

| | $Bin$ | $HI$ | $NHI$ | $Dot$ | $\rho_{bhatt}$ |
|---|---|---|---|---|---|
| mAP | 0.435 | 0.444 | 0.449 | 0.397 | 0.406 |

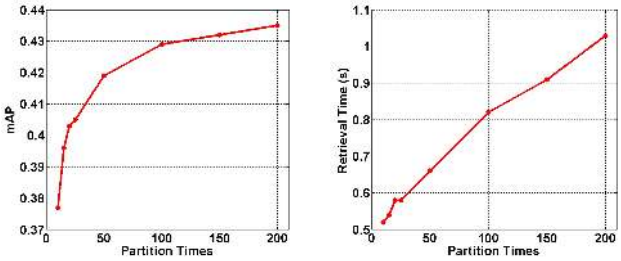
Table 2: mAP for different vector distances with $\alpha = 3.0$.

| $\alpha$ | 1.0 | 2.0 | 3.0 | 4.0 | 5.0 |
|---|---|---|---|---|---|
| mAP | 0.403 | 0.422 | 0.435 | 0.434 | 0.420 |

Table 3: mAP for different segment coefficient $\alpha$ using $Bin(\cdot)$.

| $M \times N$ | $8 \times 4$ | $16 \times 8$ | $24 \times 12$ | $32 \times 16$ |
|---|---|---|---|---|
| mAP | 0.395 | 0.435 | 0.432 | 0.425 |

Table 4: mAP for different partition parameters $M \times N$.




(a) (b)

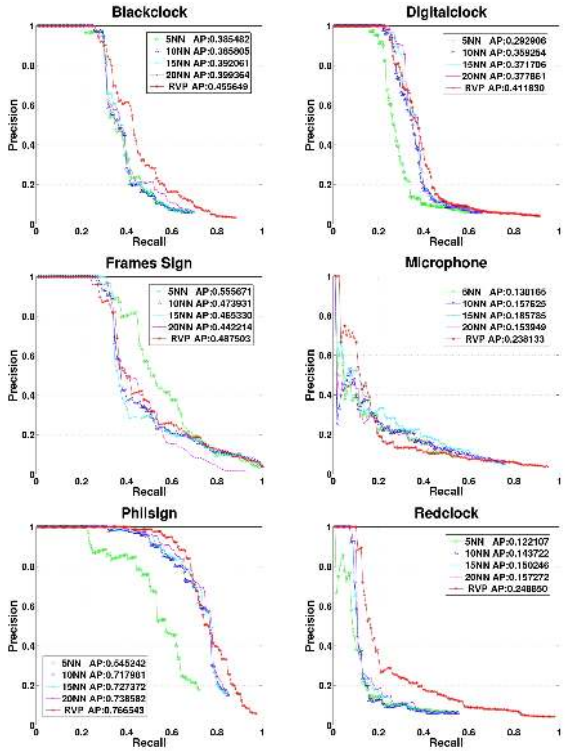
Figure 5: Performance of different number of partition times, from 10 to 200: a) the mAP curve as the number of partition times increases; b) the time cost for different number of partition times, including patch matching, voting and object segmentation.

### 4.2. Comparison with Fixed-Scale Visual Phrase Methods

First, we compare our RVP approach with the $k$-NN phrase method [17]. Here we set $k = 5, 10, 15, 20$ to test the retrieval performance when considering spatial context at different scales. As in [17], $Bin(\cdot)$ is selected as the matching kernel; and the RVPs or $k$-NN phrases are rejected if they have less than two visual words matched with the query, which means no spatial support. We fix partition parameters $\alpha = 3.0$ and $K \times M \times N = 200 \times 16 \times 8$ for all images in this database. The experimental results are shown in Fig. 6, 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 frames are much smaller than the query so that it is easier to introduce noise with a larger context scale; 2) although the optimal scale is unknown, our RVP is stable and robust to object variations, thereby achieves a better performance over the $k$-NN phrase.

Further, the RVP approach is compared with the grid-based algorithm [6] on the Belgalogo database consisting of 10K images. 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 visual phrases, 4 different grid sizes, from $8 \times 8$ to $32 \times 32$, are tested. Normalized histogram intersection $NHI(\cdot)$ is chosen as the similarity function. The top 100


Figure 4: The influence of the number of partition times. The $1_{st}$ row lists three pairs of the query object (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 voting 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 voting map becomes more salient and the object is located more accurately.




| | 5-NN | 10-NN | 15-NN | 20-NN | RVP |
|---|---|---|---|---|---|
| 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 |

Figure 6: Precision/Recall curves and AP scores for the six query objects in the movie Groundhog Day. In the bottom table, the red number in each row is the best result for the query object while the blue one is the second.

| | G-8 | G-16 | G-24 | G-32 | ESR [8] | RVP |
|---|---|---|---|---|---|---|
| Base | 0.079 | 0.093 | 0.099 | 0.116 | 0.179 | 0.208 |
| Dexia | 0.144 | 0.143 | 0.151 | 0.145 | 0.117 | 0.153 |
| Ferrari | 0.023 | 0.015 | 0.011 | 0.010 | 0.052 | 0.013 |
| Kia | 0.365 | 0.355 | 0.358 | 0.364 | 0.497 | 0.506 |
| Mercedes | 0.185 | 0.184 | 0.183 | 0.181 | 0.180 | 0.215 |
| President | 0.346 | 0.368 | 0.353 | 0.424 | 0.446 | 0.675 |
| mAP | 0.190 | 0.193 | 0.192 | 0.207 | 0.245 | **0.295** |

Table 5: AP scores of grid-based approach with different grid sizes (8×8, 16×16, 24×24 and 32×32), ESR [8], and RVP approach for the 6 query logos on the BelgaLogos database.

retrieval results are used for evaluation. The comparison results are given in the $2_{nd}$ to $5_{th}$ columns and $7_{th}$ column of Tab. 5, which show that the mAP of random partition approach 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 randomized visual phrase is superior to fixed-scale visual phrase bundled by grid.

### 4.3. Comparison with Subimage Search Methods

Subimage search algorithms employing the branch-and-bound scheme are the state-of-the-arts for object search, e.g., the efficient subimage retrieval (ESR) [8] and the efficient subwindow search (ESS) [9]. 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 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: $\hat{f} = \frac{|\bar{h}_{\mathcal{R}} \cap h_Q|}{|\underline{h}_{\mathcal{R}} \cup h_Q|}$, 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 for fast computation.

First we compare our approach with ESR on the Belgalogo database. We set the partition parameters $K \times M \times N = 200 \times 16 \times 16$ and $\alpha = 5.0$, and choose $NHI(\cdot)$ as the matching kernel. The retrieval performance is given in the $6_{th}$ and $7_{th}$ columns of Tab. 5. We can see that our approach leads to a better retrieval performance compared with the 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. 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, our approach bundles the local features into the RVPs by random patches. It favors matched points that are distributed compactly, otherwise the voting map will not produce a salient enough region. Therefore, compared with the RVP 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. 7 compares ESR and our approach by several examples. Next, our RVP algorithm is implemented in parallel and compared with ESR in retrieval speed. All algorithms are re-run for 3 times to calculate the average retrieval time, as shown in Tab. 6. As we can see, without parallel implementation our approach is comparable with ESR in speed; and the parallel implementation achieves about 7 times speedup.

| | ESR [8] | RVP | RVP (parallelized) |
|---|---|---|---|
| Time (s) | 2.97 | 2.84 | 0.44 |

Table 6: Retrieval time comparison on the Belgalogo database.

Finally to verify the scalability of our algorithm, we further perform the RVP approach on the Belgalogo+Flickr database consisting of 1M images. Both $HI(\cdot)$ and $NHI(\cdot)$ are tested with parallel implementation. Since ESR is essentially an extension of ESS to improve efficiency and we have compared RVP with ESR on the Belgalogo database, here we compare our RVP approach with ESS on this 1M database. The speed of the algorithms is evaluated by the average processing time per retrieved image. Tab.7 shows the comparison results between ESS and RVP on this 1M database, in which our RVP algorithm beats ESS in both accuracy and speed. This experimental result shows that: 1) employing either $HI(\cdot)$ or $NHI(\cdot)$ as the matching kernel, our RVP 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 pure Belgalogo database consisting of only 10K images, the retrieval performances of both RVP and ESS/ESR become worse. However, the mAP of ESS/ESR decreases much more sharply than that of RVP. 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 RVP approach adopting $HI(\cdot)$ kernel could process more than 500 images in one second, therefore it has a great potential for large-scale object search application.


Figure 7: 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 voting maps generated by the RVP 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).

| | ESS [9] | RVP (HI) | RVP (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 |

Table 7: Comparison on the Belgalogo + Flickr database.

## 5. Conclusions

We propose a scalable visual object search system based on randomized visual phrase (RVP) for robust object matching and localization. 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 fixed-scale visual phrase or subimage search method, our randomized approach achieves better

Figure 8: Examples of our search results in the movie Groundhog Day for 6 query objects: Black Clock, Digital Clock, Frames Sign, Microphone, Phil Sign and Red Clock (from top to bottom). Queries are denoted in yellow in $1_{st}$ column. The correct detections selected from different shots are denoted in green in the right columns.

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. Furthermore, the design of the algorithm makes it ready for parallelization and thus well suited for large-scale applications. We believe that as a novel way to define visual phrases, 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. European Conf. on Computer Vision*, 2010.

[2] O. Boiman, E. Shechtman, and M. Irani. In defense of nearest-neighbor based image classification. In *Proc. IEEE Conf. on Computer Vision and Pattern Recognition*, 2008.

[3] O. Chum, M. Perdoch, and J. Matas. Geometric min-hashing: finding a (thick) needle in a haystack. In *Proc. IEEE Conf. on Computer Vision and Pattern Recognition*, 2009.

[4] J. He, R. Radhakrishnan, S.-F. Chang, and C. Bauer. Compact hashing with joint optimization of search accuracy and time. In *Proc. IEEE Conf. on Computer Vision and Pattern Recognition*, 2011.

[5] H. Jegou, M. Douze, C. Schmid, and P. Perez. Aggregating local descriptors into a compact image representation. In *Proc. IEEE Conf. on Computer Vision and Pattern Recognition*, 2010.

[6] Y. Jiang, J. Meng, and J. Yuan. Grid-based local feature bundling for efficient object search and localization. In *Proc. IEEE Conf. on Image Processing*, 2011.

[7] Y.-H. Kuo, H.-T. Lin, W.-H. Cheng, Y.-H. Yang, and W. H. Hsu. Unsupervised auxiliary visualwords discovery for large-scale image object retrieval. In *Proc. IEEE Conf. on Computer Vision and Pattern Recognition*, 2011.

[8] C. H. Lampert. Detecting objects in large image collections and videos by efficient subimage retrieval. In *Proc. IEEE Intl. Conf. on Computer Vision*, 2009.

[9] C. H. Lampert, M. B. Blaschko, and T. Hofmann. Beyond sliding windows: Object localization by efficient subwindow search. In *Proc. IEEE Conf. on Computer Vision and Pattern Recognition*, 2008.

[10] T. Li, T. Mei, I. Kweon, and X. Hua. Contextual bag-of-words for visual categorization. *IEEE Transactions on Circuits and Systems for Video Technology.*, 2010.

[11] W. Liu, Y.-G. Jiang, J. Luo, and S.-F. Chang. Compact hashing with joint optimization of search accuracy and time. In *Proc. IEEE Conf. on Computer Vision and Pattern Recognition*, 2011.

[12] D. Lowe. Distinctive image features from scale-invariant keypoints. *Intl. Journal of Computer Vision*, 2004.

[13] J. Meng, J. Yuan, Y. Jiang, N. Narasimhan, V. Vasudevan, and Y. Wu. Interactive visual object search through mutual information maximization. In *Proc. ACM Multimedia*, 2010.

[14] M. Perdoch, O. Chum, and J. Matas. Efficient representation of local geometry for large scale object retrieval. In *Proc. IEEE Conf. on Computer Vision and Pattern Recognition*, 2009.

[15] J. Philbin, O. Chum, M. Isard, J. Sivic, and A. Zisserman. Object retrieval with large vocabularies and fast spatial matching. In *Proc. IEEE Conf. on Computer Vision and Pattern Recognition*, 2007.

[16] B. C. Russell, A. A. Efros, J. Sivic, W. T. Freeman, and A. Zisserman. Using multiple segmentation to discover objects and their extent in image collections. In *Proc. IEEE Conf. on Computer Vision and Pattern Recognition*, 2006.

[17] J. Sivic and A. Zisserman. Efficient visual search of videos cast as text retrieval. *IEEE Trans. on Pattern Analysis and Machine Intelligence*, 2009.

[18] X. Wang, M. Yang, T. Cour, S. Zhu, K. Yu, and T. X. Han. Contextual weighting for vocabulary tree based image retrieval. In *ICCV'11: IEEE International Conference on Computer Vision*, Barcelona, Spain, Nov. 6-13, 2011.

[19] J. Winn and N.Jojic. Locus: Learning object classes with unsupervised segmentation. In *Proc. IEEE Intl. Conf. on Computer Vision*, 2005.

[20] Z. Wu, Q. Ke, and J. Sun. Bundling features for large-scale partialduplicate web image search. In *Proc. IEEE Conf. on Computer Vision and Pattern Recognition*, 2009.

[21] J. Yuan and Y. Wu. Spatial random partition for common visual pattern discovery. In *Proc. IEEE Intl. Conf. on Computer Vision*, 2007.

[22] J. Yuan, Y. Wu, and M. Yang. Discovery of collocation patterns: from visual words to visual phrases. In *Proc. IEEE Conf. on Computer Vision and Pattern Recognition*, 2007.

[23] S. Zhang, Q. Huang, G. Hua, S. Jiang, W. Gao, and Q. Tian. Building contextual visual vocabulary for large-scale image applications. In *Proc. ACM Multimedia*, 2010.

[24] Y. Zhang, Z. Jia, and T. Chen. Image retrieval with geometry-preserving visual phrases. In *Proc. IEEE Conf. on Computer Vision and Pattern Recognition*, 2011.