Int J Comput Vis (2009) 85: 143–166
DOI 10.1007/s11263-009-0252-y

Foreground Focus: Unsupervised Learning from Partially Matching Images

Yong Jae Lee · Kristen Grauman

Received: 11 July 2008 / Accepted: 13 May 2009 / Published online: 27 May 2009

Abstract We present a method to automatically discover meaningful features in unlabeled image collections. Each image is decomposed into semi-local features that describe neighborhood appearance and geometry. The goal is to determine for each image which of these parts are most relevant, given the image content in the remainder of the collection. Our method first computes an initial image-level grouping based on feature correspondences, and then iteratively refines cluster assignments based on the evolving intra-cluster pattern of local matches. As a result, the significance attributed to each feature influences an image's cluster membership, while related images in a cluster affect the estimated significance of their features. We show that this mutual reinforcement of object-level and feature-level similarity improves unsupervised image clustering, and apply the technique to automatically discover categories and foreground regions in images from benchmark datasets.

Keywords Object recognition · Feature selection · Unsupervised learning · Feature descriptor

1 Introduction

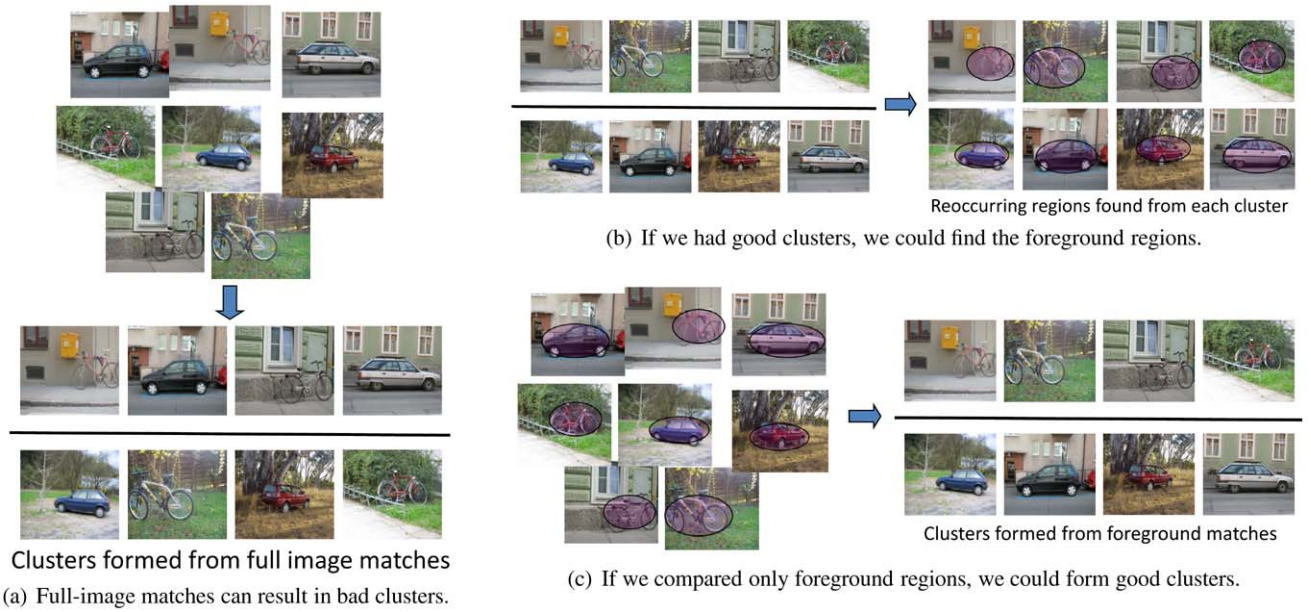
Learning to describe and recognize visual objects is a fundamental problem in computer vision that serves as a building block to many potential applications. Recent years have shown encouraging progress, particularly in terms of generic visual category learning (Weber et al. 2000; Leibe et al. 2004; Winn and Jojic 2005; Chum and Zisserman 2007; Ling and Soatto 2007) and robust local feature representations (Lowe 2004; Agarwal and Triggs 2006; Lazebnik et al. 2004). A widespread strategy is to determine the commonalities in appearance and shape amongst a group of labeled images, and then search for similar instances in new images based on those patterns. Typically one assumes that categories may be learned in a supervised setting, where the recognition method is trained with manually prepared exemplars of each class of interest. This format of the problem continues to yield good results, as evidenced by steady accuracy improvements on benchmark datasets (Everingham et al. 2006; Fei-Fei et al. 2004).

However, carefully labeled exemplars are expensive to obtain in the large numbers needed to fully represent a category's variability, and methods trained in this manner can suffer from unintentional biases imparted by dataset creators. Recognition methods stand to gain from stores of unstructured, unlabeled images and videos, if they can infer which basic visual patterns are meaningful. While recent work has begun to address the need for looser supervision requirements (Weber et al. 2000; Winn and Jojic 2005; Fergus et al. 2005; Sivic et al. 2005; Grauman and Darrell 2006), learning from completely unlabeled images remains difficult. Unsupervised learners face the same issues that plague supervised methods—clutter, viewpoint, intra-class appearance variation, occlusions—but must handle them without any explicit annotation guidance.

In this work we consider the problem of automatically identifying the foreground object(s) of interest among an unlabeled pool of images. To qualify as foreground, we say that the visual pattern must have observable support within the

Y.J. Lee (✉)
Department of Electrical and Computer Engineering,
University of Texas at Austin, Austin, TX 78712, USA
e-mail: yjlee0222@mail.utexas.edu

K. Grauman
Department of Computer Sciences, University of Texas at Austin,
Austin, TX 78712, USA
e-mail: grauman@cs.utexas.edu



(a) Full-image matches can result in bad clusters.

(b) If we had good clusters, we could find the foreground regions.

(c) If we compared only foreground regions, we could form good clusters.

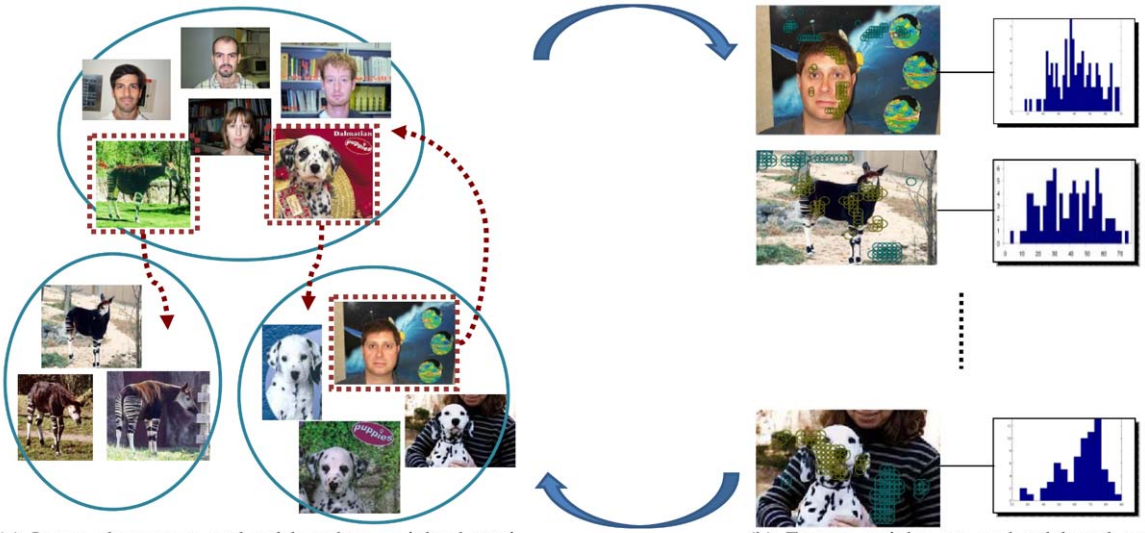
Fig. 1 Summary of the problem. The *horizontal lines* separate clusters. (**a**) When all features in an image are given equal weight and many of them belong to the background, full image matches can result in clusters that are based on similar background appearances. (**b**) If we are given clusters that agree in terms of the object of interest that each intra-cluster image holds, the reoccurring regions will be found on the foreground. (**c**) If we are given images that have their foreground features weighted higher than their background features, we can form clusters that agree on the objects' appearances (i.e., the foreground). The problem we address in this work is how to discover which features are foreground among unlabeled images, which requires simultaneously solving (**b**) and (**c**) above

collection—that is, it must re-occur repeatedly, albeit with some variation in appearance across the instances. Isolating "important" features that are responsible for generating natural image clusters would be useful to construct models to detect discovered objects in novel images, or to generate compact summaries of visual content. Thus the task is essentially unsupervised feature subset selection: to determine which portion of the features present can be used to form high quality clusters under a chosen clustering objective.

How can we learn object categories from unlabeled images? If the images contain both objects of interest as well as background clutter, a simple image clustering will likely give poor results; the matches between background and foreground features will yield noisy inter-image affinities (see Fig. 1(a)). On the other hand, if we were to know which features in each image belonged to the foreground, the clustering task would be greatly simplified—we could cluster images using only the foreground features (see Fig. 1(c)). Similarly, we cannot accurately perform foreground feature selection among a pool of images containing many different objects; however, if we were to know which groups of images contained the same object, then we could select the features they share as the foreground (see Fig. 1(b)). Thus, it is unclear which should be learned first, since one influences the outcome of the other; the selected features will dictate the clusters formed, while the image clustering will influence which features are deemed important. This "chicken-and-egg" type problem means that feature selection and data clustering must be learned *simultaneously*.

We propose a solution to this problem that seeks the mutual support between discovered objects and their defining features. Given a collection of examples, we extract semi-local descriptors throughout each image. To discover object categories, an initial image-level grouping is computed based on the correspondences between any two images' features. To determine the foreground features, we analyze the pattern of the matches within each initial group to determine the extent to which each local part agrees with parts in other images within the current cluster. From this, we compute a weight on each feature representing its significance. The groups and feature weights are then iteratively refined by alternately computing 1) the cluster membership given the re-weighted features, and 2) the feature weights given the newly refined memberships (see Fig. 2). As the common features within a group are emphasized with high weights, the foreground features increasingly become the focus of the clusters found. Due to the reciprocal reinforcement between the consistent matches and cluster assignments, the iterative process yields both a partition of the unlabeled inputs as well as their detected foreground, i.e., the regions for which the grouping is most consistent.

Our main contribution is a new approach to perform unsupervised foreground feature selection from collections of



(a) Image clusters are updated based on weighted semi-local feature matches.

(b) Feature weights are updated based on current clusters.

Fig. 2 Illustration of the proposed method. The images are grouped based on weighted semi-local feature matchings (**a**), and then image-specific feature weights are adjusted based on their contribution in the match relative to all other intra-cluster images (**b**). These two processes are iterated (as denoted by the *block arrows in the center*) to simultaneously determine foreground features while improving cluster quality. As the foreground features on repeated objects receive greater weight, the cluster memberships change, and the groups discovered more accurately reflect the objects present. In this example, the *dotted arrows* between clusters in (**a**) denote that updates to the feature weights cause the dalmatian and face examples to swap group memberships, whereas the okapi leaves the face cluster in favor of the other okapis

unlabeled images. Whereas previous feature selection methods could detect foreground or discriminative features in labeled images, our method discovers them in unlabeled images. Our secondary contribution is a new semi-local region descriptor that provides a flexible encoding of local appearance and geometry. Our results support the notion that unsupervised foreground feature detection aids in grouping similar objects, while important features are better found on objects of interest (foreground) when given partitions of partially re-occurring patterns. We compare our approach with existing unsupervised learning algorithms and show improvements on benchmark datasets.

2 Related Work

In this section we review relevant work in supervised image feature selection, weakly supervised and unsupervised category learning, and semi-local descriptors.

2.1 Supervised Feature Selection

Various recognition methods can learn categories from labeled images with segmented foreground and then detect them within cluttered images. In Leibe et al. (2004), Marszalek and Schmid (2006), the authors show how to weight features matched to a novel test image based on their agreement with known object geometry, thereby downplaying background and better segmenting the object. The paradigm of "weak supervision" suggested in Weber et al. (2000) explored the idea of simultaneous learning of feature selection and data clustering, and has since been pursued by a number of methods (e.g. Winn and Jojic 2005; Chum and Zisserman 2007). In this model, categories are learned from cluttered, unsegmented class-labeled images; one seeks the parts in each image that best fit all examples sharing the same label. The model parameters and feature selection for each image are learned iteratively using the Expectation Maximization (EM) algorithm. Discriminative feature selection strategies have also been explored to detect features that occur frequently in in-class examples but rarely on the background (Quack et al. 2007; Dorko and Schmid 2003). Our approach shares the goal of identifying consistent features in cluttered images, but unlike the above methods it does not employ any labeled examples to do so.

The problem of unsupervised feature selection has received limited attention in the machine learning community (see Dy and Brodley 2004 and references therein), but existing methods presume a vector input space, many assume the data to be generated by certain parametric distributions, and/or are specifically tailored to a particular clustering method—any of which can be ill-suited for the visual learning scenario.

2.2 Weakly Supervised and Unsupervised Category Learning

Recent work in unsupervised category learning has considered ways to discover latent visual themes in images using topic models developed for text, such as probabilistic Latent Semantic Analysis (pLSA) or Latent Dirichlet Allocation (Quelhas et al. 2005; Fei-Fei and Perona 2005; Sivic et al. 2005; Russell et al. 2006; Fergus et al. 2005; Liu and Chen 2007). The main idea is to use feature co-occurrence patterns in images to recover the underlying distributions (topics) that best account for the data. Having discovered the topics, one can express an image based on the mixture of topics it contains. Early models transferred the notion of text documents containing unordered words to images composed of "visual words" (Quelhas et al. 2005; Fei-Fei and Perona 2005; Sivic et al. 2005). Recent extensions show how to incorporate spatial constraints (Fergus et al. 2005; Liu and Chen 2007), or use segmentation to reduce the spatial extent of each "document" (Russell et al. 2006).

Our method also discovers feature co-occurrence patterns in images, however, unlike these methods, we use explicit correspondences between feature sets. These correspondences allow us to select the most distinctive features within some partition of a dataset, and assign a confidence value to each feature reflecting how relevant it is to one of the discovered categories. In contrast, latent topic models produce soft cluster assignments, where an image is explained as a mixture of all the discovered topics. Thus, a feature's confidence is influenced by the visual word distribution of the entire dataset.

Other approaches treat the unsupervised visual category learning task as an image clustering problem, where images are given a hard assignment into one of the discovered groups. In Grauman and Darrell (2006), affinities computed from local feature matches are used with spectral clustering to find object clusters and prototypes, and in Dueck and Frey (2007) a message-passing algorithm propagates non-metric affinities and identifies good exemplars. Our method also begins by computing pairwise affinities between images. In contrast to these techniques, however, the proposed approach allows common feature matches to reinforce and refine the discovered groups; as a result it provides both the groupings as well as the predicted foreground-background separation.

2.3 Semi-Local Descriptors

Local features are a favored representation of images due to their resilience under common transformations, occlusion, and clutter. However, in some cases too much locality can also be problematic: features with minimal spatial extent may be too generic and easily matched, and comparing unordered sets of local patches enforces little geometry. Researchers have therefore proposed "semi-local" feature descriptors that capture information about local neighborhoods surrounding an interest point (Lazebnik et al. 2004; Agarwal and Triggs 2006; Quack et al. 2007; Sivic and Zisserman 2004). The general idea is to build more specific features that reflect some geometry and aggregate nearby features into a single descriptor. Various aggregation strategies have been proposed: in Lazebnik et al. (2004), groups are formed from regions that remain affinely rigid across multiple views of an object, while in Agarwal and Triggs (2006) neighborhoods are collected hierarchically in space, in Quack et al. (2007) a tiled region is centered on each interest point to bin nearby visual words, and in Sivic and Zisserman (2004) the k-nearest points to the base point are included but without any spatial ordering.

These methods aggregate information in a semi-local neighborhood surrounding each interest point, but fail to capture either the neighboring features' spatial configuration, spatial ordering, or spatial count. In order to compute more reliable correspondences between images, we design a new descriptor that counts the co-occurrence of each visual word type relative to an interest point, accumulating the counts at increasingly distant spatial regions and in distinct relative configurations. Our descriptor is inspired by Ling and Soatto (2007), where a kernel is developed to compare correlogram-like distributions of visual words. In Ling and Soatto (2007), each image is described by the distribution of its visual words, whereas our descriptor describes each feature's semi-local neighborhood. Thus, our method is able to localize an object as well as allow for multiple regions to be represented in an image.

This article differs from our previous related conference papers, Grauman and Darrell (2006) and Lee and Grauman (2008a), both in technical aspects and in evaluation. Unlike Grauman and Darrell (2006), the proposed method iteratively refines the discovered groups based on feature matches and vice versa. We offer a complete analysis of how the cluster refinements influence the discovery of foreground features, and in turn, how the discovered patterns influence the refinement of groups. We have also extensively tested our method on a variety of datasets and offer more thorough comparisons to related methods relative to Lee and Grauman (2008a), including an explicit evaluation of our new semi-local descriptor.

3 Approach: Discovering Object Categories and Foreground Features by Mutual Reinforcement

The goal is to predict which regions in unlabeled images correspond to foreground, and in doing so to improve accuracy in unsupervised visual pattern discovery. In this section, we describe how to simultaneously group similar images and

discover the foreground regions using partial matching and feature weight refinements.

The main idea is as follows: Given a set of unlabeled images, our method groups similar examples based on the correspondence between their semi-local features. This yields an initial set of "discovered" categories. To discover the foreground regions within a category, we weight each feature according to its contribution to the match between the image that contains it and every other intra-cluster image. Then, the groupings and weights for the whole image collection are iteratively re-computed: the new feature weights influence each image's similarity to the rest, and the successive groupings that result influence the next set of feature weights. The end result is both a partition of the image collection that represents the discovered categories, as well as weights that reflect the degree to which a feature is believed to be foreground. Since each image is ultimately assigned to one cluster, the method discovers one primary object of interest per image.

In the following, we first describe the grouping process in detail, and then overview our semi-local descriptor.

3.1 Simultaneous Image Grouping and Foreground Detection

Given an unlabeled data set of N images, $U = \{I_1, \ldots, I_N\}$, we represent each image I_i as a set of weighted features, $X_i = \{(f_1, w_1), (f_2, w_2), \ldots, (f_{|X_i|}, w_{|X_i|})\}$, where each $f_j \in \Re^d$ is a local image descriptor weighted with some $w_j \geq 0$, where $w_j \in \Re$. The weight on a feature vector determines its importance within the image, and will affect any matching computed for the set in which it is contained. Initially, all feature weights are set to a uniform value: $w_j = 1$, for all features $j = 1, \ldots, |X_i|$ in all sets $i = 1, \ldots, N$. Subsequently, every time we cluster the images, the support (or lack of support) computed for a feature within a group will result in an increase (or decrease) of its weight. Those weight updates in turn influence the image groups found at the next iteration.

3.1.1 Clustering Weighted Feature Sets

A good clustering should group together images that have a consistent repeated appearance pattern. However, given that the images will likely be cluttered and may contain multiple objects, the pattern need not encompass the entire image. Therefore, we want to compute clusters based on the appearance agreement of some portion of each example—that is, based on a match between subsets of the local features. Further, the weight on a feature should dictate how much attention an image-to-image comparison pays to it, so that features with high weight have more influence on the measured cost of a match, and features with low weight have little effect.

To accomplish such a grouping, we perform spectral clustering with an affinity matrix that reflects the least-cost partial matching between weighted point sets. Also known as the Earth Mover's Distance (EMD) (Rubner et al. 2000), this optimal match cost $M(X, Y)$ reflects how much effort is required to transform weighted point set X into weighted point set Y:

$$M(X, Y) = \frac{\sum_i \sum_j F_{i,j} \;\; D(f_i^{(X)}, f_j^{(Y)})}{\sum_i \sum_j F_{i,j}}, \tag{1}$$

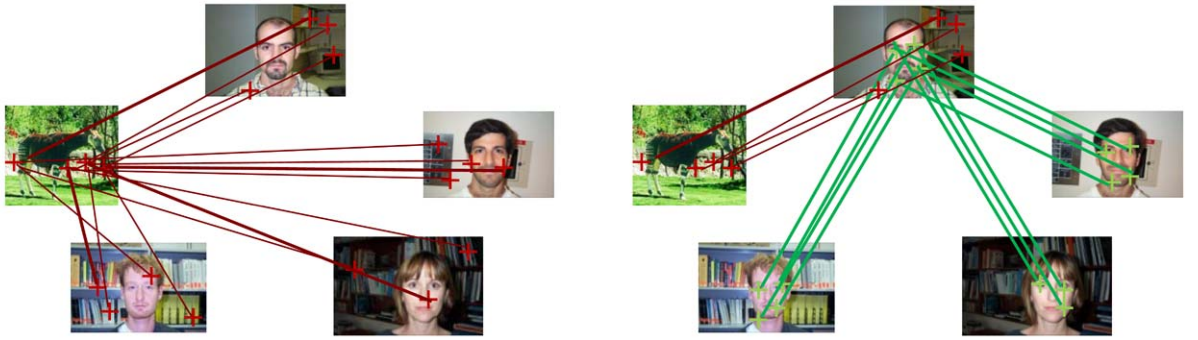
where $f^{(X)}$ and $f^{(Y)}$ denote features from sets X and Y, respectively, and $D(f_i^{(X)}, f_j^{(Y)})$ denotes the distance (typically Euclidean) between points $f_i^{(X)}$ and $f_j^{(Y)}$.

The values $F_{i,j}$ are scalars giving the *flow*, or amount of weight that is mapped from point $f_i^{(X)}$ to point $f_j^{(Y)}$. The flows indicate both the correspondences between features of two matching sets, and the contribution that each feature made to the match. The EMD takes as input the weighted features sets and produces the least cost match and the flows as output. It can be viewed as a solution for computing the minimum amount of work required to shift the mass (weights) of the larger weighted feature set to "fill" the holes (weights) of the smaller weighted feature set. Note that this measure takes into account the distance between matched points as well as the amount of weight (mass or "dirt") attached to each one. The EMD has previously been used in supervised tasks to compare textures and shapes described by local feature distributions (Lazebnik et al. 2003; Grauman and Darrell 2004).

Due to the complexity of computing the optimal matching on weighted point sets, which is super-cubic in the number of features, in practice we approximate the EMD with a variant of the Pyramid Match Kernel (PMK) algorithm (Grauman and Darrell 2005) (see the Appendix for details).

Typically, the weights for point sets given to EMD are used to denote each point's frequency of occurrence. In our case, however, we use the weights to encode priority in the matching: assuming an image's foreground features are relatively highly weighted, a second image cannot produce a low matching cost against it unless it has similar point(s) to the foreground with similar total weight(s). Likewise, a feature with low weight cannot contribute much cost to any match, so its influence is negligible.

At each clustering iteration, we compute affinities using the $N \times N$ matrix $\mathbf{A}$ of matching scores between all pairs of unlabeled images: $\mathbf{A}_{m,n} = \exp(-(M(X_m, X_n))^2/2\sigma^2)$, for $m, n = 1, \ldots, N$. These affinities are input to a spectral clustering algorithm that partitions the N examples into k groups. In our implementation we use the normalized cuts criterion (Shi and Malik 2000), which finds the optimal partitioning of the data by "cutting" the edges (similarity values) between the nodes (images) to form disjoint clusters in



(a) An incorrectly clustered image produces inconsistent matches.

(b) A correctly clustered image produces consistent matches that are on the foreground.

Fig. 3 An example of a heterogeneous cluster due to inconsistent matching between image features. The thickness of the *lines* indicate the strength of the match, i.e. *thicker lines* indicate higher matches. (**a**) The Okapi image has been grouped with the four Face images due to its high pair-wise matching similarity with each Face image. However, an okapi feature that matches highly with one face image does not necessarily match highly with another face image—the highly matching features across images are inconsistent. (**b**) The Face image in the cluster has highly matching features that are consistently on the foreground

which the intra-cluster similarity and the inter-cluster dissimilarity are maximized. The objective criterion is formulated such that the edges between the least similar nodes are removed without favoring a few isolated nodes (outliers). This allows our method to favor a broad range of similar images to be selected as a cluster in place of a few exceptionally well-matching images. We have chosen the normalized cuts criterion due both to its efficiency and the fact that it prefers farther-reaching clusters.

3.1.2 *Refining Foreground Feature Weights from Current Clusters*

Given a k-way partition of the images, we update the weights attached to each feature by leveraging any current regions of agreement among the images in a single partition. Even when all pairs of examples within a cluster have high matching similarity, because each matching can draw from different combinations of features, heterogeneous clusters are possible (see Fig. 3). To overcome this, we look to the pattern of the flows computed by (1). The idea is to use information among the "good" matches (images amongst which all pairs have similar matching points) to re-interpret the "bad" matches (images amongst which similar matching points exist, but are not consistent across all intra-cluster pairs).

The flows computed between two images specify which features best match which, and using what amount of weight. Given a cluster containing C images $\{X_1, \ldots, X_C\}$, for each example X_i, $i = 1, \ldots, C$, we define $(C-1)$ $|X_i|$-dimensional weight vectors denoted $\mathbf{w}_{ij}$, with $j = \{\{1, \ldots, C\} \setminus i\}$. That is, we compute a vector of feature weights for the i-th example against every other image within the cluster. Each of the weight entries in $\mathbf{w}_{ij}$ specifies how much its feature from X_i contributed to the match with set X_j. We define the d-th element as:

$$\mathbf{w}_{ij}(d) = \sum_{p=1}^{|X_j|} \frac{F_{d,p}}{D(f_d^{(X_i)}, f_p^{(X_j)})}, \tag{2}$$

for $d = 1, \ldots, |X_i|$. Each weight is the sum of all the flow amounts from that particular feature in X_i to any other feature in the other set X_j, normalized by the inter-feature distance between the matches (we use L_2). We compute the final weights $\{w_1, \ldots, w_{|X_i|}\}$ as the element-wise median of these $(C-1)$ vectors, normalized to maintain the original total weight. The median is robust in skewed distributions as it is less affected by outliers (unlike the mean). In this regard, the median is appropriate for our method, since our goal is to reward features that have consistently good matches and to prevent giving high weight to a feature that produces a few exceptionally high matches.

The final weights give a robust estimate of how much each feature consistently matched with other features in intra-cluster images. Highly weighted features in an image will indicate that they have good consistent matches throughout the intra-cluster images, while low weighted features will indicate that they have inconsistent matches throughout the intra-cluster images—there may be a few good matches, but the matches are not consistent enough to produce high weights. Specifically, for a feature to obtain a high weight, it must have high matches to features belonging to at least half of the intra-cluster images.

We normalize the final weights to maintain constant total weight per image, such that $\sum_{p=1}^{|X_i|} w_p = |X_i|$. This prevents weights attached to an irrelevant example from wasting away to nothing and getting stuck in their initial cluster.

Algorithm 1 The Foreground Focus algorithm

Input: Set of unlabeled images, $U = \{I_1, \ldots, I_N\}$, and # of clusters k. Each I_i is represented by a set of weighted features, $X_i = \{(f_1, w_1), (f_2, w_2), \ldots, (f_{|X_i|}, w_{|X_i|})\}$

Output: Set of k clusters and updated weights for each feature in each image

while *average % change in feature weights* $\leq$ *threshold* **do**
- **foreach** X_i, $i = 1, \ldots, N$ **do**
  - **foreach** X_j, $j = 1, \ldots, N$ **do**
    - Compute feature correspondences between X_i and X_j;
    - Obtain cost $M(i, j)$ by (1);
    - Convert cost $M(i, j)$ to affinity: $\mathbf{A}_{i,j} = \exp(-(M(i, j))^2/2\sigma^2)$;
    - Obtain $|X_i| \times |X_j|$ *flow* matrix, $\mathbf{F}_{X_i, X_j}$;
  - **end**
- **end**
- Perform *NCuts* on $\mathbf{A}$ to form k clusters;
- **foreach** *Cluster* C_l, $l = 1, \ldots, k$ **do**
  - **foreach** $X_i \in C_l$, $i = 1, \ldots, |C_l|$ **do**
    - **foreach** *weighted feature* $f_d^{(X_i)} \in X_i$, $d = 1, \ldots, |X_i|$ **do**
      - **foreach** $X_j \in C_l$, $j = 1, \ldots, |C_l|$ **do**
        - Compute feature contribution to match, $\mathbf{w}_{i,j}(d)$ by (2)
      - **end**
      - Update weight w_d of weighted feature $f_d^{(X_i)}$ by $w_d = median(\mathbf{w}_{i,1}(d), \ldots, \mathbf{w}_{i,|C_l|}(d))$
    - **end**
  - **end**
- **end**

end

Before normalization, an image that produces inconsistent matches to its intra-cluster images will end up with mostly low feature weights, enough so to prevent it from producing meaningful image similarities in the next iteration (since the match costs are influenced by both the distances as well as the weights of the matching features). By normalizing the final weights to maintain constant total weight, the image has a chance to be assigned to the correct cluster.

In general, if a clump of images in a cluster contains instances of the same category, high weights will be attributed to their consistently re-occurring parts—the foreground. To begin the next iteration, we re-compute the flows and affinities between all pairs of all N examples using the new weights, and re-cluster. As the weight distributions shift, subsequent least-cost matches are biased towards matching those features more likely to be foreground. We iterate between the matching, clustering, and re-weighting, until there is no change in the cluster assignments or until the average percent change in weight is below a threshold.

Essentially, our method updates the weights on the features of each image to produce tighter clusters in the next iteration. This is possible because our feature weight updates guarantee that the matching cost between two images decreases when compared to the matching cost obtained prior to the weight updates. Our algorithm then chooses the weights (by the median) for each image such that the overall matching cost between the intra-cluster images decreases. Therefore, subsequent iterations produce tighter clusters, and will be required to focus the matchings on those features that already have support across multiple images. In practice, we observe the most impact from the first several iterations (see Sect. 7).

Algorithm 1 gives a step-by-step summary of the Foreground Focus method.

4 Semi-Local Proximity Distribution Descriptors

Our algorithm description thus far implies an orderless set-of-features representation. We propose a novel *semi-local* region descriptor that encodes the appearance and relative locations of other features in a spatial neighborhood. Our descriptor is inspired by the proximity distribution kernel (Ling and Soatto 2007), which compares images described by cumulative histograms of nearby visual word pairs. However, while their approach summarizes an entire image with one histogram, we design a proximity distribution feature for each interest point, which makes it possi-

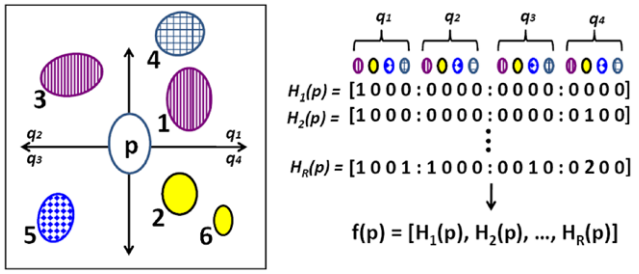


Fig. 4 (Color online) Schematic of the proposed semi-local descriptor. The base feature is p. The ellipses denote the features, their patterns indicate their corresponding visual word types, the *numbers* indicate their rank order of spatial proximity to the base feature, and the q_i's denote the four quadrants (directions) relative to p. Here the nearest neighboring feature to p is the feature in q_1 corresponding to the vertically textured word type, and so its bin in H_1 is incremented. For H_2, the bins corresponding to the word types of the two nearest neighboring features to p are incremented—the vertically textured word type feature in q_1 and the clear word type feature in q_4. The process is repeated until all R spatially nearest features to p are observed. In this example, $R = 6$

ble to use rich local configuration cues within an explicit, weighted matching (and thus calculate the flow as described above). As mentioned in Sect. 2, previous methods that encode the semi-local neighborhood information fail to either capture the spatial configuration, spatial ordering, or spatial count of the features in the semi-local neighborhood. Our descriptor is built with the motivation to capture all of the above information, and we show in our experiments that it can lead to better object localization and classification performance.

We extract local patch features at all interest points. Then we construct a standard n-word visual vocabulary by clustering a random pool of descriptors (we use SIFT (Lowe 2004)) extracted from the unlabeled image dataset, U, and record each feature's word type. We use the k-means algorithm for clustering. For each patch in an image, for each of four directions (quadrants) relative to its center, we compute a cumulative distribution that counts the number of each type of visual word that occurs within that feature's r spatially nearest neighbor features, incremented over increasing values of r (see Fig. 4).

More precisely, consider an image with patches $\{p_1, \ldots, p_m\}$ and their associated word types $\{v_1, \ldots, v_m\}$. For each p_i, we construct R total $4n$-dimensional histogram vectors $H_r(p_i)$, for $r = 1, \ldots, R$. In each, the first n bins represent quadrant 1, the next n bins represent quadrant 2, and so on. Each n-length chunk is a histogram counting the number of occurrences of each word type v_j within p_i's r spatially nearest feature points, divided into quadrants relative to p_i. Note that higher values of r produce a vector $H_r(p_i)$ covering a spatially larger region. Finally, our semi-local descriptor for p_i is the concatenation of these R histograms: $f(p_i) = [H_1(p_i), \ldots, H_R(p_i)]$.

Every patch's $R \times 4n$-length vector is a translation-invariant encoding of neighborhood appearance and coarse geometry. (We can add rotation invariance by setting quadrants based on a feature's dominant gradient; we have not yet explored this variant.) Due to the high-dimensionality and correlation among dimensions, we compute compact descriptors using Principal Components Analysis (PCA). Matching sets of our descriptors does not explicitly enforce spatially contiguous regions to be discovered. However, due to their spatial extent and overlap, individual point matches are in fact dependent.

Recent work on sampling strategies shows that the single most important criterion for recognition performance tends to be the number of patches detected in each image (Nowak et al. 2006). Hence, dense sampling is shown to often yield better recognition accuracy than interest-point detectors, because it provides more coverage of the image.

The region that our semi-local descriptor encapsulates can be quite different depending on the sampling method for the base features. Dense sampling on uniform grid points will produce semi-local regions that are consistent in area (in terms of image coordinates) for each feature (except those that lie near edges or corners) independent of the image it belongs to, which could assist in better matching if the objects occupy similarly-sized regions across images. However, if the foreground objects in different images do not have the same scale, the foreground coverage of the semi-local descriptor in each image will be different—even if the base feature covers the same part of the object in the images. To make the descriptor scale invariant for densely sampled features, multi-scale sampling can be used where the sampling points are adjusted with respect to the size of the patches, i.e., finer sampling for smaller sized patches. When computing feature correspondences between two images, matching can proceed between all features at all scales. This way, matches are made between descriptors that cover the same regions of the objects, even between images that have foreground objects of different scale.

Sparse sampling with interest point detectors will produce scale invariant semi-local descriptors that are independent of the spatial area in terms of image coordinates. An exception can occur for images containing objects at very different resolutions, since different sparse points will be detected (i.e., a high resolution view of an object may result in more detections with interest point detectors than a low resolution view of the same object). In this case, scale invariance can be approximated by using scale-invariant feature detectors and considering only similarly sized features in the base feature's neighborhood. This way, the semi-local neighborhoods in different images would capture similarly sized regions with respect to the scale of the objects, *independent* of the image resolution. The tradeoff with dense sampling is that sparse sampling produces less coverage of the image,

and not all parts of the foreground object may be captured by the descriptor. In our experiments in Sect. 7, we analyze the practical tradeoffs between building our descriptor with densely or sparsely sampled points.

Related methods for encoding the appearance of semi-local neighborhoods have been previously proposed. The authors of Quack et al. (2007) employ a neighborhood-based image description of visual words in which the scale of the neighborhood is determined by the size of the region of interest (detected feature). Multiple instances of the same visual word are not counted, and the neighborhood descriptions are not used explicitly for matching. Instead, the object category is determined by the *set* of words in a region using data mining tools. Similarly, in Sivic and Zisserman (2004) the neighborhood of each region of interest is represented by encoding the set of the R spatially nearest words to a base feature as a n-d vector, where n is the size of the vocabulary. The authors of Agarwal and Triggs (2006) construct hyperfeatures—descriptors collected hierarchically in increasing neighborhoods of the image space. In contrast to these methods, our descriptor considers the order of spatial proximity as well as the spatial direction in which the neighboring features are located with respect to the patch center. This is a richer description of the semi-local neighborhood of a feature; in order for two descriptors to have a high match, having similar features in their semi-local neighborhoods is not enough—the neighboring features must also have similar geometric configurations.

The authors of Lazebnik et al. (2006) propose to represent an image as a spatial pyramid. An image is repeatedly subdivided and histograms of features are computed over the sub-regions, thereby capturing both the appearance information as well as the spatial layout information of the features. However, their representation is global—the partitioning of the regions is based on the image coordinates, and the image as a whole is represented. In contrast, our semi-local descriptor captures information specific to each feature's neighborhood. This allows our descriptor to be used for object localization and grants more robustness to clutter and occlusion for image classification tasks.

5 Computational Complexity of the Algorithm

In this section we analyze the computational complexity of our Foreground Focus method and the memory requirements for computing and storing our semi-local proximity distribution descriptors.

Let N be the number of feature sets in the dataset and T be the number of features in each feature set (without loss of generality, assume all feature sets have the same number of features). Let L be the number of levels used to construct the pyramid tree for partitioning the feature space for the modified PMK algorithm that approximates the EMD (see the Appendix).

If EMD is modeled as a network flow problem, it can be computed in $O(T^3 \log(T))$ time (Rubner et al. 2000). This is the worst-case complexity for our algorithm to compute the least cost distance and flows between two feature sets, which occurs if all the points in feature space fall in the same node of the pyramid tree. In practice, most of the features are spread out across the nodes since the tree is constructed by directly sampling features from the feature sets of the dataset. The best-case complexity is $O(LT)$, which occurs if each non-empty node in the tree is occupied by a single feature from each feature set. Empirically, the typical runtime per approximate EMD and flow computation is about quadratic in T.

Our method's clustering stage forms k groups from an $N \times N$ affinity matrix using the normalized cuts algorithm (Shi and Malik 2000). The computational complexity at each iteration is $O(N^3)$, which is the time required for eigen-decomposition of the matrix.[1] The memory requirement for storing the matrix is $O(N^2)$.

When computing our proximity distribution descriptors, we use PCA to reduce their dimensionality. To compute the subspace bases efficiently, we sample the features from the dataset (typically about 5% to 10% of the total number of features). For V sampled features, this requires storage of V vectors of length $R * 4 * n$, where R is the neighborhood parameter and n is the vocabulary size.

Once the subspace bases are computed from PCA on the covariance matrix formed from these V vectors, each feature in the dataset is projected down to have a dimensionality that is comparable to standard descriptors, such as SIFT which has 128-dimensions. In our experiments, we project down to 100 and 130 dimensional descriptors.

6 Discussion and Assumptions

What are the assumptions of our approach? For a pattern to be discovered, it must have support among multiple examples in the collection. Further, only visual patterns that share some configuration of similar semi-local regions can ever be found (e.g., using standard gradient-based region descriptors, our method will not discover a single cluster consisting of both soccer balls and volleyballs, but it can discover a group comprised of different people's faces). Finally, *some* support for a pattern must be detected in the initial iteration for progress towards refining that pattern to be made in the remaining iterations.

[1] To further improve efficiency for larger datasets, we could employ the equivalent kernel k-means formulation developed in Dhillon et al. (2004), which also minimizes the normalized cut but does not require eigen-decomposition.

Note that features that are strictly speaking "background" can also earn high weights, if they happen to consistently re-occur with the same foreground class. So, what is learned depends on what the collection U contains: for example, if bikes are typically against a bike rack, then we can expect the pattern to be found as a single entity. The same holds for images with multiple objects that repeatedly co-occur—for example, if computer monitors always exist on desks. This is a natural outcome for unsupervised learning from static images (e.g., nothing can indicate that the bike and rack are not one composite object unless they often occur separately), and satisfies the problem definition.

This also means that the discovered patterns will not always correspond to the foreground objects, i.e., the dataset will not necessarily be partitioned in concurrence with object class labels. This is because the feature weight updates depend strictly on the intra-cluster matches. For two objects that typically occur in the same setting, e.g., cows and sheep, our method may find the co-occurring visual pattern to be part of the background, e.g., grass. The dataset will be partitioned accordingly, in which case we may not end up with a cow-cluster and/or sheep-cluster. This is still a perfectly reasonable outcome, since our method will have found the most consistently co-occurring visual patterns.

In our current implementation, we leave the method completely unsupervised. However, semi-supervision can be added to guide the algorithm to learn objects under a certain criterion. For the cows and sheep example, we could take a few images from each category, and remove all features on the grass (the background) by setting their weights to 0. This way, our algorithm would be biased towards finding the re-occurring patterns that fall on the foreground. We could also enforce high (low) affinity in the kernel matrix between some examples that are constrained to be similar (dissimilar). This would be especially helpful for examples that are often misclassified due to high background clutter.

7 Results: Evaluation of the Foreground Focus Method and Proximity Distribution Descriptor

In this section we present experiments both to analyze the mutual reinforcement of foreground and clusters, and to compare against existing unsupervised methods. We work with images from the Caltech-101 (Fei-Fei et al. 2004) and Microsoft Research Cambridge v1 (MSRC-v1) (Winn et al. 2005) datasets. We chose these datasets both because they provide object segmentations that we need as ground truth to evaluate our foreground detection, and because previous related unsupervised techniques were tested with this data. We also evaluate our semi-local descriptor's foreground discovery on the Caltech Cars Rear, TUD Motorbikes, and GRAZ Bikes datasets. Unless otherwise specified below, we sample SIFT features at regular image intervals.

7.1 Implementation Details

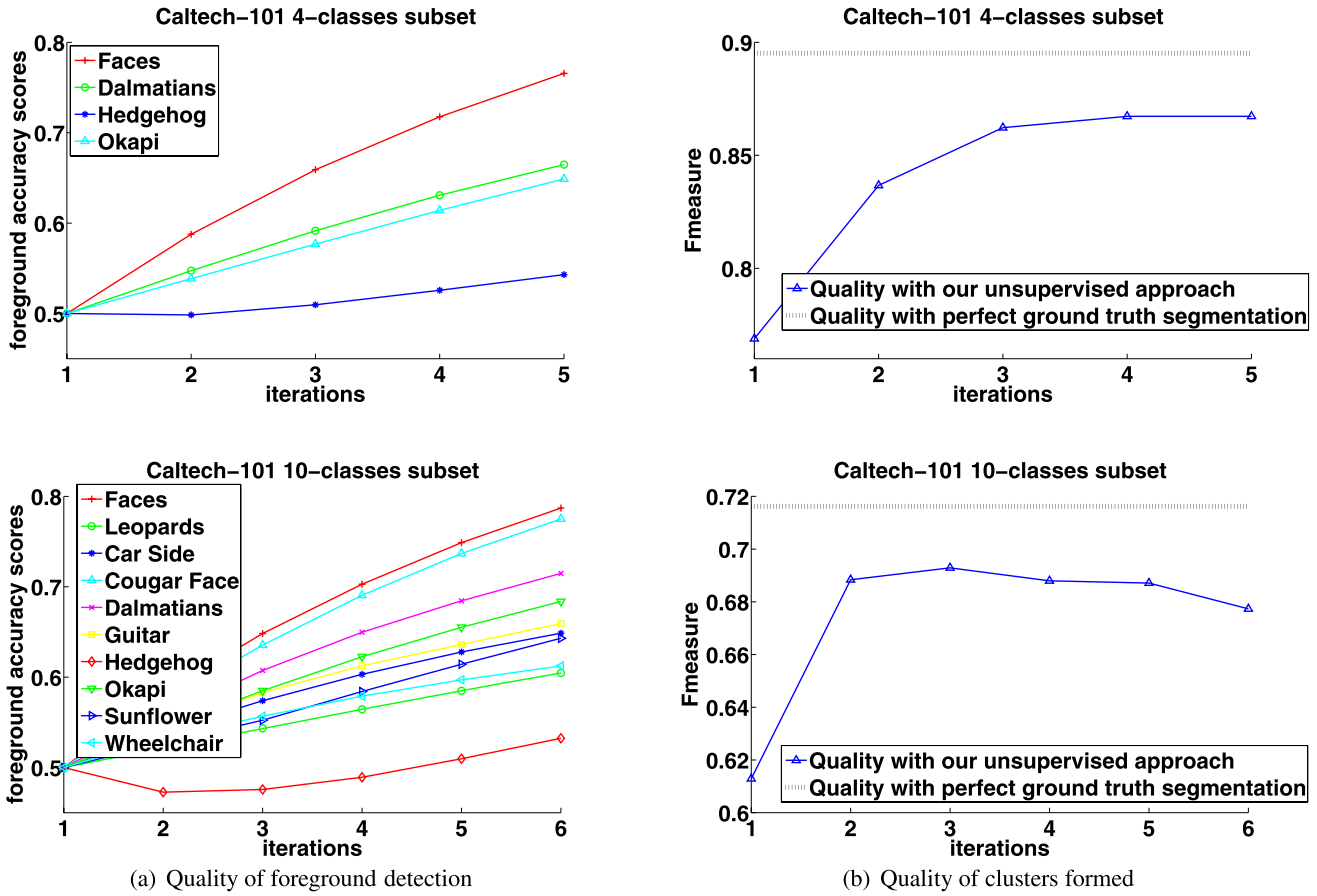
To determine when to stop iterating, we measure the percent change in the average feature weight change in all images from one iteration to the next, and stop once it slows to 15% or less (a threshold we set arbitrarily). When clustering, we set k as the number of classes present in the dataset in order to evaluate how well the true objects are discovered. Note that k can be set higher to allow sub-categories, e.g., rear-view, side-view of the car category, to be discovered. The number of clusters can be automatically determined by the self-tuning spectral clustering method (Zelnik-Manor and Perona 2004), which was demonstrated in Lee and Grauman (2008b) to find different aspects/views of tourist attractions. In practice, we have found that setting the value of k to be equal to the number of categories produces the best clusters (consistent with the images' class labels).

We fix the neighborhood parameter at $R = 64$, following (Ling and Soatto 2007), which means that each descriptor covers about $\frac{1}{4}$th to $\frac{1}{5}$th of the image in width and height. The vocabulary size n as well as the final dimensionality d (corresponding to the eigenvectors with the d largest eigenvalues after PCA) of the spatial descriptors are set roughly depending on the number of input images in an attempt to get good coverage; however, they are not optimized for each dataset. For a dataset that has many object categories (each having distinct appearances and shape), we expect to need a larger vocabulary to capture the variability of the data. The values we use are in line with typical choices for similarly sized datasets (Ling and Soatto 2007; Fergus et al. 2005; Lazebnik et al. 2006). The specific values for n and d are given below in the appropriate sections.

7.2 Analyzing the Effects of Mutual Foreground/Clustering Reinforcement

7.2.1 Caltech-101 Images

While some classes in the Caltech-101 are fairly clutter-free, we purposely select categories with the highest clutter in order to demonstrate our method's impact. To do this, we first built *supervised* classifiers on all 101 categories: one trained with all image features, and one trained using only foreground features. Then we ranked the classes for which segmentation most helped the supervised classifier, since these directed us to the classes with the most variable and confusing backgrounds. In this way, we formed a four-class (Faces, Dalmatians, Hedgehogs, and Okapi) and 10-class (previous four plus Leopards, Car_Side, Cougar_Face, Guitar, Sunflower, and Wheelchair) set. For each class, we use the first 50 images. Figures 6 and 7 show example images of the two sets. We set n and d to 200 and 100, respectively, for



(a) Quality of foreground detection

(b) Quality of clusters formed

Fig. 5 Evaluation of feature selection and category discovery on the Caltech dataset. (**a**) The average foreground scores over iterations for all images from the 4-class (*top*) and 10-class (*bottom*) sets from the Caltech-101. (**b**) The cluster quality for those sets. The *black dotted lines* indicate the best possible quality that could be obtained if the ground truth segmentation were known (see text)

the four-class set, and 400 and 130, respectively, for the 10-class set. For this experiment, we discard any contrast-free regions.

If our algorithm correctly identifies the important features, we expect those features to lie on the foreground objects, since that is what primarily re-occurs in these datasets. To evaluate this, we compare the feature weights computed by our method with the ground truth list of foreground features. We quantify accuracy by the percentage of total feature weight in an image that our method attributes to true foreground features. To make values comparable across images and classes, we compute $\frac{fg}{fg+bg}$, where fg and bg denote the sums of all foreground (background) weights normalized by the number of all foreground (background) features, respectively. If all weights were on foreground, the score would be 1, while if all weights were on background, the score would be 0. If each feature's weight is set uniformly to 1, then the score would be 0.5 since fg and bg would both be equal to 1 (regardless of the actual number of features that are on the foreground or background).

As discussed above, our method gives the highest weights to the most commonly reoccurring features throughout the intra-cluster images. Therefore it is possible for so-called "background" features to also be weighted highly, for example, when the background consists of repeated contextual features (e.g., street features often appear with car features). However, since we purposely choose the Caltech categories which have the highest clutter *and* show the most improvement in classification accuracy when using only foreground features, the $\frac{fg}{fg+bg}$ evaluation score is appropriate. In the case that contextual background features do get high weights, this metric can only underestimate the accuracy of our method.

Figure 5(a) evaluates our method's unsupervised foreground selection for the two datasets across iterations. All features start with uniform weights, which yields a base score of 0.5. Then each image's weights continually shift to the foreground, with significant gains for most classes as the clusters continue to be refined. In the 10-class set, the Hedgehog class improves more slowly. Upon examination, we found that this was due to many hedgehog images dis-

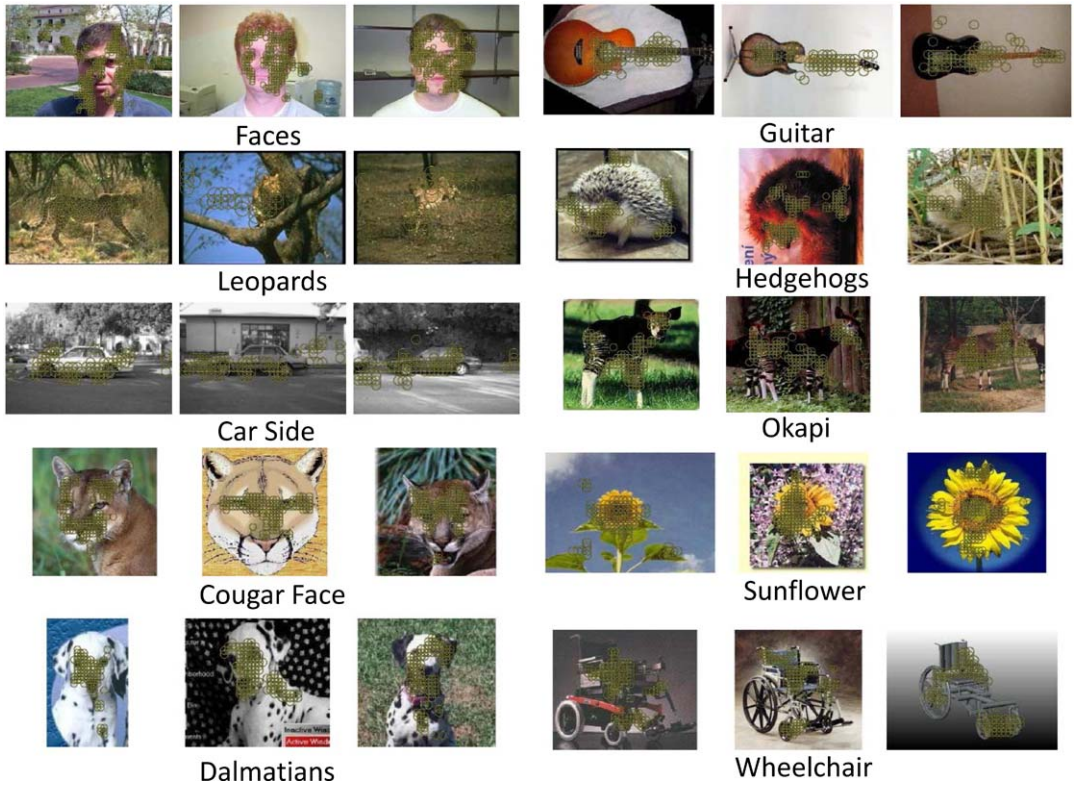


Fig. 6 (Color online) Examples showing the highest weighted features per image. In these examples, our method attributes weight almost only to foreground features. Note that we show the base features to our semi-local descriptors

persed across the initial clusters, resulting in more gradual convergence and cluster swaps.

As our method weights foreground features more highly, we also expect a positive effect on cluster quality. Since we know the true labels of each image, we can use the F-measure to measure cluster homogeneity. The F-measure measures the degree to which each cluster contains only and all objects of a particular class: $F = \sum_i \frac{N_i}{N} \max_j F'(i, j)$, where $F'(i, j) = \frac{2\times\mathcal{R}(i,j)\times\mathcal{P}(i,j)}{\mathcal{R}(i,j)+\mathcal{P}(i,j)}$, and $\mathcal{P}$ and $\mathcal{R}$ denote precision and recall, respectively, and i indexes the classes and j indexes the clusters. High values indicate better quality. Figure 5(b) shows the impact of foreground detection on cluster quality. To provide an upper bound on what quality level would result if we were to have *perfect* foreground segmentation, we also evaluate clusters obtained using *only* the foreground features (black dotted lines). Note that without any supervision or foreground/background annotation, our approach clusters almost as well as the ideal upper bound. Also, as we iterate, the better foreground weights incrementally improve the clusters, until quality levels out.

Figure 6 illustrates example results in which our method finds good support on the foreground. These examples have the highest foreground scores in each category and are always associated with the correct cluster, e.g., the guitar image belonging to a cluster almost entirely comprised of guitars. Considering the fact that we use densely sampled features, a great deal of irrelevant features have been discarded (i.e., assigned low weight) by our method. Figure 7 illustrates example results where our method weights foreground features highly, but also mistakenly finds good support for some background. These examples have the lowest foreground scores in each category and are almost always associated with the incorrect cluster, e.g., the guitar image belonging to a cluster almost entirely comprised of leopards. These results confirm what is expected, since the majority-class images in a cluster will have the highest weighted features on the foreground, while the outlier images will have the highest weighted features on regions other than (or possibly in addition to) the foreground.

Note that our algorithm finds *meaningful* features—by definition, re-occurring visual patterns in the cluster images. Therefore, this does not imply that *all* foreground features are meaningful. Only those that re-occur across images in a cluster are meaningful. Furthermore, even some background features may be meaningful, e.g., features that capture parts of the street in car images, since cars are commonly found on the street. (If this were a supervised feature selection task, it may even be favorable to include background features if we knew that those features were part of regions that were visually re-occurring patterns across the foreground images.)

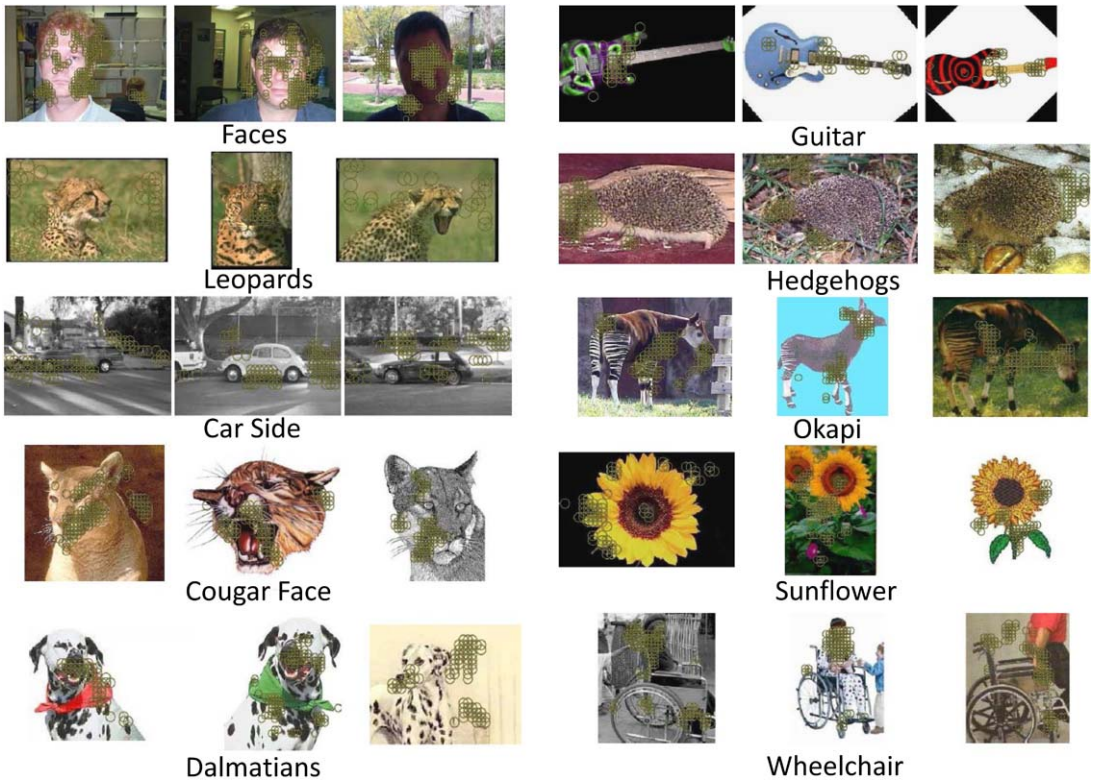


Fig. 7 (Color online) Examples our method does most poorly on: it weights foreground features highly, but also (mistakenly) finds good support for some background. Note that we show the base features to our semi-local descriptors

Our assumption for this experiment is that the backgrounds are uncorrelated, and therefore the foreground features are the *only* meaningful features. This property does not always hold in real images, but we can still expect to get better clusters with our method as long as the re-occurring patterns are weighted highly.

Figure 8 shows an example of the refinement of clusters and weights over iterations on the four-class set. The highest weighted features are shown in yellow, and are features that have weight greater than 1.75—a high value given the distribution of weights in the data. Each iteration produces improvements in both cluster quality and discovery of foreground features. At each iteration, intra-cluster images that have similar visual patterns produce highly weighted foreground features. On the other hand, "outlier" images in each cluster have weights distributed fairly evenly across foreground and background features—the highly matching features for an outlier image are inconsistent over pairwise matches to intra-cluster matches and will not be reflected since the median values are taken among all intra-cluster matching features' weights. The combination of highly weighted foreground features on the similar images, and (approximately) evenly distributed weights on the outlier images produces better quality clusters in the following iteration.

7.2.2 MSRC-v1 Images

We also evaluate our method's unsupervised foreground discovery and category learning on the MSRC-v1 dataset. The dataset is comprised of 240 images belonging to 9 object classes, and has more clutter and variability in the objects' appearances than the Caltech-101 dataset. The object categories are Horse, Sheep, Tree, Building, Airplane, Cow, Face, Car, Bicycle. The dataset creators state that there are not enough training regions to learn reasonable models of horses and sheep—we remove the first 30 images of the dataset which correspond largely to these classes. Therefore, our revised dataset consists of seven classes with 30 images each. Examples of images in this dataset are shown in Fig. 9. For this experiment, we set n to 400 and d to 130.

Figure 10(a) evaluates our method's unsupervised foreground selection for the dataset across iterations. Again, improvements in foreground discovery over iterations is evident. We also evaluate overall cluster quality using the F-measure, which is shown in Fig. 10(b). The black dotted lines indicate the upper bound on the quality level which is found by evaluating clusters obtained using only foreground features.

As our method weights foreground features more highly, we see improvement in cluster quality. However, the increase is not as significant compared to that seen on the

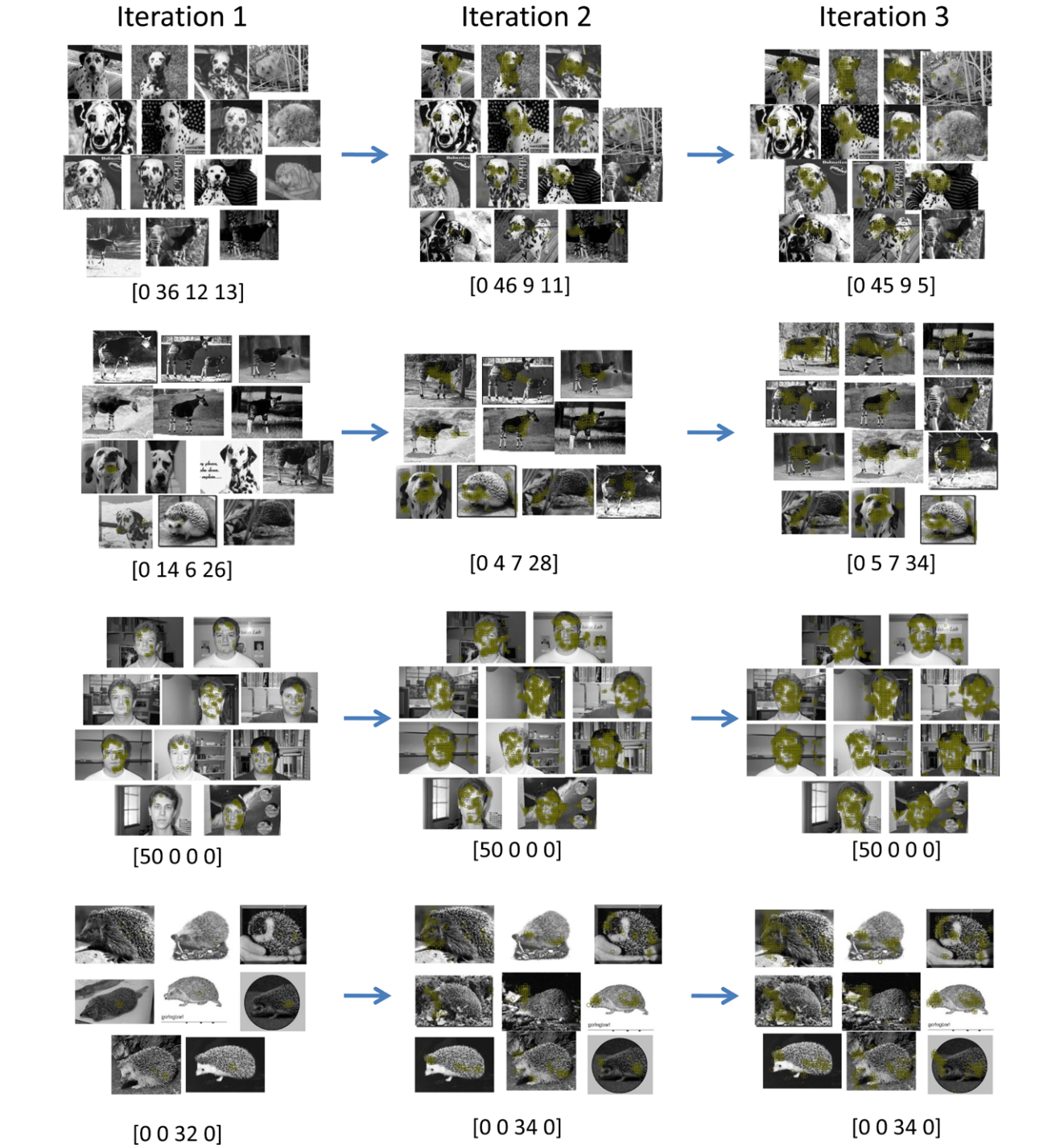


Fig. 8 (Color online) Example cluster and weight refinement on the 4-class set. The highest weighted features are shown in *yellow*. The actual number of images in a cluster for [faces, dalmatians, hedgehog, okapi] is shown below. Images displayed are sampled proportionally to the actual number of images per class in each cluster. Note that as the cluster quality improves, our method weights features on the foreground more highly

Caltech-101 dataset. Categories which have slower improvement, e.g., Tree, are those that are more likely to be confused with background, e.g., grass. This is because the MSRC-v1 has a lot of correlated background—many objects are situated on grass—and therefore the meaningful features found are often on the background. This resulted in some of the initial clusters being grouped based on the background features and improvement on those clusters (in terms of class

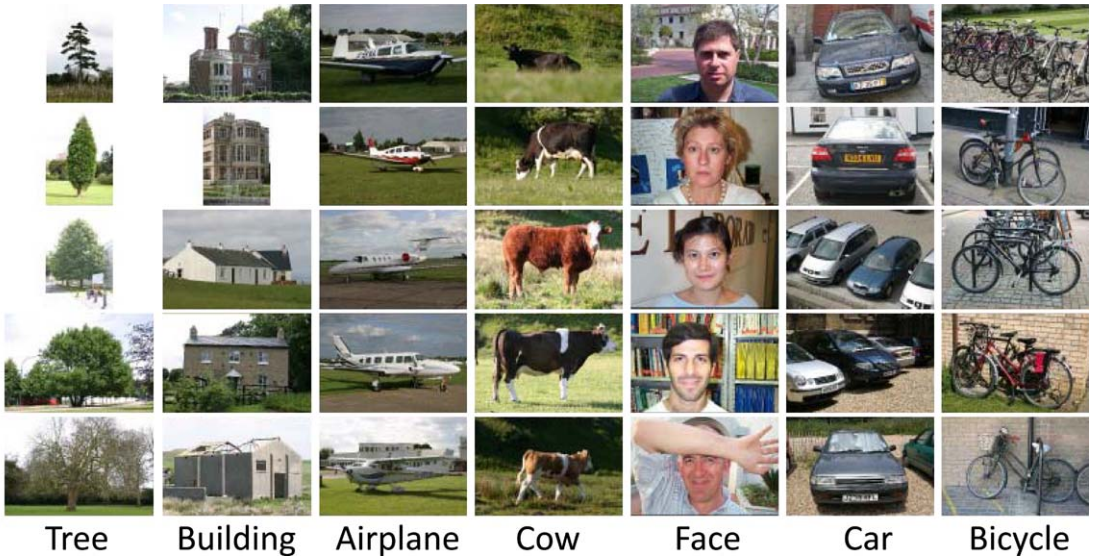
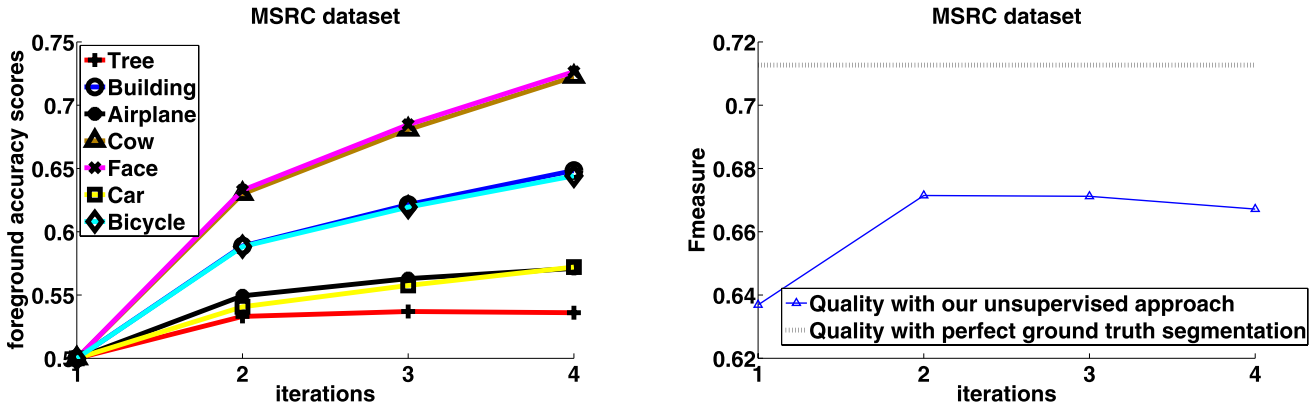


Fig. 9 Examples of images belonging to the MSRC-v1 dataset



(a) Quality of foreground detection

(b) Quality of clusters formed

Fig. 10 Evaluation of feature selection and category discovery on the MSRC dataset. (**a**) The average foreground scores over iterations for all images from the seven classes of the MSRC dataset. (**b**) The cluster quality for those sets. The *black dotted lines* indicate the best possible quality that could be obtained if the ground truth segmentation were known (see text)

labels) could not be made over iterations. (If we were to label the images in which the background features were given the highest weight with "grass" or "sky" (instead of cow, airplane, etc.), then we would see an improvement in cluster quality since the background is the most consistently re-occurring visual pattern in those images.) In order to tune our algorithm to a particular user's goals, we could add semi-supervision by removing the background features on some images to bias our algorithm to capture the foregrounds, or adjusting the kernel matrix based on some paired constraints to enforce grouping or separation of some images. We would like to explore these areas in future work.

The results in this section on the Caltech and MSRC datasets show that our method's mutual reinforcement of unsupervised foreground discovery and category learning can in practice benefit both tasks.

7.3 Comparison with Existing Unsupervised Methods

Next we empirically compare our approach against published results from alternative unsupervised visual learning methods (Dueck and Frey 2007; Grauman and Darrell 2006; Liu and Chen 2006, 2007).

The authors of Dueck and Frey (2007) propose a clustering algorithm called affinity propagation, where messages between data points are exchanged to find a good partition. The method considers all data points as candidate exemplars and iteratively finds the best set of exemplars that partitions the data. They chose two subsets of the Caltech-101: a 20-class subset composed of:

Table 1 Comparison with affinity propagation (Dueck and Frey 2007) for the seven-class and 20-class subsets of the Caltech 101 dataset in terms of the purity cluster quality measure. We test our method (abbreviated here as "FF" for Foreground Focus) with three different features: 1) FF-Dense, in which our semi-local descriptor uses densely sampled SIFT descriptors as base features, 2) FF-Sparse, in which our semi-local descriptor uses the SIFT descriptors detected with DoG as base features (the same features as in Dueck and Frey 2007), and 3) FF-SIFT, in which the DoG SIFT features are used without our semi-local descriptor. Overall, our method performs much better than affinity propagation

	Dueck and Frey 2007	FF-Dense	FF-Sparse	FF-SIFT
Purity (7-class) (%)	59.41	78.91	77.51	70.75
Purity (20-class) (%)	36.91	65.61	41.79	38.94

Table 2 Comparison with Grauman and Darrell (2006) for unsupervised category learning and recognition performance on novel images for the Caltech-4 dataset. Unsupervised category learning is measured in terms of overall cluster purity, and recognition on unseen images is measured in terms of the mean diagonal of the confusion matrix. Results are mean values with standard deviations, averaged over 10 runs with randomly selected training/testing pools. We test our method (abbreviated here as "FF" for Foreground Focus) with two different features: 1) FF-Dense, in which our semi-local descriptor uses densely sampled SIFT descriptors as base features, and 2) FF-Sparse, in which our semi-local descriptor uses the SIFT descriptors detected with the shape-adapted and maximally stable region detectors as base features. Our method performs better than Grauman and Darrell (2006) for both base feature types. Note the robustness of our method shown by the small standard deviations

	Grauman and Darrell 2006	FF-Dense	FF-Sparse
Purity (%)	85.00 ± 4.72	88.82 ± 0.86	91.10 ± 1.10
Prediction rate (%)	84.10 ± 5.07	87.13 ± 0.37	92.29 ± 1.07

Faces, Leopards, Motorbikes, Binocular, Brain, Camera, Car_Side, Dollar_Bill, Ferry, Garfield, Hedgehog, Pagoda, Rhino, Snoopy, Stapler, Stop_Sign, Water_Lilly, Windsor_Chair, Wrench, Yin_Yang, and a seven-class subset composed of: Faces, Motorbikes, Dollar_Bill, Garfield, Snoopy, Stop_Sign, Windsor_Chair. The first 100 images are taken from each class, and n and d are set to 200 and 100, respectively, for both subsets.

In Table 1 we compare our method with the same data, using the "purity" cluster quality measure used in Dueck and Frey (2007). Purity measures the extent to which a cluster contains images of a single dominant class, Purity $= \sum_j \frac{N_j}{N} \max_i \mathcal{P}(i, j)$, where i indexes the classes and j indexes the clusters, and again $\mathcal{P}$ is precision. We first produce results using our algorithm with two base feature types for our semi-local descriptor: 1) densely sampled SIFT descriptors, and 2) SIFT descriptors detected using Lowe's Difference of Gaussians (DoG) scale space selection (the same setting as in Dueck and Frey 2007). A strength of the affinity propagation method is that non-metric affinities are allowed, and so the authors compare images with SIFT features and a voting-based match, which is insensitive to clutter (Lowe 2004). Still, the clusters found by our method are significantly more accurate, indicating the strength of both our refinement process and semi-local descriptor. Our method using dense base features performs much better than when using sparse (DoG) base features. Since most of the objects in the seven-class and 20-class subsets are of similar size, dense sampling produces consistently sized regions described by our semi-local descriptor throughout the images. More importantly, dense sampling provides better coverage of the objects (foreground) than sparse sampling.

We also measure purity for the two subsets using *only* the DoG SIFT features (without the semi-local descriptor). This is to observe the gain that our semi-local descriptor provides over the local base descriptors. Results are shown in Table 1, fourth column. The clusters found using only the DoG SIFT features still produce higher accuracy than that obtained by Dueck and Frey (2007), but lower accuracy than when the semi-local descriptors are built on top of the base features. On the seven-class set, the visual patterns captured in the first iteration corresponded mostly to the foreground objects such that the clusters were of high quality. This improved the foreground detection and cluster quality over iterations. However, on the 20-class set, the cluster quality after the first iteration was not as good—the visual patterns found did not entirely correspond to the foreground. Since the features did not capture any spatial and/or geometrical information, many were erroneously matched, i.e., foreground feature to background feature. This produced weak clusters such that less improvement could be made over iterations. These results show the value of the proposed semi-local descriptor, since they confirm that capturing both appearance as well as semi-local structure improves matching quality.

In Table 2 we compare against the method of Grauman and Darrell (2006), which also forms groups with partial-match spectral clustering, but does not attempt to mutually improve foreground feature weights and clusters

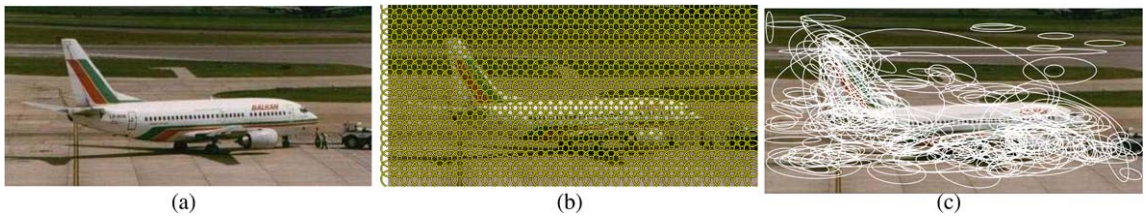


Fig. 11 (**a**) An airplane image. (**b**) With dense sampling, 720 features are detected, of which 119 belong on the foreground object (16.53%). (**c**) With sparse sampling using interest point detectors, 253 features are detected, of which 125 belong on the foreground object (49.41%)

as our method does. We use two feature types for base features to our semi-local descriptor: 1) densely sampled 128-dimensional SIFT descriptors (denoted as FF-Dense), and 2) 72-dimensional SIFT features detected with shape-adapted and maximally stable region detectors (denoted as FF-Sparse). Base feature type one will produce many more semi-local descriptors on average per image, than base feature type two. It also allows the semi-local descriptors to cover similarly sized regions across images, which could be useful if the foreground object is consistent in size across images. Base feature type two is more sparse—there is less coverage of foreground, but also of background which could potentially eliminate spurious matches—yet distinctive.

We perform the same unsupervised category learning and classification experiments as prescribed in Grauman and Darrell (2006). In the object category learning experiment, four categories are learned from the Caltech-4 database comprised of 1155 rear views of cars, 800 images of airplanes, 435 images of frontal faces, and 798 images of motorbikes. We set n and d to 800 and 130, respectively, to account for the large number of images. Results are averaged over 10 runs with randomly selected learning pools of 100 images per class. We achieve better cluster purity with both feature types than Grauman and Darrell (2006), where sparse 10-dimensional Harris-affine SIFT features are used.

In the classification experiment, we use the learned categories to predict labels for novel images. We train Support Vector Machines with the PMK using the labels produced by the unsupervised category learning. We classify the remaining images of the dataset (2788 images, ranging from 300 to 1000 per class), where recognition performance is computed as the mean diagonal of the resulting confusion matrix, and average results over 10 runs with the randomly selected pools of training images from the object category learning experiment. The second row of Table 2 shows that our method gives better prediction for novel examples than Grauman and Darrell (2006). Our algorithm's very small standard deviations in accuracy for both experiments indicate that it is less sensitive to the composition of the unlabeled data, and provides significantly more reliable groupings.

In these experiments, our method performed better using sparse base features rather than dense base features for the semi-local descriptors. Most of the error for either feature type occurred for the airplanes being confused as cars or motorbikes. Upon further examination, we found that the confused images had more background clutter than other Airplane images, and their background features were similar to those found on the Motorbike and Car images. We also noticed that the Airplane images on average had the least number of foreground features among the four classes in the dataset, which could have resulted in semi-local descriptions occupied by a lot of background features (see Fig. 11). Thus, it makes sense that the sparse base features perform better, as the dense sampling detects many more background features.

Table 3 Comparison between semi-local descriptors and SIFT descriptors for unsupervised category learning and recognition performance on novel images for the Caltech-4 dataset. Unsupervised category learning is measured in terms of overall cluster purity, and recognition on unseen images is measured in terms of the mean diagonal of the confusion matrix. Results are mean values with standard deviations, averaged over 10 runs with randomly selected training/testing pools. We test our method (abbreviated here as "FF" for Foreground Focus) with two different features: 1) FF-Sparse, in which our semi-local descriptor uses the SIFT descriptors detected with the shape-adapted and maximally stable region detectors as base features, and 2) FF-Sparse-Local, in which the same shape-adapted and maximally stable region detected SIFT features are used without our semi-local descriptor. Our method performs better with semi-local descriptors

	FF-Sparse	FF-Sparse-Local
Purity (%)	91.10 ± 1.10	67.48 ± 3.50
Prediction accuracy (%)	92.29 ± 1.07	71.35 ± 3.12

In addition to the semi-local descriptor features, we also evaluate our method using only the 72-dimensional SIFT features detected with shape-adapted and maximally stable region detectors. In Table 3, we compare our method using these features as base features to our semi-local descriptor, and our method using only these features (without semi-local information). The interest point detected SIFT features (FF-Sparse-Local) performs much worse than the semi-local descriptors (FF-Sparse). This again shows the value of our semi-local descriptors—that capturing both appearance and geometry in a local neighborhood can produce more informative and descriptive features that are less likely to spuriously match. The initial clusters produced from the sparse

features had low overall cluster purity, and not much improvement could be made over iterations.

Finally, we compare the accuracy of our method's foreground discovery to that of several latent topic models for the Caltech motorbike class, as reported in Liu and Chen (2007). Foreground features are determined by ground-truth bounding box segmentations. In Liu and Chen (2007), two probabilistic Latent Semantic Analysis (pLSA) topic models are learned from a combined dataset of the Caltech motorbike class (826 images) and the Caltech background class (900 images). Similarly, we learn two object categories (clusters). The foreground detection rate is computed by varying the threshold among the top 20% most confident features as prescribed in Liu and Chen (2007). Interest points are detected using the Hessian-affine detector (Mikolajczyk and Schmid 2004), and described by SIFT descriptors. The descriptors are projected down by PCA to 30 dimensions, and n is set to 500. These features and parameters are consistent with those used in Liu and Chen (2007).

We compare our method with a 1) standard pLSA model, 2) a pLSA model with spatial information (Liu and Chen 2006) that hypothesizes the location and scale of the object, and 3) a correspondence-based pLSA variant (Liu and Chen 2007) that considers the configuration of patches belonging to the object. The pLSA models compute foreground confidence based on the probability of the topic given the patch. We test our method with two feature types: 1) Hessian-affine SIFT, as described above, and 2) semi-local descriptors with Hessian-affine SIFT as base features. We set d to 130 for both feature types. Results are shown in Fig. 12. Our approach outperforms the others for most points on the detection curve, providing much better precision for low false positive rates. Using the semi-local descriptors performs overall slightly better than using the local descriptors.

The gap in performance between the pLSA methods and our approach for this experiment makes sense due to the primary differences between the two approaches. The pLSA methods model each image as a mixture across all topics: they produce a soft-clustering of the data and every visual word in an image contributes to the topic discovery. As a result, the foreground confidence of a feature (visual word) depends both on how likely it is part of the described topic across the dataset as well as how likely its parent image belongs to that topic. This means that a motorbike image can have highly weighted features on the background (in addition to those on the foreground), if the motorbike topic does not dominate the background topic. Furthermore, background images that have high probability of belonging to the motorbike topic can adversely influence the background features in motorbike images to have high weights.

In contrast, our method selects the most distinctive features by iteratively updating the feature weights and image clusters. By hard-clustering the data, our method determines

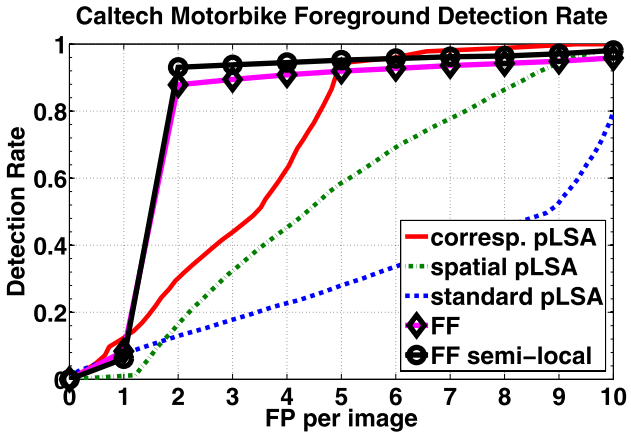


Fig. 12 Comparison with several latent topic models for foreground discovery for the Caltech motorbike class. For each method, the foreground detection rate is computed by varying the threshold among the top 20% most confident features in each image. We compare our method with a standard pLSA model, pLSA with spatial information—"spatial pLSA" (Liu and Chen 2006), and a correspondence-based pLSA variant—"corresp. pLSA" (Liu and Chen 2007). We test our method (abbreviated as FF for "Foreground Focus") with two different features: 1) FF, with Hessian-affine SIFT features (same feature setting as in Liu and Chen (2007)), without our semi-local descriptor, 2) FF semi-local, in which our semi-local descriptor uses the Hessian-affine SIFT features as base features

the feature weights solely based on intra-cluster matches: it computes the weight of a feature by taking the median weight among all intra-cluster matches. Therefore, the correctly clustered motorbike images will have their highest weighted features on the regions that consistently match well (i.e., the foreground). Likewise, our approach mitigates the impact of an incorrectly clustered background image, since matches made to the background features will likely result in low weights and be ignored by the median.

Overall, the partitioning of images that our method provides makes it possible to discover distinctive features at the object level, whereas topic models must account for an image with a soft assignment to each topic, and thus can less reliably select the most confident per-topic features. Whether a soft or hard partitioning of the unlabeled image collection is preferable would depend on the ultimate application.

7.4 Evaluation of our Semi-Local Proximity Distribution Descriptor

In previous sections, we have analyzed our method's classification and foreground discovery accuracy on various datasets. In this section, we analyze the performance of our semi-local descriptor by making a direct comparison to the neighborhood-based image descriptor of Quack et al. (2007) and to a local alternative, a simple bag-of-words description. The goal of these experiments is to determine how well our semi-local descriptor discovers the same visual patterns (foreground objects) in novel images that occur in the

training images. We tie this in with a modified version of our Foreground Focus method for determining the feature weights. The difference with our original method is that, unlike previous experiments, the following are supervised tasks and hence no clustering is involved.

We perform the same experiment as in Quack et al. (2007), where a *bounding box hit rate* (BBHR) is measured over the positive test sets. A bounding box hit (BBH) is counted if more than h of the selected features lie on the object (i.e., inside the bounding box) and the BBHR is the total BBH normalized by the total number of object instances in the positive test set. The BBHs are measured by using the ground truth bounding box annotations. The BBHR is measured with respect to the False Positive Rate (FPR) which is the number of selected features lying outside of the bounding box divided by the number of selected features, averaged over the entire test set. The selected features are determined by varying the selection threshold over the feature confidences. The idea behind the BBHR is that there should be at least a certain number of features selected for later processes (such as an object recognition system) to operate effectively. Therefore, the metric (i.e., BBHR vs FPR) compares the tradeoff of the number of selected foreground features at the expense of false positives (background features that are mistakenly thought to be on the foreground).

The experiment is conducted on three object categories: Bikes, Motorbikes, and Cars Rear. We use the same images from the publicly available datasets and set h to five, as in Quack et al. (2007). We use densely sampled SIFT base features for our semi-local descriptor, and set n to 200 and d to 130. The positive training images use only regions of the image that correspond to the object (inside the bounding box), except for the Motorbikes where full images are used since no ground truth annotation is available.

More detail on the datasets corresponding to the object categories are as follows:

Bikes. This dataset has 250 positive training, 250 negative training, and 125 testing images of bikes taken from the GRAZ-01 (100 positive training, 100 negative training, and 50 testing) and GRAZ-02 (150 positive training, 150 negative training, and 75 testing) datasets (Opelt et al. 2006), respectively.

Motorbikes. This dataset has 826 positive training images taken from the Caltech-4 Motorbikes database and 200 images randomly taken from the Caltech-256 (Griffin et al. 2007) background class. There are 115 testing images taken from the TUD Motorbikes (Everingham et al. 2006) dataset.

Cars Rear. This dataset has 126 positive training images and 526 testing images of rear-views of cars from the Caltech-4 dataset. The negative training images are 1155 images of street scenes without cars.

We compare our method with Quack et al. (2007), in which a tiled region is centered on each interest point to bin nearby visual words. The scale of the neighborhood is determined by the size of the region of interest (detected feature). The confidence of a feature in a test image is measured by counting how often it is part of a neighborhood that matches to mined configurations of the training images. The more matched configurations the feature is part of, the higher its confidence. We also compare our method with the baseline bag-of-words scheme (as defined in Quack et al. 2007), where each visual word is given a weight based on how often it appears in the training images of the given category. Then, each feature in a test image is given a confidence that corresponds to the weight of its matching visual word.

In order to make a fair comparison to Quack et al. (2007), we modify our unsupervised Foreground Focus method to work with labeled data, and devise a method to set weights on our features discriminatively. Since we are only working with a single dataset at a time, no clustering is involved (and hence no iterations).

We compute the weights on our descriptors in the test images as follows. First, each test image is matched to all images in the positive training set. The feature weights for a test image are computed in exactly the same way as in our original Foreground Focus method; we take the median over the weights (which take into account both the mass of the matching features as well as their distances) obtained from the pair-wise matches between the test image and all positive training images. We only select features that have weight greater than one—essentially, we are only keeping the features that our method determines to be on the foreground, since all features initially have weights set to one. The remaining feature weights are set to zero. Then, the process is repeated, but with the test images matched to the *negative* training images. Again, we select features that have weight greater than one, which are those that our method determines to be on the background (since this time the test images are matched to the negative training examples). Finally, among the features that our method had determined to be on the foreground, we discard the features that are also determined to be on the background.

We are left with a set of features that have high confidence of belonging to the foreground, while at the same time have low confidence of belonging to the background. This procedure is analogous to the way the confidences for the features of Quack et al. (2007) are determined, where both the positive and negative training examples are used to find the most distinctive and frequent foreground features that are unlikely to be part of the background. Results on the three datasets are shown in Fig. 13. The results of Quack et al. (2007) are shown as curves with square markings, the baseline bag-of-words scheme results are shown as curves with diamond markings, and our method's results are shown as curves with triangles.

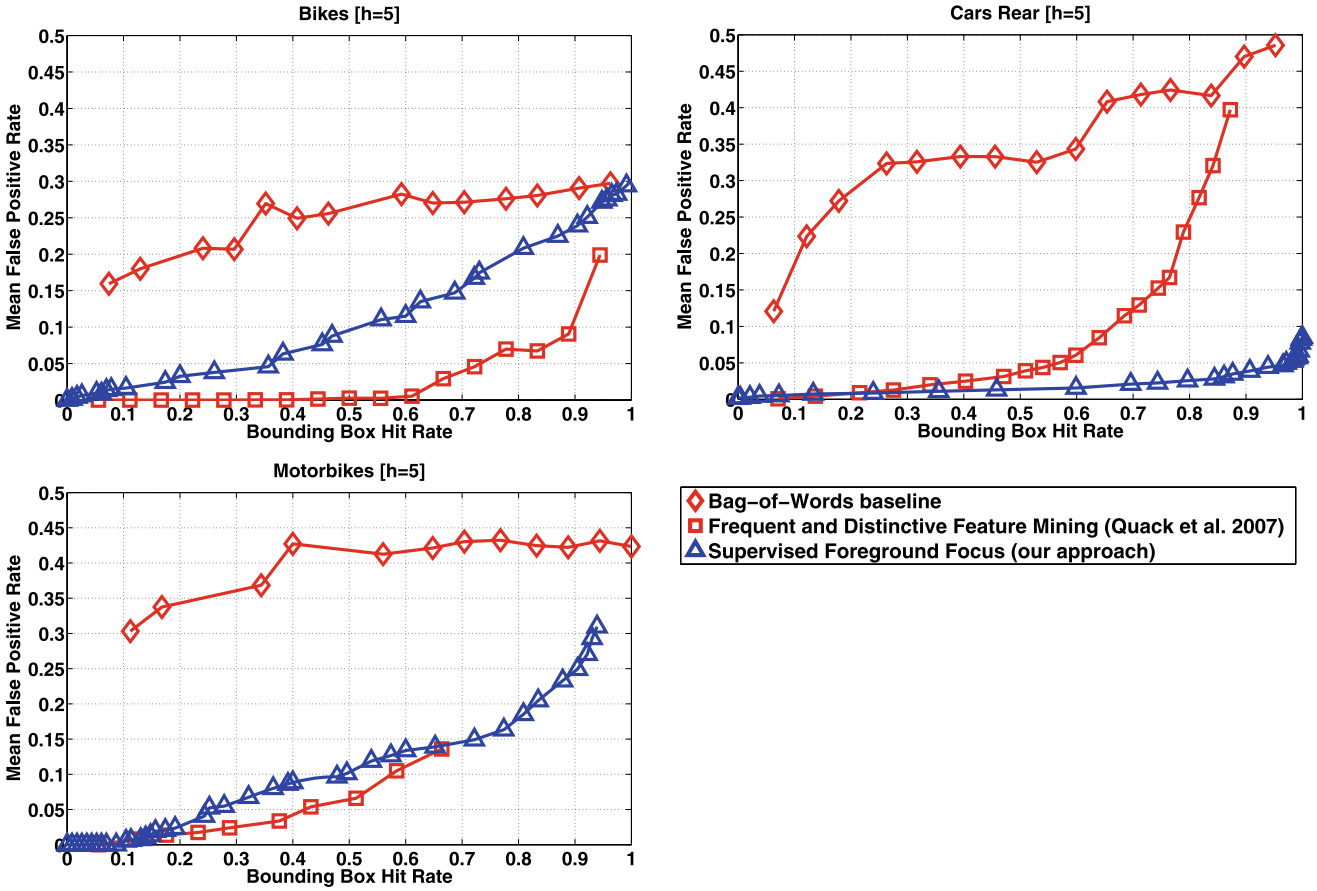


Fig. 13 Bounding box hit rates (BBHR) vs. mean false positive rates (FPR) for Bikes, Motorbikes, and Cars Rear. *Lower curves* are better. Overall, our method significantly outperforms the baseline bag-of-words scheme and achieves higher BBHRs than the method of Quack et al. (2007). On the Cars Rear and Motorbikes datasets, the higher BBHRs are obtained with lower and comparable mean FPRs, respectively, to (Quack et al. 2007)

Our semi-local descriptor significantly outperforms the baseline bag-of-words scheme on all datasets, which confirms that our method for adding geometric and spatial layout information to local appearance descriptors results in better foreground discovery. When comparing to Quack et al. (2007), our method performs better on the Cars Rear dataset, achieves higher BBHRs (with comparable FPR at low BBHR) on the Motorbikes dataset, and performs worse on the Bikes dataset. We achieve the best results on the Cars Rear dataset, because the negative training images specifically match the backgrounds of the positive training images. Hence, any background feature that may (incorrectly) have high weight of belonging to the foreground (when matching to the positive training images) would have high weight of belonging also to the background (when matching to the negative training images) and would be discarded by our modified Foreground Focus method. This method only considers those features that have high weight of belonging to the foreground and low weight of belonging to the background.

The negative training images help less on the Motorbikes and Bikes datasets, because they are not specifically chosen to be similar in appearance to the backgrounds that appear in the positive training images. These datasets also have severe clutter in many images, and large object scale and appearance variations. Still, our semi-local descriptor performs well, as can be seen by the low mean FPR even as the BBHR increases.

The method of Quack et al. (2007) is much more selective, as can be seen by the lower maximum BBHR that is obtained on all datasets compared with our method. While a more discriminative approach may reduce false positives, it can also hurt performance for the ensuing object recognition system if there are too few features to work with (even if all of them lie on the foreground). Our semi-local descriptor is able to find many foreground features and still achieve low FPR even at high BBHRs. For example, on the Motorbikes dataset, the curves produced from our method and the method of Quack et al. (2007) are similar for low BBHRs, but then our method achieves better total foreground hits, as seen by our curve extending to higher BBHRs (at relatively

low mean FPRs) while the curve produced by Quack et al. (2007) stops when the BBHR is approximately 0.67.

In terms of the differences in neighborhood-descriptions, we use ranked nearest neighbor (in image space) feature descriptions while in Quack et al. (2007) a tiled grid is used. The tiled neighborhood description does not count the number of occurrences of the same visual word that falls in a tile—it is a set description rather than a bag description. This means that the same descriptions are produced for a tile that has many occurrences of the same visual word and a tile that has just a single occurrence of that same visual word. In contrast, our descriptor counts multiple occurrences of the same visual word, regardless of the regions in which they actually fall with respect to the base feature. Therefore, our descriptor can be more specific.

In terms of orientation, however, our descriptor has a coarser description of where each neighboring feature lies (we only determine the spatial order and in which quadrant the feature lies with respect to the base feature), while the tiled neighborhood description of Quack et al. (2007) is more rigid since it depends on the actual tile in which a feature falls. In this sense, our descriptor provides a more flexible encoding, which can be more robust to deformable objects, e.g., people or animals, or when there are in-plane and/or out-of-plane rotations of the object.

Finally, our descriptor encodes all R neighbors of a base feature's neighborhood in its description, while in Quack et al. (2007) the most discriminative features in each tiled neighborhood are selected. While our descriptor can be considered to be more specific (since all nearest neighboring features in a base feature's neighborhood are considered), it can also lead to more clutter features being part of the description (for base features that fall near the edge of the object). This can potentially hurt our descriptor and is the main reason for its poorer performance on the Bikes dataset. In this dataset, there are many bikes that occupy small and narrow regions of the image (see Table 4). Furthermore, most of the bikes occupy irregularly-shaped regions in the image where the background intertwines with the object (e.g., region between handle and seat). A lot of background clutter is considered for each base feature in these regions, which leads to incorrect high matches between descriptors of the foreground and background regions, and incorrect low matches between descriptors of the foreground and foreground regions.

When the objects occupy regions that have regular shape, however, most of the features considered for the base feature will belong to the foreground object. In these situations, our descriptors can be very discriminative (and almost clutter-free) and can lead to very precise foreground matching. This is confirmed by the good performance of our descriptor on the Cars Rear dataset, where the objects also occupy a small portion of the image but the regions covered by the object are wide and tall rectangularly-shaped regions.

Table 4 The percentage of the image that is occupied by the foreground object for each category. The percentages are computed by taking the ratio between the area occupied by the object and the area of the entire image

	Bikes	Motorbikes	Cars rear
Train set (%)	24.80	85.30	24.87
Test set (%)	23.93	36.20	21.91

Figure 14 illustrates example results in which our method finds good support on the foreground. Figure 15 illustrates example results where our method weights foreground features highly, but also mistakenly finds good support for some background. In these examples, we show the highest (top 10%) weighted base features to our semi-local descriptors. Images in which our method does not perform as well on are those that have large scale variations (our current implementation uses only one scale for dense features). Nonetheless, the high maximum BBHRs at low mean FPRs achieved by our method indicate that our method finds good foreground support on most images, at relatively low cost of finding clutter features.

7.5 Summary of Results

We have analyzed the mutual reinforcement of foreground and clusters, and have made comparisons against existing unsupervised methods on images from the Caltech-101 and MSRC-v1 datasets. Our results show that foreground discovery can lead to better cluster quality, and vice versa. We have also made comparisons with different types of base features to our semi-local descriptors—specifically, we have compared the densely sampled SIFT features to the interest point detected SIFT features—and have found that the performance of the sampling type is dependent on the specific distribution of the features and images of the dataset. We have shown good performance compared to the latent topic models for foreground segmentation tasks on the Caltech-4 Motorbikes dataset. Finally, we have evaluated our semi-local descriptor by measuring foreground discovery on the Bikes, Motorbikes, and Cars Rear datasets. Our descriptor significantly outperforms a baseline bag-of-words scheme on all datasets, and offers some advantages relative to a state-of-the-art frequent configurations descriptor.

8 Conclusion

We have introduced a novel unsupervised method for discovering foreground features in images. Clusters are determined by matching weighted feature sets, and weights are iteratively adjusted based on contributions to intra-cluster image matches. We show that this mutual reinforcement improves both cluster quality and foreground

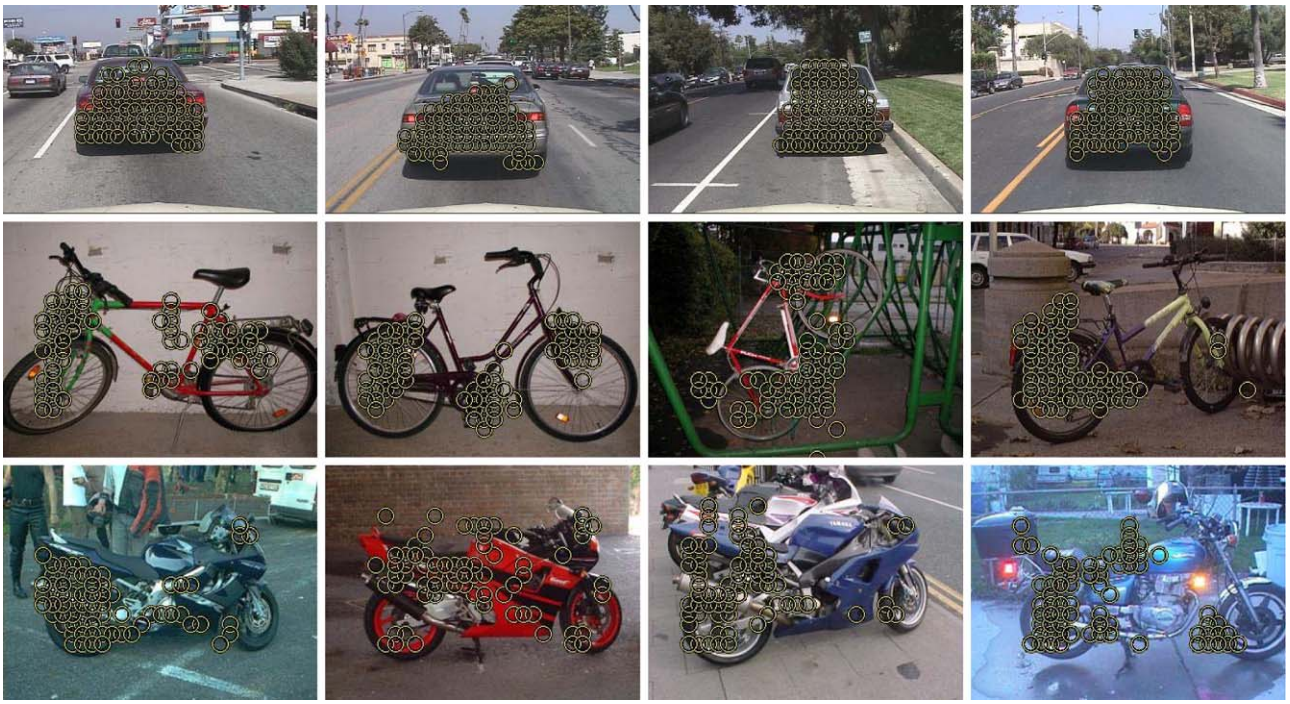


Fig. 14 Examples showing the highest weighted features per image found by our modified (supervised) Foreground Focus method. In these examples, our method attributes weight almost only to foreground features. Note that we show the base features to our semi-local descriptors

Fig. 15 Examples our modified (supervised) Foreground Focus method does most poorly on: it weights foreground features highly, but also (mistakenly) finds good support for some background. Note that we show the base features to our semi-local descriptors

detection, with datasets containing four to twenty categories.

In future work, we will investigate how our algorithm could accept incremental updates to the unlabeled pool. Feature weights and image clusters will be updated incrementally, specific to each added instance. This approach is appealing because it does not attempt to *fix* clusters, but rather would let the discovered visual patterns adjust to the new

data. We would also like to extend our method to multiple-label cluster assignments. A soft assignment to clusters could be made to allow multiple patterns in a single image influence the feature weight updates and resulting clusters.

We plan to consider sparser affinity matrices to improve the spectral clustering's computational complexity for dealing with very large datasets. Finally, while we have focused on category discovery here, it would be interesting to see how our unsupervised feature selection could be used to automatically construct summaries of unstructured image collections.

Acknowledgements The authors would like to thank the anonymous reviewers for providing excellent suggestions, and Delbert Dueck, David Liu, and Till Quack for sharing their experimental data and results. We also gratefully acknowledge support for this research provided in part by National Science Foundation CAREER Award 0747356, a Microsoft Research New Faculty Fellowship, Texas Higher Education Coordinating Board ARP 003658-01-40-2007, National Science Foundation EIA-0303609, and the Henry Luce Foundation.

Appendix

Given the expense of computing the EMD, we developed a variant of the Pyramid Match algorithm to approximate both the partial match cost as well as the flow between two weighted point sets.

The Pyramid Match Kernel (PMK) approximates the least cost match for unweighted sets in linear time in the number of points in a set by intersecting multi-resolution histograms computed in the feature space (see Grauman and Darrell 2005). Given a set of feature vectors, $\mathbf{S} = \{\mathbf{X}_1, \ldots, \mathbf{X}_{|S|}\}$ where $\mathbf{X}_i \in \Re^d$, $\forall i$, an L-level multi-resolution histogram $\mathbf{H} = [H_o, \ldots, H_{L-1}]$ is computed. The PMK value between two features sets $\mathbf{X}_i$ and $\mathbf{X}_j$ is defined as the weighted sum of the number of matches that occur at each resolution:

$$K(\mathbf{X}_i, \mathbf{X}_j) = \sum_{k=0}^{L} w_k (I_k(\mathbf{X}_i, \mathbf{X}_j) - I_{k-1}(\mathbf{X}_i, \mathbf{X}_j)), \tag{3}$$

$$I_k(\mathbf{X}_i, \mathbf{X}_j) = \sum_{n=1}^{b_k} \min(H_k(\mathbf{X}_i^{(n)}), H_k(\mathbf{X}_j^{(n)})), \tag{4}$$

where $H_k(\mathbf{X}_i^{(n)})$ is the count in bin n of multi-dimensional histogram $H_k(\mathbf{X}_i)$ having b_k bins, and w_k is a weight reflecting the similarity between points matched at level k. Note that $I_{-1}(\mathbf{X}_i, \mathbf{X}_j) = 0$.

Though defined for unweighted point sets, for this work, we propose a variant to the PMK in which we apply weights by scaling every histogram bin increment by the weight attached to that point. Given two multi-resolution histograms computed from two feature sets, for every intersecting bin, we compute the optimal matching between the features from both sets that share the bin. We record the flow and cost that each point at the current resolution level contributes to the match; any remaining weight is propagated to the next coarser pyramid level and can be used in future matchings. Zero-weighted features at any level do not contribute to the match. In the end, when all bins have been intersected, we have accumulated the approximate flow and match cost. Each per-bin flow computation is super-linear in the intersection value, but feature space partitions given by the pyramid result in small and gradually increasing intersection counts.

To construct the pyramid tree for this modified PMK algorithm, we randomly sample a representative corpus of features from the data, and partition the feature space with hierarchical k-means clustering (with Euclidean distance). The number of levels, L, and branches, B, of the tree are user-defined parameters—we typically use 10 branches with four or five levels.

References

Agarwal, A., & Triggs, B. (2006). Hyperfeatures multilevel local coding for visual recognition. In *European conference on computer vision*.

Chum, O., & Zisserman, A. (2007). An exemplar model for learning object classes. In *Conference on computer vision and pattern recognition*.

Dhillon, I., Guan, Y., & Kulis, B. (2004). Kernel k-means: spectral clustering and normalized cuts. In *ACM SIGKDD international conference on knowledge discovery and data mining*.

Dorko, G., & Schmid, C. (2003). Selection of scale-invariant parts for object class recognition. In *International conference on computer vision*.

Dueck, D., & Frey, B. (2007). Non-metric affinity propagation for unsupervised image categorization. In *International conference on computer vision*.

Dy, J., & Brodley, C. (2004). Feature selection for unsupervised learning. *Journal of Machine Learning Research*, *5*, 845–889.

Everingham, M., Zisserman, A., Williams, C. K. I., & Van Gool, L. (2006). The PASCal visual object classes challenge 2006 (VOC2006) Results.

Fei-Fei, L., & Perona, P. (2005). A Bayesian hierarchical model for learning natural scene categories. In *Conference on computer vision and pattern recognition*.

Fei-Fei, L., Fergus, R., & Perona, P. (2004). Caltech 101 image database.

Fergus, R., Fei-Fei, L., Perona, P., & Zisserman, A. (2005). Learning object categories from Google's image search. In *International conference on computer vision*.

Grauman, K., & Darrell, T. (2004). Fast contour matching using approximate Earth mover's distance. In *Conference on computer vision and pattern recognition*.

Grauman, K., & Darrell, T. (2005). The pyramid match kernel: Discriminative classification with sets of image features. In *International conference on computer vision*.

Grauman, K., & Darrell, T. (2006). Unsupervised learning of categories from sets of partially matching image features. In *Conference on computer vision and pattern recognition*.

Griffin, G., Holub, A., & Perona, P. (2007). Caltech 256 image database.

Lazebnik, S., Schmid, C., & Ponce, J. (2003). A sparse texture representation using affine-invariant regions. In *Conference on computer vision and pattern recognition*.

Lazebnik, S., Schmid, C., & Ponce, J. (2004). Semi-local affine parts for object recognition. In *British machine vision conference*.

Lazebnik, S., Schmid, C., & Ponce, J. (2006). Beyond bags of features: Spatial pyramid matching for recognizing natural scene categories. In *Conference on computer vision and pattern recognition*.

Lee, Y. J., & Grauman, K. (2008a). Foreground focus: Finding meaningful features in unlabeled images. In *British machine vision conference*.

Lee, Y. J., & Grauman, K. (2008b). Discovering multi-aspect structure to learn from loosely labeled image collections. Technical report, UT-Austin, May 2008b.

Leibe, B., Leonardis, A., & Schiele, B. (2004). Combined object categorization and segmentation with an implicit shape model. In *Wkshp on statistical learning in computer vision*.

Ling, H., & Soatto, S. (2007). Proximity distribution kernel for geometric context in recognition. In *International conference on computer vision*.

Liu, D., & Chen, T. (2007). Unsupervised image categorization and object localization using topic models and correspondences between images. In *International conference on computer vision*.

Liu, D., & Chen, T. (2006). Semantic-shift for unsupervised object detection. In CVPR *Wkshop on Beyond Patches*.

Lowe, D. (2004). Distinctive image features from scale-invariant keypoints. *International Journal of Computer Vision* 60(2).

Marszalek, M., & Schmid, C. (2006). Spatial weighting for bag-of-features. In *Conference on computer vision and pattern recognition*.

Mikolajczyk, K., & Schmid, C. (2004). Scale and affine invariant interest point detectors. *International Journal of Computer Vision*, *1*(60), 63–86.

Nowak, E., Jurie, F., & Triggs, B. (2006). Sampling strategies for bag-of-features image classification. In *European conference on computer vision.*

Opelt, A., Fussenegger, M., Pinz, A., & Auer, P. (2006). Generic object recognition with boosting. *Transacations on Pattern Analysis and Machine Intelligence* 28(3).

Quack, T., Ferrari, V., Leibe, B., & Gool, L. V. (2007). Efficient mining of frequent and distinctive feature configurations. In *International conference on computer vision*.

Quelhas, P., Monay, F., Odobez, J.-M., Gatica-Perez, D., Tuytelaars, T., & Van Gool, L. (2005). Modeling scenes with local descriptors and latent aspects. In *International conference on computer vision*, Beijing, China, October 2005.

Rubner, Y., Tomasi, C., & Guibas, L. (2000). The earth mover's distance as a metric for image retrieval. *International Journal of Computer Vision*, *40* (2), 99–121.

Russell, B., Efros, A., Sivic, J., Freeman, W., & Zisserman, A. (2006). Using multiple segmentations to discover objects and their extent in image collections. In *Conference on computer vision and pattern recognition*.

Shi, J., & Malik, J. (2000). Normalized cuts and image segmentation. *Transacations on Pattern Analysis and Machine Intelligence*, *22*(8), 888–905.

Sivic, J., & Zisserman, A. (2004). Video data mining using configurations of viewpoint ivariant regions. In *Conference on computer vision and pattern recognition*.

Sivic, J., Russell, B., Efros, A., Zisserman, A., & Freeman, W. (2005). Discovering object categories in image collections. In *International conference on computer vision*.

Weber, M., Welling, M., & Perona, P. (2000). Unsupervised learning of models for recognition. In *European conference on computer vision*.

Winn, J., & Jojic, N. (2005). LOCUS: Learning object classes with unsupervised segmentation. In *International conference on computer vision*.

Winn, J., Criminisi, A., & Minka, T. (2005). Object categorization by learned universal visual dictionary. In *International conference on computer vision*.

Zelnik-Manor, L., & Perona, P. (2004). Self-tuning spectral clustering. In *Advances in neural information processing (NIPS)*, Vancouver, Canada, December 2004.