Discovering Object Instances from Scenes of Daily Living

Hongwen Kang, Martial Hebert, and Takeo Kanade
School of Computer Science
Carnegie Mellon University
{hongwenk,hebert,tk}@cs.cmu.edu

Abstract

We propose an approach to identify and segment objects from scenes that a person (or robot) encounters in Activities of Daily Living (ADL). Images collected in those cluttered scenes contain multiple objects. Each image provides only a partial, possibly very different view of each object. An object instance discovery program must be able to link pieces of visual information from multiple images and extract the consistent patterns.

Most papers on unsupervised discovery of object models are concerned with object categories. In contrast, this paper aims at identifying and extracting regions corresponding to specific object instances, e.g., two different laptops in the laptop category. By focusing on specific instances, we enforce explicit constraints on geometric consistency (such as scale, orientation), and appearance consistency (such as color, texture and shape). Using multiple segmentations as the basic building block, our program processes a noisy "soup" of segments and extracts object models as groups of mutually consistent segments.

Our approach was tested on three different types of image sets: two from indoor ADL environments and one from Flickr.com. The results demonstrate robustness of our program to severe clutter, occlusion, changes of viewpoint and interference from irrelevant images. Our approach achieves significant improvement over with two existing methods.

1. Introduction

We tackle the problem of discovering object instances from ADLs (Activities of Daily Living) [17]. Imagine a personal robotic assistant [1] that accompanies a user to different scenes during her daily activities, such as kitchen, living room and office space. Every time the user operates in a scene, the robotic assistant takes a few pictures of the environment without the user explicitly showing the objects to the robot. After one or two weeks of data gathering, we would like the robotic assistant to automatically discover and model objects from the images it has collected.

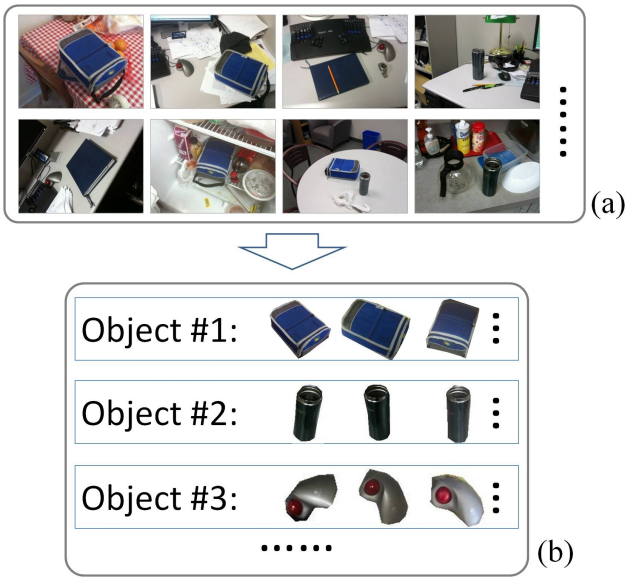


Figure 1. **From (a) a set of unlabeled images in ADLs, our program discovers (b) object instances that appear repeatedly.**

1.1. Example problem

As an example of such scenarios, a user collected 175 images while performing daily-living activities (e.g., Figure 1(a)). These images capture scenes of two offices, two kitchens, and one living room, taken when she enters or leaves a room.

From this ADL dataset, our goal is to automatically 1) identify distinct object instances, e.g., we need to distinguish particular mugs that a person uses at home or office; and, 2) recover the spatial extent of the objects in the images (Figure 1(b)).

1.2. Background and related work

Many approaches have been developed for unsupervised object discovery [3, 6, 10, 16, 33, 27, 29, 36, 39, 44]. An extensive study and comparison of the state-of-the-art techniques is reported in [41]. These previous papers aim at discovering/grouping visually different object instances ac-

cording to some definition of categorization, e.g., motorbikes of different makes and models, etc [3, 33, 27]. Because of the large intra-category variance, they resorted to use generic techniques that are less discriminative, such as topic models [33, 36]. Despite substantial efforts, these approaches can only identify limited types of objects where the *inter*-category variance outweighs the *intra*-category variance, e.g., faces, cars, airplanes and motorbikes [3, 10, 16, 33, 27, 36, 39], or can only handle one class of objects at a time [3, 6, 10, 39]. Because of their lack of discriminative power, neither will they be suitable for our task that requires distinguishing specific object instances.

Another style of object discovery uses videos [20, 34]. However, existing approaches largely rely on the continuous observation of the same object at one location, and do not take advantage of the fact that the same object instance can appear at multiple locations. In this paper, we use images collected from monocular cameras, which is different from approaches using stereo (e.g., [38]) or 3D range data (e.g., [25, 40]). Our problem can also be considered as a large co-segmentation problem that segments common objects from a set of images. However, the current co-segmentation methods [11, 15, 31] can only handle one object at a time and their inputs are a small set of images that are known to contain the same object. [4] extended the co-segmentation idea to extracting identical objects. However, because this approach relies on dense pairwise feature matching, it is more suitable for the scenario of finding identical texture-rich objects from a smaller set of images, similar to [21, 29, 30, 32, 43].

Recently, we have seen significant progress in understanding object instances. An "object" instance can be either a "model image" that captures an object with a clean background [22, 9], or a region of interest (ROI) that a user selects from an image [37]. Many techniques applied in category-level applications are also applied to tasks that deal with instances, but substantial changes to these techniques are required. For example, in a category scenario, small codebooks can be used in bag-of-words models [37] to compensate the intra-category variance; however, in an instance scenario, significantly larger codebooks (e.g., *millions* of visual-words) have been used to improve the discriminative power [26, 28], i.e., constraining the allowed changes in object appearance. Further, geometric verification can be conducted to enforce the geometric consistency [14, 28]. These approaches have been applied to image clustering [5]. However, they are not suitable for our problem. Because many of our input images are captured in the same environment (similar rooms), large portions of the features from the background will dominate the clustering. Fundamentally, this is because these approaches use a "loose" bag of features instead of a concrete definition for the spatial extent of an object.

1.3. Proposed approach

The object discovery problem requires the correct identification and segmentation of each object instance. We propose an approach that uses bottom-up image segmentation as the basic building block to attack the identification and segmentation problems simultaneously (Figure 2). Image segmentation is typically noisy. Among hundreds of segments, only a few might belong to meaningful objects (e.g., Figure 2(a)), which is impossible to tell from a single image. However, from multiple images, the segments that belong to the same object will display stronger correlation than the ones that belong to different objects or backgrounds. For specific instances, we measure these correlations explicitly as geometric consistency (scale, orientation), and appearance consistency (color, texture and shape).

Based on this observation, our approach processes a noisy "soup" of segments [12, 33] and extracts object candidates as groups of mutually consistent segments (Figure 2(b)). We develop a procedure that iteratively groups and refines segments (Figure 2(c)). We also use the co-occurring information between object segments to generate models of objects with high complexity (Figure 2(d)). Compared with two existing methods [16, 33], our approach demonstrates significant advantage in attacking this unique problem.

2. Algorithms

2.1. Generating a pool of object segments using multiple segmentations

The goal of image segmentation is to extract objects, each of which represented by exactly one image region. However, in real images, many of the regions generated by typical segmentation algorithms fall short of capturing meaningful objects (e.g., Figure 2(a)). Instead of relying on one segmentation to segment all the objects correctly, [12] and [33] propose to vary the parameters of the segmentation program and combine different segmentation results together. Improved recall of objects was obtained through this process, because for most objects there exists at least one combination that segments some of them reasonably well. To extend this further, we propose to combine multiple segmentation programs with multiple parameter settings to generate a more diverse pool of object segments. In this paper, we combine the segmentations of [7] and [24]. For the ADL dataset, we start with over 35000 segments. After merging overlapping segments and filtering out small segments, on average 25 segments are retained per image (e.g., Figure 2(a)), 4390 segments in total.

2.2. Extracting groups of mutually consistent segments

Using a larger segment pool increases the recall of objects, but it also substantially increases the number of segments that do not belong to actual objects. We observed

that, in multiple images, the regions that correctly segment the same object are consistent up to certain transformations, e.g., scale, rotation, while the segments that belong to different objects or backgrounds do not display the same level of consistency (e.g., Figure 1(a)). Therefore, we can identify object candidates by dividing the pool of segments into multiple groups, each of which contains mutually consistent segments. Each group will be treated as an object instance candidate.

To achieve such grouping of the segments, we: 1) compute the consistency of each pair of segments; and, 2) extract groups of segments that are mutually consistent.

2.2.1 Computing pairwise segment consistency

Given two segments, we want to calculate a measure of pairwise consistency to quantify how likely the segments belong to the same object. There are two challenges in deriving such a consistency measure.

First, the same object can have different appearance in multiple images because of different placements and viewpoints; our consistency measure should be robust to such changes. We found that the following three features yield a good balance between robustness and discriminative power:

- **Color:** We calculate the RGB color histogram (H_s) and the mean RGB color values (M_s) for a segment s.
- **Texture:** We extract SIFT [22] features for the whole image and represent s using the quantized SIFT features (V_s) located inside the segment. We use a bag-of-words (BoW) representation [28, 37].
- **Shape**: We choose the shape descriptor and matching algorithm proposed in [2]. The shape matching metric measures how likely the two segments are equivalent, up to a similarity transformation.

Second, everyday objects have heterogeneous appearance: some objects with strong local texture are well-suited for descriptors like SIFT, while others are completely featureless. For objects with sparse features, [8] proposed an approach that assumes the existence of small texture-rich regions. However, for the texture-less objects that we handle, this assumption is rarely valid. Instead, it is more appropriate to represent these objects with color and shape descriptors. To cope with such heterogeneity, we develop a consistency measure that adapts to the appearance of different segments. For each segment s, we classify it as either texture-rich or texture-less based on the number of SIFT features detected in it. For different types of segments, we define different consistency measures. Our adaptive consistency measure consists of two stages. The first stage uses color and texture features. Given a segment s, we compute

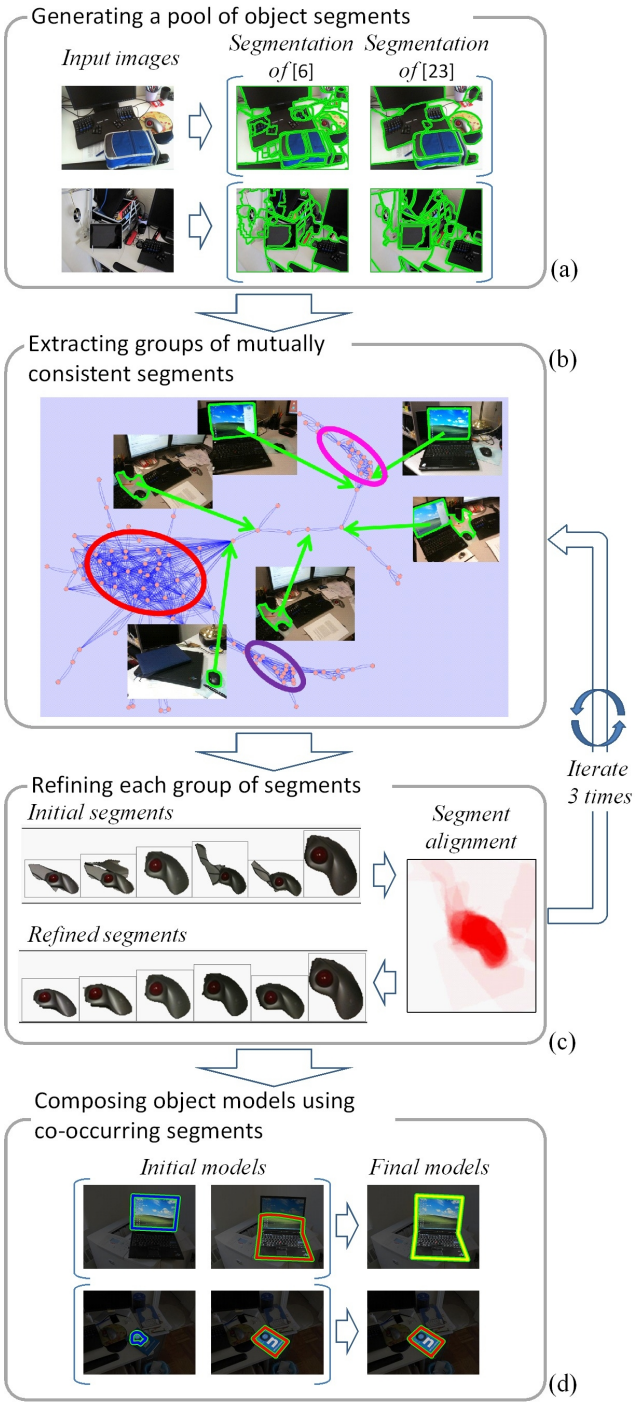


Figure 2. **Proposed approach for object instance discovery. (Best viewed in color.)**

its consistency with another segment t as:

$$c_1(s,t) = \begin{cases} 1 - \alpha d(H_s, H_t) \\ -\beta d(V_s, V_t) \\ 1 - d(M_s, M_t), \end{cases} \begin{array}{l} , \quad s \text{ is texture-rich} \\ \\ \quad\quad s \text{ is texture-less} \end{array} , \tag{1}$$

where $d(\cdot,\cdot)$ is the Euclidean distance. We learn α and β

using 1000 pairs of segments. These parameters are learned once and fixed across all the experiments.

$c_1(\cdot,\cdot)$ can be computed efficiently[1]. Since our ultimate goal is to find the mutually consistent segments, we use c_1 as the first step to quickly eliminate a large number of "obviously" non-similar segments. The most similar segments ($\mathbf{T}_1(s) = \{t|c_1(s,t) > C_1, C_1 = 0.5\}$) are passed to the second step of shape matching. We measure the shape consistency $c_2(s,t)$ using the metric $d_s(s,t)$ from [2]:

$$c_2(s,t) = 1 - d_s(s,t), \quad t \in \mathbf{T}_1(s), \tag{2}$$

and we use an adaptive threshold to determine if two segments' shapes are similar:

$$C_2 = \begin{cases} 0.5 \text{ (suggested in [2])}, & s \text{ is texture-rich} \\ 0.75, & s \text{ is texture-less} \end{cases}. \tag{3}$$

The performance of the whole algorithm is robust to different choices of these thresholds.

We represent the pairwise consistency of segments using a graph structure (Figure 2(b)), in which each segment is a node and the edges link pairs of segments (s,t) that are consistent, i.e., $c_1(s,t) > C_1$ and $c_2(s,t) > C_2$. For each segment s in the ADL dataset, on average, about 20 segments t satisfy $c_1(s,t) > C_1$. In the final consistency graph, 709 ($\sim 16\%$) segments are connected with other segments and their average degree is 3.1. This means that our pairwise consistency measure is filtering out most of the segments that are not likely to belong to any objects. This makes the consistency graph sparse (density$\approx 1.14e-4$), a desirable property for discovering objects from large datasets.

2.2.2 Grouping mutually consistent segments

If the segment consistency measurement is perfect, then only segments of the same object should be connected. Unfortunately, matching individual pairs of segments is prone to errors because: 1) the bottom-up segmentation results are imperfect, e.g., two segments of different objects might intersect (Figure 2(b)); and, 2) different objects might look similar from some special viewpoints, e.g., a cup might appear similar to a computer mouse. On the other hand, even if individual matches are imperfect, segments belonging to the same object are more consistently connected as a group. A similar phenomenon has been observed in studies of collaboration networks [42], where groups of people are mutually connected through their email exchanges. By analogy, we propose to extract segments of the same object that form mutually consistent "communities".

In this paper, we apply a graph-based method [42] to this community discovery task. We select this method because it does not require knowing the desired number of groups. In scenarios where the group numbers are known, other methods [23, 35] should also be applicable.

For the ADL dataset, we extracted 180 groups that contain at least 2 segments, among which 21 groups contain more than 5 segments, and the largest group contains 16 segments. Visual inspection confirmed that the larger groups are more likely to belong to the same objects. In practice, we use the group size as a criterion to select the most promising object candidates.

2.3. Iteratively grouping and refining segments

After groups of mutually consistent segments are extracted, we can now use the grouping information to refine the segmentation. A straightforward approach, e.g., [18], is to generate two ensemble models, one using the pixels from the segments, and the other using those from the background and to re-segment the images based on these ensemble foreground/background models. However, this approach is sensitive to errors in the initial segmentation.

Instead, for object instances, we can explicitly model the geometric transformations between object segments and enforce segmentation constraints based on the pixel correspondences. For a group of segments, we warp each pair of aligned segments using the transformation estimated during the segment shape matching step [2] and we detect the regions that the segments intersect. We initialize the image regions that all the segments intersect as "foreground", the regions that none of the segments intersect as "background", and all the other regions as "unknown". Using the initial "foreground"/"background" regions, we can calculate how likely a pixel belongs to the object or the background. We also enforce that corresponding pixels in different images should have the same identity.

We formulate the segment refining problem as a graph-cut problem. We construct a graph using all the pixels of the images from which the segments originated. Each pixel is treated as a node i, and the union of all the pixels is V. An edge is created if two pixels are in a 4-neighborhood (ϵ) of the same image, or they are corresponding pixels (Π) of two images. The segmentation minimizes the energy function:

$$E(X) = \sum_{i \in V} E_u(x_i) + \lambda \sum_{(i,j)\in\epsilon} E_p(x_i, x_j) + \gamma \sum_{(k,l)\in\Pi} E_c(x_k, x_l), \tag{4}$$

where X is the labeling of all the pixels. $E_u(x_i)$ is the unary energy based on the similarity of i to the "foreground"/"background" regions, and $E_p(x_i, x_j)$ is the image-wise smoothness energy. The definitions of these two energies are the same as in [19]. We propose to use a new energy term, $E_c(x_k, x_l)$, to penalize labeling two corresponding pixels differently:

$$E_c(x_k, x_l)|_{(k,l)\in\Pi} = \begin{cases} 0, & x_k = x_l \\ 1, & x_k \neq x_l \end{cases}. \tag{5}$$

[1] We found that color and BoW matching is about two to three orders of magnitude faster than shape matching.

Figure 3. **Example of a complex object. Our program compose the final model using multiple parts that co-occur. (We darkened the background for visualization. For reference, we include a picture of the object on the right. Best viewed in color)**

Some representative examples comparing the segmentation results before and after the refinement are shown in Figure 2(c). In this case the initial pixel-wise precision/recall are $72\%/98\%$, and after refinement they are $93\%/98\%$, a 20% improvement. We replace the initial segments with the refined ones and iterate this grouping/refining process 3 times.

2.4. Composing object models using co-occurring segments

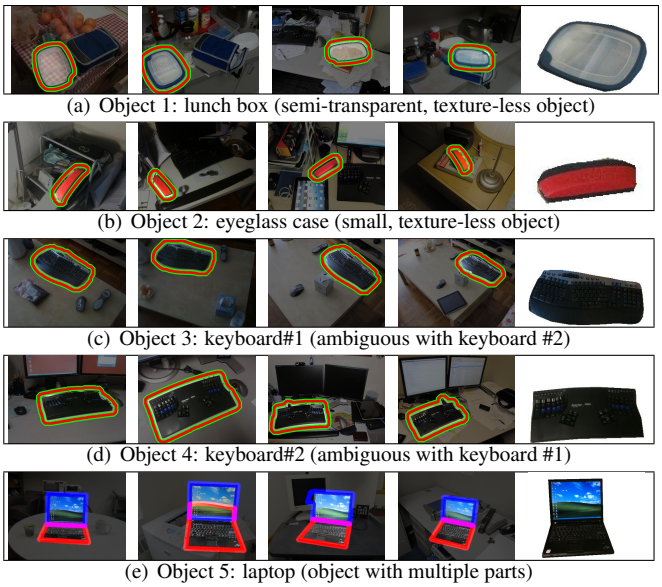
The extracted groups of segments are good enough for representing most of the object instances, especially the ones with a single part. However, some complex objects can be fragmented as several groups. This could happen if: 1) the object has complex textures, e.g., a book with large characters on its cover; and, 2) objects are made of multiple parts, e.g., an opened laptop, whose screen and keyboard are divided into two groups of segments (Figure 2(d)). Our program solves this fragmentation issue by composing objects parts that appear coherently in multiple images.

In each image, we identify segments that co-occur and are adjacent. If a significant portion (80%) of two segment groups co-occur, we compose new segments using the co-occurring segments and generate a new group using the composed segments. Figure 3 shows a multi-part chair model that our program extracted from a set of images independently collected, by composing different parts, i.e., back, seat and two handles. Due to large errors in the initial segmentation, our program did not discover the legs.

3. Result and analysis on the ADL dataset

From the ADL dataset, we first extract 4390 segments, from which our program produces 113 segment groups, each containing at least 2 segments, among which 18 groups contain more than 5 segments each. Some examples of objects that the program discovers were shown in Figure 2(d). Figure 4 shows some more discovered objects, including semi-transparent, texture-less objects, ambiguous objects, and objects made of multiple parts.

To quantitatively evaluate our program, we ask the user to label the objects that appear in more than 5 images, move relative to the environment at least once, and are larger than 1600 pixels (i.e., about 1% of the image size). 16 objects satisfy these criteria. We apply a hit-miss criterion used in object detection to decide if a segment is correct, i.e., whether the segment overlaps an object over 50%. We calculate the group purity of each segment group as the portion of correct segments that belong to the same object [41]. If there are multiple objects in the group, we compute the group purity as the percentage of the correct segments of the most frequent object. To qualify as a correct object candidate, we require that a segment group's purity be over 80%. We found that the larger the group, the more likely the seg-



(a) Object 1: lunch box (semi-transparent, texture-less object)

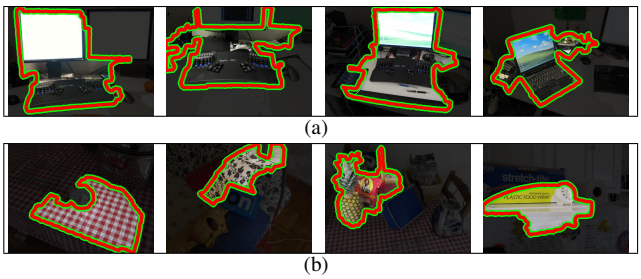
(b) Object 2: eyeglass case (small, texture-less object)

(c) Object 3: keyboard#1 (ambiguous with keyboard #2)

(d) Object 4: keyboard#2 (ambiguous with keyboard #1)

(e) Object 5: laptop (object with multiple parts)

Figure 4. **Example of objects discovered by our program from the ADL object dataset. The backgrounds darkened for visualization.**



(a)

(b)

Figure 5. **Examples of objects discovered by the modified baseline system ("Russell, et.al. (modified)") [33].**

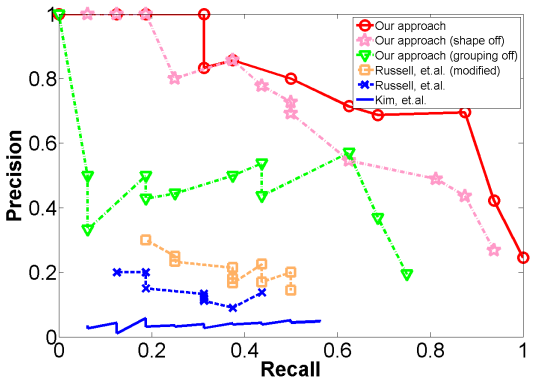


Figure 6. **Comparison of our approach in 3 different configurations and [16, 33] on the ADL dataset.**

ments belong to the same object. Therefore we use the size of a group as the threshold to choose object candidates. We calculate object-wise precision/recall with respect to each choice of the group size threshold. We define precision as the number of correct object candidates divided by the total number of groups, and recall as the number of unique objects discovered by the program divided by the total 16 groundtruth objects. Figure 6 shows the precision/recall profile of our program. The group size threshold increases from right to left.

We also want to understand the effect of different components on the discovery process. For example, Figure 6 shows the precision/recall profile of our program when we do not use shape consistency, i.e., using color and texture only ("Our approach (shape off)"), and if we disable the segment grouping component, i.e., treating each connected component as a group ("Our approach (grouping off)"). Comparing "Our approach (shape off)" with the full system, i.e., "Our approach", we see that shape information helps when combined with color and texture features. However, our further experiments show that using shape alone does not discover any meaningful objects at all. This is because, unlike the objects in [27] that have distinctive contours (e.g., swans, horses), in an ADL environment, many objects have similar shapes (e.g., rectangle, cylinder). We also notice that grouping is vital, because pairwise segment matching is prone to errors, and groups of consistent segments can be linked by these erroneous pairwise links.

For comparing the results of our method with previous work, two existing methods [16, 33] were applied on the same dataset. These methods are the closest to our method, because [33] ("Russell, et.al.") uses multiple segmentations as object hypothesis and [16] ("Kim, et.al.") uses graph analysis to find mutually coherent feature correspondences.

We used programs provided by the authors of [16, 33]. [33] uses normalized cut to generate multiple segmentations. For a consistent comparison, we also compare with a modified system ("Russell, et.al. (modified)") that uses our segmentation results as the input to their program. [16] was tuned to the Caltech101 dataset in which each image contains only one object. Since each of ADL images contains multiple objects, when applied directly, [16] does not discover any meaningful object models. Instead, we use our segments as the input, each treated as an image, and we discard links between segments from the same image.

Figure 5 shows examples of the objects that the "Russell, et.al. (modified)" system discovers. Because this baseline system relies solely on a BoW model, it cannot distinguish objects with similar texture patterns but different shapes (Figure 5(a), Figure 5(b)). We also measure the quantitative performance of the baseline systems and compare them to our method in Figure 6. For the baseline systems, we vary the number of topics/clusters ($20 - 300$); increasing the number generates more fragmented small clusters, and as a result precision generally increases. In Russell's systems, we keep the top 5 segments that have the highest likelihood in each topic. In Kim's system, we keep 20% of the segments with the highest Page-rank and discard clusters that contain less than 5 segments. The quantitative comparison shows that our approach significantly outperforms the baseline systems. Because Kim's method relies heavily on distinctive features, it performs badly on the ADL dataset.

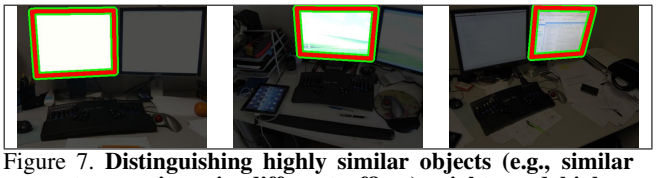


Figure 7. **Distinguishing highly similar objects (e.g., similar computer monitors in different offices) might need higher-level, e.g., contextual, information.**

Figure 7 shows a typical failure case of our program, where computer monitors at different offices are grouped together. In this case, contextual information might be useful to distinguish each of them.

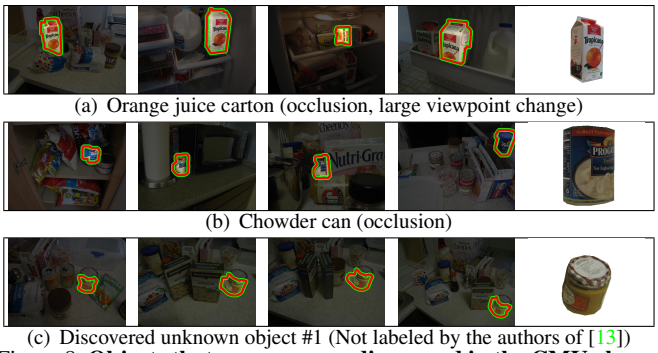
4. Further tests on other datasets

In addition to the ADL dataset, we evaluate our program on two other datasets: the CMU object dataset that includes a large amount of clutter, occlusion and viewpoint changes; and a set of Flikr images that include a large amount of interference from irrelevant images.

4.1. Cluttered environment, occlusion and viewpoint changes (CMU object dataset)

The CMU object dataset [13] consists of objects used in a kitchen environment with severe clutter, occlusion and viewpoint changes. We extract about 10000 segments from 500 images and the average degree of the consistency graph is 2.7, the density of the graph is $2.62e-4$. Figure 8 shows some objects that our program discovers. Figure 10 shows the precision/recall profile of our program. Since only 10 of the objects in [13] were labeled, we calculate the recall score as the percentage of the labeled objects that are successfully identified. We manually inspect the result segment groups, and count how many of them satisfy the 80% purity requirement and belong to meaningful objects. We calculate a precision score by dividing this number with the total number of extracted groups.

For comparison, the baseline systems used on the ADL dataset are also applied to this dataset. Figure 9 shows examples of objects that the "Russell, et.al. (modified)" system discovered. The baseline system is confused by the cluttered environment and repetitive texture patterns. Quantitative comparison demonstrates that our method achieves significant improvement compared with the baseline systems (Figure 10). We noticed that Kim's approach outperforms Russell's systems here, probably because this dataset contains many texture-rich objects, and Kim's approach

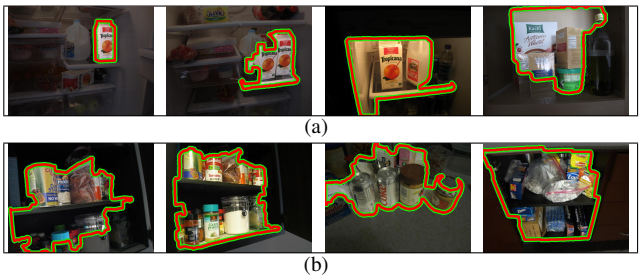


(a) Orange juice carton (occlusion, large viewpoint change)

(b) Chowder can (occlusion)

(c) Discovered unknown object #1 (Not labeled by the authors of [13])

Figure 8. **Objects that our program discovered in the CMU object dataset [13]. Our program successfully discovers objects despite the severe clutter, occlusion and changes of viewpoints.**



(a)

(b)

Figure 9. **Examples of objects discovered by the modified baseline system [33]. The method cannot distinguish objects with the similar BoW representations but with different shapes (a), and, is not suitable for environments where different objects have similar textures (b).**

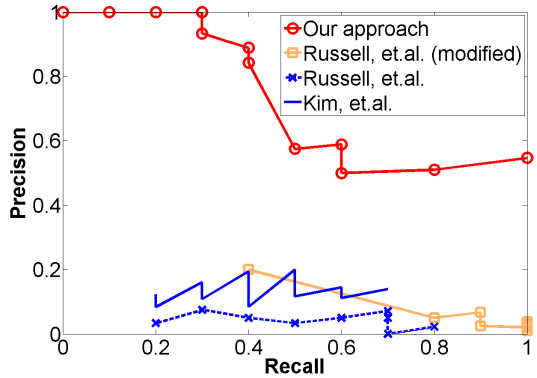


Figure 10. **Comparison of our approach and the baseline systems [16, 33] on the CMU object dataset.**

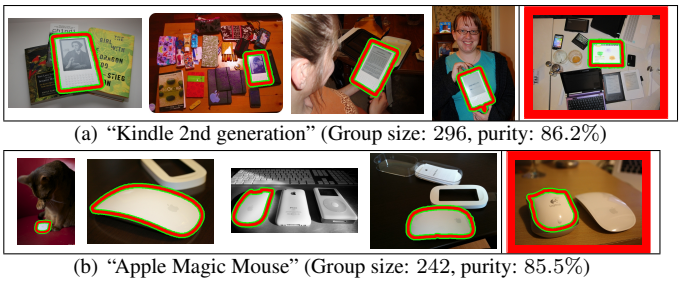
matches raw features instead of the quantized bag-of-words used in Russell's systems.

4.2. Interference from irrelevant images (Flickr)

We also want to evaluate how our program scales to larger datasets and copes with interference from irrelevant images, for scenarios such as object discovering from videos and Internet images.

We construct a dataset using images retrieved from Flickr.com. First, we retrieve images tagged with "Kindle 2nd generation" (Kindle) and "Apple Magic Mouse" (mouse), e.g., Figure 11. The Kindle represents texture-rich objects with regular rectangular shapes and the mouse represents texture-less objects with more distinctive shapes. For each object, we keep the top 100 most relevant images returned by Flickr.com. Then we download 1000 interference images from the scenes that might contain similar objects, such as "office", "living room", and "kitchen". In the test, we measure the performance of our program against the different number of interference images, added 100 each time until all the 1000 interference images are used (about 20000 segments in total, the average degree of the consistency graph is 2.8, and the density of the graph is $1.56e-4$).

We measure the maximum purity for segment groups larger than 15, since, in practical applications, a good object model should contain many segments from the same object. Because the same interference images are added to each object, the measurement also reflects how badly each object is confused with the objects in the interference images. Figure 12 shows that at 50% purity, the Kindle set can be mixed up with an interference set 7 times as large as itself, and the mouse set can be mixed up with an interference set 10 times its size. The difference is probably because the Kindle's rectangular shape is confused with many objects in the man-made environment. It again shows that our program is robust to heterogeneity of object appearance.



(a) "Kindle 2nd generation" (Group size: 296, purity: 86.2%)

(b) "Apple Magic Mouse" (Group size: 242, purity: 85.5%)

Figure 11. **Objects discovered by our program from 1000 most relevant images returned by Flickr.com for (a) "Kindle 2nd generation" and (b) "Apple Magic Mouse". The last columns are typical false positives for each object.**

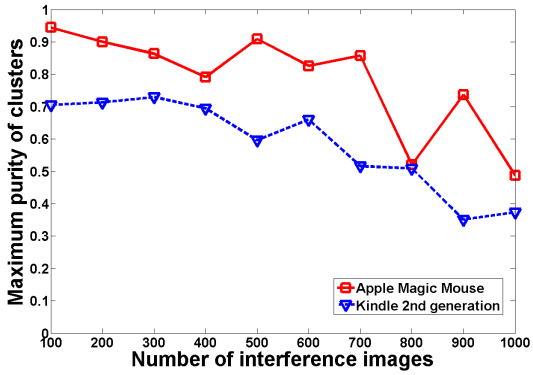


Figure 12. **Starting with 100 relevant images for each object, the maximum purity changes with respect to the number of interference images.**

5. Conclusion

While most of the previous papers on object discovery deal with categories [41], this paper tackled the problem of object instance discovery. We have shown how stricter constraints on the object geometric and appearance consistency can be used to discover object instances. We handle the heterogeneity of real life objects; we overcome the initial "noisy" segmentation, by enforcing the geometrical correspondence consistency; and we compose object models using spatial co-occurrence. The proposed approach demonstrated robustness to severe clutter, occlusion, changes of viewpoint, and interference from irrelevant images; the approach achieved significant improvement compared with two existing methods.

6. Acknowledgements

The authors would like to thank anonymous reviewers for helpful suggestions. Hongwen Kang thanks Minsu Cho and Gunhee Kim for providing their results and discussions. This work is partially supported by NSF Grant No. EEC-0540865.

References

[1] The future of humanoid robots. *DISCOVER*, 2000. 1

[2] E. M. Arkin, L. P. Chew, D. P. Huttenlocher, K. Kedem, and J. S. B. Mitchell. An efficiently computable metric for comparing polygonal shapes. *PAMI*, 1991. 3, 4

[3] H. Arora, N. Loeff, D. A. Forsyth, and N. Ahuja. Unsupervised segmentation of objects using efficient learning. *CVPR'07*. 1, 2

[4] M. Cho, Y. M. Shin, and K. M. Lee. Unsupervised detection and segmentation of identical objects. In *CVPR*, 2010. 2

[5] O. Chum and J. Matas. Large-scale discovery of spatially related images. *PAMI10*. 2

[6] T. Deselaers, B. Alexe, and V. Ferrari. Localizing objects while learning their appearance. In *ECCV'10*. 1, 2

[7] P. F. Felzenszwalb and D. P. Huttenlocher. Efficient graph-based image segmentation. *IJCV*, 2004. 2

[8] V. Ferrari. Affine invariant regions++. *PhD Dissertation*, pages 121–130, 2004. 3

[9] V. Ferrari, T. Tuytelaars, and L. Gool. Simultaneous object recognition and segmentation from single or multiple model views. *IJCV*, 2006. 2

[10] M. Fritz and B. Schiele. Decomposition, discovery and detection of visual categories using topic models. In *CVPR08*. 1, 2

[11] D. S. Hochbaum and V. Singh. An efficient algorithm for co-segmentation. In *ICCV*, 2009. 2

[12] D. Hoiem, A. A. Efros, and M. Hebert. Geometric context from a single image. In *ICCV*, 2005. 2

[13] E. Hsiao, A. C. Romea, and M. Hebert. Making specific features less discriminative to improve point-based 3d object recognition. In *CVPR*, 2010. 6, 7

[14] H. Jegou, M. Douze, and C. Schmid. Hamming embedding and weak geometric consistency for large scale image search. In *European Conference on Computer Vision*, 2008. 2

[15] A. Joulin, F. Bach, and J. Ponce. Discriminative clustering for image co-segmentation. *CVPR10*. 2

[16] G. Kim, C. Faloutsos, , and M. Hebert. Unsupervised modeling of object categories using link analysis techniques. In *CVPR*, 2008. 1, 2, 5, 6, 7

[17] K. Krapp. Activities of daily living evaluation. *Encyclopedia of Nursing & Allied Health*, 2002. 1

[18] Y. J. Lee and K. Grauman. Collect-cut: Segmentation with top-down cues discovered in multi-object images. In *CVPR10*. 4

[19] Y. Li, J. Sun, C.-K. Tang, and H.-Y. Shum. Lazy snapping. *ACM Trans. Graph.*, 23(3):303–308, 2004. 4

[20] D. Liu and T. Chen. A topic-motion model for unsupervised video object discovery. In *CVPR*, 2007. 2

[21] H. Liu and S. Yan. Common visual pattern discovery via spatially coherent correspondences. In *CVPR*, 2010. 2

[22] D. G. Lowe. Distinctive image features from scale-invariant keypoints. *IJCV*, 2004. 2, 3

[23] U. Luxburg. A tutorial on spectral clustering. *Statistics and Computing*, 17(4):395–416, 2007. 4

[24] A. Mishra and Y. Aloimonos. Active segmentation with fixation. In *ICCV*, 2009. 2

[25] J. Modayil and B. Kuipers. Bootstrap learning for object discovery. In *IROS-04*, 2004. 2

[26] D. Nister and H. Stewenius. Scalable recognition with a vocabulary tree. In *CVPR*, 2006. 2

[27] N. Payet and S. Todorovic. From a set of shapes to object discovery. In *ECCV'10*. 1, 2, 6

[28] J. Philbin, O. Chum, M. Isard, J. Sivic, and A. Zisserman. Object retrieval with large vocabularies and fast spatial matching. In *CVPR*, 2007. 2, 3

[29] T. Quack, V. Ferrari, and L. V. Gool. Video mining with frequent itemset configurations. In *CIVR*, 2006. 1, 2

[30] T. Quack, V. Ferrari, B. Leibe, and L. V. Gool. Efficient mining of frequent and distinctive feature configurations. In *ICCV'07*, October 2007. 2

[31] C. Rother, T. Minka, A. Blake, and V. Kolmogorov. Cosegmentation of image pairs by histogram matching - incorporating a global constraint into mrfs. *CVPR*, 2006. 2

[32] F. Rothganger, S. Lazebnik, C. Schmid, and J. Ponce. 3d object modeling and recognition using local affine-invariant image descriptors and multi-view spatial constraints. *International Journal of Computer Vision*, 2006. 2

[33] B. C. Russell, A. A. Efros, J. Sivic, W. T. Freeman, and A. Zisserman. Using multiple segmentations to discover objects and their extent in image collections. In *CVPR*, 2006. 1, 2, 5, 6, 7

[34] B. C. S. Sanders, R. C. Nelson, and R. Sukthankar. A theory of the quasi-static world. In *ICPR*, pages 1–6, 2002. 2

[35] J. Shi and J. Malik. Normalized cuts and image segmentation. In *CVPR '97*. 4

[36] J. Sivic, B. C. Russell, A. A. Efros, A. Zisserman, and W. T. Freeman. Discovering object categories in image collections. In *ICCV*, 2005. 1, 2

[37] J. Sivic and A. Zisserman. Video Google: A text retrieval approach to object matching in videos. In *ICCV*, 2003. 2, 3

[38] G. Somanath, M. Rohith, D. Metaxas, and C. Kambhamettu. D - clutter: Building object model library from unsupervised segmentation of cluttered scenes. *CVPR09*. 2

[39] S. Todorovic and N. Ahuja. Extracting subimages of an unknown category from a set of images. In *CVPR*, 2006. 1, 2

[40] R. Triebel, J. Shin, and R. Siegwart. Segmentation and unsupervised part-based discovery of repetitive objects. In *Robotics: Science and Systems*, 2010. 2

[41] T. Tuytelaars, C. H. Lampert, M. B. Blaschko, and W. Buntine. Unsupervised object discovery: A comparison. *IJCV10*. 1, 5, 8

[42] J. R. Tyler, D. M. Wilkinson, and B. A. Huberman. Email as spectroscopy: Automated discovery of community structure within organizations, 2003. 4

[43] A. Vedaldi and S. Soatto. Local features, all grown up. In *CVPR*, 2006. 2

[44] M. Weber, M. Welling, and P. Perona. Towards automatic discovery of object categories. *CVPR*, 2000. 1