Connecting Missing Links: Object Discovery from Sparse Observations Using 5 Million Product Images

Hongwen Kang, Martial Hebert, Alexei A. Efros, and Takeo Kanade

School of Computer Science
Carnegie Mellon University
{hongwenk,hebert,efros,tk}@cs.cmu.edu

Abstract. Object discovery algorithms group together image regions that originate from the same object. This process is effective when the input collection of images contains a large number of densely sampled views of each object, thereby creating strong connections between nearby views. However, existing approaches are less effective when the input data only provide sparse coverage of object views.

We propose an approach for object discovery that addresses this problem. We collect a database of about 5 million product images that capture 1.2 million objects from multiple views. We represent each region in the input image by a "bag" of database object regions. We group input regions together if they share similar "bags of regions." Our approach can correctly discover links between regions of the same object even if they are captured from dramatically different viewpoints. With the help from these added links, our proposed approach can robustly discover object instances even with sparse coverage of the viewpoints.

1 Introduction

Object discovery systems group input image regions into clusters that represent individual objects [1–9]. The grouping quality relies on finding correct matches between occurrences of the same object. When each object is observed seamlessly from all of its possible viewpoints, such matches can be recovered by existing techniques. In reality, the volume of input data is limited to a sparse sampling of views for each object. The appearance variance between different views makes it hard to recover the matches correctly. This happens frequently in scenarios such as at the beginning of a data collection process, or on websites such as Flickr and eBay, where most users only upload pictures of objects from a few viewpoints.

In this paper, we propose a data-driven approach that matches objects despite of large viewpoint changes. We are inspired by the fact that, given an unseen object, a person can easily reason about the appearance of the occluded views [10]. We emulate this ability with a data-driven approach that leverages a large prior "knowledge-base" of how objects appear in different views. Since similar objects appear similar from different views, we use the prior knowledge to "augment"

A. Fitzgibbon et al. (Eds.): ECCV 2012, Part VI, LNCS 7577, pp. 794–807, 2012.

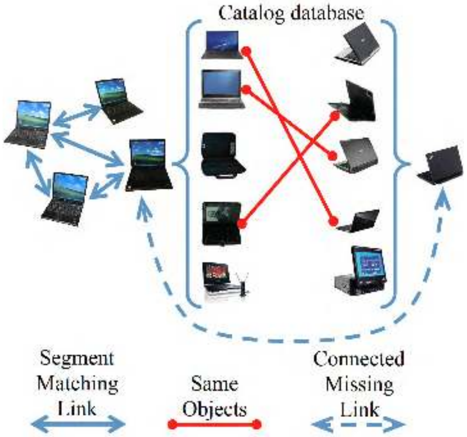


Fig. 1. Object discovery is difficult when appearance changes dramatically, such as the front view and rear view of the same laptop. Our approach connects the missing links between two object views if they are similar to the same set of objects in a large catalog database.

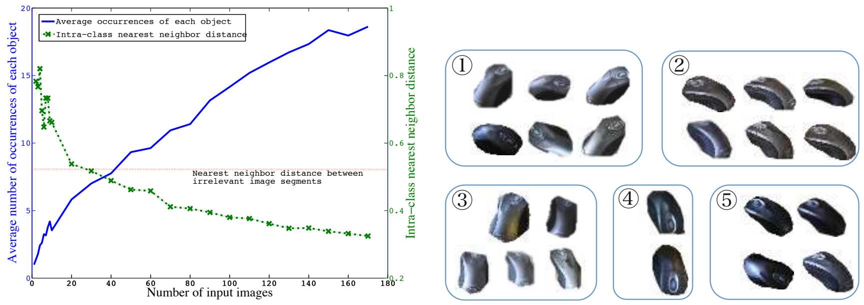
the observations of new unseen objects. For example, we observe that two object occurrences are likely from the same object, if they are similar to the same set of database objects (Figure 1). We develop a data collection procedure that takes advantage of multiple data sources. We collect a large image database of known objects (5 million product image of 1.2 million objects), where each object is imaged from several different viewpoints. We develop "data-driven similarity" based on this object database. We show that this data-driven similarity is effective in finding matches between views of similar objects. In this paper, we demonstrate specifically its effectiveness in improving over the state-of-art in object discovery.

Our approach is related to the object bank work of [11], where images are represented as bags of detector responses. However, instead of training detectors for a small number of object categories, we propose a data-driven approach that handles millions of objects and tens of thousands of categories. Our work is also related to transfer learning [12], which uses borrowed examples to address the data sparsity issue, with the key difference that we do not explicitly learn object classifiers. The problem of matching input images with product image databases have been attacked in many commercial systems, such as SnapTell and Google Goggles. Our paper explores a new application of using such existing product datasets as the prior to process new images, i.e., augmenting object similarities between sparse viewpoints. A method in face recognition [13] is most related to our approach, the work compares two faces using a ranked lists of faces that each matches in an existing face image database. To the best of our knowledge, our paper is the first to apply it to object discovery.

2 The Problem of Sparse Observation in Object Discovery

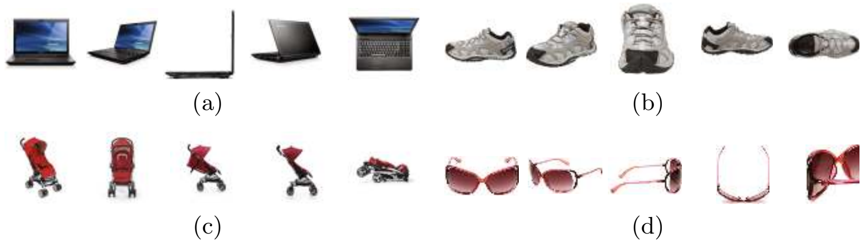
In the computer vision literature, similarities are generally defined by directly comparing the image regions, the feature points or both. Approaches from previous work on object discovery [1–9] form clusters of image regions based on these similarities. The quality of the clustering step has two aspects, 1) high purity, i.e., only regions belonging to the same object should be in the same cluster; 2), low fragmentation, i.e., the regions of the same object should be separated into as few clusters as possible.

To achieve high clustering quality, the similarity between regions of the same object should be significantly higher than the similarity between the segments belonging to different objects. Unfortunately, in real world scenarios with sparse observations, image segments of the same object can appear different, and sometimes more dissimilar than random pairs of image segments (Figure 2(a)). For example, the front view of a laptop is different from its rear view (Figure 1) and it is hard to match them based on visual features alone. As a result, in the absence of additional information, the discovery results in high fragmentation clusters (e.g., Figure 2(b)).



(a) (b)

Fig. 2. The phenomenon of sparse observation and its impact on the object discovery problem. (a) As the number of images increases, the number of observations of each object increases (blue solid curve from left to right), and as a result, the shortest distance between occurrences of the same object decreases (dashed-cross curve). For comparison, we also show the average nearest neighbor distance between random image segments on the same dataset. The statistics are generated from the dataset of [9]. (b) Image segments from different views of the same object may be separated into different groups by existing object discovery approaches, such as [9]. In this case, image segments of a computer mouse are fragmented into 5 separate groups, each group roughly corresponds to ① the *right* 45° view, ② the *left* 45° view, ③ the *rear* view, ④ the *front* view, and ⑤ the *left* 45° view captured under different lighting condition. In this paper, we focus on the fragmentation issue due to view point changes, i.e., the first 4 cases.


(a) (b)

(c) (d)

Fig. 3. Given a large database of objects captured from different viewpoints, we can leverage the knowledge of object correspondences to match input images of objects captured from very different viewpoints.

Table 1. Categories of query terms from Craigslist.org used in our data collection. $487,139$ query terms are collected from 6 U.S. cities (Boston, Chicago, New York, Pittsburgh, San Francisco, Seattle) during 2 weeks' time.

appliances	computers & tech	furniture	general for sale	sporting goods
tools	arts & crafts	baby & kid stuff	cell phones	clothing & accessories
electronics	household items	musical instruments	photo/video	toys & games

3 Collecting 5 Million Product Images to Model the Connections of Objects between Different Views

Directly solving the matching problem with sparse observations is hard but we can reason about the geometry of unseen objects by exploiting our knowledge of known similar objects. In this paper we exploit the prior knowledge of commonly used objects and the commonality shared between different objects.

For this prior knowledge database to be effective, it needs to have high *coverage*, i.e., it must capture most of the objects encountered in typical environments. A handcrafted list of objects would likely be incomplete and would introduce bias due to personal experience and preference. Instead, we need to develop a "surveying" mechanism that collects lists of object names. To minimize bias, the process of generating this list of objects should be independent to the sources of object images. In the following, we explore ways to harvest such a dataset by making use of several existing web sources.

Collecting Textual Names of Objects Used in Daily Environment from Independent Web Sources. First, we collect the names of daily-used objects. We need to take special care to address the dataset bias issue [14], by choosing text terms from independent data sources. At the same time, we need to explore a data source that users "voluntarily" report common objects in their surroundings. To this end, we choose the Craigslist.org as the data source. Specifically, we collect the titles of the classified advertisements posted in 15 categories of 6 major US cities during two weeks' time (Table 1). We found that object names are usually dominant in these advertisement titles. In total we collected $487,139$ terms.

Table 2. The coverage of the database based on 70 users' feedback, where the numbers represent the percentages of times that a user sees objects in the database that are similar to what she sees in her environment. Examples of objects that users did not find in our dataset includes: "research paper" in office, "egg" in kitchen and "voting machine" in conference room. For examples of objects contained in our database, please explore: `http://bit.ly/cmu5million`

Office	Kitchen	Living room	Bedroom	Conference room	Overall
94.0%	93.0%	96.7%	95.9%	91.2%	93.6%

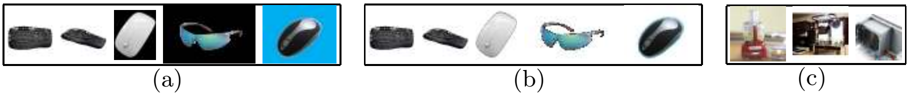


Fig. 4. We create a database of segmented objects by automatically identifying "clean" database images (a), segmenting the corresponding foreground objects (b), and removing the images taken without a clean background(c).

Harvesting Millions of Product Using Product Catalogs. After collecting the names of objects, we need to collect the corresponding images of these objects. We use product images from online stores, such as Amazon.com or Walmart.com. These data sources have three advantages: first, the products displayed in their catalogs are commonly found in daily environment; second, each product in these catalogs is captured in several images from different views, which provide useful information on the appearance of objects from different views; third, most of the product images contain objects captured on clean backgrounds, i.e., the object can be segmented out easily, which improves matching quality.

In this paper, we used the Amazon product search engine to collect relevant product images. We use each text term collected before as a query. For each query, we extract the top 20 most relevant products. We remove duplicates using the product ID returned by the web store. We retrieved about 5 million catalog images for about 1.2 million products. Most of the products are captured multiple times in different poses. Each product in the database is also assigned to one of about $15,000$ categories. We surveyed multiple independent persons living in multiple geographical locations to illustrate the coverage of the dataset for five different environments as shown in Table 2.

Extracting Object Regions from "clean" Product Images. For reliable matching with input images, we segment the foreground objects from the product images with clean background. To identify such "clean" images, we build a logistic classifier based on the color variance along the image boundaries, similar to [15]. Since these images only contain the objects and clean background, we can use a simple background subtraction algorithm. Figure 4 shows some examples of the "clean" images and the corresponding foreground objects. Figure 3 shows more examples of object regions from different viewpoints.

4 Overview of the Object Discovery Framework

In the following, we briefly review the object discovery framework proposed in [9]. We then introduce and integrate data-driven similarity with the object discovery process. First, we generate object proposals using multiple segmentation algorithms [16, 17]. Each individual image segment forms an object hypothesis s_i. We take two steps to discover objects from these image segments: 1) we calculate the similarity between image segments; and 2) we discover objects as groups of image segments that are mutually consistent.

We estimate the similarity between s_i and s_j by combining two measurements, $c_1(s_i, s_j)$ and $c_2(s_i, s_j)$, calculated from color-texture features and shape features respectively. For each consistency measure, we find a threshold that maximizes the separation of two independent sets of segments that are labeled as *consistent* and *inconsistent*. Finally, a binary graph is generated as creating each link $c(i, j)$:

$$c(i,j) = c_1(i,j) > t_1 \wedge c_2(i,j) > t_2, \tag{1}$$

With the binary similarity graph established, we extract groups of mutually similar segments such that each group corresponds to an object. Because the features used for calculating $c_1(s_i, s_j)$ and $c_2(s_i, s_j)$ are sensitive to viewpoint changes. The image segments of an object might be fragmented into several clusters due to sparse observations, e.g., Figure 2(b).

In this paper, we introduce a "data-driven similarity" measure that augments the matching of objects despite large viewpoint changes, using the product image database we have collected. Given an image segment, s_i, we find the set of most similar product objects, $\Psi(s_i)$. We compute the data-driven similarity, denoted by $c_a(i, j)$, based on the portion of common objects shared between the two sets, $\Psi(s_i)$, $\Psi(s_j)$. We assign high similarity value to $c_a(i, j)$ if $\Psi(s_i)$ share with $\Psi(s_j)$ a large portion of common objects.

We combine the data-driven similarity measure with the existing binary similarity graph by modifying $c(i, j)$:

$$c(i,j) = (c_1(i,j) > t_1 \wedge c_2(i,j) > t_2) \vee (c_a(i,j) > t_a)\,. \tag{2}$$

In other words, we use the data-driven similarity to add extra links to the graph. Because in our database, each object could be captured from multiple views. The shared objects could capture different aspects of the same object. The added links effectively connect views of objects despite the difficulty of finding direct visual correspondences. As a result, data-driven similarity succeeds in matching objects despite of large viewpoint changes.

5 Calculating Data-Driven Similarity

Generating new links based on the data-driven similarity involves three main steps. First, we find similar database object regions, $\Psi(s_i)$ for each input image segment, s_i. Second, we select the input segments that yield the most consistent matches. Third, we calculate the data-driven similarity, $c_a(i, j)$ as in (2), based on the intersection of $\Psi(s_i)$ and $\Psi(s_j)$.

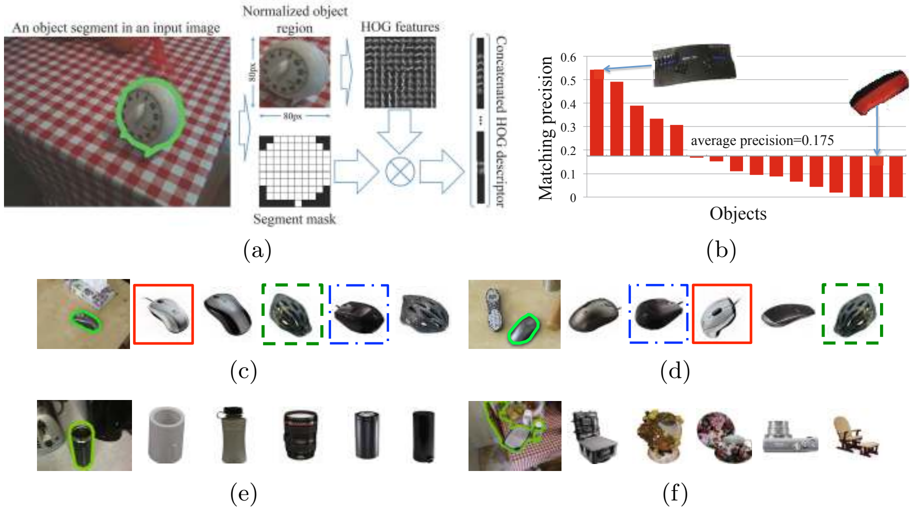


Fig. 5. Finding similar database regions using a modified HOG feature. (a) Generating HOG descriptors from each image segment. (b) Precisions of the K nearest neighbor matching, evaluated under a recognition setup. Note that the “recognition” rate for certain objects is close to zero. However, this does not prevent them from being successfully discovered. The representation based on the K-nearest neighbors is robust to mismatches: For example, a computer mouse from viewpoint 1 is similar to other computer mice and some bicycle helmets (c); the same computer mouse from viewpoint 2 is similar to other computer mice and some bicycle helmets (d); some of these similar database objects are common, which are labeled with different colored bounding boxes. Another example is that a water bottle could be similar to a pipe, a SLR lens, and some trash cans (e). In each row, the image segments used for query are outlined in green.

5.1 Matching Input Regions with Prior Database of Regions

Given an input image segment s_i, we want to find the set of database regions, $\Psi(s_i)$, that are most similar. We consider color, texture and shape as the three criteria to select K nearest neighbors from the database, i.d., $|\Psi(s_i)| = K$. We measure color similarity by using color histogram matching. We propose a modified HOG descriptor that retrieves objects with similar texture and shape. This modified HOG descriptor has three advantages, 1) it addresses the heterogeneity of object appearance, i.e., some objects are texture-rich (e.g., keyboard), while some other objects are texture-less (e.g., mouse); 2) it is computationally efficient for retrieving objects with similar shapes (unlike other approaches for direct shape comparison, such as [18]); 3) using the gradients makes our approach more robust to illumination changes.

We illustrate the process of generating the HOG descriptors in Figure 5(a). First, we crop the minimum bounding rectangle regions of image segments from the input and database images, and we normalize each rectangle region to a canonical size (80 × 80 pixel). Second, we extract HOG descriptors from the normalized image patch, using 8 × 8 pixel cells [19][1]. Each cell is represented as a 31 dimensional histogram of gradients vector. Third, using the segment mask, we set the cells in the background region to zeros. Finally, we concatenate all the HOG descriptors into a 3100-dimensional vector normalized to unit length. For each input image segment, we find the most similar database regions using the cosine similarity between HOG descriptors. Figure 5 shows several examples of the database regions similar to the image segments in the input images.

There are interesting differences and similarities between our matching task and object recognition. Although the purpose of finding similar database regions is relevant to the task of recognition, it addresses a different objective. As an object recognition task, the retrieved objects have to be the same as the input image segment to be considered correct, while in our task the matched objects are merely used as an intermediate representation. Since objects share appearance features, it is perfectly fine for a computer mouse to be matched with other computer mice as well as bike helmets (Figure 5(c)); similarly a water bottle may be matched with pipes, SLR lenses, and trash cans (Figure 5(e)).

On the other hand, it seems reasonable that improving the matching quality as a recognition task will probably improve our performance in object discovery. As a way to illustrate the matching quality, we evaluate the matching algorithm under a recognition setup, where a match is counted as correct if at least one object of the same type as the input appears in the top 10 matches (Figure 5(b)). While the "recognition" rates for certain objects are close to zero, they can still be successfully discovered because the data-driven similarity remains informative.

5.2 Selecting Segments with Consistent Matches

We select a subset of the segments that yield high quality matches for calculating data-driven similarity. We found this has two advantages: first, it improves the quality of the added links; second, it reduces the computational cost for pairwise comparison.

A straightforward approach is to use the averaged similarity of a region with its K nearest neighbors. However, when the database size increases, the likelihood of finding a set of unrelated objects that happen to be similar to an image segment increases [20, 21] (e.g., Figure 5(f)).

We propose to use the meta-data category information to cross-validate the visual matches. Specifically, we select the segments with most consistent nearest neighbors (e.g., Figure 5(c), (d), (e)). We measure the matching consistency of an image segment s_i as the category entropy of $\Psi(s_i)$:

$$H(s_i) = -\sum_{c=1}^{C} p_c \ln p_c, \tag{3}$$

[1] To prevent boundary crossing, we pad the original region with 8 pixel-wide blank stripes in each boundary direction.

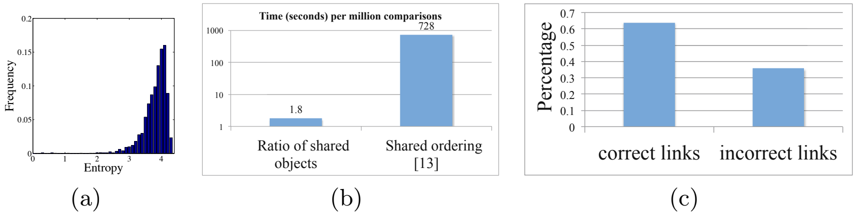


Fig. 6. (a) shows the distribution of the category entropy of different bags of matched regions. (b) Computational cost of computing the consistency measure over a million pairs. (c) The percentages of the discovered links being correct or incorrect based on the ground truth matches.

where C is the number of unique object categories in the list of $\Psi(s_i)$, and p_c is the percentage of objects in category c. In our experiments, we fix $K = 20$. We keep the segments with the 10% lowest matching entropy. Figure 6(a) shows the distribution of entropy generated from the dataset used in our experiment.

5.3 Calculating Data-Driven Similarity Using Bags of Regions

Given a pair of input image segments s_i and s_j, and the K nearest neighbors $\Psi(s_i)$ and $\Psi(s_j)$. We measure the data-driven similarity as the percentage of shared common objects:

$$c_a(i,j) = \frac{|\Psi(s_i) \bigcap \Psi(s_i)|}{K}. \tag{4}$$

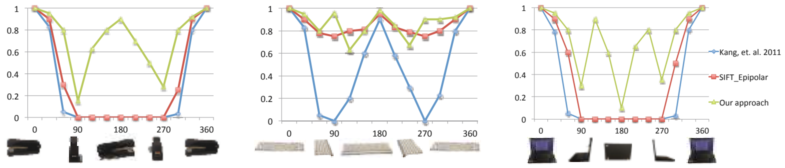
This is a more efficient version of the similarity measure used in [13], which considers also the ordering of the shared nearest neighbors. We found that it improves the computational efficiency, (Figure 6(b)), which is crucial in the discovery task that requires millions of pairwise comparison. Empirically, we found that these two measures perform similarly in the discovery performance. We create links between segments according to (2). We found that the data-driven similarity links aligned well with the ground truth matches (Figure 6(c)).

6 Experiments

In this section, we perform rigorous experimental evaluation of our algorithm on multiple datasets to quantify the benefits of our approach. We show that our approach does help in recovering matches despite large viewpoint changes and we demonstrate the effectiveness of our approach in object discovery.

6.1 Quantitative Evaluation of Robustness to Viewpoint Changes

First, we investigate the effectiveness of our approach in establishing links between object regions from widely separated viewpoints of an object. In this



(a) (b) (c)

Fig. 7. Comparison of different approaches in recovering links between regions in the presence of large viewpoint changes. The horizontal axis shows the change of viewpoints ($0° - 360°$ in increments of 30°). For visualization, we normalize the similarity score estimated by each method between 0 and 1.

experiment, we select two objects, stapler and keyboard, from the UoW's RGB-D dataset [22]. We collect another image sequence of a laptop, in addition to these two objects, to increase matching difficulty. For each object, we sample the image sequence for viewpoints at 30° increments. We use the ground truth segmentations to make sure that the only factor involved in this experiment is the variation of appearance due to viewpoint.

We compare three techniques for computing similarity. The first technique of [9] calculates similarity according to color, texture and shape. This technique is sensitive to non-planar transformation since it uses a similarity transformation model for matching shapes [18]. The second technique (SIFT_Epipolar) uses sparse SIFT feature matching with epipolar constraints [23]. It still can not handle the extreme viewpoint changes that completely changes the appearance of the object.

For quantitative evaluation, we normalize all scores between images (similarity measure for [9], number of correspondence features for SIFT_Epipolar, and data-driven similarity measure for our approach) to be between 0 and 1. We plot these normalized scores in Figure 7. For the stapler and the laptop, the number of correspondences recovered by the baseline techniques decrease quickly with respect to the amount of viewpoint change. Neither method can find any correspondences if the viewpoint change is larger than 90°. The SIFT_Epipolar technique performance improves on the keyboard, which is planar and texture-rich. Our approach maintains reliable correspondence between large viewpoint changes in most cases. Exceptions are when one of the input regions is uninformative. For example, when the stapler is rotated by 90°, the rear view of the stapler becomes unformative and matched with generic objects different from that of the more distinctive side view, the same happens for the rear view of the laptop.

6.2 Quantitative Evaluation of Object Discovery on the CMU ADL Dataset

Now we evaluate the effectiveness of our approach in the object discovery application. We evaluate three separate properties of our discovery algorithm. 1:

Each cluster should contain image segments that correspond to a single object (*precision*). 2: All of the objects in the environment should be discovered (*recall*). 3: Each object should be found as a single cluster (*fragmentation*).

Following prior work [3, 7, 9], we measure precision by estimating the purity of each cluster, which is defined as the percentage of segments in the cluster which come from a single object. Precision measures the percentage of pure clusters out of all the extracted clusters. We define the recall as the percentage of objects that are correctly discovered. We define object fragmentation as the average number of segment clusters per discovered object.

We compare our approach to the object *instance* discovery method of [9]. We also compare with [3] and [5], which used multiple segmentations and link analysis for object discovery. We also investigated [6] as a potential baseline system but it turned out to be too computationally expensive for our application. The approach of [7] is also an interesting baseline, but it requires supervision whereas we are interested in fully unsupervised discovery.

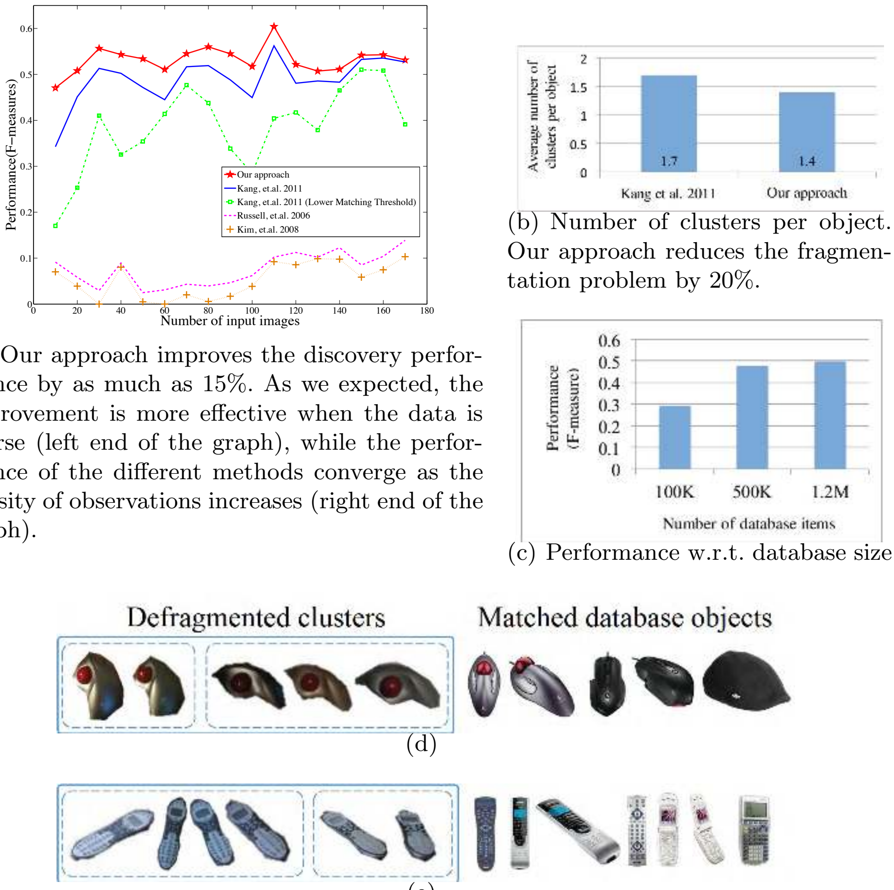
We evaluate our approach on the ADL dataset [9]. Since our approach is essentially adding new links to the similarity graph, it is important to compare it with a baseline in which we simply lower the thresholds on the appearance similarity measures c_1 and c_2 to include more links. We adjusted the thresholds so that the baseline system include a majority (80%) of the new links that our approach adds. We quantify the performance of each approach using the F-measure.

We evaluate the effects of dataset size by performing experiments on subsets of images from the ADL dataset (ranging in size from 10 to 175 images). Figure 8 shows a comparison of different approaches for this task. Our approach consistently outperforms the baseline system ("Kang, et al. 2011") by 6% on average; moreover, the performance improvement grows to over 15% as the number of input images decreases. In addition, we outperform the modified version of ("Kang, et al. 2011 (Lower Matching Threshold)"). This is intuitive, since lowering the matching threshold results in a large number of spurious links that degrade the quality of clusters. We also outperform the baseline system in terms of fragmentation. Our approach generates fewer clusters per object and lowers the fragmentation factor by about 20% (Figure 8(b)).

6.3 Qualitative Evaluation on Flickr Images

We also evaluate our approach in the scenario of online photosharing, where users tend to take pictures of the same objects from few iconic views, resulting in sparse observations, for example the 23 images corresponding to the query "Vibram five fingers," downloaded from Flickr, shown in Figure 9. The sparse observation problem is a challenge in many applications, such as web image clustering and categorization. Existing definitions of visual similarity are very limited in handling such large viewpoint changes.

In this experiment, we applied the clustering algorithm in [9], which results in highly fragmented clusters, i.e., 3 segment clusters, each containing less than 4 image segments. By comparison, our algorithm is able to discover additional links



(a) Our approach improves the discovery performance by as much as 15%. As we expected, the improvement is more effective when the data is sparse (left end of the graph), while the performance of the different methods converge as the density of observations increases (right end of the graph).

(b) Number of clusters per object. Our approach reduces the fragmentation problem by 20%.

(c) Performance w.r.t. database size

(d)

(e)

Fig. 8. We compare our approach with existing techniques in discovering objects from sparse observations. In this experiment, we randomly select a proportion, $p \in (0, 1]$, of all the ADL images dataset. We run 10 rounds of experiments for each p and measure the performance of each instance discovery approach. As we can see, our approach performs reliably even with a small number of input images while the performance of the baseline approaches degrades more rapidly as the number of images decreases.

between segments by matching with the product database. Figure 9 shows the cluster of 15 segments that our program discovered. We not only merge all the original fragmented clusters, but also include some segments that were originally left out due to large visual difference. Analysis on the database objects used for generating the data-driven similarity links shows that objects of different types that share similar attributes also contributes to the matching process, such as gloves and keyboards (Figure 9(b)).

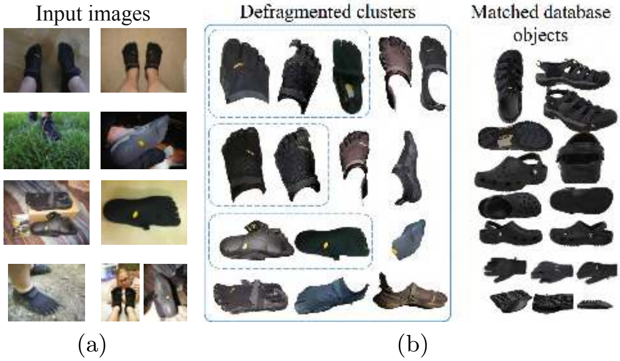


Fig. 9. Examples of images retrieved from Flickr.com using the query "Vibram five fingers." Users tend to capture the same objects from a few iconic views, while leaving out the intermediate views, which makes it impossible to group regions based on appearance alone (fragmentation). Our approach correctly groups the regions into a single cluster.

7 Conclusion

In this paper, we proposed a data-driven approach for measuring object similarities that is robust to sparse observations. We use a large product image database to represent the appearance of objects in different viewpoints. We match image segments that are similar to the same set of database objects, augmenting the existing visual matches. We demonstrate that our approach recovers matches despite large viewpoint changes, specifically, for the application of object discovery. This work shows the value of data-driven approaches with a large body of prior knowledge. In the future, we would like to explore the application of our approach to handle the sparse observation issues due to other imaging conditions, such as difference in illumination.

Acknowledgments. The authors would like to thank anonymous reviewers for helpful suggestions. This work is partially supported by NSF Grant No. EEC-0540865.

References

1. Tuytelaars, T., Lampert, C.H., Blaschko, M.B., Buntine, W.: Unsupervised object discovery: A comparison. IJCV (2009)
2. Sivic, J., Russell, B.C., Efros, A.A., Zisserman, A., Freeman, W.T.: Discovering object categories in image collections. In: ICCV (2005)
3. Russell, B.C., Efros, A.A., Sivic, J., Freeman, W.T., Zisserman, A.: Using multiple segmentations to discover objects and their extent in image collections. In: CVPR (2006)
4. Quack, T., Ferrari, V., Leibe, B., Van Gool, L.: Efficient mining of frequent and distinctive feature configurations. In: ICCV 2007 (2007)

5. Kim, G., Faloutsos, C., Hebert, M.: Unsupervised modeling of object categories using link analysis techniques. In: CVPR (2008)
6. Cho, M., Shin, Y.M., Lee, K.M.: Unsupervised detection and segmentation of identical objects. In: CVPR (2010)
7. Lee, Y.J., Grauman, K.: Object-graphs for context-aware category discovery. In: CVPR (2010)
8. Payet, N., Todorovic, S.: From a Set of Shapes to Object Discovery. In: Daniilidis, K., Maragos, P., Paragios, N. (eds.) ECCV 2010, Part V. LNCS, vol. 6315, pp. 57–70. Springer, Heidelberg (2010)
9. Kang, H., Hebert, M., Kanade, T.: Discovering object instances from scenes of daily living. In: ICCV (2011)
10. Huffman, D.A.: Impossible Objects as Nonsense Sentences. Machine Intelligence 6, 295–323 (1971)
11. Li, L.J., Su, H., Xing, E.P., Fei-Fei, L.: Object bank: A high-level image representation for scene classification & semantic feature sparsification. In: NIPS (2010)
12. Lim, J., Salakhutdinov, R., Torralba, A.: Transfer learning by borrowing examples. In: NIPS (2011)
13. Schroff, F., Treibitz, T., Kriegman, D., Belongie, S.: Pose, illumination and expression invariant pairwise face-similarity measure via doppelgänger list comparison. In: ICCV (2011)
14. Torralba, A., Efros, A.A.: Unbiased look at dataset bias. In: CVPR 2011 (2011)
15. Ke, Y., Tang, X., Jing, F.: The design of high-level features for photo quality assessment. In: CVPR (2006)
16. Felzenszwalb, P.F., Huttenlocher, D.P.: Efficient graph-based image segmentation. IJCV (2004)
17. Mishra, A., Aloimonos, Y.: Active segmentation with fixation. In: ICCV (2009)
18. Arkin, E.M., Chew, L.P., Huttenlocher, D.P., Kedem, K., Mitchell, J.S.B.: An efficiently computable metric for comparing polygonal shapes. PAMI (1991)
19. Dalal, N., Triggs, B.: Histograms of oriented gradients for human detection. In: CVPR (2005)
20. Hays, J., Efros, A.A.: Scene completion using millions of photographs. In: SIGGRAPH (2007)
21. Torralba, A., Fergus, R., Freeman, W.T.: 80 million tiny images: A large data set for nonparametric object and scene recognition. PAMI (2008)
22. Lai, K., Bo, L., Ren, X., Fox, D.: A large-scale hierarchical multi-view rgb-d object dataset. In: ICRA (2011)
23. Brown, M., Lowe, D.: Recognising panoramas. In: Proceedings of the 9th International Conference on Computer Vision, Nice, vol. 2, pp. 1218–1225 (2003)