# Spatial Random Partition for Common Visual Pattern Discovery

Junsong Yuan, Ying Wu
EECS Department, Northwestern University
2145 Sheridan Road, Evanston, IL, USA 60208
{j-yuan, yingwu}@northwestern.edu

## Abstract

*Automatically discovering common visual patterns from a collection of images is an interesting but yet challenging task, in part because it is computationally prohibiting. Although representing images as visual documents based on discrete visual words offers advantages in computation, the performance of these word-based methods largely depends on the quality of the visual word dictionary. This paper presents a novel approach base on spatial random partition and fast word-free image matching. Represented as a set of continuous visual primitives, each image is randomly partitioned many times to form a pool of subimages. Each subimage is queried and matched against the pool, and then common patterns can be localized by aggregating the set of matched subimages. The asymptotic property and the complexity of the proposed method are given in this paper, along with many real experiments. Both theoretical studies and experiment results show its advantages.*

## 1. Introduction

It is of great interest to automatically discover common visual patterns (if any) in a set of unlabeled images. Recent research has suggested its applicability in many potential applications, such as content-based retrieval [8], image categorization [5], object discovery [10, 6, 13], recognition [11] and segmentation [3, 10, 9, 15, 14], image irregularity detection [1] and similarity measure [2].

Because no prior knowledge on the common patterns is provided, this task is very challenging, even for our human eyes. Let's look at the example in Fig. 1. This is much more difficult than pattern detection and retrieval, because the set of candidates for possible common patterns is enormous. Validating a single candidate (which is equivalent to pattern detection) has been computationally demanding, and thus evaluating all these candidates will inevitably be prohibiting, if not impossible.

This difficulty may be alleviated by developing robust partial image matching methods [1, 5, 13], but this is not a trivial task. Another idea is to transform images into visual documents so as to take advantage of text-based data mining techniques [12, 10, 16]. These methods need to quantize continuous primitive visual features into discrete labels (i.e., "visual words") through clustering. The matching of two image regions can be efficiently performed by comparing their visual-word histograms while ignoring their spatial configurations. Although these methods are efficient, their performances are largely influenced by the quality of the visual word dictionary. It is not uncommon that the dictionary includes visual synonyms and polysemys that may significantly degrade the matching accuracy. In addition, since a large number of images is generally required to determine the dictionary, these methods may not be suitable if pattern discovery needs to be performed on a small number of images.

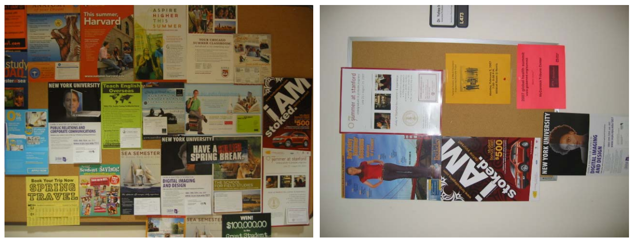

Figure 1. Can you find those common posters (patterns in this case) in the two images? There are three of them (see Sec. 4). It is not an easy task, even for our human eyes.

This paper presents a novel approach to efficient pattern discovery based on spatial random partition. Each image is represented as a set of continuous visual primitives, and is randomly partitioned into *subimages* for a number of times. This leads to a pool of subimages for the set of images given. Each subimage is queried and matched against the subimage pool. As each image is partitioned many times, a common pattern is likely to be present in a good number of subimages across different images. The more matches a subimage query can find in the pool, the more likely it contains a common pattern. And then the pattern can be localized by aggregating these matched subimages. In addition, the proposed method for matching image regions is word-free as it is performed directly on the continuous visual primitives. An approximate solution is proposed to efficiently match two subimages by checking if they share enough similar visual primitives. Such an approximation provides an upper bound estimation of the optimal matching score.

This new method offers several advantages. (1) It does not depend on good image segmentation results. According to its asymptotic property, the patterns can be recovered regardless of its scale, shape and location. (2) It can automatically discover multiple common patterns without knowing the total number *a priori*, and is robust to rotation, scale changes and partial occlusion. The robustness of the method only depends on the matching of visual primitives. (3) It is word-free but still computationally efficient, because of the use of the locality sensitive hash (LSH) technique.

## 2. Proposed Approach

### 2.1. Algorithm overview

Given a number of $T$ unlabeled images, our objective is to discover common spatial patterns that appear in these images. Such common patterns can be identical objects or categories of objects. The basic idea of the proposed spatial random partition method is illustrated in Fig. 2.

We extract a set of visual primitives $\mathbf{V}_{\mathcal{I}} = \{v_1, ..., v_m\}$ to characterize each image $\mathcal{I}$. Each visual primitive is described by $v = \{x, y, \vec{f}\}$, where $(x, y)$ is its spatial location and $\vec{f} \in \Re^d$ is its visual feature vector. Collecting all these visual primitives, we build the visual primitive database $\mathbf{D}_v = \mathbf{V}_{\mathcal{I}_1} \cup \mathbf{V}_{\mathcal{I}_2} \cup ... \cup \mathbf{V}_{\mathcal{I}_T}$, whose size is denoted by $N = |\mathbf{D}_v|$, where $T$ is the total number of images. To index visual primitives, each $v$ is associated with a unique integer $z$ $(1 \leq z \leq N)$ for retrieving it from $\mathbf{D}_v$. Our algorithm is summarized in Alg. 1.

**Algorithm 1**: Spatial Random Partition for Pattern Discovery

```
input  : a collection of unlabeled images: $\mathbf{D}_{\mathcal{I}} = \{\mathcal{I}_i\}$
output : a set of subimage regions that correspond to
         common spatial patterns
1 Hashing: $\forall$ image $\mathcal{I}_i \in \mathbf{D}_{\mathcal{I}}$, randomly partition it into
  $G \times H$ subimages for $K$ times. This outputs the
  subimage database $\mathbf{D}_{\mathcal{R}}$ (Sec. 2.2).
2 Matching: $\forall$ subimage $\mathcal{R}_i \in \mathbf{D}_{\mathcal{R}}$, query it in $\mathbf{D}_{\mathcal{R}}$.
  This leads to a small set of popular subimages that have
  enough matches in $\mathbf{D}_{\mathcal{R}}$ (Sec. 2.3).
3 Voting: $\forall \mathcal{I}_i \in \mathbf{D}_{\mathcal{I}}$, vote the corresponding regions of
  the discovered popular subimages $\mathcal{R} \subset \mathcal{I}_i$ and
  accumulate all the votes to form a voting map (Sec. 2.4)
4 Localization: $\forall \mathcal{I}_i \in \mathbf{D}_{\mathcal{I}}$, segment its voting map to
  localize the common patterns (Sec. 2.4)
```

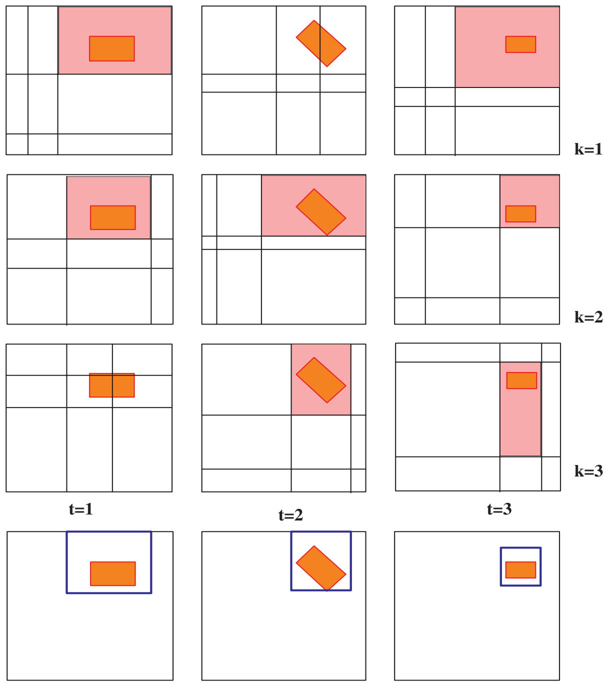

Figure 2. Illustration of the basic idea of spatial random partition. There are three images, each of which contains the same common pattern $\mathcal{P}$, which is represented by the orange rectangle. Each column corresponds to a same image ($t = 1, 2, 3$). Note this common pattern exhibits variations like rotation (the second image) and scale changes (the third image). We perform a $G \times H$ (e.g. $3 \times 3$) partitions for each image for $K$ (e.g. 3) times. Each of the first three rows shows a random partition ($k = 1, 2, 3$). The highlight region of each image indicates a *good* subimage (7 in total out of 81 candidates) that contains the common pattern. All of these good subimages are also *popular* ones as they can find enough matches (or supports) in the pool. The bottom row is a simple localization of the pattern, which is the intersection of the popular subimages in the corresponding image.

### 2.2. Spatial random partition

For each image $\mathcal{I} \in \mathbf{D}_{\mathcal{I}}$, we randomly partition it into $G \times H$ non-overlapping subimages $\{\mathcal{R}_i\}$ and perform such partition $K$ times independently. We end up with in total $M = G \times H \times K \times T$ subimages and form a *subimage database* $\mathbf{D}_{\mathcal{R}} = \{\mathcal{R}_i\}_{i=1}^{M}$. Each generated subimage is characterized by a "bag of visual primitives": $\mathcal{R} = (\mathbf{V}_R, C_{\mathcal{R}})$, where $\mathbf{V}_R \subset \mathbf{V}_{\mathcal{I}}$ denotes the set of visual primitives contained in $\mathcal{R}$ and $C_{\mathcal{R}}$ is the bounding box of $\mathcal{R}$. Under a certain partition $k \in \{1, 2, ..., K\}$, the $G \times H$ subimages are non-overlapping, and we have $\mathbf{V}_{\mathcal{I}} = \mathbf{V}_{R_1} \cup \mathbf{V}_{R_2} \cup ... \cup \mathbf{V}_{R_{G\times H}}$. However, subimages generated from different partitions possibly overlap.

Under each partition, we are concerned on whether there exists a *good subimage* that contains the common pattern $\mathcal{P}$. This depends on if pattern $\mathcal{P}$ is broken under this partition. Without losing generality, we assume that the pattern $\mathcal{P}$ appears at most once in each image. Supposing the spatial size of the image $\mathcal{I}$ is $(I_x, I_y)$ and the bounding box of the common pattern $\mathcal{P}$ is $C_{\mathcal{P}} = (P_x, P_y)$, we calculate the *non-broken probability* for $\mathcal{P}$ as the probability that none of the $(G-1)+(H-1)$ partition lines penetrates $C_{\mathcal{P}}$:

$$p = (1 - \frac{P_x}{I_x})^{G-1}(1 - \frac{P_y}{I_y})^{H-1}. \tag{1}$$

Given a partition of an image, we can find at most one good subimage with probability $p$, if the image contains no more than one such pattern. For instance in Fig. 2, there are in total 7 good subimages and 2 other ones are missed.

## 2.3. Matching and discovering popular subimages

The objective is to match subimages pair-wisely and discover "popular" ones from the pool $\mathbf{D}_{\mathcal{R}}$. Here a *popular subimage* is the one that contains a common pattern $\mathcal{P}$ and has enough matches in $\mathbf{D}_{\mathcal{R}}$.

**subimage matching**

Unlike the "bag of words" method, we cannot match subimages through "histogram of words" since our visual primitives are not quantized into words. Instead, $\forall\, \mathcal{R}, \mathcal{Q} \in \mathbf{D}_{\mathcal{R}}$, we measure the similarity by matching their visual primitives $\mathbf{V}_{\mathcal{R}}$ and $\mathbf{V}_{\mathcal{Q}}$ directly. Matching can be formulated as an assignment problem:

$$\mathbf{Sim}(\mathbf{V}_{\mathcal{R}}, \mathbf{V}_{\mathcal{Q}}) = \max_{\mathcal{F}} \sum_{i=1}^{|\mathbf{V}_{\mathcal{R}}|} s(v_i, \mathcal{F}(v_i)), \tag{2}$$

where $\mathcal{F}(\cdot)$ is the assignment function $\mathcal{F} : \mathbf{V}_{\mathcal{R}} \rightarrow \mathbf{V}_{\mathcal{Q}}$, i.e., for each $v_i \in \mathbf{V}_{\mathcal{R}}$, $\mathcal{F}$ assigns its matching $u_j = \mathcal{F}(v_i) \in \mathbf{V}_{\mathcal{Q}}$. Each $v_i$ can match only one $u_j$ and verce vice; $s(v_i, \mathcal{F}(v_i))$ is the similarity measure between $v_i$ and its assignment $u_j$. Two subimages are matched if their similarity $\mathbf{Sim}(\mathbf{V}_{\mathcal{R}}, \mathbf{V}_{\mathcal{Q}}) \geq \lambda$, where $\lambda > 0$ is the subimage matching threshold. Generally, it is non-trivial and computationally demanding to solve this assignment problem.

In this paper, we present an approximate solution to this problem with a linear complexity. Firstly, we perform a pre-processing step on the visual primitive database $\mathbf{D}_v$. This is the overhead of our pattern discovery method. For each $v \in \mathbf{D}_v$, we perform the $\epsilon$- Nearest Neighbors ($\epsilon$-NN) query and define the retrieved $\epsilon$-NN set of $v$ as its *match-set* $\mathcal{M}_v = \{u \in \mathbf{D}_v : \left\| \vec{f}_v - \vec{f}_u \right\| \leq \epsilon\}$. In order to reduce the computational cost in finding $\mathcal{M}_v$ for each $v$, we apply LSH [4] that performs efficient $\epsilon$-NN queries.

After obtaining all the match-sets, $\forall\, v, u \in \mathbf{D}_v$, we define their similarity measure $s(v, u)$ as:

$$s(v,u) = \begin{cases} \exp^{-\frac{\|\vec{f}_v - \vec{f}_u\|^2}{\alpha}}, & \textit{if } v \in \mathcal{M}_u \\ 0, & \textit{otherwise} \end{cases}, \tag{3}$$

where $\alpha > 0$ is a parameter and $\mathcal{M}_u$ depends on the threshold $\epsilon$. $s(v,u)$ is a symmetric measure as $v \in \mathcal{M}_u \Leftrightarrow u \in \mathcal{M}_v$. This visual primitive matching is illustrated in Fig. 3.

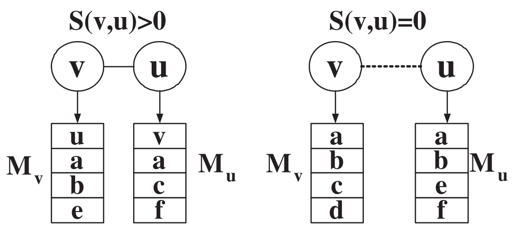

Figure 3. Similarity measure of two visual primitives $s(v,u)$, where $a$,$b$,$c$,$d$,$e$,$f$ denote visual primitives. We notice $s(v,u) = 0$ when $v \notin \mathcal{M}_u$ and $u \notin \mathcal{M}_v$.

Now suppose that $\mathbf{V}_{\mathcal{R}} = \{v_1, v_2, ..., v_m\}$ and $\mathbf{V}_{\mathcal{Q}} = \{u_1, u_2, ..., u_n\}$ are two sets of visual primitives. We can approximate the match between $\mathbf{V}_{\mathcal{R}}$ and $\mathbf{V}_{\mathcal{Q}}$ in Eq. 4, by evaluating the size of the intersection between $\mathbf{V}_{\mathcal{R}}$ and the match-set of $\mathbf{V}_{\mathcal{Q}}$:

$$\begin{aligned} \widetilde{Sim}(\mathbf{V}_{\mathcal{R}}, \mathbf{V}_{\mathcal{Q}}) &\triangleq |\mathbf{V}_{\mathcal{R}} \cap \mathcal{M}_{\mathbf{V}_{\mathcal{Q}}}| && (4) \\ &\geq \max_{\mathcal{F}} \sum_{i=1}^{|\mathbf{V}_{\mathcal{R}}|} s(v_i, \mathcal{F}(v_i)) && (5) \\ &= \mathbf{Sim}(\mathbf{V}_{\mathcal{R}}, \mathbf{V}_{\mathcal{Q}}), && (6) \end{aligned}$$

where $\widetilde{Sim}(\mathbf{V}_{\mathcal{R}}, \mathbf{V}_{\mathcal{Q}})$ is a positive integer; $\mathcal{M}_{\mathbf{V}_{\mathcal{Q}}} = \mathcal{M}_{u_1} \cup \mathcal{M}_{u_2} ... \cup \mathcal{M}_{u_n}$ denotes the match-set of $\mathbf{V}_{\mathcal{Q}}$. We apply the property that $0 \leq s(v,u) \leq 1$ to prove Eq. 5. As shown in Fig. 4, $\widetilde{Sim}(\mathbf{V}_{\mathcal{R}}, \mathbf{V}_{\mathcal{Q}})$ can be viewed as the approximate *flow* between $\mathbf{V}_{\mathcal{R}}$ and $\mathbf{V}_{\mathcal{Q}}$. Based on the approximate similarity score, two subimages are matched if $\widetilde{Sim}(\mathbf{V}_{\mathcal{R}}, \mathbf{V}_{\mathcal{Q}}) \geq \lambda$. Since we always have $\widetilde{Sim}(\mathbf{V}_{\mathcal{R}}, \mathbf{V}_{\mathcal{Q}}) \geq \mathbf{Sim}(\mathbf{V}_{\mathcal{R}}, \mathbf{V}_{\mathcal{Q}})$, the approximate similarity score is a safe bounded estimation. The intersection of two sets $\mathbf{V}_{\mathcal{R}}$ and $\mathcal{M}_{\mathbf{V}_{\mathcal{Q}}}$ can be performed in a linear time $O(|\mathbf{V}_{\mathcal{R}}| + |\mathcal{M}_{\mathbf{V}_{\mathcal{Q}}}|) = O(m + nc)$, where $c$ is the average size of the match-set for all $v$. Since $m \approx n$, the complexity is essentially $O(mc)$.

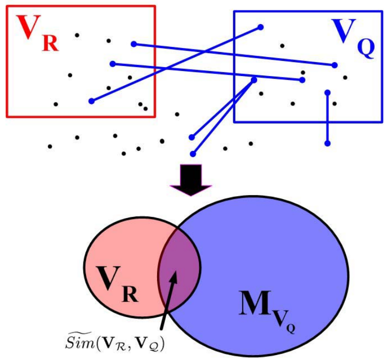
V_R
V_Q
V_R
M_{V_Q}
$\widetilde{Sim}(\mathbf{V}_{\mathcal{R}}, \mathbf{V}_{\mathcal{Q}})$

Figure 4. Similarity matching of two subimages. Each point is a visual primitive and edges show correspondences between visual primitives. The flow between $\mathbf{V}_{\mathcal{R}}$ and $\mathbf{V}_{\mathcal{Q}}$ can be approximated by the set intersection $\widetilde{Sim}(\mathbf{V}_{\mathcal{R}}, \mathbf{V}_{\mathcal{Q}})$.

**Finding popular subimages**

Based on the matching defined above, we are ready to find *popular subimages*. Firstly, we denote $\mathbf{G}_{\mathcal{R}} \subset \mathbf{D}_{\mathcal{R}}$ as the set of good subimages which contain the common pattern $\mathcal{P}$: $\forall\ \mathcal{R}_g \in \mathbf{G}_{\mathcal{R}}$, we have $\mathcal{P} \subset \mathcal{R}_g$. A good subimage becomes a *popular subimage* if it has enough matches in the pool $\mathbf{D}_{\mathcal{R}}$. As we do not allow $\mathcal{R}_g$ to match subimages in the same image as $\mathcal{R}_g$, its *popularity* is defined as the number of good subimages in the rest of $(T-1) \times K$ partitions. As each partition $k$ can generate one good subimage with probability $p$ (Eq. 1), the total matches $\mathcal{R}_g$ can find is a binomial random variable: $\mathbf{Y}_{\mathcal{R}_g} \sim B(K(T-1), p)$, where $p$ depends on the partition parameters and the shape of the common pattern (Eq. 1). The more matches $\mathcal{R}_g$ can find in $\mathbf{D}_{\mathcal{R}}$, the more likely that $\mathcal{R}_g$ contains a common pattern and more significant it is. On the other hand, unpopular $\mathcal{R}$ may not contain any common spatial pattern as it cannot find supports from other subimages.

Based on the expectation of matches that a good subimage can find, we apply the following truncated 3-$\sigma$ criterion to determine the threshold for the popularity:

$$\tau = \mu - 3\sigma = (T-1)Kp - 3\sqrt{(T-1)Kp(1-p)}, \quad (7)$$

where $\mu = E(\mathbf{Y}_{\mathcal{R}_g}) = (T-1)Kp$ is the expectation of $\mathbf{Y}_{\mathcal{R}_g}$ and $\sigma^2 = Var(\mathbf{Y}_{\mathcal{R}_g}) = (T-1)Kp(1-p)$ is the variance.For every subimage $\mathcal{R} \in \mathbf{D}_{\mathcal{R}}$, we query it in $\mathbf{D}_{\mathcal{R}} \backslash \mathcal{I}_t$ to check its popularity, where $\mathcal{I}_t$ is the image that generates $\mathcal{R}$. If $\mathcal{R}$ can find at least $\lceil \tau \rceil$ matches, it is a popular one.

### 2.4. Voting and locating common patterns

After discovering all the popular subimages (denoted by set $\mathbf{S}_{\mathcal{R}} \subset \mathbf{G}_{\mathcal{R}}$), they vote for the common patterns. For each image, we select all popular subimages that are associated with this image. Aggregating these popular subimages must produce overlapped regions where common patterns are located. A densely overlapped region is thus the most likely location for a potential common pattern $\mathcal{P}$. Each popular subimage votes its corresponding pattern in a *voting map* associated with this image.

Since we perform the spatial random partition $K$ times for each image, each pixel $l \in \mathcal{I}$ has up to $K$ chances to be voted, from its $K$ corresponding subimages $\mathcal{R}_l^k (k = 1, ..., K)$ that contains $l$. The more votes a pixel receives, the more probable that it is located inside a common pattern. More formally, for the common pattern pixel $i \in \mathcal{P}$, the probability it can receive a vote under a certain random partition $k \in \{1, 2, ..., K\}$ is:

$$\begin{aligned} & Pr(x_i^k = 1) = Pr(\mathcal{R}_i^k \in \mathbf{S}_{\mathcal{R}}) \\ = \ & Pr(\mathcal{R}_i^k \in \mathbf{G}_{\mathcal{R}}) Pr(vote(\mathcal{R}_i^k) \geq \lceil \tau \rceil \,|\mathcal{R}_i^k \in \mathbf{G}_{\mathcal{R}}) \\ = \ & pq, \end{aligned} \quad (8)$$

where the superscript $k$ indexes the partition and the subscript $i$ indexes the pixel; $\mathcal{R}_i^k$ is the subimage that contains $i$; $p$ is the prior that $i$ is located in a *good* subimage, i.e. $Pr(\mathcal{R}_i^k \in \mathbf{G}_{\mathcal{R}})$, the non-broken probability of $\mathcal{P}$ under a partition (Eq. 1); $q$ is the likelihood that a *good* subimage $\mathcal{R}_i^k$ is also a *popular* one, which depends on the number of matches $\mathcal{R}_i^k$ can find. Specifically, under our popular subimage discovery criterion in Eq. 7, $q$ is a constant. Given a pixel $i$, $\{x_i^k, k = 1, 2, ..., K\}$ is a set of independent and identically distributed (i.i.d.) Bernoulli random variables. Aggregating them together, the votes that $i \in \mathcal{P}$ can receive is a binomial random variable $\mathbf{X}_i^K = \sum_{k=1}^K x_i^k \sim B(K, pq)$. Thus we can determine the common pattern regions based on the number of votes they receive.

Under each partition $k$, $\mathcal{P}$ is voted by the popular subimage $\mathcal{R}_{\mathcal{P}}^k \in \mathbf{S}_{\mathcal{R}}$. Since $\mathcal{R}_{\mathcal{P}}^k$ contains $\mathcal{P}$, it gives an estimation of the location for $\mathcal{P}$. However, a larger size of $\mathcal{R}_{\mathcal{P}}^k$ implies more uncertainty it has in locating $\mathcal{P}$ and thus its vote should take less credit. We thus adjust the weight of the vote based on the size of $\mathcal{R}_{\mathcal{P}}^k$. $\forall\ i \in \mathcal{P}$, we weight the votes:

$$\mathbf{X}_i^K = \sum_{k=1}^K w_i^k x_i^k, \quad (9)$$

where $w_i^k > 0$ is the weight of the $k_{th}$ vote. Among the many possible choices, in this paper we set $w_i^k = \frac{area(\mathcal{I})}{area(\mathcal{R}_i^k)}$, meaning the importance of the popular subimage $\mathcal{R}_i^k$. The larger the $area(\mathcal{R}_i^k)$, the smaller weight its vote counts. Sec. 3.1 will discuss the criteria and principle in selecting a suitable $w_i^k$. Finally, we can roughly segment the common patterns given the voting map, based on the expected number of votes a common pattern pixel should receive. This rough segmentation can be easily refined by combining it with many existing image segmentation schemes, such as the level set based approach.

## 3. Properties of the Algorithm

### 3.1. Asymptotic property

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

**Theorem 1** *Asymptotic property*
*We consider two pixels $i, j \in \mathcal{I}$, where $i \in \mathcal{P} \subset \mathcal{I}$ is located inside one common pattern $\mathcal{P}$ while $j \notin \mathcal{P}$ is located outside any common patterns (e.g. in the background). Suppose $\mathbf{X}_i^K$ and $\mathbf{X}_j^K$ are the votes for $i$ and $j$ respectively, considering $K$ times random partitions. Both $\mathbf{X}_i^K$ and $\mathbf{X}_j^K$ are discrete random variables, and we have:*

$$\lim_{K \to \infty} P(\mathbf{X}_i^K > \mathbf{X}_j^K) = 1. \quad (10)$$

The above theorem states that when we have enough times of partitions for each image, a common pattern region $\mathcal{P}$ must receive more votes, so that it can be easily discovered and located. The proof of Theorem 1 is given in the Appendix. We briefly explain its idea below.

**Explanation of Theorem 1**

We consider two pixels $i \in \mathcal{P}$ and $j \notin \mathcal{P}$ as stated in Theorem 1. We are going to check the total number of votes that $i$ and $j$ will receive after $K$ times partitions of $\mathcal{I}$.

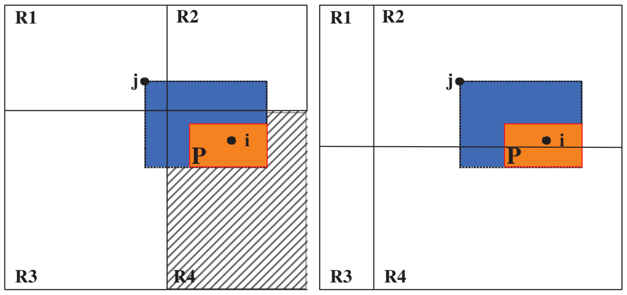

Figure 5. Illustration of the EVR. The figures show two different random partitions on the same image. The small orange rectangle represents the common pattern $\mathcal{P}$. We compare two pixels $i \in \mathcal{P}$ and $j \notin \mathcal{P}$. The large blue region represents $\mathcal{R}'_j$, the EVR of $j$; while $\mathcal{R}'_i = \mathcal{P}$. In the left figure, $\mathcal{R}'_j$ is broken during the partition while $\mathcal{R}'_i$ is not. Thus $i$ get a vote because $\mathcal{R}_4$ (shadow region) is a popular subimage and the whole region is voted; while $j$ does not receive the vote. In the right image, both $\mathcal{R}'_i$ and $\mathcal{R}'_j$ are broken during the partition, so neither $i$ and $j$ is voted as no popular subimage appears.

For each pixel $l \in \mathcal{I}$, we define its *Effective Vote Region* (EVR) as:

$$\mathcal{R}'_l = \underset{\mathcal{R}}{\operatorname{argmin}}\, area(\mathcal{R}|\mathcal{P} \subseteq \mathcal{R}, l \in \mathcal{R}), \tag{11}$$

where $\mathcal{R}$ is a rectangle image region that contains both the common pattern $\mathcal{P}$ and the pixel $l$. Fig. 5 illustrates the concept of EVR. Based on the definition, both EVR $\mathcal{R}'_i$ and $\mathcal{R}'_j$ contain $\mathcal{P}$. For the "positive" pixel $i \in \mathcal{P}$, we have: $\mathcal{R}'_i = \mathcal{P}$. On the other hand, for the "negative" pixel $j \notin \mathcal{P}$, it corresponds to a larger EVR $\mathcal{R}'_j$, and we have $\mathcal{R}'_i \subset \mathcal{R}'_j$.

Like pixel $i$, whether $j \notin \mathcal{P}$ can get a vote depends on whether its subimage $\mathcal{R}^k_j$ is a popular one. Suppose the spatial size of the EVR $\mathcal{R}'_j$ is $(B_x, B_y)$. Similar to Eq. 1, the non-broken probability of $\mathcal{R}'_j$ is:

$$p_j = (1 - \frac{R_x}{I_x})^{G-1}(1 - \frac{R_y}{I_y})^{H-1}. \tag{12}$$

Following the same analysis in Eq. 8, $x^k_j$ is a Bernoulli random variable:

$$Pr(x^k_j) = \begin{cases} p_j q, & x^k_j = 1, \\ 1 - p_j q, & x^k_j = 0, \end{cases} \tag{13}$$

where $q$ is the likelihood of the good subimage being a popular one, which is a constant unrelated with $p_j$ (Eq. 7). Thus whether a pixel $j \notin \mathcal{P}$ can receive a vote depends on the size of its EVR. When considering $K$ times random partitions, the total number of votes for pixel $j \notin \mathcal{P}$ is also a binomial random variable $\mathbf{X}_j = \sum_{k=1}^{K} x^k_j \sim B(K, p_j q)$.

Since $\mathcal{R}'_i \subset \mathcal{R}'_j$, we have $B_x > P_x$ and $B_y > P_y$. It is easy to see $p_i > p_j$ by comparing Eq. 1 and 12. When we consider the unweighted voting (i.e. $w^k_i = w^k_j = 1$), $i$ is expected to receive more votes than $j$ because $E(\mathbf{X}^K_i) = p_i qK > E(\mathbf{X}^K_j) = p_j qK$. In the case of the weighted voting, we can estimate the expectation of $\mathbf{X}^K_i$ as:

$$E(\mathbf{X}^K_i) = \sum_{k=1}^{K} E(w^k_i x^k_i) = \sum_{k=1}^{K} E(w^k_i) E(x^k_i) \tag{14}$$

$$= \sum_{k=1}^{K} pqE(w^k_i) = pqKE(w_i), \tag{15}$$

where we assume $w^k_i$ be independent to $x^k_i$ and $E(w_i)$ is only related to the average size of the popular subimage. Therefore to prove Theorem 1, we need to guarantee that $E(\mathbf{X}^K_i) = p_i qKE(w_i) > p_j qKE(w_j) = E(\mathbf{X}^K_j)$. It follows that we need to select suitable weighting strategy such that $p_i E(w_i) > p_j E(w_j)$. A possible choice is given in Sec. 2.4.

It is worth mentioning that the expected number of votes $E(\mathbf{X}^K_i) = p_i qKE(w_i)$ depends on the spatial partition scheme $G \times H \times K$, where $p_i$ depends on $G$ and $H$ (Eq. 1), $q$ depends on both $p$ and $K$ (Eq. 7), and $w_i$ depends on $G$ and $H$ as well. Our method does not need the prior knowledge of the pattern, but knowing the shape of the pattern can help choose better $G$ and $H$, which leads to faster convergence (Theorem 1). A larger $K$ results in more accurate patterns but needs more computation. In general, $G$ and $H$ are best selected to match the spatial shape of the hidden common pattern $\mathcal{P}$ and the larger the $K$, the more accurate our approximation is but more computation is required.

## 3.2. Computational complexity analysis

Let $M = |\mathbf{D}_\mathcal{R}| = G \times H \times K \times T$ denote the size of the subimage database $\mathbf{D}_\mathcal{R}$. In general, $M$ is much less than $N = |\mathbf{D}_v|$, the total number of visual primitives, when selecting hashing parameters suitably. Because we need to evaluate $\binom{M}{2}$ pairs, the complexity for discovering popular subimages in $\mathbf{D}_\mathcal{R}$ is $O(M^2(mc))$, where $mc$ is the cost for matching two sets of visual primitives $m = |\mathbf{V}_\mathcal{R}|$ and $n = |\mathbf{V}_\mathcal{Q}|$ as analyzed before, where $c$ is a constant. The overhead of our approach is to find the $\mathcal{M}_v$ for each $v \in \mathbf{D}_v$ (formation of $\mathbf{D}_\mathcal{R}$ is of a linear complexity $O(N)$ and thus ignored). By applying LSH, each query complexity can be reduced from $O(dN)$ to less than $O(dN^{\frac{1}{a}})$ where $a > 1$ is the approximation factor [4] and $d$ is the feature dimension. As we have in total $N$ such queries, the total overhead cost is $O(dN^{1+\frac{1}{a}})$.

We further compare the computational complexity of our method to two existing methods [1] [12] in Table 1. The overhead of [12] comes from clustering visual primitives

| Method | overhead | matching | discovering |
|---|---|---|---|
| [12] | $O(dNki)$ | $O(k)$ | $O(N^2k)$ |
| [1] | none | $O(Nd+mb)$ | $O(N(Nd+mb))$ |
| Ours | $O(dN^{1+\frac{1}{a}})$ | $O(mc)$ | $O(M^2mc)$ |

Table 1. Computational complexity comparision. The total cost is the overhead part plus the discovering part.

into $k$ types of words through the $k$-means algorithm, and $i$ is the number of iterations. We estimate the discovering complexity of [1] by assuming that there are in total $O(N)$ number of queries for evaluation, each time applying the fast inference algorithm proposed in [1], where $b$ is the constant parameter. It is clear that our method is computationally more efficient as $M << N$. In the example shown in Fig. 1, we detect $N = 3416$ visual primitives for two images. When performing a hashing scheme as $G \times H \times K = 4 \times 3 \times 30$, we generate $M = 360 \times 2$ subimages. The CPU cost of the overhead of performing LSH is around 7.6 seconds (without parameter optimization), and the CPU cost of pattern discovery is around 4.7 seconds where the average cost of each subimage matching is less than 0.1 millisecond.

## 4. Experiments

We apply Scale Invariant Features (SIFT) [7] as the visual primitives although other local invariant features are certainly possible. Each SIFT descriptor $\vec{f_v}$ is a 128-$d$ vector which characterizes the local invariance of a visual primitive $v$. We set $\epsilon = 200$ in the LSH-based $\epsilon$-NN query; the subimage matching threshold is $\lambda = 40$. All the images are of size $640 \times 480$ and each image can generate 1000 to 3000 visual primitives. Without specific indication, we apply the random partition scheme $G \times H \times K = 4 \times 3 \times 50$, where $G \times H$ is selected according to the aspect ratio of each image. All the experiments are performed on a Pentium-4 3.19GHz machine with 1GB RAM. The algorithm is implemented in C++.

### 4.1. Common pattern discovery

The results of the pattern discovery on the two images in Fig. 1 are presented in Fig. 6. As each voting subimage is a rectangle, the voting map is a union of weighted rectangles. The brightness of the regions indicates the total votes each pixel receives, i.e. the likelihood belonging to a common pattern. At the moment, pattern boundary is not accurate enough as a very naive fixed threshold $E(X_i^k)$ (Eq. 15) is applied. Depending on applications, accurate segmentation may not matter much. But if required, the boundaries can be further improved by incorporating other image segmentation techniques.

To evaluate our algorithm, we use 8 image datasets, where each dataset contains 1 to 3 different common patterns and each common pattern has 2 to 4 instances within

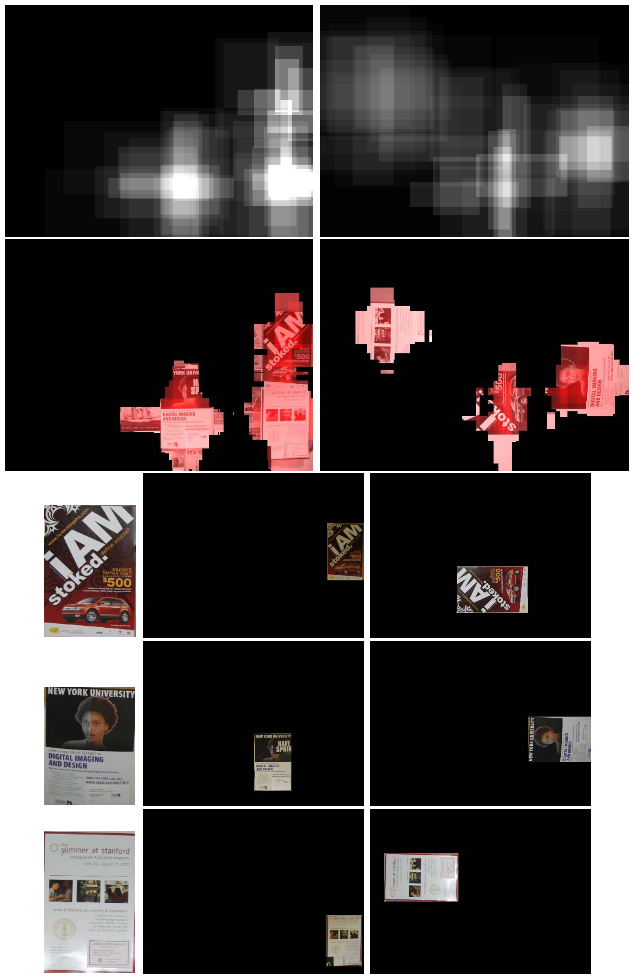

Figure 6. The input is from Fig. 1. The $1_{st}$ and the $2_{nd}$ row show the voting maps and the rough segmentations respectively. The three common patterns are listed from the $3_{rd}$ to the $5_{th}$ row. Besides variations like rotation and scale change, the second poster suffers from partial occlusion in the left image. The hit ratio and the background ratio are $h_r = 0.76$ and $b_r = 0.29$ respectively.

the dataset. Each dataset contains 4 to 8 images where some of them may not contain any common pattern. Around half of the images have multiple common patterns. The instances of each common pattern exhibit possible variations like rotation, partial occlusion, scale and slightly viewpoint changes and are located in clutter backgrounds. Given an image dataset, we evaluate the performance by checking how accurate the regions containing the common patterns are recovered. Let $R$ and $GT$ be respectively the discovered common pattern regions and the bounding boxes of ground truth patterns. The performance is evaluated by using two criteria: hit ratio $h_r = \frac{|GT \cap R|}{|GT|}$ and the background ratio $b_r = \frac{|R| - |R \cap GT|}{|R|}$. Table 2 presents the performance. Since common patterns are of different shapes and sizes, it is unfair to apply a unique non-broken proba-

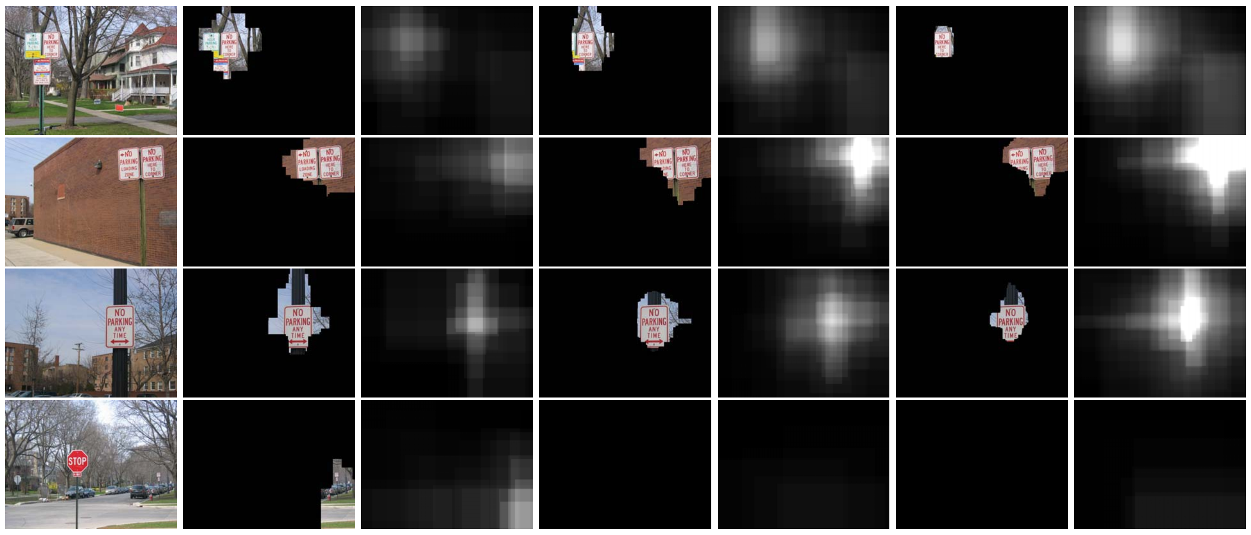

Figure 7. The input is a set of four images (the $1_{st}$ column). Each row corresponds to an image. The common pattern "no parking sign" appears in the first three images. A comparison of applying the different partition times $K$ is shown in the $2_{nd}$ ($K = 25$), $4_{th}$ ($K = 50$) and $6_{th}$ ($K = 75$) columns. The $3_{th}$, $5_{th}$ and $7_{th}$ columns are voting maps associated with the corresponding images.

bility $p$ for all image dataset. For each image dataset, the best result is reported by searching optimal $p$ from $0.15$ to $0.30$. We describe each dataset as the common pattern it contains: $Dataset_A$ ={Fedex}, $Dataset_B$ ={Stop Sign}, $Dataset_C$={No Parking Sign}, $Dataset_D$={Tea Box}, $Dataset_E$={Tag}, $Dataset_F$={Book, Tea Box and Tag}, $Dataset_G$={Books}, $Dataset_H$={Posters}.

| Dataset | **A** | **B** | **C** | **D** | **E** | average |
|---|---|---|---|---|---|---|
| Hit Ratio | 0.95 | 0.98 | 0.92 | 0.45 | 0.91 | 0.84 |
| Bk Ratio | 0.41 | 0.55 | 0.45 | 0.15 | 0.36 | 0.38 |
| Dataset | **F**$^*$ | **G**$^*$ | **H**$^*$ | | | average |
| Hit Ratio | 0.80 | 0.85 | 0.76 | | | 0.80 |
| Bk Ratio | 0.53 | 0.37 | 0.29 | | | 0.40 |

Table 2. Performance evaluation. The superscript $*$ denotes the dataset containing multiple common patterns. See text for details.

Another example of pattern discovery in multiple images is presented in Fig. 7. Three of the four input images contain the common pattern: "no parking sign", and the fourth image does not. We compare results under various partition times $K$ ($K$=25,50,75). One false alarm appears when $K = 25$. This is caused by fake popular subimages. However when a larger $K$ is selected, these fake popular subimages can be rejected as the threshold for popularity increases with $K$. In general, larger $K$ produces more accurate localization of the patterns, which supports Theorem 1.

### 4.2. Image irregularity detection

The spatial random partition method can also be applied for single image irregularity detection. Firstly, we perform the $G \times H \times K$ spatial partitions on the input image to generate a subimages database. Secondly, for each generated subimage, we match it with all others. In contrast to common pattern discovery, we select the *unpopular* subimages instead of the popular ones for irregularity detection. The subimage matching is performed based on Eq. 4, but a lower threshold is applied. Therefore common subimages are more easily to be matched. If there exists an irregular pattern in the image, then a subimage containing this irregular pattern is very likely to be an *unpopular* one as it cannot match other subimages. After discovering all unpopular subimages, they vote for the irregular pattern and build the accumulated voting map similar to the discussion in Sec. 2.4. The only difference lies in the weighting strategy of voting. Now we weight the vote in proportion to the size of the unpopular subimage, i.e. $w_i^k \propto area(\mathcal{R}_i^k)$, because a larger unpopular subimage is more likely to include an irregular pattern. Fig. 8 shows an example of irregularity detection. We perform $K = 75$ partitions to obtain a good boundary. In order to avoid detecting the blank background, subimages that contain very few visual primitives are ignored.

## 5. Conclusion

We present a novel method based on spatial random partition for common pattern discovery and image irregularity detection. Our method is robust to various pattern variations. Although no word dictionary is needed, our method is still efficient because it employs LSH for fast $\epsilon$-NN query and uses the bounded approximation for subimage matching. The asymptotic property of the proposed algorithm provides a theoretical guarantee for its performance. Partition times $K$ trades-off between the accuracy of the localization and the speed. Although only SIFT features

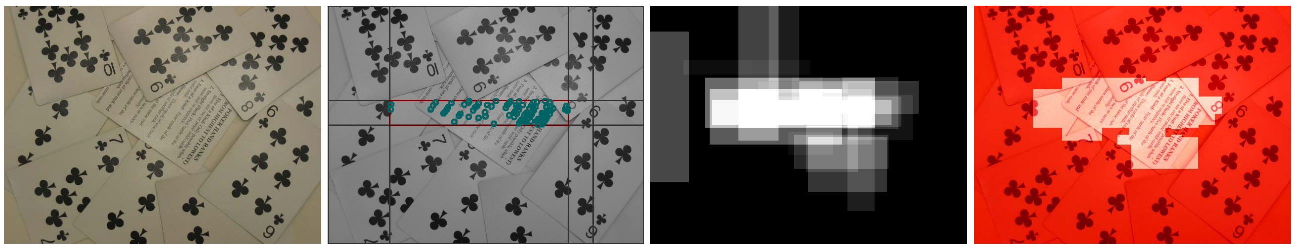

Figure 8. Spatial random partition for image irregularity detection. An instance of the random partition and the unpopular subimage are shown in the $2_{nd}$ image. The subimage that contain green circles (denoting visual primitives) is the *unpopular* one. After segmenting the voting map (the $3_{rd}$ image), we obtain the final irregularity detection result (the $4_{th}$ image).

are used in the experiments reported, our method is generally applicable in using many other types of visual primitives (e.g. over-segment regions) and features (e.g. color histograms). The only requirement is that two subimages should be matched when they share a common pattern.

## 6. Appendix

We prove theorem 1 here. Given two pixels $i$ and $j$, let both $\{x_i^k\}_{k=1}^K$ and $\{x_j^k\}_{k=1}^K$ be sequences of i.i.d. Bernoulli random variables indicating whether the corresponding pixel can receive a vote at the partition $k$. Assuming $x_i^k$ and $x_j^k$ are independent given $i$ and $j$, we have:

$$\mathbf{X}_i^K - \mathbf{X}_j^K = \sum_{k=1}^{K} x_i^k - \sum_{k=1}^{K} x_j^k = \sum_{k=1}^{K} (x_i^k - x_j^k), \qquad (16)$$

where elements $\{(x_i^k - x_j^k)\}_{k=1}^K$ are also i.i.d. random variables. It is easy to see that $E(x_i^k - x_j^k) = p_i - p_j$. Thus according to the weak law of large numbers, we have $\forall\ \epsilon > 0$

$$\lim_{K\to\infty} Pr(|\frac{\mathbf{X}_i^K - \mathbf{X}_j^K}{K} - p_i + p_j| < \epsilon) = 1. \qquad (17)$$

Now let $\epsilon = \frac{p_i - p_j}{2} > 0$, we have

$$\lim_{K\to\infty} Pr(\frac{(\mathbf{X}_i^K - \mathbf{X}_j^K)}{K} > \frac{p_i - p_j}{2}) = 1. \qquad (18)$$

Since $p_i - p_j > 0$, it follows

$$\lim_{K\to\infty} Pr(\mathbf{X}_i^K - \mathbf{X}_j^K > 0) = 1. \qquad (19)$$

## Acknowledgment

This work was supported in part by National Science Foundation Grants IIS-0347877 and IIS-0308222.

## References

[1] O. Boiman and M. Irani. Detecting irregularities in images and in video. In *Proc. IEEE Intl. Conf. on Computer Vision*, 2005. 1, 5, 6

[2] O. Boiman and M. Irani. Similarity by composition. In *Proc. of Neural Information Processing Systems*, 2006. 1

[3] E. Borenstein and S. Ullman. Learning to segment. In *Proc. European Conf. on Computer Vision*, 2004. 1

[4] M. Datar, N. Immorlica, P. Indyk, and V. Mirrokni. Locality-sensitive hashing scheme based on p-stable distribution. In *Proc. of Twentitieth Annual Symposium on Computational Geometry*, 2004. 3, 5

[5] K. Grauman and T. Darrell. Unsupervised learning of categories from sets of partially matching image features. In *Proc. IEEE Conf. on Computer Vision and Pattern Recognition*, 2006. 1

[6] P. Hong and T. S. Huang. Spatial pattern discovery by learning a probabilistic parametric model from multiple attributed relational graphs. *Discrete Applied Mathematics*, pages 113–135, 2004. 1

[7] D. Lowe. Distinctive image features from scale-invariant keypoints. *Intl. Journal of Computer Vision*, 2004. 6

[8] 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. 1

[9] C. Rother, V. Kolmogorov, T. Minka, and A. Blake. Cosegmentation of image pairs by histogram matching: incorporating a global constraint into mrfs. In *Proc. IEEE Conf. on Computer Vision and Pattern Recognition*, 2006. 1

[10] 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. 1

[11] U. Rutishauser, D. Walther, C. Koch, and P. Perona. Is bottom-up attention useful for object recognition? In *Proc. IEEE Conf. on Computer Vision and Pattern Recognition*, 2004. 1

[12] J. Sivic and A. Zisserman. Video data mining using configurations of viewpoint invariant regions. In *Proc. IEEE Conf. on Computer Vision and Pattern Recognition*, 2004. 1, 5, 6

[13] K.-K. Tan and C.-W. Ngo. Common pattern discovery using earth mover's distance and local flow maximization. In *Proc. IEEE Intl. Conf. on Computer Vision*, 2005. 1

[14] S. Todorovic and N. Ahuja. Extracting subimages of an unknown category from a set of images. In *Proc. IEEE Conf. on Computer Vision and Pattern Recognition*, 2006. 1

[15] J. Winn and N.Jojic. Locus: Learning object classes with unsupervised segmentation. In *Proc. IEEE Intl. Conf. on Computer Vision*, 2005. 1

[16] 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. 1