In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), June, 2010.

Collect-Cut: Segmentation with Top-Down Cues Discovered in Multi-Object Images

Yong Jae Lee and Kristen Grauman
University of Texas at Austin
yjlee0222@mail.utexas.edu, grauman@cs.utexas.edu

Abstract

We present a method to segment a collection of unlabeled images while exploiting automatically discovered appearance patterns shared between them. Given an unlabeled pool of multi-object images, we first detect any visual clusters present among their sub-regions, where inter-region similarity is measured according to both appearance and contextual layout. Then, using each initial segment as a seed, we solve a graph cuts problem to refine its boundary—enforcing preferences to include nearby regions that agree with an ensemble of representative regions discovered for that cluster, and exclude those regions that resemble familiar objects. Through extensive experiments, we show that the segmentations computed jointly on the collection agree more closely with true object boundaries, when compared to either a bottom-up baseline or a graph cuts foreground segmentation that can only access cues from a single image.

1. Introduction

Unsupervised learning from images (also referred to as "discovery") entails detecting the visual patterns that occur with some regularity within an unlabeled collection of examples. Reliable discovery methods would be useful for a number of practical applications—such as generating compact summaries of large photo collections, organizing image or video data for content-based similarity search, identifying the rarer instances, or even to supplement traditional supervised object recognition systems. Recent progress on the discovery problem has yielded methods able to cluster images according to their primary object category [8, 12], to rank the "topics" present somewhere in each image [25, 5], to mine for object descriptors [19], and summarize iconic landmarks in tourist photos [18].

However, discovering generic object categories from natural images remains a considerable challenge, for two primary reasons. First, generic categories lack the strict geometric consistency and distinctive features inherent to specific objects, forcing a discovery method to simultaneously identify natural groups while estimating their variability.[1] Second, with multiple objects present within a single image, a method must identify those segments among all possible image decompositions that will reveal common objects, as well as the common object types themselves—yet both tasks influence the other.

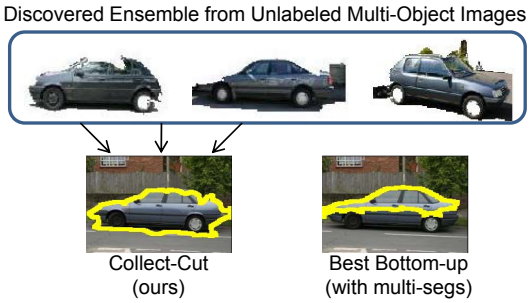


Figure 1. Our method discovers and exploits the shared structure in a collection of unlabeled images in order to segment the objects more accurately than is possible with bottom-up methods.

Most approaches have avoided the second problem by imposing the (usually unrealistic) restriction that each image contain only a single object of interest, or else by forgoing localization of what is discovered. An exception is the method of [22], which deals with multi-object images by first decomposing each into multiple segmentations, and then looking for common patterns among the pool of segments (*sub-images*), rather than the pool of images. The assumption is that each semantic object will have a corresponding sub-image somewhere in the pool.

Unfortunately, this strategy loses the context offered by the image content surrounding each region. Furthermore, while using multiple segmentations helps safeguard against missing "good" regions, there is still a risk of omitting meaningful segments from the pool, and thus never having the chance to detect their regularity. Bottom-up segmentation by definition has no concept of object categories, and so cannot reliably produce coherent regions that agree with

[1] We use *generic* in the usual sense, to refer to basic categories of objects (e.g., cows, cars), in contrast to *specific* objects (e.g., the Eiffel tower, my car). By *multi-object* we mean an image with multiple objects of interest.

true object boundaries. In fact, recent studies [16] suggest that in practice close to 10,000 segments *per image* need to be generated to ensure a "good" segment exists for each object—an enormous number considering that on average a natural image contains only 10 objects.

In the unsupervised realm, this implies a notable computational burden for existing methods, especially those that require pairwise distance computations between all regions in the unlabeled pool (e.g., spectral or agglomerative clustering); at $10K$ segments per image, a meager collection of only 100 images would already require one trillion comparisons! Computation aside, simply increasing the pool of candidate segmentations is bound to add many distracting "noise" regions, with a negative impact on discovery. A polluted pool with a low signal-to-noise ratio will make it harder for the algorithm to find the matches among the good segments in order to group them.

Our idea is to discover shared top-down cues from a collection of unlabeled multi-object images, and use them to refine both the segments and discovered objects. Rather than commit to a pool of candidate segments, our method allows any initially discovered shared appearances to influence segmentation boundaries, and then in turn, lets the refined regions influence the category-level grouping. Given an initial set of bottom-up segmentations, we first detect any clusters (or visual "themes") that agree in terms of appearance and contextual layout. Then, for each discovered pattern we form an *ensemble* model consisting of its representative regions. We design an energy function amenable to graph cuts [10] to revise the spatial extent of each initial segment. This step essentially favors keeping pixels that agree with the appearance of any part of the cluster's ensemble model; meanwhile, it favors losing pixels that either agree with the remaining background in its original image, or are likely attributable to a familiar previously learned class.

Unlike existing applications of graph cuts for segmentation (e.g., [20, 21]), our method generates the "foreground" model in a data-driven way, from the patterns shared across the unlabeled images. Further, it permits the inclusion of somewhat heterogeneous instances within a generic category, due both to our use of an ensemble foreground model, as well as our integration of a context-aware discovery algorithm [13] to find the initial groups. Finally, by favoring cuts that separate familiar and unfamiliar regions, our discovery approach can be exploited in semi-supervised situations where direct class-specific knowledge is available, but only for a partial set of categories appearing in the image collection.

We demonstrate our method with two datasets, and show that segmentation results are significantly closer to ground truth object boundaries when we leverage the shared discovered structure, as compared to either bottom-up segmentation or a graph cuts baseline that lacks access to the full collection. Further, we illustrate the positive impact the refined segmentation has on unsupervised category discovery.

2. Related Work

In this section, we review relevant work in unsupervised visual discovery and image segmentation.

The goal in *unsupervised discovery* is to detect recurring visual patterns within a collection of unlabeled images [28]. Several methods use topic models e.g., [25, 22, 5], while others use graph-based clustering [8, 12, 18, 13]. Some work considers scalable techniques for mining common feature patterns in large image collections [19, 18, 4]. Related to these methods, our end goal is to decompose large un-annotated image collections into their common visual patterns, though we focus on generic object categories [25, 22, 12, 8, 13]. In contrast to previous work, our approach actively refines the segmentation of the discovered objects as it detects their similarities, resulting in improved localized category discovery.

Weakly-supervised methods can segment out the foreground region in cluttered images, with the assumption that each image has the same single prominent object [29, 11, 27, 1] or scene type [26]. Such methods leverage the statistics across the weakly-labeled collection to better identify the true object boundaries, though they assume more supervision than discovery methods. A recent method performs *unsupervised* image segmentation with data-driven scene matching [23]. *Top-down segmentation* methods exploit class-specific knowledge, often combining supervised object detectors with low-level grouping cues [2, 9, 7]. In the proposed setting, the learner must discover (pseudo)-top-down cues from recurring visual patterns in unlabeled images.

Recent work shows how to use link-analysis techniques [8] or consistent feature matches [12] to select foreground SIFT features during discovery. Related to these methods, we let the discovery process influence which image parts are emphasized. However, in contrast, our approach removes the assumption of having single-object images, and it provides region segmentation rather than feature selection among interest points.

Graph-cuts methods [3, 10] have been developed for human-guided foreground-background segmentation [20] and for *co-segmentation* of objects in a pair of images [21, 17]. The latter requires that the same specific object appear in both images, and that the two backgrounds be distinct in appearance. An extension of these methods initializes the foreground automatically using pLSA [14]. An approach to co-segment clothing regions is developed in [6]: it constructs a foreground model from the average appearance within clothing segments identified under faces, and then applies graph cuts to refine the boundaries.

Our method can also be viewed as a form of co-

segmentation—although it segments unlabeled images into multiple unfamiliar objects. A key part of our contribution is to design graph-cut energy terms that are well-suited for joint segmentation where (a) the allowable foreground appearance is heterogeneous (i.e., at the generic category-level), (b) the background regions may not be distinct across images, and (c) some familiar objects may be present.

3. Approach: Collect-Cut

The goal is to discover top-down cues from recurring visual patterns within an unlabeled image collection, and to use them to refine the segmentations such that they better agree with object boundaries. We call the method "Collect-Cut" since it uses the image *collection* to estimate the graph cut-based segmentation.

Given a pool of unlabeled images, we decompose each into multiple segmentations. After clustering the segments from all images (Sec. 3.1), for each group the method chooses representative instances to act as an *ensemble* of possible appearance models (Sec. 3.2). The ensemble serves as (pseudo) top-down cues for that cluster's segments. For every initial "seed" segment, we refine its spatial extent at the granularity of superpixels, promoting the inclusion of regions that (a) resemble any instance of that segment's cluster ensemble, and (b) are unlikely to correspond to an instance of a familiar class. We formulate these preferences in an energy function amenable to graph cuts algorithms (Sec. 3.3). Finally, having refined each region, we recompute a clustering on all regions (Sec. 3.4). The final output is a set of segmented discovered objects.

3.1. Context-Aware Region Clustering

The first step consists of mapping an unlabeled collection of images to a set of clusters or visual topics; we employ our algorithm for "context-aware" visual category discovery [13]. The main idea of that method is to leverage the object-level context provided by previously learned categories when trying to discover novel (un-trained) categories in a pool of unlabeled images. When new instances of those familiar objects occur with some spatial regularity relative to the novel objects, their presence can help the clustering algorithm perform more reliable discovery. (For example, if we had previously learned models for *grass*, *driveway*, and *house* categories, we can better discover a novel cluster of *mailboxes* by representing their spatial inter-relationships when clustering.) The algorithm is summarized in the box above, and details are in [13]. In the proposed approach, we treat it as a black box to produce clusters of regions.

We chose this method because it significantly outperforms appearance-only approaches when a set of previously learned categories (distinct from those to be discovered) is available to build the object-context. Please note, however, that in the following the discovery of top-down segmentation cues will only be performed and evaluated on those regions that the method deems to be unknown. Thus, while we expect to be able to capture more variable intra-class instances with our context-aware method, this clustering step is interchangeable with an existing appearance-based technique (e.g., [25, 22]), as we illustrate in our experiments.

Train a set of region-based classifiers for N categories, denoted as the "known" objects.
Input: Unlabeled image collection, number of groups k.

- Obtain multiple segmentations for each image with NCuts; describe each region with a bag-of-features.
- Classify each region as either "known" or "unknown".
- For each unknown region, compute an *object-graph* descriptor that encodes the context of surrounding familiar objects.
- Compute affinities between all pairs of unknowns (using object-graphs and appearance features), and apply spectral clustering.

Output: Set of k clusters of regions.

Algorithm 1: Context-aware region clustering [13].

3.2. Assembling Ensemble Models

Given the initial clustering results from above, we can now proceed to build the ensemble models that will be used to refine the spatial extent of each individual region. An ensemble is a set of regions that represents a cluster. We use an ensemble because each cluster may itself contain some variety, for two reasons: First, the clusters are comprised of segments produced from bottom-up segmentation methods (e.g., [24]), and so may contain partial segments from the full object (for example, a single cluster may consist of both cow heads and cow bodies). Second, since we allow the regions' context to influence their grouping, a given cluster may contain somewhat heterogeneous-looking instances of objects occurring in similar contexts; for instance, the context-aware grouping might produce a cluster with both red and blue buildings, or side views and rear views of cars.

Thus, for each of the k discovered groups, we extract r representative region exemplars to serve as its top-down model of appearance. Specifically, we take those regions with the highest total affinity relative to the rest of the cluster instances. Let s_{C_i} denote the i-th segment belonging to cluster C, and let $K(\cdot,\cdot)$ denote the similarity function used for clustering. For each segment in cluster C, we compute its intra-cluster degree: $L(s_{C_i}) = \sum_j K(s_{C_i}, s_{C_j})$, sort the values, and take the top r (from unique images). This yields the ensemble model of object appearance $\{s_{C_1}, \ldots, s_{C_r}\}$ for cluster C, where for convenience of notation we are re-using the indices $1, \ldots, r$ to denote the selected top r. Though individually the ensemble's regions may be short of an entire object, as a group they represent the variable appearances that arise within generic intra-category instances (see Fig. 2(c) for an example). When refining a region's boundaries, the idea is to treat resemblance to *any one* of

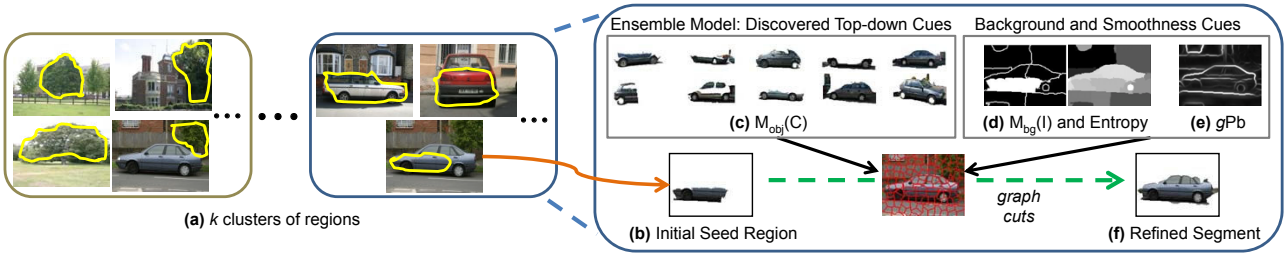


Figure 2. Overview of the proposed method. We use graph cuts to minimize an energy function that encodes top-down object cues, entropy-based background cues, and neighborhood smoothness constraints. In this example, suppose the familiar object categories are *building* and *road*. (a) A set of k clusters of regions. (b) An initial region from the pool generated from multiple-segmentations. (c) Ensemble cluster exemplars which we use to encode top-down cues. (d) Background exemplars and entropy map to encode background preference for familiar objects. Darker regions are more "known", i.e., more likely to be background. (e) Soft boundary map produced by the gPb [15] detector. (f) Our final refined segmentation for the region under consideration. Note that a single-image graph-cuts segmentation using the initial seed region as foreground and the remaining regions as background would likely have oversegmented the car, due to the top half of the car having different appearance from the seed region.

the representative ensemble regions as support for the object of interest, as described in the following section.

3.3. Collective Graph-Cut Segment Refinement

Given the discovered ensemble models, we take each initial "seed" region and refine its segmentation using graph cuts. We use a mix of large and small segments for the original multiple-segmentation pool, with the intent of capturing reasonable portions of objects; however, when computing the refinement we break each image into finer-scale superpixels so that the resulting label assignment may more closely follow true object boundaries. We generate ∼120 superpixels per image. In the following, we refer to the segments from the initial multiple segmentations as "regions", the smaller superpixel segments as "superpixels", and reserve "segment" as a generic term for either one.

We describe all segments with color and texton histograms. To compare two segments s_1 and s_2, we average the χ^2 distances of both their feature types:

$$\chi^2(s_1, s_2) = \frac{1}{2}(\chi^2_{color}(s_1, s_2) + \chi^2_{texton}(s_1, s_2)). \quad (1)$$

A seed region has both an image and cluster membership. Below we use subscripts to refer to either a region's image or its cluster; s_{C_i} refers to the i-th region in cluster C, and s_{I_j} refers to the j-th region in image I.

The idea is to compute a refined labeling over the superpixels in the image to separate the object that overlaps with the current "seed" region from the background.[2] Both the initial region itself and the cluster's ensemble model guide the assignment of "object" superpixel labels, while the originating image alone guides the assignment of "background" superpixel labels. The output labeling will serve as the refinement for that initial region.

We define a graph over an image's superpixels: a node corresponds to a superpixel, and an edge between two nodes corresponds to the cost of a cut between two superpixels. The energy function we minimize is:

$$E(f, s_{seed}) = \sum_{i \in \mathcal{S}} D_i(f_i) + \sum_{i,j \in \mathcal{N}} V_{i,j}(f_i, f_j), \quad (2)$$

where f is a labeling of the superpixel nodes, $\mathcal{S} = \{p_1, \ldots, p_n\}$ is the set of n superpixels in the image, $\mathcal{N}$ consists of neighboring (adjacent) superpixels, and i and j index the superpixels. Each superpixel p_i is assigned to $f_i \in \{0, 1\}$, where 0 corresponds to background and 1 corresponds to object.[2] The data cost term is $D_i(f_i)$, and the smoothness cost term is $V_{i,j}(f_i, f_j)$. Note that the energy is parameterized by s_{seed}, since we will optimize this function once for each seed region.

We define the data term as:

$$D_i(f_i) = \begin{cases} \exp\left(-d(p_i, M_{obj}(C))\right), & \text{if } f_i = 0; \\ \exp\left(-d(p_i, M_{bg}(I))\right), & \text{if } f_i = 1. \end{cases} \quad (3)$$

where $M_{obj}(C)$ and $M_{bg}(I)$ denote the foreground ensemble model and background model, respectively. Note that the foreground model is a function of the cluster C, and the background model is a function of the image I. We let $M_{obj}(C)$ consist of the r exemplars in the ensemble *plus* the initial seed region: $M_{obj}(C) = \{s_{C_1}, \ldots, s_{C_r}, s_{seed}\}$. We let $M_{bg}(I)$ consist of the regions from image I *minus* the seed region: $M_{bg}(I) = \{s_{I_1}, \ldots, s_{I_v}\} \backslash \{s_{seed}\}$, where v is the number of regions in image I's segmentation. Our data term assigns a high cost when a superpixel is labeled as background (object) but has a low distance to the ensemble model (image's background).

When computing the distances $d(p_i, M)$ above, we take the minimum distance between p_i and any instance in the set M. We want to exploit the diversity of object parts in the ensemble, and to let each model instance contribute only when needed. For example, if there are red and blue

[2]We use the terms "foreground object" and "background" to be consistent with familiar uses of graph-cuts segmentation, though in this case their meanings are relative. That is, since we work with multi-object images, each region from the initial segmentation will be considered separately as a possible "foreground object" in turn. The "object" label is the given cluster C, and "background" is the remaining objects in the image.

cars among the exemplars, a refinement of a red car region would benefit from the red exemplars rather than a single combined (e.g., average of red and blue) model. Specifically, we compute:

$$d(p_i, M_{obj}(C)) = \min_j \chi^2(p_i, s_{C_j}), \text{ for } s_{C_j} \in M_{obj}(C),$$
$$d(p_i, M_{bg}(I)) = \chi^2(p_i, s_{I_k^*}), \text{where}$$
$$k^* = \underset{k}{\operatorname{argmin}}\, w_k\, \chi^2(p_i, s_{I_k}), \quad (4)$$

where the argmin serves to keep $d(p_i, M_{obj}(C))$ and $d(p_i, M_{bg}(I))$ on the same scale.

The last equation above imposes the weight w_k on region s_{I_k} from the image's background set. The purpose of the weighting is to modulate the distances between a superpixel and the $M_{bg}(I)$ regions, so as to prefer that *familiar* objects be treated as background. It is defined as follows:

$$w_k = (-\log H(s_{I_k}))^{-1}, \text{where} \quad (5)$$
$$H(s_{I_k}) = -\frac{1}{\log_2 N} \sum_{n=1}^{N} P(o_n|s_{I_k}) \log_2 P(o_n|s_{I_k}),$$

and $o_1, \ldots, o_N$ are the N familiar object models used by the context-aware discovery in Alg. 1. Note that $H(s_{I_k})$ is the (normalized) entropy for segment s_{I_k}. The lower the entropy under the "known" models, the more familiar we consider the region (see Fig. 2 (d)). The weight w_k has a sharp peak for a normalized entropy value of 1, and then a gradual fall-off as the entropy decreases. Thus, if w_k is small (more "known"), it downweights the χ^2 distance, and makes the region k more likely to be selected as the superpixel's most similar background region. In this way, we account for the relative certainty of detected familiar objects to hone the segmentation for novel unfamiliar objects.

Finally, we define the smoothness term in Eqn. 2 as:

$$V_{i,j}(f_i, f_j) = |f_i - f_j| \cdot \exp(-\beta \cdot z(p_i, p_j)), \quad (6)$$

where $z(p_i, p_j) = \frac{1}{2}(\chi^2(p_i, p_j) + \text{Pb}(p_i, p_j))$, and Pb (Probability of boundary) is determined by the probability outputs given by the gPb [15] detector (see Fig. 2 (e)). For each pair of neighboring superpixels, we look at their boundary pixels and the associated gPb confidences. We compute a single value, $\text{Pb}(p_i, p_j)$, by averaging over those boundary confidences. Our smoothness term is standard and favors assigning the same label to neighboring superpixels that have similar color and texture and have low probability of an intervening contour.

We minimize Eqn. 2 with graph cuts [3], and use the resulting label assignment as the refined segmentation for region s_{seed} (see Fig. 2 (f)).

Fully Unsupervised Variant: We briefly explain how to apply our framework in the fully unsupervised setting where no previously learned category models are available.

Input: Unlabeled image collection, parameter k.

- Obtain a set of k clustered regions via context-aware region clustering (Sec. 3.1).
- Establish discovered top-down segmentation cues by selecting ensemble of exemplars from each cluster (Sec. 3.2).
- Refine each region with graph cuts by encoding discovered top-down cues, background preference via entropy, and smoothness constraints (Sec. 3.3).
- Repeat the discovery process using the refined segmentations as input (Sec. 3.4).

Output: Segmented images and k discovered objects.

Algorithm 2: Summary of the Collect-Cut method.

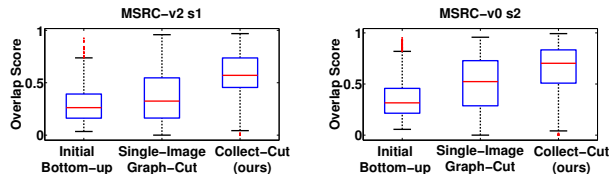
We replace the context-aware clustering from Sec. 3.1 with an appearance-based algorithm, and swap out the entropy-based background weighting with a distance-based background weighting. We use the method of [22], which uses Latent Dirichlet Allocation to discover visual topics among the regions. To compose our ensemble model, we take the r instances (from unique images) with the smallest KL-divergence to the given topic.

When comparing a superpixel to $M_{bg}(I)$, we replace entropy $H(s_{I_k})$ with a weighting $J(s_{I_k})$ that depends on the spatial distance between the centroids of region k and the initial seed region. The idea is that, in absence of any knowledge of familiar categories, we should prefer regions that are far away from the seed region to be background. Specifically, we place a Gaussian centered at the seed region center $\boldsymbol{x}_c$, with σ equal to the mean of the region's width and height: $g(\boldsymbol{x}) = \exp(-\|\boldsymbol{x} - \boldsymbol{x}_c\|^2/(2\sigma^2))$. Then, we compute a single weight $J(s_{I_k})$ by averaging $g(\boldsymbol{x})$ within that region.

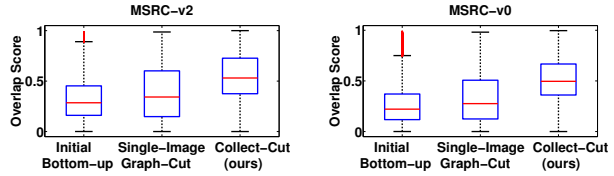
Discussion: We choose a binary-label formulation to refine each image region, instead of a multiple-label formulation to compute a single image segmentation. Aside from computational cost, the advantage of the binary formulation is its robustness to the initial discovery procedure; if the number of clusters found is greater than the true number of categories, the labeling would risk oversegmenting an object, since two or more ensembles could represent the same category. Similarly, if the clusters found are fewer than the true categories, some categories would not be represented, leading to incorrect segmentations. With a binary formulation, we only enforce that each instance resembles its ensemble model instances and differs from its own image's background regions.

3.4. Iterating the Discovery Process

Once we refine all the segmentations, we remove the cluster associations, and compute new appearance features for the refined regions. Then we provide the resulting descriptors as input to the discovery algorithm. Having improved the segmentation boundaries with the collective



(a) Semi-supervised via familiar-object context



(b) Fully unsupervised

Figure 3. Segmentation overlap scores for both datasets, when tested (a) with the context of familiar objects or (b) without. Higher values are better—a score of 1 would mean 100% pixel-for-pixel agreement with ground truth object segmentation. By collectively segmenting the images, our method's results (right box-plots) are substantially better aligned with the true object boundaries, as compared to both the initial bottom-up multiple segmentations (left box-plots), as well as a graph cuts baseline that can use only cues from a single image at once (middle box-plots).

graph-cut, the discovery procedure can (potentially) form better groups than were possible at the previous iteration. Alg. 2 summarizes the steps of our method.

4. Results

In this section, we evaluate our method's segmentation performance and analyze how it affects discovery accuracy.

Datasets: We use the MSRC v0 and v2 datasets.[3] Both consist of natural scenes containing instances from 21 generic object categories. The MSRC-v0 contains 4,325 images; we used Mechanical Turk to obtain pixel-level ground-truth on all images with multiple objects (3,457 total). The MSRC-v2 contains 591 images, and high-quality pixel-level ground-truth[4]. These datasets seem most appropriate given that the images contain multiple repeating objects from generic categories, and allow segmentations to be scored at the pixel-level.

When evaluating the semi-supervised form of our method, N previously learned categories are used as context during the region clustering. To demonstrate the stability of the results with respect to which categories are familiar, we consider two known/unknown splits for each dataset (see Table 1 for each split's unknown classes). When evaluating the method without using familiar objects as context, all 21 classes are considered as unknown.

Implementation Details: We use Normalized Cuts [24] to generate multiple segmentations for each image; we vary the number of segments from 3 to 12. We obtain contour estimates with the gPb detector [15]. To represent each segment's appearance, we compute texton and color histograms. We generate the texton codebook with k-means on filter responses from 18 bar and 18 edge filters (6 orientations and 3 scales each), 1 Gaussian, and 1 LoG, with $k = 400$ texton codewords. We use Lab color space, and 23 bins per channel.

We fixed $\beta = 10$ for the smoothness term after examining a few image examples (we did not optimize the value). When including previously learned categories for context-aware region clustering, we train SVM classifiers on texton, color, and pHOG histograms (see [13]). We set $r = 5, 10$ for the MSRC-v2 and v0, respectively. These are chosen arbitrarily based on the relative dataset sizes. In discovery experiments, we weight the appearance features four-times as much as the context features for the context-aware clustering, since we expect the refined segments to improve appearance support more than spatial context. We average all clustering results over five runs.

As usual in segmentation or discovery, the model selection task (choosing k, the number of objects to be discovered) is difficult without prior knowledge. In our initial rounds of experiments, we tested the segmentation as a function of k. We found that our method's gains relative to the baselines were very stable as the value of k varies; thus, due to space restrictions, we present here results for a single value, $k = 30$. (Fig. 1 in the supplementary file illustrates the consistency of our results when varying k from 1 to 35.)

	building	tree	cow	airplane	bicycle
v2-$s1$	.31 (116%)	.28 (89%)	.37 (114%)	.23 (86%)	.35 (123%)
	cow	sheep	airplane	car	bicycle
v0-$s1$	.30 (65%)	.28 (60%)	.13 (36%)	.23 (65%)	.27 (95%)
	tree	sheep	car	bicycle	sign
v2-$s2$	.33 (109%)	.38 (127%)	.30 (105%)	.28 (100%)	.26 (90%)
	tree	sheep	chimney	door	window
v0-$s2$	.41 (145%)	.29 (62%)	.19 (43%)	.21 (44%)	.29 (81%)

Table 1. Mean overlap score *improvement* per category, for each split (s1 and s2) of the two datasets (MSRC v0 and v2). Gains are measured between each initial bottom-up segment and our refinement; both the absolute and percentage increases are shown. Our collectively segmented regions are more accurate for all categories, including those with heterogeneous appearance (cars, bicycles), which are most challenging.

4.1. Object Segmentation Accuracy

We first evaluate our method's segmentation results. To quantify accuracy, we use the pixel-level segmentation *overlap score*, OS. The quality of segmented region R with respect to the ground-truth object segmentation GT is measured as: $OS = \frac{|GT \cap R|}{|GT \cup R|}$, where we take as GT the full object region associated with region R's majority pixel label. We only score segments that initially belong to an unknown category, to focus our evaluation on the contribution of our full model (i.e., using familiarity estimates). This amounts to a total of 1,921, 1,202, 1,018, and 572 regions for the v0 $s1$, v0 $s2$, v2 $s1$, and v2 $s2$, respectively.

We compare our **Collect-Cut** method against two base-

[3] http://research.microsoft.com/en-us/projects/objectclassrecognition/

[4] http://www.cs.cmu.edu/∼tmalisie/projects/bmvc07/ (see [16])

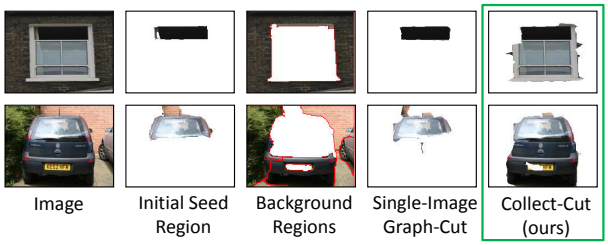


Figure 4. Two illustrative results comparing our Collect-Cut to the single-image graph-cuts baseline. If the initial seed region captures only a single part of a multi-part object (i.e., heterogeneous appearance), a method restricted to using only the single image for segmentation may fail. In contrast, by integrating the ensemble of discovered shared cues found in the collection, our approach can more fully recover the object's segmentation.

lines: (1) the original bottom-up segmentation provided by the NCut multiple segmentations (denoted **Initial Bottom-up**), and (2) a graph-cuts segmentation that uses only information in the single originating image to assign costs for labeling superpixels (denoted **Single-Image Graph-Cut**). Specifically, for the foreground model the single-image method uses only the initial seed region, and for the background model it uses the outermost regions along the image boundaries from the same image. We modeled this baseline after a model devised in [30], and it represents the best we could do if trying to refine the segmentation independently for each image.

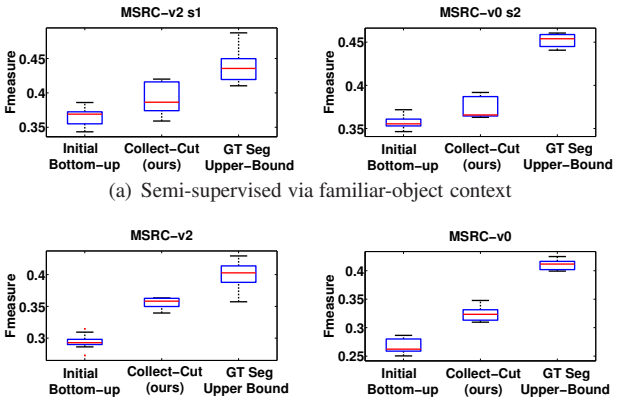
Fig. 3 shows the results. We evaluate our method both when (a) using the familiar categories during context-aware region clustering, and (b) using no familiar categories. In either case, note that *no* supervision is used for the regions/categories that are scored.[5] The low initial scores for the bottom-up regions confirms the well-known difficulty in generating object-level segments while relying only on low-level feature similarity (color, texture). The single-image baseline improves over this, exploiting the contrasts between the seed and its surrounding superpixels, as well as the prior to prefer outer regions as background. However, by leveraging the shared structure in the *collection* of images, our method produces significantly better segmentations than either baseline.

We noticed that the single-image baseline has particular difficulty in refining segmentations for objects with heterogeneous appearance (see Fig. 4).

Table 1 shows our method's average improvements for each of the unknown categories. Overall, there is consistent and significant gains for all categories when compared to the original bottom-up regions. The smaller improvements (e.g., airplane: 36%) seem to occur when the initial clusters are less homogeneous, leading to weaker ensembles.

Fig. 6 shows representative (good and bad) qualitative segmentation examples, where we compare against the *best* segment from the initial pool of multiple-segmentations.

[5]Results for the other two splits are similar; see supp. file.



(a) Semi-supervised via familiar-object context

(b) Fully unsupervised

Figure 5. Impact of collective segmentation on discovery accuracy, as evaluated by the F-measure (higher values are better). For discovery, we plug in both (a) our context-aware clustering algorithm [13], and (b) an appearance-only discovery method [22]. In both cases, using our Collect-Cut algorithm to refine the original bottom-up segments yields more accurate grouping.

Fig. 7 shows good multi-object segmentation examples, where we aggregate our method's refined object regions into a single image-level segmentation.

4.2. Category Discovery Accuracy

We next analyze the extent to which segmentation refinement improves category discovery. We use the *F-measure* to quantify clustering accuracy: $F = \frac{2 \cdot P \cdot R}{P+R}$, where P denotes precision and R denotes recall. This scoring reflects the coherency (precision) of the clusters, while taking into account the recall of the same-category instances. To score an arbitrary segment, we consider its ground truth label to be that which the majority of its pixels belong to.

Fig. 5 shows the results. We compare three variants: (1) running discovery with the initial bottom-up multiple segmentations pool as input, (2) running discovery with our method's results as input, and (3) running discovery with the ground truth object segments, which provides an upper bound on accuracy. Our method yields a significant gain in clustering accuracy over the initial segmentations. This can be attributed to the fact that the spatial extent of the refined regions more closely matches that of the true objects, thereby allowing more complete appearance features to be extracted per region, and then clustered. The upper bound on accuracy in this experiment is imperfect—showing the limits of clustering multiple generic object categories.

Note that segment refinement can cause changes to an instance's majority label, though approximately 80% of the instances retain their initial labels. These changes mean that the absolute differences between the F-measures are not one-to-one. However, the absolute values and distributions are themselves meaningful.

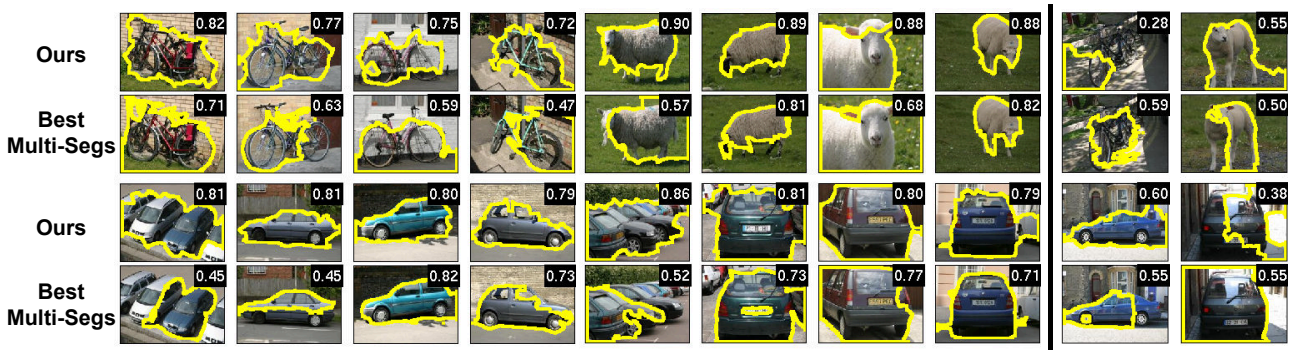


Figure 6. Qualitative comparison: our results vs. the best corresponding segment available in the pool of multiple-segmentations. Numbers above denote overlap scores. The **first 8 columns** are examples where our method performs well, extracting the true object boundaries much more closely than the bottom-up segmentation can. We stress that the "best multi-segs" shown are picked using the ground truth, meaning there is no better region for the object available in the pool of segments; thus, it should be viewed as a generous upper bound on the quality of the regions we can get for the baseline. The **last 2 columns** show failure cases for our method. It does not perform as well for images where the multiple objects have very similar color/texture, or when the ensembles are too noisy. (Best viewed on pdf.)

Figure 7. Examples of high quality multi-object segmentation results. We aggregate our refined segmentations into a single segmentation of the image.

5. Conclusions

Overall our results illustrate the proposed method's advantage of discovering shared structure in the unlabeled set of images when computing segmentations. We have also demonstrated the value of (optionally) introducing knowledge about previously learned categories. The results indicate that when some recurring objects are present in the image collection, exploiting their repetition leads to high quality segmentations that better capture full objects.

Acknowledgements: This research is supported in part by NSF CAREER 0747356, NSF EIA-0303609, Texas HECB 003658-01-40-2007, the Luce Foundation, and Google and Microsoft Research.

References

[1] H. Arora, N. Loeff, D. Forsyth, and N. Ahuja. Unsupervised Segmentation of Objects using Efficient Learning. In *CVPR*, 2007.
[2] E. Borenstein, E. Sharon, and S. Ullman. Combining Top-down and Bottom-up Segmentation. In *CVPR*, 2004.
[3] Y. Boykov, O. Veksler, and R. Zabih. Efficient Approximate Energy Minimization via Graph Cuts. *TPAMI*, 20(12):1222–1239, 2001.
[4] O. Chum, M. Perdoch, and J. Matas. Geometric min-Hashing: Finding a (Thick) Needle in a Haystack. In *CVPR*, 2009.
[5] M. Fritz and B. Schiele. Decomposition, Discovery and Detection of Visual Categories Using Topic Models. In *CVPR*, 2008.
[6] A. Gallagher and T. Chen. Clothing Cosegmentation for Recognizing People. In *CVPR*, 2008.
[7] S. Gould, R. Fulton, and D. Koller. Decomposing a Scene into Geometric and Semantically Consistent Regions. In *ICCV*, 2009.
[8] G. Kim, C. Faloutsos, and M. Hebert. Unsupervised Modeling of Object Categories Using Link Analysis Techniques. In *CVPR*, 2008.
[9] P. Kohli, L. Ladicky, and P. Torr. Robust Higher Order Potentials for Enforcing Label Consistency. In *CVPR*, 2008.
[10] V. Kolmogorov and R. Zabih. What Energy Functions can be Minimized via Graph Cuts? *TPAMI*, 26(2):147–159, 2004.
[11] M. Kumar, P. Torr, and A. Zisserman. OBJCUT. In *CVPR*, 2005.
[12] Y. J. Lee and K. Grauman. Foreground Focus: Unsupervised Learning From Partially Matching Images. *IJCV*, 85, 2009.
[13] Y. J. Lee and K. Grauman. Object-Graphs for Context-Aware Category Discovery. In *CVPR*, 2010.
[14] D. Liu and T. Chen. Background Cutout with Automatic Object Discovery. In *ICIP*, 2007.
[15] M. Maire, P. Arbelaez, C. Fowlkes, and J. Malik. Using Contours to Detect and Localize Junctions in Natural Images. In *CVPR*, 2008.
[16] T. Malisiewicz and A. A. Efros. Improving Spatial Support for Objects via Multiple Segmentations. In *BMVC*, 2007.
[17] L. Mukherjee, V. Singh, and C. R. Dyer. Half-Integrality Based Algorithms for Cosegmentation of Images. In *CVPR*, 2009.
[18] J. Philbin and A. Zisserman. Object Mining using a Matching Graph on Very Large Image Collections. In *ICVGIP*, 2008.
[19] T. Quack, V. Ferrari, B. Leibe, and L. V. Gool. Efficient Mining of Frequent and Distinctive Feature Configurations. In *ICCV*, 2007.
[20] C. Rother, V. Kolmogorov, and A. Blake. Grabcut: Interactive Foreground Extraction using Iterated Graph Cuts. In *ACM Transactions on Graphics*, 2004.
[21] C. Rother, V. Kolmogorov, T. Minka, and A. Blake. Cosegmentation of Image Pairs by Histogram Matching - Incorporating a Global Constraint into MRFs. In *CVPR*, 2006.
[22] B. Russell, A. Efros, J. Sivic, W. Freeman, and A. Zisserman. Using Multiple Segmentations to Discover Objects and their Extent in Image Collections. In *CVPR*, 2006.
[23] B. Russell, A. Efros, J. Sivic, W. Freeman, and A. Zisserman. Segmenting Scenes by Matching Image Composites. In *NIPS*, 2009.
[24] J. Shi and J. Malik. Normalized Cuts and Image Segmentation. *TPAMI*, 22(8):888–905, August 2000.
[25] J. Sivic, B. Russell, A. Efros, A. Zisserman, and W. Freeman. Discovering Object Categories in Image Collections. In *ICCV*, 2005.
[26] E. Sudderth and M. Jordan. Shared Segmentation of Natural Scenes Using Dependent Pitman-Yor Processes. In *NIPS*, 2008.
[27] S. Todorovic and N. Ahuja. Extracting Subimages of an Unknown Category from a Set of Images. In *CVPR*, 2006.
[28] T. Tuytelaars, C. Lampert, M. Blaschko, and W. Buntine. Unsupervised Object Discovery: A Comparison. *IJCV*, 88(2):284–302, 2010.
[29] J. Winn and N. Jojic. LOCUS: Learning Object Classes with Unsupervised Segmentation. In *ICCV*, 2005.
[30] S. Yu and J. Shi. Segmentation Given Partial Grouping Constraints. *TPAMI*, 26(2):173–183, 2004.