The Dissertation Committee for Yong Jae Lee
certifies that this is the approved version of the following dissertation:

Visual Object Category Discovery in Images and Videos

Committee:

Kristen Grauman, Supervisor

Joydeep Ghosh, Co-Supervisor

J. K. Aggarwal

Al Bovik

Alexei Efros

Wilson Geisler

Visual Object Category Discovery in Images and Videos

by

Yong Jae Lee, B.S.; M.S.E.

DISSERTATION

Presented to the Faculty of the Graduate School of

The University of Texas at Austin

in Partial Fulfillment

of the Requirements

for the Degree of

DOCTOR OF PHILOSOPHY

THE UNIVERSITY OF TEXAS AT AUSTIN

May 2012

Dedicated to my parents.

Acknowledgments

I feel extremely fortunate and grateful to have had Kristen Grauman as my advisor. Her passion for research and dedication to her students have been truly inspiring. I have learned so much from her about how to be a good researcher. Thank you for believing in me, and for all the support and guidance throughout the past five years that have made this thesis possible.

I would like to thank my co-advisor Joydeep Ghosh, who was always available to chat, and to share his invaluable thoughts and advice. I have learned a lot from attending the IDEAL group meetings. I am also grateful to my thesis committee members, Alyosha Efros, J. K. Aggarwal, Al Bovik, and Bill Geisler for their insightful comments and feedback that strengthened this thesis.

My labmates deserve much thanks for making my PhD student life an enjoyable and memorable one. Thanks Sudheendra Vijayanarasimhan for being there from the beginning, and setting such a good example to follow. Thanks Jaechul Kim for all the valuable discussions about research and life. Thanks Adriana Kovashka for providing balance in our group, and the discussions about random and various topics over beer. Thanks Sung Ju Hwang for your kindness and the many laughs. Thanks Chao-Yeh for always being optimistic and being the designated driver. Thanks Sunil Bandla for reigniting my interest in soccer. Finally, thanks Lu Zheng for all the good advice and fun times.

Thanks to Yoshihisa Shinagawa for introducing me to computer vision,

Ben Kuipers for co-advising my Masters thesis, and Larry Zitnick and Michael Cohen for mentoring my internship at Microsoft Research. Thanks also to my friends Heeseok Koo, Sumi Kim, Hyejung Noh, Minchan Lee, Sungho Yun, Michael Ryoo, Jong-Taek Lee, Josh Harguess, Birgi Tamersoy, Chia-Chih Chen, Changhai Xu, Dam Sunwoo, and Sanmi Koyejo.

Finally, this thesis would not have been possible without the unconditional love and support of my family: my father Sang-Pal Lee, my mother Young-Hee Lee, and my brother Song Jae Lee. I dedicate this thesis to them.

Special thanks to Yaewon Kang who has been by my side throughout my entire graduate studies. This thesis, and especially Chapter 5, would not have been possible without her help. The last few years have been the happiest of my life. Thank you for your love, support, encouragement, and allowing a poor graduate student to live a luxurious life, including providing yummy food to eat.

Visual Object Category Discovery in Images and Videos

Publication No. _____

Yong Jae Lee, Ph.D.
The University of Texas at Austin, 2012

Supervisor: Kristen Grauman
Co-Supervisor: Joydeep Ghosh

The current trend in visual recognition research is to place a strict division between the supervised and unsupervised learning paradigms, which is problematic for two main reasons. On the one hand, supervised methods require training data for each and every category that the system learns; training data may not always be available and is expensive to obtain. On the other hand, unsupervised methods must determine the optimal visual cues and distance metrics that distinguish one category from another to group images into semantically meaningful categories; however, for unlabeled data, these are unknown a priori.

I propose a visual category discovery framework that transcends the two paradigms and learns accurate models with few labeled exemplars. The main insight is to automatically focus on the prevalent objects in images and videos, and learn models from them for category grouping, segmentation, and summarization.

To implement this idea, I first present a context-aware category discovery framework that discovers novel categories by leveraging context from previously learned categories. I devise a novel object-graph descriptor to model

the interaction between a set of known categories and the unknown to-be-discovered categories, and group regions that have similar appearance and similar object-graphs. I then present a collective segmentation framework that simultaneously discovers the segmentations and groupings of objects by leveraging the shared patterns in the unlabeled image collection. It discovers an ensemble of representative instances for each unknown category, and builds top-down models from them to refine the segmentation of the remaining instances. Finally, building on these techniques, I show how to produce compact visual summaries for first-person egocentric videos that focus on the important people and objects. The system leverages novel egocentric and high-level saliency features to predict important regions in the video, and produces a concise visual summary that is driven by those regions.

I compare against existing state-of-the-art methods for category discovery and segmentation on several challenging benchmark datasets. I demonstrate that we can discover visual concepts more accurately by focusing on the prevalent objects in images and videos, and show clear advantages of departing from the status quo division between the supervised and unsupervised learning paradigms. The main impact of my thesis is that it lays the groundwork for building large-scale visual discovery systems that can automatically discover visual concepts with minimal human supervision.

Table of Contents

List of Tables

List of Figures

Chapter 1

Introduction

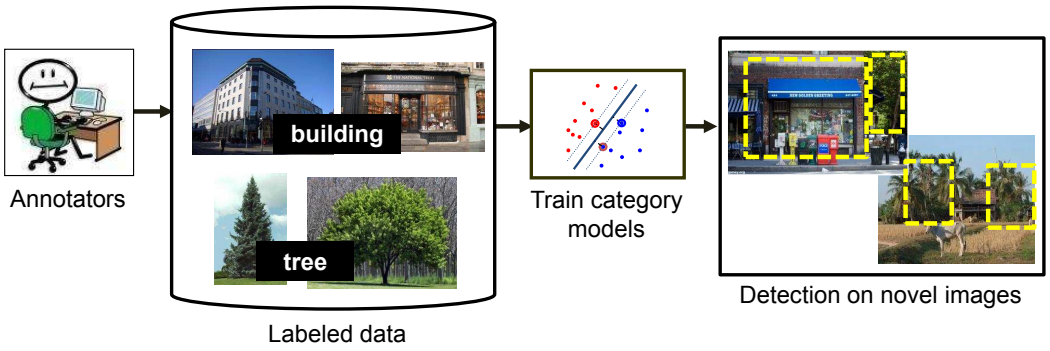
The goal in computer vision research is to produce autonomous systems that can meaningfully perceive visual data. One of the central problems in computer vision is object recognition, the task of identifying objects in images or videos. While effortless for humans, object recognition for machines can be difficult for several reasons. Background clutter, illumination effects, occlusion, appearance variations due to scale, translation, and rotation are common in natural images and are challenges that a computer vision system must overcome. Recent years have shown encouraging progress, particularly in terms of generic visual category learning [43, 85, 94, 160] and robust local feature representations [1, 84, 101].

The learning paradigms of existing object recognition methods can be largely divided into two groups: *supervised* and *unsupervised* methods. In the supervised setting, the recognition system trains with manually prepared exemplars of each class of interest. The most common forms of annotations are pixel-level labelings or bounding boxes surrounding each object. In this setting, the system can learn discriminative properties of each given category from their training examples, which often leads to high recognition accuracy. Useful applications for supervised object recognition include face recognition for automated visual surveillance, face-tagging in consumer photo collections, content-based image or video retrieval for search engines, automatic inspec-

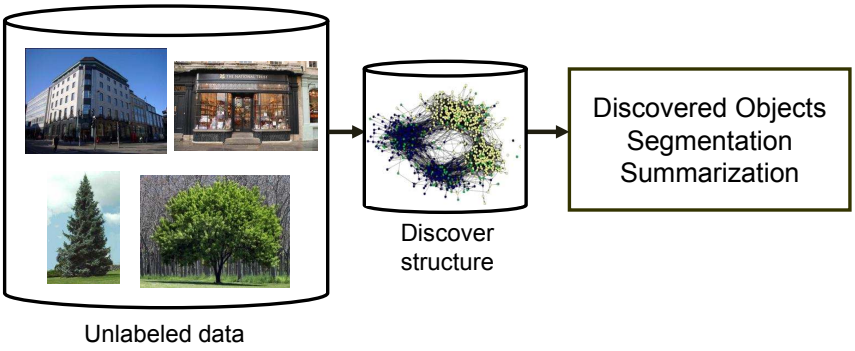
tion of products in manufacturing applications, and medical imaging analysis to detect viruses or diseases. However, carefully labeled exemplars are expensive to obtain in the large numbers needed to fully represent a category's variability, and methods trained in this manner can suffer from unintentional biases imparted by dataset creators [147].

In contrast, in the unsupervised setting, the system completely forgoes human annotation to learn category models. Also referred to as "discovery", existing methods mine for recurring appearance patterns in the unlabeled image collection to group images or image regions into discovered categories (e.g., [55, 72, 86, 127]). 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, robot-navigation in unexplored territory, or even to supplement traditional supervised object recognition systems. While recent work has shown great progress, learning from completely unlabeled images remains difficult. Unsupervised learners face the same issues that plague supervised methods—clutter, viewpoint, intra-class appearance variation, occlusions—but must handle them without any explicit annotation guidance. Consequently, supervised learners often outperform their unsupervised counterparts since they can focus on the task for which they are specifically trained to handle. See Figure 1.1 for flow-charts comparing the two learning paradigms.

The current trend in visual recognition research is to place a strict division between the two learning paradigms. However, this is problematic for two main reasons. On the one hand, supervised methods require training data for each and every category that the system learns; training data may not always



(a) Supervised visual learning



(b) Unsupervised visual discovery

Figure 1.1: The contrast between (a) supervised and (b) unsupervised visual category learning.

be available and is expensive to obtain. For example, providing appropriate training data for a robot navigating a newly discovered planet may be infeasible. As another example, for video summarization, one cannot identify the categories of the main objects that appear without having seen the video first. On the other hand, it is unrealistic to expect unsupervised methods to learn hundreds or even thousands of categories without any supervision. Grouping images into semantically meaningful categories requires the unsupervised

learner to determine the optimal visual cues (i.e., image representation) and distance metrics that distinguish one category from another; however, for unlabeled data, these are unknown a priori. Furthermore, the set of optimal cues and metrics could differ for each category. Naturally, the task becomes more difficult as the number of categories increases. Without providing any human guidance, the unsupervised learner may not be able to produce semantically meaningful groups.

I propose a visual category discovery framework that transcends the two paradigms and learns accurate models with few labeled exemplars. The main insight is to automatically focus on the prevalent objects in images and videos, and learn models from them for category grouping, segmentation, and summarization. Any available labeled exemplars can be used to train models to help the system more accurately target the prevalent objects in the unlabeled data for discovery. Specifically, I propose to use two forms of pre-trained models: (1) category independent detectors (i.e., models that are trained to detect any category) to provide better candidate segments—especially for objects with heterogeneous appearance—than purely bottom-up methods, and (2) classifiers for the known categories (i.e., those that the system has training data) to provide object-level context cues to detect patterns whose correct grouping may be too ambiguous if relying on appearance alone.

To realize this goal of building visual category discovery systems that can learn accurate models from unlabeled data, there are several key challenges that must be addressed:

- Generic categories (e.g., people, cars, buildings) lack the strict geometric consistency and distinctive features inherent to specific objects (e.g.,

Barack Obama, my car, the Eiffel tower), forcing a discovery method to simultaneously identify natural groups while estimating their unknown variability in appearance.

- When there are multiple objects in each image in the unlabeled data collection, the system must simultaneously estimate the objects' proper segmentation, as well as their correct grouping across images. This is a chicken-and-egg problem: correct category groupings will depend heavily on the system having proper object segmentations, while proper object segmentations will depend heavily on the system having top-down category-level knowledge of the objects. In the unsupervised discovery setting, neither is known.

- Finally, since some objects will be more important than others, the system must identify the key components in that data that are worth discovering. For example, given a day's worth of wearable camera data, the system must be able to discern the important foreground objects (e.g., the camera wearer's friends and pets) from the irrelevant background clutter (e.g., cars passing by on the street).

Addressing these important challenges will lay the groundwork for building large-scale visual discovery systems that can automatically learn object categories, produce accurate segmentations, and summarize large collections of unlabeled visual data with minimal human supervision. Throughout the chapters in this thesis, I will introduce novel components of my visual discovery framework that bring us closer to realizing these goals. I will compare against existing state-of-the-art methods for category discovery and segmentation on several challenging benchmark datasets. I will demonstrate that we can

discover visual concepts more accurately by focusing on the prevalent objects in images and videos, and show clear advantages of departing from the status quo division between the supervised and unsupervised learning paradigms.

1.1 Overview of Thesis

In this section, I provide a summary of the main components of my thesis. The summaries discuss the main insights to how I have addressed the important challenges raised in the previous section. I first present a context-aware category discovery framework that discovers novel categories by leveraging context from previously learned categories. I next introduce a collective segmentation framework that simultaneously discovers the segmentations and groupings of objects by leveraging the shared patterns in the unlabeled visual collection. Finally, building on these techniques, I describe how to produce compact visual summaries for first-person egocentric videos that focus on the important people and objects. In the ensuing chapters, I will provide more detail on the technical ideas and experimental results for each component.

1.1.1 Context-Aware Category Discovery

Existing unsupervised category discovery methods [55, 72, 86, 99, 127, 134, 151] mine for frequently recurring appearance patterns in the unlabeled image collection, typically employing a clustering algorithm to group local features across images according to their texture, color, shape, etc. Unfortunately, learning multiple visual categories simultaneously from unlabeled images remains difficult, especially in the presence of substantial clutter and scenes with multiple objects. While appearance is a fundamental cue for object recognition, it can often be too weak of a signal to reliably detect visual

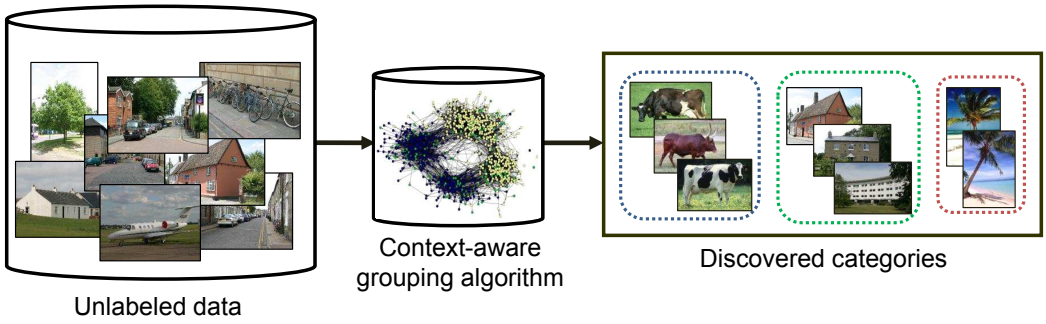


Figure 1.2: Proposed framework for context-aware category discovery. In most real settings, the unlabeled data contains a mix of known and unknown objects. Instead of mining for all categories from scratch, my approach leverages context from the familiar objects to group the unfamiliar ones to discover novel categories.

themes in unlabeled, unsegmented images in the face of occluded objects, large intra-category variations, or low-resolution data.

Furthermore, existing methods assume no prior information about categories and thus tend to perform poorly for cluttered scenes with multiple objects. In most real settings, we cannot predefine all categories of interest. For example, we cannot prescribe training data for all categories that a robot might encounter when navigating a new environment. The robot should be able to detect instances of the familiar objects for which it has training data, but should also be able to discover novel, unfamiliar objects.

Therefore, instead of relying only on recurring appearance patterns and mining for all categories from scratch, I propose a *context-aware* approach to discover novel categories that occur amid known objects within unannotated images [89, 90, 92] (see Figure 1.2). How can knowing about some categories help us to discover new ones in unlabeled images? The idea is that the context from familiar objects surrounding less familiar image regions can help

to detect patterns whose correct grouping may be too ambiguous if relying on appearance alone. Specifically, I devise a novel *object-graph descriptor* to model the interaction between a set of known categories and the unknown to-be-discovered categories. My system discovers novel categories by grouping regions that have similar appearance and similar category-level context.

The framework strikes a useful balance between current recognition strategies at either end of the supervision spectrum. The norm for supervised image labeling methods is forced-choice classification, with the assumption that the training and test sets are comprised of objects from the same pool of categories. On the other hand, the norm for unsupervised recognition is to mine for all possible categories from scratch. In my approach, the system need not know how to label every image region, but instead can draw on useful cues from familiar objects to better detect novel ones.

I evaluate my framework for two applications: object discovery in natural images (Section 3.1) and face discovery in consumer photo collections (Section 3.3). I perform extensive experiments on several benchmark datasets, and compare it to baseline methods including state-of-the-art discovery methods that only use appearance information. By modeling the interaction between an image's known and unknown objects, my method leads to significantly better detection of new visual categories compared to the conventional approach. It produces groups that tend to be more inclusive of intra-class appearance variation than those that could be found with appearance alone. For example, it discovers both side views and rear views of cars as a single category, and groups together face instances of the same person in different poses and different expressions.

The basic context-aware framework as described thus far allows discov-

ery to be considered in a more realistic scenario in which the unlabeled visual data collection can have a mix of known and unknown categories. However, it treats unsupervised category discovery as a one-pass "batch" procedure in which the input is a set of unlabeled images, and the output is a set of k discovered categories. All existing discovery methods adhere to this batch framework [72, 89, 99, 127, 151], which implicitly assumes that all categories are of similar complexity and that all information relevant to learning is available at once. However, paying equal attention to all instances makes the grouping sensitive to outlier regions that have incorrect object segmentations, and can skew the resulting category models unpredictably. Furthermore, it denies the possibility of exploiting inter-object context cues during discovery, since one cannot detect the typical relationships between objects if models for the component objects are themselves not yet formed.

Instead, in Section 3.2, I show how to consider visual discovery as a *self-paced, continuous* learning process [91]. Building on the context-aware framework I described above, I propose to focus on the "easier" objects first, and gradually discover new models of increasing complexity. What makes some image regions easier than others? Given that our goal is to group objects with similar appearance and context, regions that have consistent appearance patterns and are surrounded by familiar objects (i.e., those with stronger object-level context) will be easier to group. Why should it matter in what order objects are discovered? After each discovery, the system can update the set of familiar categories by training a detector for the newly found object class, which will allow it to produce a richer context model for each remaining (harder) unfamiliar instance.

I validate this self-paced variant of my approach on realistic natural

images, and show clear advantages for category discovery compared to conventional state-of-the-art batch clustering algorithms. The results indicate that learning categories through a self-paced curriculum is critical to being robust to outliers and to fully utilize inter-object context cues. Further, I show a practical application for discovery, in which the discovered categories are used to train models to predict instances in novel images. My approach achieves competitive results to fully supervised baselines at a fraction of the required human labeling cost.

1.1.2 Segmentation with Discovered Top-Down Cues

Thus far, I have overviewed an approach to discover novel categories from unlabeled image collections containing multiple categories. We assumed that performing multiple bottom-up segmentations will produce candidate object regions that roughly agree with true boundaries for each object in each image. However, this assumption may not always hold, especially if the images contain objects with heterogeneous appearance. Therefore, we can go further by not only discovering what categories exist, but also discovering how to segment their object instances (see Figure 1.3). The main challenge of discovering objects and their proper segmentations in unlabeled multi-object image collections is that generating correct category groupings depends on the system having proper object segmentations, while generating proper object segmentations depends on the system having top-down category-level knowledge of the objects in hand. Unfortunately, in the unsupervised discovery setting, neither is known.

Existing unsupervised segmentation methods can only group pixels with similar color or texture [6, 29, 40, 132], and can fail to group heteroge-

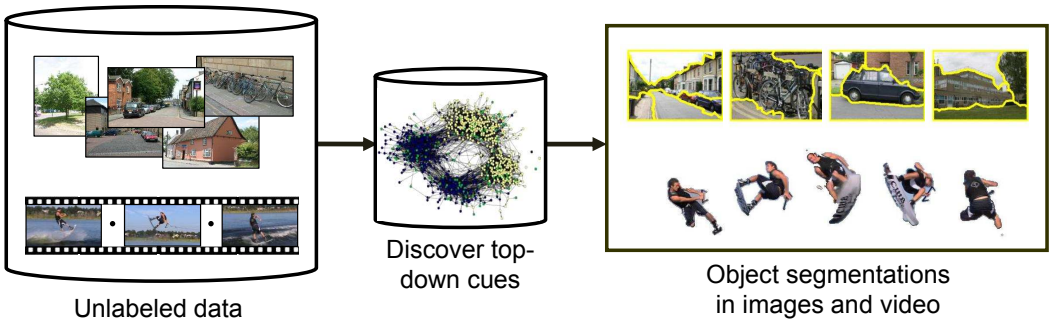


Figure 1.3: Proposed framework for unsupervised object segmentation with discovered top-down cues. When there are multiple objects in each image in the unlabeled collection, the system must simultaneously estimate each objects proper segmentation, as well as their correct grouping across images. My approach discovers the shared appearance patterns in the image collection, and builds models from them to segment the objects.

neous objects. For example, a car could be over-segmented into its wheels, windows, and body. Given a single image or video frame, this is the best we could do with a state-of-the-art segmentation method. Given a *collection* of images or frames, however, we can expect to find some accurate object segments that have recurring structure across the collection. If we can detect the good segments that correspond to coherent objects, we can build models from them to refine the inaccurate segments that correspond to object fragments or a mix of objects.

To implement this idea, I propose an approach that simultaneously segments a collection of unlabeled images while exploiting automatically discovered appearance patterns shared between them [88]. The goal is to discover an ensemble of representative instances for each category, and build top-down models from them to refine the segmentation of the remaining instances. To discover the ensemble models, the method clusters regions that have similar

appearance and contextual layout given by the object-graph descriptor from Section 1.1.1. Then, using each initial segment as a seed, the method refines its boundary by enforcing preferences to include nearby regions that agree with the ensemble regions, and exclude those regions that resemble familiar objects.

My results show that the segmentations computed jointly on the collection agree more closely with true object boundaries, when compared to bottom-up baselines that can only access cues from a single image. Furthermore, I show that the refined segmentations produce even more accurate clusters when provided to the context-aware discovery algorithm I discussed above.

Building on this idea, I next generalize the approach to the video domain. Unlike an image collection, which contains generic categories, a single video has recurring object instances. Thus, in this setting, the goal is to segment the recurring foreground objects in the video while ignoring the irrelevant background clutter. The problem is challenging because the background can be moving and changing, and the categories of the objects are unknown in advance. Existing unsupervised methods lack an explicit notion of what a foreground object should look like in video data [19, 21, 56, 65, 152] and rely only on low-level appearance and motion cues to group the pixels, which usually results in an over-segmentation of the objects.

To overcome these limitations, I show how to automatically discover a set of *key-segments* (similar to the ensemble regions described above) to explicitly model likely foreground regions for video object segmentation. The idea is to leverage both static and dynamic cues to detect persistent object-like regions, and then estimate a complete segmentation of the video using those regions and a novel localization prior that uses their partial shape matches

across the sequence. To find the key-segments, I introduce an object-like measure that reflects a region's likelihood of belonging to a foreground object using static intra-frame properties and dynamic inter-frame properties. The system groups object-like regions to generate multiple object hypotheses, and builds segmentation models from them to produce a pixel-wise segmentation for each hypothesis. To focus on the foreground objects while ignoring the irrelevant background, the system automatically ranks the discovered hypotheses according to their average object-like measure.

Important novel components of the proposed technique include (1) a new motion-based measure of object-like regions in video that complements existing image-based cues, (2) a localization prior using partial shape matches in video, and (3) a space-time graph segmentation that accommodates the key-segments. I apply my unsupervised approach to challenging benchmark videos, analyze its components in detail, and show state-of-the-art results compared to existing unsupervised and supervised methods.

I discuss my discovery framework for collective segmentation for images and videos in Sections 4.1 and 4.2, respectively.

1.1.3 Discovering Important People and Objects for Egocentric Video Summarization

Building on many of the techniques that I have introduced in the previous sections, I will finally present an approach to summarize egocentric videos captured from a wearable camera. The key insight is to discover the important people and objects in the data, and use them to drive the summarization. Existing video summarization methods extract keyframes [52, 164, 170], create montages of still images [5, 25], or generate compact dynamic sum-

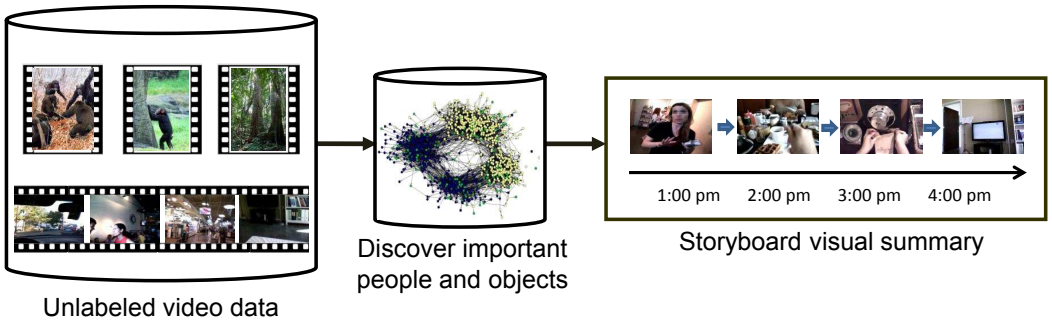


Figure 1.4: Proposed framework for egocentric video summarization. A system that lacks high-level information on *which objects matter* may produce a summary that consists of irrelevant frames or regions. Instead, my approach discovers the important regions in the video, and then produces a concise visual summary that is driven by those regions.

maries [118, 122]. Despite promising results, they assume a static background or rely on low-level appearance and motion cues to select what will go into the final summary. However, in many interesting settings, such as egocentric videos, YouTube style videos, or feature films, the background is moving and changing. More critically, a system that lacks high-level information on *which objects matter* may produce a summary that consists of irrelevant frames or regions. In other words, existing methods do not perform *object-driven* summarization and are indifferent to the impact that each object has on generating the "story" of the video.

Instead, I propose to learn category-independent *importance* cues designed explicitly to target the *key objects and people* in the video. The main idea is to leverage novel egocentric and high-level saliency features (including the motion-based measure of object-like regions from Section 1.1.2) to train a model that can predict important regions in the video, and then to produce a concise visual summary that is driven by those regions (see Figure 1.4). By

learning to predict important regions, the system can focus the visual summary on the main people and objects, and ignore the irrelevant or redundant information.

I apply my method to challenging real-world videos captured by users in uncontrolled environments, and process a total of 17 hours of video—orders of magnitude more data than previous work in egocentric analysis. Evaluating the predicted importance estimates and summaries, I find my approach outperforms state-of-the-art saliency measures for this task, and produces significantly more informative summaries than traditional methods unable to focus on the important people or objects.

I discuss my summarization framework for egocentric videos in Section 5.

1.2 Main Contributions

The main impact of my thesis is that it shows how to discover important content in large collections of unlabeled visual data with minimal human supervision. The key components in support of this are:

- *leveraging knowledge about previously learned categories to enable context-aware discovery.* I introduce novel descriptors to encode the familiar object-level context relative to an unfamiliar region, and show that by using them to model the interaction between an image's known and unknown objects we can better detect new visual categories. Furthermore, I formulate context-aware discovery in a self-paced, continuous framework in which easier categories are targeted first. At each cycle of the

continuous discovery process, the system bootstraps a model of object-level context based on the categories it has already discovered. In turn, the detected contextual cues aid in identifying the "harder" categories that remain in the data. In this way, my approach accumulates discovered models over time, benefitting from the context provided by those seen before.

- *discovering key object-like regions to perform accurate segmentations of the unknown objects in images and videos.* I show how to segment a collection of unlabeled images and video frames while exploiting automatically discovered appearance patterns shared between them. I devise novel energy functions amenable to graph cuts that use the discovered share structure to efficiently refine the object segmentations. For video segmentation, I introduce a novel motion-based measure of object-like regions, and a novel partial shape matching technique to localize the object in each frame of the video.

- *a real-world egocentric video summarization approach that is driven by predicted important people and objects.* I introduce novel egocentric features that model the camera wearer's interaction with an object, the camera wearer's gaze, and the frequency of the object-of-interest. I show how to build a regression model using those features to predict important people and objects, irrespective of their category (i.e., in a category-independent way), and to use those predictions to produce a concise visual summary of the data.

- for all new algorithm contributions, I provide extensive experimental results on benchmark datasets and compare against state-of-the-art meth-

ods and relevant baselines for visual category discovery, segmentation, and summarization. I also demonstrate practical applications for discovery, including generalization to novel instances and summarization for tens of hours of real-world videos captured by users in uncontrolled environments.

1.3 Road Map

In the following chapter, I describe related work to my thesis. In Chapter 3, I first present my context-aware discovery framework, which shows how to discover categories from unlabeled images by leveraging knowledge from previously learned categories, and extend it to perform self-paced, continuous discovery. Then, in Chapter 4, I directly build on this work to perform unsupervised object segmentation in images and videos by exploiting the shared discovered visual patterns in the visual collection. In Chapter 5, using many of the techniques I developed for context-aware discovery and collective segmentation, I present a summarization approach that is driven by the discovered important objects and people in egocentric videos. In Chapter 6, I summarize my future work, and finally, in Chapter 7, I conclude by discussing the main contributions of my thesis.

Chapter 2

Related Work

In this chapter, I review related work to the research presented in this thesis. Five main topics of computer vision research are especially relevant. The first thread is visual category recognition, where the task is to learn semantic object categories in images or videos. I focus mainly on unsupervised visual category discovery methods that learn categories with minimal human supervision. Since my thesis proposes unsupervised learning in the context of familiar objects, I also briefly review the state-of-the-art in supervised recognition methods. The second topic is using context to improve object recognition performance. The third line of research is image segmentation where the goal is to partition an image into its constituent objects. The fourth and fifth topics are video summarization and novelty detection, respectively. At the end of each section, I will discuss important similarities and differences of existing work to my proposed approach.

2.1 Visual Category Recognition

The level of supervision is a key implementation choice when devising an object recognition system. In this section, I briefly review related supervised and unsupervised methods.

2.1.1 Supervised Object Recognition

In supervised learning of category models, the system trains on manually prepared image examples. These may be in the form of annotating object parts [30, 42], providing bounding boxes [38, 155], labeling each pixel (i.e., providing a complete image segmentation) [133], or labeling the image with the main object of interest [22, 23]. The system then uses the trained models to perform image classification (i.e., labeling an image with a category) or object detection (i.e., localizing an object in the image) of the learned categories on novel test images.

In order to recognize an object in the image, we must first have a representation of the image. Researchers have explored various image representations. An image can be represented globally as a single feature vector, usually in the form of a histogram that captures color or filter responses at the pixel level. For example, in [141], each image is represented as a single histogram of color counts of all pixels, while in [146] oriented filter responses are summarized to capture coarse texture and spatial layout. Global representations provide simple image representations which lead to efficient matching. However, due to their equal consideration to foreground and background image regions, they are more suitable for scenes in which the global image structure is roughly fixed.

Recent work shows that decomposing an image into local features provides a robust representation which is resilient to object appearance variations, occlusions, and image transformations. In particular, the "bag-of-words" (BoW) model [32, 135] has shown state-of-the-art results in various object recognition tasks. In this model, the system first extracts local image patches with an interest point detector or densely-sampled in a grid pattern, and com-

putes a feature descriptor (e.g., SIFT [101]) for each patch to describe its appearance. The descriptors are designed to be robust to scale, translation, and rotation transformations as well as photometric changes across images. The system then forms a visual "vocabulary" by clustering the descriptors and assigns each patch to its corresponding visual "word" (i.e., cluster center). The bag-of-words model has been applied for various tasks ranging from object recognition [54] to activity recognition [130].

More recently, researchers have explored sparse coding representations [16, 158, 167] for local features, which have shown to outperform the BoW model. In sparse coding, each local feature is encoded as a weighted combination of multiple visual words. This provides a more complete representation of the original features by reflecting their distance to multiple vocabulary words (unlike the BoW model's hard vector quantization). Typically, the weights are then pooled within a region of interest using the max function, which has been shown to provide better discriminability of the features amid high-variance clutter [158].

The critical disadvantage of the above models is that they completely ignore spatial information of the local features. Researchers have proposed ways to alleviate this issue at the feature-level with semi-local features that capture information about local neighborhoods surrounding an interest point [1, 84, 119], or at the image-level by partitioning the image into a grid [85], or describing the image by cumulative histograms of nearby visual word pairs [96]. Given these image representations, image matching is performed with standard metrics (e.g., L1, L2, χ^2 distance, histogram intersection). Often, the resulting metrics are kernelized to allow better separation between categories in feature space, and are inputted to machine learning algorithms such as the

Support Vector Machine (SVM) classifier.

Others have explored deformable part-based representations [4, 31, 41, 43, 45, 95, 160] to explicitly model the part configurations of an object. These methods represent objects as a constellation of parts. In [45], Fergus et al. probabilistically encode object shape with the mutual position of parts, appearance, and scale with patches surrounding the interest points. In [43], Felzenszwalb et al. present a multiscale, deformable part model for object detection. A model for an object consists of a global root filter and several part filters that are automatically learned from annotated bounding boxes on training data. This method has consistently produced detection results that are among the best on the recent PASCAL VOC challenges.

For object detection, a drawback of part-based models is computational cost: a window must be swept across the image at various scales and sizes. Furthermore, the objects (and their parts) are typically represented with rectangular shaped bounding-boxes. Therefore, to alleviate detection costs and to naturally encode the shape and size of an object, researchers have proposed to represent objects with regions [50, 57, 104, 121, 145]. In [104], Malisiewicz and Efros represent objects as segments and use various region-based features that broadly describe shape, texture, color, and location. Similarly, in [145], Todorovic and Ahuja capture region properties such as area, mean pixel value, boundary shape context, and region saliency for object discovery. Finally, in [57], Gu et al. perform object detection and segmentation by representing an image as a "bag-of-regions", where regions are extracted from the segmentation algorithm of [6]. Each region is represented by its color, texture, contour shape, and edge shape.

Current state-of-the-art recognition methods typically require hundreds

of manually labeled images (either in the form of pixel-level labels or image tags). Recognition performance is still well below that of humans but has dramatically improved over the years. For example, on the Caltech-101 dataset [22], which is a benchmark for image classification, the highest accuracy to date is about 74% [70] with 15 training examples per class. In comparison, the first reported result in 2003 was less than 20%. Similarly, detection, classification, and segmentation results on the PASCAL benchmark challenge [37] have shown significant improvements each year since its introduction in 2005. Recent methods use context to improve recognition performance, which I will discuss in Section 2.2.

Discussion: The norm for supervised image labeling methods is forced-choice classification, with the assumption that the training and test sets are comprised of objects from the same pool of categories. These methods aim to either classify the image as a whole, label every pixel with a category, or localize a particular object. However, in many real-world settings, it is not enough to have a preset list of categories. For example, when a robot is exploring a new environment, it will likely encounter many objects that are unfamiliar. In this case, it is more natural for the robot to classify only those objects for which it has trained models, and assign the remaining objects to be "unknown". If there are any repeating instances of the unknown objects across images, they can then be clustered to form new categories. I address this issue in Chapter 3. Specifically, I show how to identify known and unknown objects, and how to group the unknown objects to discover novel categories by using both appearance information as well as contextual information from their relationships to the known objects. To establish any known objects, I will borrow existing supervised methods. The specific methods I use will be

described in the relevant chapters.

2.1.2 Unsupervised Object Discovery

In contrast to the supervised learning setting, which trains models with annotated image data, unsupervised category discovery methods mine for recurring visual patterns in unlabeled images.

Perhaps the first work to approach this problem is that of Weber et al. [160], which introduced the paradigm of "weak supervision" and explored the idea of simultaneously performing feature selection while learning the object's parts-and-structure model. This model learns categories from cluttered, unsegmented class-labeled images; one seeks the parts in each image that best fit all examples sharing the same label. The model parameters and feature selection for each image are learned iteratively using the Expectation Maximization (EM) algorithm [33]. The method in [45] extends this work by improving the part-based model representation and learning algorithm to be more robust to variations in appearance and scale.

Recent work for unsupervised learning of multiple categories has considered ways to discover latent visual themes in images using topic models developed for text, such as probabilistic Latent Semantic Analysis (pLSA) or Latent Dirichlet Allocation (LDA) [44, 99, 120, 134]. The main idea is to use feature co-occurrence patterns in images to recover the underlying distributions (topics) that best account for the data. Having discovered the topics, one can express an image based on the mixture of topics it contains. Early models transferred the notion of text documents containing unordered words to images composed of "visual words". Note that in contrast to the weakly supervised setting above, these approaches use no image annotations whatso-

ever.

Recent extensions use segmentation to reduce the spatial extent of each "document" [127]. Russell et al. decompose each image into multiple-segmentations to increase the likelihood that each object in the image is represented by a segment. The key idea is that segments corresponding to coherent objects will produce strong matches while noisy segments will produce noisy matches. After discovering categories with LDA, the intra-cluster segments are sorted according to their Kullback-Leibler divergence to the cluster topic distribution to reveal the representative category instances. Other extensions show how to incorporate spatial constraints by jointly modeling location and appearance [44] or by feature correspondences [99].

Other approaches treat the unsupervised visual discovery task as an image clustering problem [34, 55, 72]. In [55], Grauman and Darrell use the Pyramid Match Kernel (PMK) [54] to efficiently compute local feature correspondences between all pairs of images, and then apply spectral clustering with the resulting affinity matrix to discover categories. In [72], Kim et al. build a large-scale network using link analysis techniques (e.g., PageRank [20]) that captures the interactions of all visual features across the training set. The system discovers object categories using Normalized Cuts [132], and detects probable object regions for each image. A recent extension [73] shows how to detect the region-of-interests (ROIs) in each unlabeled image by iteratively refining the selected exemplars across the dataset and ROI hypotheses in each image. In [34], a message-passing algorithm propagates non-metric affinities and identifies good exemplars. It takes as input measures of similarity between pairs of data instances and identifies clusters by iteratively exchanging messages that encode the strength of similarity between pairs of instances.

When the image collection is large (e.g., thousands or millions of images), measuring affinity between images (e.g., via local feature correspondences) can be computationally expensive. Thus, some works consider scalable techniques for mining common feature patterns in large image collections [27, 115, 119]. In [119], Quack et al. discover the features that frequently occur on the foreground objects and rarely on the background. They use frequent itemset and association rules from data mining to efficiently mine for recurring patterns. In [27], Chum et al. propose a hashing scheme called Geometric min-Hashing that combines visual appearance with semi-local geometry cues. Unlike most previous work which uses a bag-of-words model, their incorporation of geometry provides a discriminative description of the object while preserving repeatability (high probability that similar instances collide in the same hash bin). They show how to use the method to discover objects from large datasets on the order of 10^5 images.

Discussion: Existing unsupervised category discovery methods assume no prior knowledge of existing categories, and attempt to cluster images or regions using only appearance information. I contend that this is an unnecessarily difficult and even unrealistic scenario, since many labeled images and trained models are available for common categories such as faces, trees, grass, etc. Furthermore, while appearance is a fundamental cue for recognition, it can often be too weak of a signal to reliably detect visual themes in unlabeled, unsegmented images. In particular, appearance alone can be insufficient for discovery in the face of substantial clutter, occluded objects, large intra-category variations, scenes with multiple objects, or low-resolution data.

In contrast to existing methods that discover all categories from scratch, I propose a novel context-aware approach that leverages familiar (i.e., previ-

ously learned) category models to perform discovery. In Chapter 3, we will see that combining appearance with context from familiar object predictions using trained object detectors leads to significant improvements in object category discovery.

2.2 Context-Based Object Categorization

I next discuss related work that uses context for object recognition. I use the term "context" to broadly refer to object interactions in a given image. For supervised methods that learn from labeled images, several types of context have been proposed including global scene context, 3D geometric context, and spatial and co-occurrence context between objects.

The framework in [146] models the relationship between object properties and global scene context. Torralba proposes the "gist" descriptor, which is a holistic, low-dimensional representation of the entire image. The method learns contextual features from a set of training images where the correlation between the statistics of low-level features across the entire scene is used to capture object-specific properties such as its type, location, and scale. For example, for street scenes, a face is likely to be located in the center of the image with a small scale, while for indoor scenes a face may be found near the top of an image having a larger scale. As another example, it can signal that a boat is unlikely to be found in a bedroom.

The approach in [63] captures the overall 3D scene context for object detection by modeling the interdependence of objects, surface orientations, and camera viewpoint. By probabilistically estimating the 3D geometry of the scene in terms of both surface orientations and world coordinates, the authors model the scale and location variance of the objects in the image.

They iteratively refine the probabilistic hypotheses from object detectors and scene geometry estimates, and apply their framework to detect pedestrians and cars in street scenes.

Spatial context is more specific than global scene context; it models neighboring object interactions. In [59], contextual information captured at local-region and global-image scales are aggregated in a probabilistic framework for pixel labeling. The method enforces category label consistency between neighboring pixels except at discontinuities (i.e., object boundaries). Similarly, spatial context from inter-region texton statistics has been explored for supervised image segmentation [133]. The method selects informative regions surrounding a category object. Given a test image, the boosting classifier outputs class posterior probabilities for each pixel. These object estimates are incorporated into a conditional random field (CRF) to enforce smoothness in neighborhood labels along with boundary and color cues. In another approach to spatial context [60], Heitz and Koller combine rigid object (e.g., cars) and amorphous object (e.g., trees, sky) recognition in a unified graphical model framework. Image regions are clustered based on their ability to serve as context for the detection of the rigid objects.

More recently, Malisiewicz and Efros [105] present an exemplar-based model to capture object relationships. The authors present the Visual Memex, which encodes both appearance and 2D spatial layout between object instances. Unlike previous methods that model context between object categories, this method models context between object exemplars. The authors show that their model outperforms category-based baselines for the task of predicting hidden objects in a given scene.

The benefit of high-level semantic context based on objects' co-occurrence

and relative locations has also been demonstrated. Object co-occurrence can be used as a post-processing step to refine object labels [121], with relative spatial location and appearance information to further improve results [50]. The method by Tu [149] iteratively uses appearance information on local image patches and contextual information on classification maps for high-level vision tasks. A recent image decomposition method performs joint inference on objects, regions, and scene geometry to produce state-of-the-art results [53]. In [83], Lazebnik and Raginsky show how to recover contextual information on-the-fly from test images by exploiting the data's statistical redundancy. The approach uses the empirical Bayes technique of statistical inversion to recover a contextual model from the test data instead of learning it from training images.

Finally, researchers have shown that context is especially critical when objects have impoverished appearance due to low resolution [113]. In such cases, recognition results using only appearance can be quite poor; context in the form of co-occurrence, relative layout, and scale is shown to be necessary to obtain high accuracy.

Discussion: Supervised learning of context has proven to be effective for scene understanding and object recognition. Existing methods have shown context to be especially helpful to disambiguate objects that are very similar in appearance (e.g., an electronic screwdriver and blow dryer are similar in appearance, but one appears in a workshop while the other appears in a bathroom). However, there also some limitations. Context alone is usually insufficient to classify an object, and can even be harmful for classifying objects that are out of context (e.g., a television in a grass field). Despite these limitations, overall, the wide range and depth of research in this area have

shown that context plays a valuable role in image understanding tasks.

A central component of my thesis is to explore how high-level semantic context can be used for unsupervised category discovery. In Chapter 3, I show how to identify contextual information in a data-driven manner, by detecting patterns in the relative layout of known and unknown object regions within unlabeled images. Unlike the above supervised methods, my method does not learn about inter-category interactions from a labeled training set. Instead, it discovers the object relationships on the unlabeled test data in a data driven way and uses the context from the familiar categories to group the unfamiliar objects, which leads to improved category discovery performance.

2.3 Image and Video Object Segmentation

In this section, I discuss object segmentation, which is a crucial component of a visual category discovery system. I review state-of-the-art methods in both unsupervised and supervised segmentation for image and video data.

Unsupervised image segmentation methods group pixels that are similar in color and/or texture with the goal that each resulting segment corresponds to a coherent object. Also known as *bottom-up* methods, some representative algorithms are Normalized Cuts [132], Mean-Shift [29], and the hierarchical segmentation method of [6]. While they produce reasonable results for objects with homogeneous appearance (e.g., grass or sky), for more complex objects with heterogeneous appearance (e.g., people, cars, or bicycles) they tend to fail to group each object into a single segment and instead produce oversegmentations.

Therefore, to overcome the limitations of a single bottom-up segmenta-

tion, researchers have proposed to generate *multiple* bottom-up segmentations of the image [62, 103, 121, 127], with the expectation that although some regions will fail to agree with object boundaries, some will be good segments that correspond to coherent objects. Each segmentation is the result of varying the parameters to the segmentation algorithm (i.e., number of regions, image scale).

More recently, a non-parametric method by Russell et al. performs unsupervised image segmentation with data-driven scene matching [128]. The authors exploit the fact that for scene matching, some parts of the image match better than others. The method matches an image to similar scenes in the database, and determines the object boundaries by grouping regions that produce consistently good matches and separating those that produce inconsistent matches.

In order to produce higher quality segments that correspond to semantic objects, researchers have developed methods that incorporate *top-down* category knowledge. Several types of top-down approaches have been proposed. These include combining top-down object detections with bottom-up low-level cues, weakly supervised segmentation, co-segmentation, human-guided segmentation, and category-independent object segmentation, as I describe in the following.

Many approaches that combine supervised object detectors with low-level grouping cues have been proposed [14, 53, 59, 77, 133, 150]. In [150], Tu et al. propose an approach to parse images into object regions by simultaneously performing object recognition and image segmentation. The method combines bottom-up information with top-down generative models using a data-driven Markov Chain Monte Carlo algorithm. In [14], figure-ground segmentation is

performed by minimizing a global cost function that combines both top-down and bottom-up requirements. The top-down approach uses object representations learned from training examples, while the bottom-up approach uses low level features to produce consistent regions that belong to the figure or background. The combined model produces segmentations that agree closely with top-down learned object models yet also agrees with natural image boundaries (i.e., discontinuities). In [77], Kohli et al. combine multiple segmentations in a higher order CRF that incorporates top-down unary potentials from the TextonBoost algorithm [133]. The higher order potentials enforce label consistency in image regions captured by bottom-up segmentations, and are generalizations of the commonly used pairwise contrast-sensitive smoothness potentials that enforce label consistency between neighboring pixels.

To minimize the costs of supervision while capturing the benefits of top-down category knowledge, researchers have proposed *weakly-supervised* segmentation methods that segment the foreground object in cluttered images. These methods assume that each image contains the same foreground object [8, 80, 145, 163], and leverage statistics across the weakly-labeled collection to better identify true object boundaries. For example, in [163], a generative probabilistic model combines bottom-up cues with top-down object cues of shape and pose. The belief in the object's position, segmentation, pose, and size are iteratively refined. The object's appearance is allowed to vary from image to image, which allows significant intra-class variations across the dataset. In [145], Todorovic and Ahuja represent each image as a tree that captures a hierarchical, multi-scale image segmentation. The foreground object segments are retrieved by matching the image trees and finding the best matched subtrees.

Co-segmentation methods aim to segment the foreground object in two or more images [12, 48, 98, 110, 126, 154]. The idea is to simultaneously segment the foreground objects in each image, while enforcing that the appearance (e.g., color) histograms of the foregrounds be similar. Co-segmentation methods usually require that the same specific object appear across the images, and that their backgrounds be distinct in appearance. An extension of these methods initializes the foreground automatically using pLSA [98]. An approach to co-segment clothing regions for person recognition is developed in [48]: it constructs a foreground model using the average appearance of the clothing segments under faces predicted to be of the same person, and applies graph cuts to refine the segment boundaries.

Human-guided foreground-background segmentation methods [12, 125] typically employ graph-cuts [18, 78], which provides efficient approximations for energy minimization tasks for pixel-label assignments with the constraint that labels vary smoothly while preserving sharp discontinuities, e.g., at object boundaries. For these approaches, a user selects some foreground and background pixels to initialize the models for single image segmentation [125], or co-segmentation of multiple images [12].

Category-independent segmentation methods [3, 24, 35, 100] explore finding object-like regions in the image. These works draw on classic Gestalt cues to learn higher-level object-like region cues from labeled data of segmented objects. In particular, interesting approaches to generate and rank an image's multiple figure-ground segmentation hypotheses are explored in [24, 35], with results showing that higher-ranked figure proposals are more likely to be objects in an image.

Whereas most prior work considers segmenting individual static images,

increasingly researchers are exploring techniques for segmenting videos, which can be considered as stacks of images. *Video object segmentation* is often performed in an interactive or supervised way. Interactive methods require a user to annotate object boundaries in some key frames, which are then propagated to other frames while a user stands by to adjust errors [10, 117, 169]. Tracking-based methods attempt to reduce the supervision to a manual segmentation on only the first frame (e.g., [124, 148]). However, all such methods demand user input drawing regions of interest, and may suffer from sensitivity to a user's annotation expertise.

Bottom-up approaches can segment videos in a fully automatic manner, based on cues like motion and appearance similarity. Motion segmentation methods (e.g., [131]) cluster pixels in video using bottom-up motion cues. Recent methods either perform pixel-level segmentation in a spatio-temporal video volume from scratch [56], begin with an image segmentation per frame and then match segments across nearby frames, e.g., [19, 65, 152], or use dense flow to cluster long-term motion trajectories [21]. Without any top-down notion of objects, however, such methods tend to over-segment, yielding regions that taken alone may lack semantic meaning.

Discussion: Top-down methods have shown promising results for image and video segmentation. As part of my goal to accurately discover novel visual categories in images and videos containing multiple objects, I identify a novel setting where *pseudo* top-down cues discovered from a collection of unlabeled images or video frames can be used in conjunction with bottom-up grouping cues for object segmentation. While bottom-up methods cannot guarantee good object segmentations for any given image (even with multiple-segmentations by varying the parameters of the segmentation algorithm), it

will likely produce good object segmentations for *some* images. Using this idea, in Section 4, I show how to group recurring visual patterns in a collection of images to discover representative object instances, and to use them to refine the remaining instances.

2.4 Video Summarization

In this section, I discuss video summarization, focusing specifically on video data. Summarization is particularly relevant to unsupervised visual category discovery: In many real-world settings, it is difficult to have a predefined list of interesting categories (i.e., those that are relevant for the application) in large image collections or videos. Thus, unsupervised summarization methods are useful to automatically extract the key content in the visual data.

Static keyframe selection methods use motion stability from pixel-level optical flow aggregates [164] or color differences between selected frames [170] to choose the most informative frames in the video. The selected keyframes can be used to create a "storyboard" that summarizes the main content of the video [52]: Given a set of keyframes and a set of foreground and background keypoints manually selected by a user, the algorithm creates an extended frame layout, and composites the frames using foreground object mattes.

Video or images sequence summarization can take the form of a single montage of still images [5, 25, 46]. The basic idea in [5] is to create a montage by choosing a reference frame, computing affine transformations between successive frames, and projecting them onto the reference frame. In [25], the system segments the foreground object with user input and automatically selects key-poses based on the object's motion. The key-poses are placed sequentially in a single image that depicts the dynamic motion patterns of the object. In [46],

Freeman and Zhang show how to capture the shape relationships of an object over time in a single image. Both range and image information are used to display the pixels showing the surfaces closest to the viewer among all surfaces seen over the entire sequence.

In contrast to still-image summaries, compact dynamic summaries simultaneously show several actions that occur in different times of the original video [118, 122]. The authors of these works achieve this by minimizing an energy function to maximize activity, minimize overlap, and maximize temporal consistency between the foreground object "tubes" in the video. The drawback is that the framework is limited to videos taken from a static camera in which there is little background motion.

Recent methods aim to discover scene or action categories in visual data captured from a wearable camera [67, 76]. In [67], Jojic et al. develop a novel generative model called the "structural element epitome", which discovers scene categories such as kitchen, office, etc. The images are mapped to a larger epitome matrix, where the amount of overlap indicates image similarity. The model represents image similarity based on the spatial configuration of the objects or scene, rather than their appearance similarity. In [76], Kitani et al. cluster first person sports videos with a stacked Dirichlet process mixture model that infers both the representation of actions (i.e., the motion histogram codebook) and ego-action categories.

Discussion: Despite promising results, existing video summarization approaches assume a static background or rely on low-level appearance and motion cues to select what will go into the final summary. However, in many interesting settings, such as egocentric videos, YouTube style videos, or feature films, the background is moving and changing. More critically, a system that

lacks high-level information on *which objects matter* may produce a summary that consists of irrelevant frames or regions. In other words, existing methods do not perform *object-driven* summarization and are indifferent to the impact that each object has on generating the "story" of the video. In Chapter 5, I propose an approach that learns category-independent *importance* cues designed explicitly to target the *key objects and people* in the video. By learning to predict important regions, my system focuses the visual summary on the main people and objects, and ignores the irrelevant or redundant background clutter.

2.5 Novelty Detection

Finally, I briefly review relevant work in novelty detection, the task of identifying novel, unknown instances in the data; more detailed reviews can be found in [108, 109]. Realistically, we cannot expect to train a system to recognize all possible categories that it will encounter. Therefore, the ability to differentiate between known and unknown instances in the data is critical.

Some methods model statistical properties of the data and estimate whether a test instance belongs to the modeled distribution. In [107], a density function is constructed for a given class, and the probability that a test instance belonging to that class is computed. If the resulting probability is below a threshold, the test instance is classified to be novel. The number of standard deviations from the data mean can also be used to signal novelty [106]. Novelty detection using other statistics (e.g., distance to median) to reject outliers has been explored in [82].

Researchers have explored using classifiers to predict novel instances [81, 129, 161]. One-class SVMs [129] train using examples from a single class and

classify instances that fall outside of the classification boundary to be novel, while minimax probability machines (MPM) [81] identify outliers falling outside of a convex set given the mean and covariance of the data. In [161], Weinshall et al. focus on "incongruent events", which are defined to be conflicting predictions between a general-level and specific-level classifier. As a concrete example, a general classifier can be trained on the face images of many individuals. Another classifier can be trained using a specific smaller set of Einstein's faces. An incongruous instance (i.e., a novel face) is expected to have a smaller posterior probability as estimated by the specific classifier relative to that of the more general classifier.

Novelty detection for video has been of particular interest to the computer vision community, due to its potential use in numerous practical applications ranging from detecting abnormal activities in surveillance data to summarizing key content on a day's worth of web-cam data. Existing unsupervised methods explore tracking objects to determine the abnormality for each object's trajectory [11, 138], detecting abnormal activities with low-level measurements using Hidden Markov Models (HMM) [166], Bayesian topic models [159], or Markov Random Fields (MRF) [74], and clustering to find outlier sequences [58, 174].

Discussion: While novelty detection is a difficult problem, it is critical for the success of recognition methods, whether they be supervised or unsupervised. The method needs to identify which instances of the data are novel and which are familiar. Nonetheless, the problem of distinguishing known instances from unknown instances has not directly been addressed in the recognition literature, as most methods assume forced choice or sparse detections in a binary setting where everything else is treated to be part of a nebulous "background"

class. This is seen clearly in the form of benchmark challenges that have become a central focus in recognition research, such as the Caltech and PASCAL challenges. Instead, my work acknowledges that novel visual data can contain a mix of both familiar and unfamiliar objects. Furthermore, my system need not know how to label every image region, but instead can draw on useful cues from familiar objects to better detect novel ones. In Chapter 3, I show that entropy can provide reasonable estimates for novelty detection and perform discovery only on instances that are deemed to be unknown.

Having summarized related work, I next describe my approach to addressing these problems. The next chapter introduces one of the central ideas of my thesis, that of *context-aware visual category discovery*.

Chapter 3

Context-Aware Category Discovery

I propose to discover novel categories that occur amidst *known* objects within un-annotated images. How could visual discovery benefit from familiar things? The idea is that the relative layout of familiar visual objects surrounding less familiar image regions can help to detect patterns whose correct grouping may be too ambiguous if relying on appearance alone. Specifically, I propose to model the interaction between a set of detected categories and the unknown to-be-discovered categories, and show how a grouping algorithm can yield more accurate discovery if it exploits both object-level context cues as well as appearance descriptors. In addition, I show how this context-aware framework can be enhanced through a self-paced curriculum, where the system focuses on the easiest instances first, and progressively expands its repertoire to include more complex objects. In the ensuing sections, I develop and validate this novel context-aware category discovery approach for two applications: (1) object discovery in natural image collections, and (2) face discovery in consumer photo collections.[1]

[1] I first presented the ideas in this chapter in [89–92].

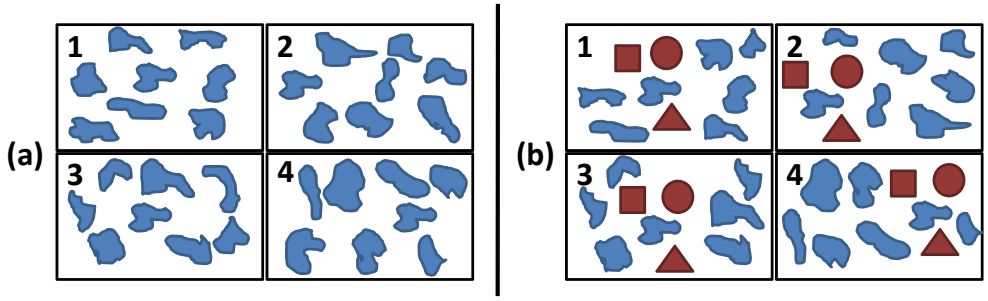


Figure 3.1: Toy example giving the intuition for context-aware discovery. First cover (b) and try to discover the common object(s) that appear in the images for (a). Then look at (b) and do the same. (*Hint: the new object resembles an 'r'.*) **(a)** When all regions in the unlabeled image collection are unfamiliar, the discovery task can be daunting; appearance patterns alone may be insufficient. **(b)** However, the novel visual patterns become more evident if we can leverage their relationship to things that are familiar (i.e., the circles, squares, triangles). I propose to discover visual categories within unlabeled natural images by modeling interactions between the unfamiliar regions and familiar objects.

3.1 Object-Graphs for Context-Aware Discovery

As the toy example in Figure 3.1 illustrates, novel recurring visual patterns ought to be more reliably detected in the presence of familiar objects. Studies in perception show that humans use contextual cues from familiar objects to learn entirely new categories [69]. The use of familiar things as context applies even for non-vision tasks. As a rough analogy for this visual process, take natural language learning: when we encounter unfamiliar words, their definition can often be inferred using the contextual meaning of the surrounding text [162].

To implement this idea, I introduce a context-aware object category

discovery algorithm.[2] My method first learns category models for some set of known or "familiar" categories. Given a new set of completely unlabeled images, it predicts occurrences of the known classes in each image (if any), and then uses those predictions as well as the image features to mine for common visual patterns. For each image in the unlabeled input set, we generate multiple segmentations in order to obtain a pool of regions likely to contain some full objects. We classify each region as known (if it belongs to one of the learned categories) or unknown (if it does not strongly support any of the category models). We then group the unknown regions based on their appearance similarity and their relationship to the surrounding known regions. To model the inter-category interactions, I propose a novel *object-graph* descriptor that encodes the layout of the predicted classes (see Figure 3.2). The output of the method is a set of discovered categories—that is, a partitioning of the unfamiliar regions into coherent groups.

The proposed method strikes a useful balance between recognition strategies at either end of the supervision spectrum. As discussed in Chapter 2, the norm for supervised image labeling methods is forced-choice classification, with the assumption that the training and test sets are comprised of objects from the same pool of categories. On the other hand, the norm for unsupervised recognition is to mine for all possible categories from scratch [55, 72, 86, 99, 127]. In my approach, the system need not know how to label every image region, but instead can draw on useful cues from familiar objects to better detect novel ones.

The key contribution is the idea of context-aware visual category discovery; my technique introduces a method to determine whether regions from

[2] I published the work described in this section in [89, 92].

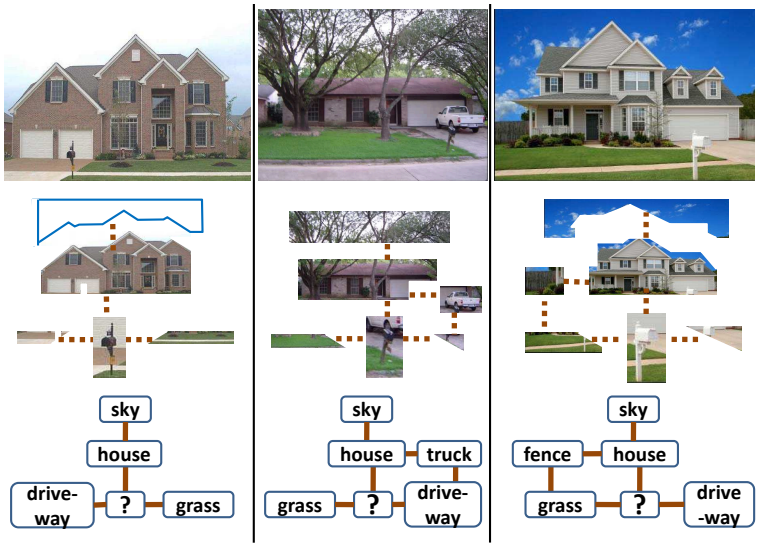


Figure 3.2: We want to encode the layout of known categories relative to an unknown object. In this example, the unknown region is the *mailbox*. The goal is to form clusters on the basis of the similarity of the unknown regions' appearance, as well as the similarity between the structural relationships with surrounding familiar objects.

multiple segmentations are known or unknown, as well as a new object-graph descriptor to encode object-level context. Most importantly, unlike existing approaches, my method allows the interaction between known and unknown objects to influence discovery. I evaluate my approach on five datasets, and show that it leads to significant improvements in category discovery compared to traditional methods that rely only on appearance information and perform discovery from scratch.

3.1.1 Approach

There are three main steps to the approach: (1) detecting instances of known objects in each image while isolating regions that are likely to be

unknown; (2) extracting object-level context descriptions for the unknown regions; and (3) clustering the unfamiliar regions based on these cues. In the following, I describe each step in turn.

3.1.1.1 Identifying Unknown Objects

Any image in the unlabeled collection may contain multiple objects, and may have a mixture of familiar and unfamiliar regions. In order to describe the interaction of known and unknown objects, we must first predict which regions are likely instances of the previously learned categories.

Ideally, an image would first be segmented such that each region corresponds to an object; then we could classify each region and take only those with the most confident outputs as "knowns". In practice, due to the non-homogeneity of many objects' appearance, bottom-up segmentation algorithms (e.g. Normalized Cuts [132]) cannot produce such complete regions. Therefore, following [103, 127], we generate *multiple segmentations* per image, with the expectation that although some regions will fail to agree with object boundaries, some will be good segments that correspond to coherent objects. Each segmentation is the result of varying the parameters to the segmentation algorithm (i.e., number of regions, image scale).

We first compute the confidence that any of these regions correspond to a previously learned category. Assuming reliable classifiers, we will see the highest certainty for the "good" regions that are from known objects, and lower responses on regions containing a mix of known and unknown objects or regions comprised entirely of unknown objects (see Figure 3.3). Using this information to sort the regions, we can then determine which need to be sent to the grouping stage as candidate unknowns, and which should be used to

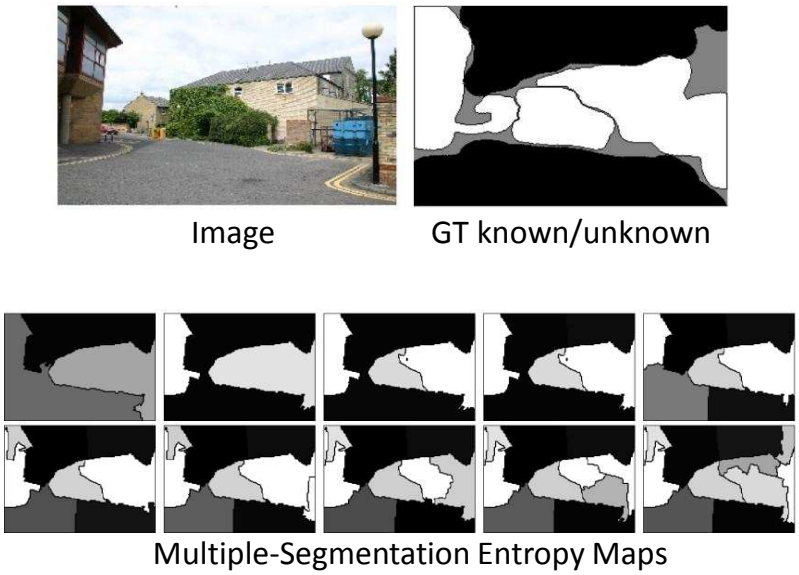


Figure 3.3: An example image, its ground-truth known/unknown label image, and our method's predicted entropy maps for each of its 10 segmentations. For the ground-truth, black regions denote **known** classes (sky, road), and white regions denote **unknown** classes (building, tree). (Gray pixels are "void" regions that were not labeled in the MSRC-v2 ground-truth). In the entropy maps, lighter/darker colors indicate higher/lower entropy, which signals higher/lower uncertainty according to the known category models. Note that the regions with highest uncertainty (whitest) correspond correctly to unknown objects, while those with the lowest uncertainty (darkest) are known. Regions that are comprised of both known and unknown objects are typically scored in between (gray). By considering confidence rates among multiple segmentations, we can identify the regions that are least strongly "claimed" by any known model.

construct the surrounding object-level cues.

We use a labeled training set to learn classifiers for N categories, $C = \{c_1, \ldots, c_N\}$. The classifiers must accept an image region as input and provide a confidence of class membership as output. We combine texture, color, and shape features using the multiple kernel learning (MKL) framework of [9] and

obtain posterior probabilities for any region with an SVM classifier; i.e., the probability that a segment s belongs to class c_i, $\mathrm{P}(c_i|s)$. (Details on the features we use in our results are given in Section 3.1.2.)

The familiarity of a region is captured by the list of these posterior probabilities for each class, which reflect the class-label confidences given the region. Segments that look like a learned category c_i will have a high value for $\mathrm{P}(c_i|s)$, and low values for $\mathrm{P}(c_j|s)$, $\forall j \neq i$. These are the known objects. Unknown objects will have more evenly distributed values among the posteriors. To measure the degree of uncertainty, we compute the entropy E for a segment s, $E(s) = -\sum_{i=1}^{N} \mathrm{P}(c_i|s) \cdot \log_2 \mathrm{P}(c_i|s)$.

The lower the entropy, the higher the confidence that the segment belongs to one of the known categories. Similarly, higher entropy regions have higher uncertainty and are thus more "unknown". This gives us a means to separate the known regions from the unknown regions in each image. Note that entropy ranges from 0 to $\log_2(N)$; we simply select a cutoff threshold equal to the midpoint in this range, and treat regions above the threshold as unknown and those below as known. Figure 3.3 shows the entropy maps we computed for the multiple segmentations from a representative example image. Note the agreement between the highest uncertainty ratings and the true object boundaries.

3.1.1.2 Object-Graphs: Modeling the Topology of Category Predictions

Given the unknown regions identified above, we would like to model their surrounding contextual information in the form of object interactions. Specifically, we want to build a graph that encodes the topology of adjacent

regions relative to an unknown region (see Figure 3.2). Save the unknown regions, the nodes are named objects, and edges connect adjacent objects. With this representation, one could then match any two such graphs to determine how well the object-level context agreed for two candidate regions that might be grouped. Regions with similar surrounding context would have similar graphs; those with dissimilar context would generate dissimilar graphs.

If we could rely on perfect segmentation, classification, *and* separation of known and unknown regions, this is exactly the kind of graph we would construct—we could simply count the number and type of known objects and record their relative layout. In practice, we are limited by the accuracy and confidence values produced by our classifier as well as the possible segments. While we cannot rectify mislabeled known/unknown regions, we can be more robust to misclassified known regions (e.g., sky that could almost look like water) by incorporating the uncertainty into the surrounding object context description.

I propose an *object-graph* descriptor that encodes the likely categories within the neighboring segments and their proximity to the unknown base segment. Rather than form nodes solely based on a region's class label with the maximum posterior probability, we create a histogram that forms localized counts of object presence weighted according to each class's posterior. For each segment, we compute a distribution that averages the probability values of each known class that occurs within that segment's r spatially nearest neighboring segments (where nearness is measured by distance between segment centroids), incremented over increasing values of r (see Figure 3.4). We retain the superpixel segment centered on the unknown segment and remove the remaining segments that overlap with the unknown segment.

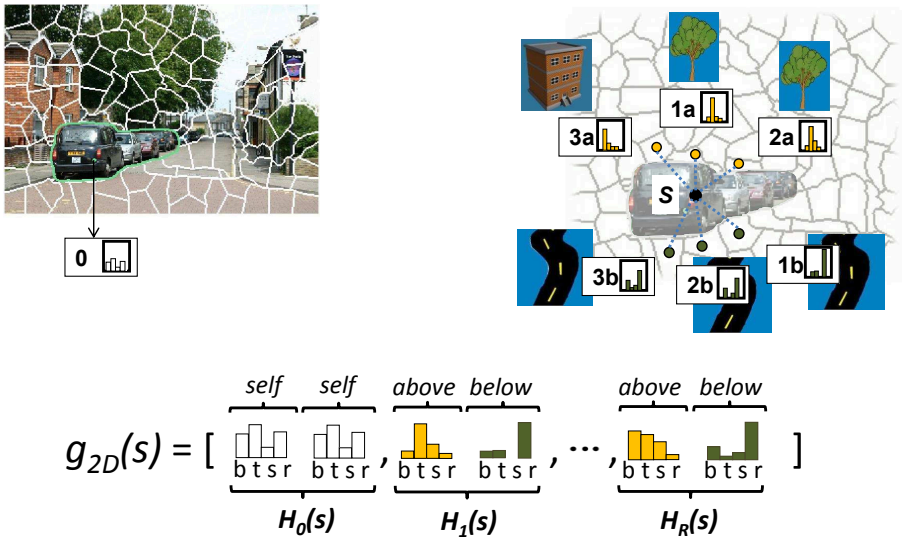


Figure 3.4: Schematic of the proposed 2D object-graph descriptor. The base segment is s. The numbers indicate each region's rank order of spatial proximity to s for two orientations, *above* and *below*. The circles denote each segment's centroid. In this example, there are four known classes: building (**b**), tree (**t**), sky (**s**), and road (**r**). Each histogram $H_r(s)$ encodes the average posteriors for the r neighboring segments surrounding s from above or below, where $0 \leq r \leq R$. (Here, $R = 3$, and bars denote posterior values.) Taken together, $g_{2D}(s)$ serves as a soft encoding of the likely classes that occur relative to s, from near to far, and at two orientations.

Specifically, for each unknown segment s, we compute a series of histograms using the posteriors computed within its neighboring superpixels. Each component histogram $H_r(s)$ accumulates the average probability of occurrences of each class type c_i within s's r spatially nearest segments for each of two orientations, *above* and *below* the segment. We concatenate the component histograms for $r = 0, \ldots, R$ to produce the final object-graph descriptor:

$$g_{2D}(s) = [H_0(s), H_1(s), \ldots, H_R(s)], \tag{3.1}$$

where $H_0(s)$ contains the posteriors computed within s's central superpixel.

The result is an $((R+1)\cdot 2N)$-dimensional vector, where N denotes the number of familiar classes. Note that higher values of r produce a component $H_r(s)$ covering a larger region, and the descriptor softly encodes the surrounding objects present in increasingly further spatial extents. Our representation can detect partial context matches (i.e., partially agreeing spatial layouts), since the matching score between two regions is proportional to how much their context agrees. Due to the cumulative construction, discrepancies in more distant regions have less influence.

There are a couple of implementation details that will help ensure that similar object topologies produce similar object-graph descriptors. First, we need to maintain consistency in the size and relative displacement of nodes (regions) across different object-graphs; to do this, we use superpixel segments as nodes (typically about 50 per image). Their fairly regular size and shape tessellates the image surrounding the unknown region well, which in turn makes a centroid-based distance between nodes reliable.[3] As usual, the superpixels may break non-homogeneous objects into multiple regions, but as long as the oversegmentation effect is fairly consistent in different images (e.g., the dark roof and light wall on the building are often in different superpixels), the object-graph will avoid misleading double-counting effects. Empirically, we have observed that this consistency holds.

Second, we need to obtain robust estimates of the known objects' posterior probabilities, and avoid predicting class memberships on regions that

[3]Note that our descriptor assumes images have similar scene depth, and thus that the relative placement of surrounding objects depends only on the scale of the object under consideration (as do most existing recognition methods using object co-occurrence context, e.g., [60, 133]). In Section 3.1.1.3, we relax this assumption to encode a 3D object-graph descriptor that utilizes scene depth.

are too local (small). For this we exploit the multiple segmentations: we estimate the class posteriors for each segment, then for each image, we stack its segmentation maps, and compute a per-pixel average for each of the N posterior probabilities. Finally, we compute the posteriors for each superpixel node by averaging the N-vector of probabilities attached to each of its pixels. Note that this allows us to estimate the known classes' presence from larger regions, but then summarize the results in the smaller superpixel nodes.

We select a value of R large enough to typically include all surrounding regions in the image. We limit the orientations to above and below (as opposed to also using left and right) since we expect this relative placement to have more semantic significance; objects that appear side-by-side can often be interchanged from left-to-right (e.g., see the mailbox example in Figure 3.2). For images that contain multiple unknown objects, we do not exclude the class-probability distributions of the unknown regions present in another unknown region's object-graph. Even though the probabilities are specific to known objects, their distributions still give weak information about the appearance of unknown objects.

Previous methods have been proposed to encode the appearance of nearby regions or patches [60, 86, 133, 153], however our object-graph is unique in that it describes the region neighborhood based on object-level information, and explicitly reflects the layout of previously learned categories. In Section 3.1.2 we demonstrate the comparative value for the discovery task. Relative to existing graph kernels from the machine learning literature [51, 71], our approach allows us to represent object topology without requiring hard decisions on object names and idealized segmentations.

3.1.1.3 3D Object-Graphs

In this section, I show how to extend the object-graph descriptor to model 3D spatial layout.

The 2D object-graph described thus far is often sufficient to model the scene context and relative locations of objects, since general photographer biases lead to similar 2D layouts across image instances (e.g., sky is above, ground is below, camera distance to objects is within a similar range). However, in some cases the spatial relationships between objects in the 2D image plane can appear to be quite different from their true relationships in the 3D world. Explicitly modeling the 3D scene geometry can resolve potential discrepancies in spatial relationships between objects in images with different scene depths. For example, in a close-up photo of a car, a part of the road that is actually behind the car could be placed *above* the car in the 2D image plane. By modeling scene geometry, we can infer that the road is actually *below* the car in the 3D world plane, and thus make its scene context comparable to that of a car in a broader street scene image. Thus, to account more explicitly for the depth ordering of objects in the scene, we next introduce a 3D variant of the object-graph that uses single-view estimates of occluding boundaries to estimate the proximity and relative orientations of surrounding familiar objects.

Given a depth ordering of the objects in the image, the object-graph descriptor can be adapted to capture the relationships between the objects in the 3D world. To estimate depth, we employ the method of [64], which infers occlusion boundaries in a single image using edge and region cues together with 3D surface and depth cues. It computes a segmentation of the image, classifies each region as belonging to either the *sky*, *ground*, or *vertical* planes,

and produces pixel-level depth estimates. We compute a single depth estimate for each region by averaging its pixel-level depth values.

To create our 3D object-graph descriptor, we again encode the likely categories within the neighboring segments and their proximity to the unknown base segment with cumulative posterior probability histograms. However, unlike the 2D object-graph descriptor, which ranks neighboring regions based on their centroid distances in the image plane, the 3D object-graph descriptor measures region nearness using 3D depth estimates, explicitly accounting for the surface planes (e.g., *sky*, *ground*, and *vertical*) that each region resides in. Furthermore, we use regions rather than superpixels for the 3D object-graph nodes since (1) the regions generated using [64] cover objects quite well, and (2) we no longer assume similar scene depth across images and thus do not benefit from the superpixels' consistency in size and relative displacements. Instead, for each surface plane, we accumulate the posterior probability distributions of neighbors in increasing displacement in depth (as measured by L2 distance) relative to the central unknown object. We then concatenate the posterior distributions to create a single 3D object-graph descriptor for the unknown region:

$$g_{3D}(s) = [H_{sky}(s), H_{ground}(s), H_0(s), \ldots, H_R(s)]. \tag{3.2}$$

Figure 3.5 shows a schematic of the 3D object-graph descriptor.

In Section 3.1.2.6 we compare the 2D and 3D object-graph variants. The performance of the 3D object-graph is influenced by the accuracy of the underlying scene depth estimate algorithm. In our experiments, we observe that the method of [64] produces best results for scene images with multiple objects (including sky and ground) and a visible horizon, and it is less reliable

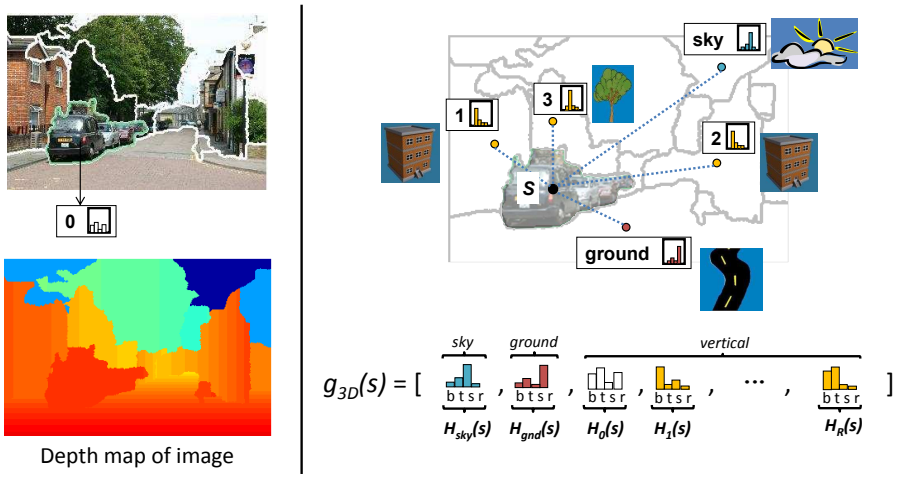


Figure 3.5: Schematic of the 3D object-graph descriptor. The base segment is s. The numbers indicate each region's rank order of estimated depth proximity to s. We compute the depth map using the method of [64]. $H_{sky}(s)$ and $H_{gnd}(s)$ encode the posteriors for the sky-plane and ground-plane segments, respectively. Each $H_r(s)$ encodes the average posteriors for the r neighboring vertical-plane segments surrounding s, where $0 \leq r \leq R$. Taken together, the object-graph $g_{3D}(s)$ serves as a soft encoding of the likely classes that occur relative to s, from near to far in terms of scene depth, and at three surface orientations.

for images of close-up objects. While we focus on single-view estimates of relative depth to avoid making assumptions about the original sensor, of course if stereo data were available our method could similarly exploit it to form the 3D object-graph descriptor.

3.1.1.4 Category Discovery Amidst Familiar Objects

Now that we have a means to compute object-level context, we can combine this information with region-based appearance to form homogeneous groups from our collection of unknown regions. We define a similarity function between two regions s_m and s_n that includes both region appearance and

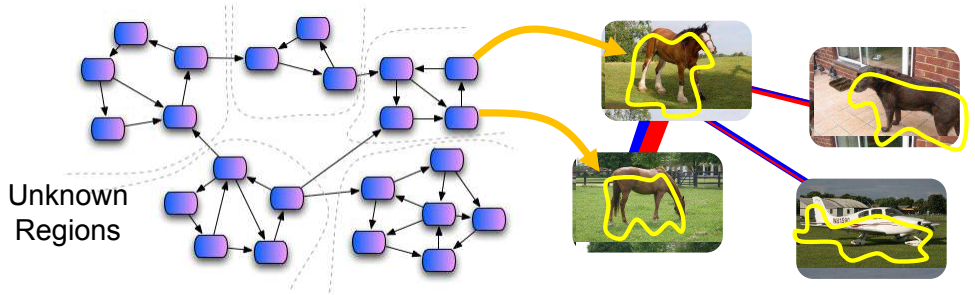


Figure 3.6: The method forms clusters on the basis of the similarity of the unknown regions' appearance (given by K_{app}), as well as the similarity between the structural relationships with surrounding familiar objects (given by our object-graph descriptor $K_{obj-graph}$). The nodes indicate the unknown regions, the arrows indicate the affinity between nodes, and the dotted lines separate discovered clusters.

known-object context:

$$K(s_m, s_n) = \frac{1}{|\mathrm{u}|} \sum_{\mathrm{u}} K_{\chi^2}\left(a_{\mathrm{u}}(s_m), a_{\mathrm{u}}(s_n)\right) + K_{\chi^2}\left(g(s_m), g(s_n)\right), \qquad (3.3)$$

where $g(s_m)$ and $g(s_n)$ are the object-graph descriptors (either of the 2D or 3D variants), and each $a_{\mathrm{u}}(s_m)$ and $a_{\mathrm{u}}(s_n)$ denotes an appearance-based feature histogram extracted from the respective region (which will be defined in Section 3.1.2). Each $K_{\chi^2}(\cdot, \cdot)$ denotes a χ^2 kernel function for two histogram inputs: $K_{\chi^2}(x, y) = \exp(-\frac{1}{2}\sum_i \frac{(x_i - y_i)^2}{x_i + y_i})$, where i indexes the histogram bins. While we have fixed the relative weighting between the appearance and context terms, one could instead learn the weights in an unsupervised manner through multiple kernel clustering [171]. I discuss this issue in more detail in Section 3.4 at the end of this chapter.

We compute affinities between all pairs of unknown regions to generate an affinity matrix, which is then given as input to a clustering algorithm

Input: Set of classifiers for N known category models, set of novel unlabeled images, and k.
Output: Set of k discovered categories (clusters).
1. Obtain multiple segmentations for each image.
2. Compute posteriors for each region. (Section 3.1.1.1)
3. Compute the entropy for each region to classify as "known" or "unknown". (Section 3.1.1.1)
4. Construct an object-graph for each unknown region. (Sections 3.1.1.2 & 3.1.1.3)
5. Compute affinities between unknown regions with the object-graph and appearance features, and cluster to discover categories. (Section 3.1.1.4)

Algorithm 1: The object-graphs discovery algorithm

to group the regions (see Figure 3.6). We use the spectral clustering method developed in [111], which clusters instances using the eigenvectors of the Laplacian matrix of the data. We choose this method due to its simplicity and ability to group instances that do not form convex regions in feature space. Because we use multiple segmentations, if at least one "good" segment of an unknown object comes out of an image, then it may be matched and clustered with others that belong to the same category. Since our unknown/known separation for novel images may be imperfect, some discovered groups may contain objects that actually belong to a known class. Importantly, since affinity can be boosted by either similar appearance *or* similar context of known objects, we expect to be able to discover objects with more diverse appearance.

To recap, I summarize the main steps of the approach in Algorithm 1.

3.1.2 Results

In this section, I (1) evaluate our method's discovery performance and compare against two appearance-only baselines, (2) analyze our entropy-based known-unknown separation measure, (3) compare the object-graph with an

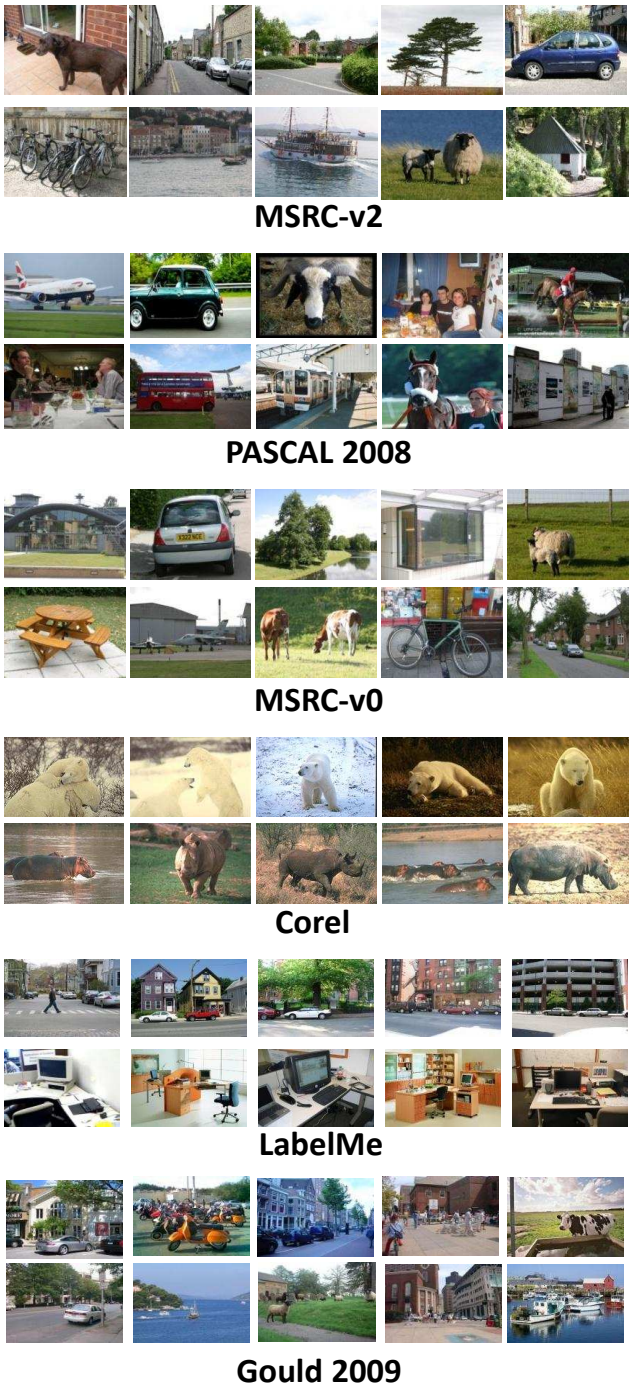


Figure 3.7: Example images of the datasets used for context-aware object discovery.

appearance-based context baseline, (4) compare the 2D and 3D object-graph variants, and (5) show qualitative examples of real object-graphs and discovered categories.

Datasets We validate our approach with five datasets: MSRC-v0, MSRC-v2, PASCAL VOC 2008, Corel, and Gould 2009 [53] (see Figure 3.7 for examples). We want to evaluate how sensitive our method is with respect to which classes are considered familiar (or unfamiliar), and how many (or few) objects are in the "known" set of models. Thus for each dataset, we form multiple splits of known/unknown classes, for multiple settings of both the number of knowns (N) and the number of true unknowns present.

Implementation details We use Normalized Cuts [132] for segmentation, and vary the number of segments from 3 to 12 to obtain 10 segmentations (75 segments) per image. To form the appearance descriptor $a_{\mathrm{u}}(s)$ for a region s, we use several types of bag-of-features histograms: Texton Histograms (TH), Color Histograms (CH), and pyramid of HOG (pHOG) [15]. These features encode region texture, color, and shape, respectively. To compute class probabilities, we use one-vs-all SVM classifiers trained using MKL, and obtain posteriors using [116]. For our 2D object-graph descriptor, we generate an over-segmentation with roughly 50 superpixels per image, and fix $R = 20$. We normalize each $a_{\mathrm{u}}(s)$, $g_{2D}(s)$, and $g_{3D}(s)$ to sum to 1.

Evaluation metrics We use both *purity* [142] and *mean Average Precision* (mAP) to quantify accuracy. Purity rates the coherency of the clusters discovered: $purity = \sum_{i=1}^{k} \frac{N_i}{N} \max_j N_{i,j}$, where N is the total number of instances, N_i is the total number of instances in cluster i, and $N_{i,j}$ is the total number of instances with ground-truth label j in cluster i. mAP reflects how well we

have captured the affinities between intra-class versus inter-class instances (independent of the clustering algorithm): $mAP = \frac{1}{N}\sum_{q=1}^{N}\int_0^1 p_q(r)\,dr$, where N is the total number of instances, $p_q(r)$ is the precision for instance q at recall rate r, and $precision = \frac{tp}{tp+fp}$ where tp and fp are the number of true positive and false positive retrieved instances, respectively. We only consider regions with ground-truth labels (i.e., no "voids" from MSRC). To score an arbitrary segment, we consider its ground truth label to be that which the majority of its pixels belong to.

These metrics reward discovery of object parts as well as full objects (e.g., we would get credit for discovering cow heads and cow legs as separate entities). This seems reasonable for the unsupervised category discovery problem setting, given that the part/object division is inherently ambiguous without external human supervision. We report purity values as a function of the number of clusters, since we cannot assume prior knowledge on the number of novel categories. Since the spectral clustering step [111] uses a random initialization, we average all results over 10 runs.

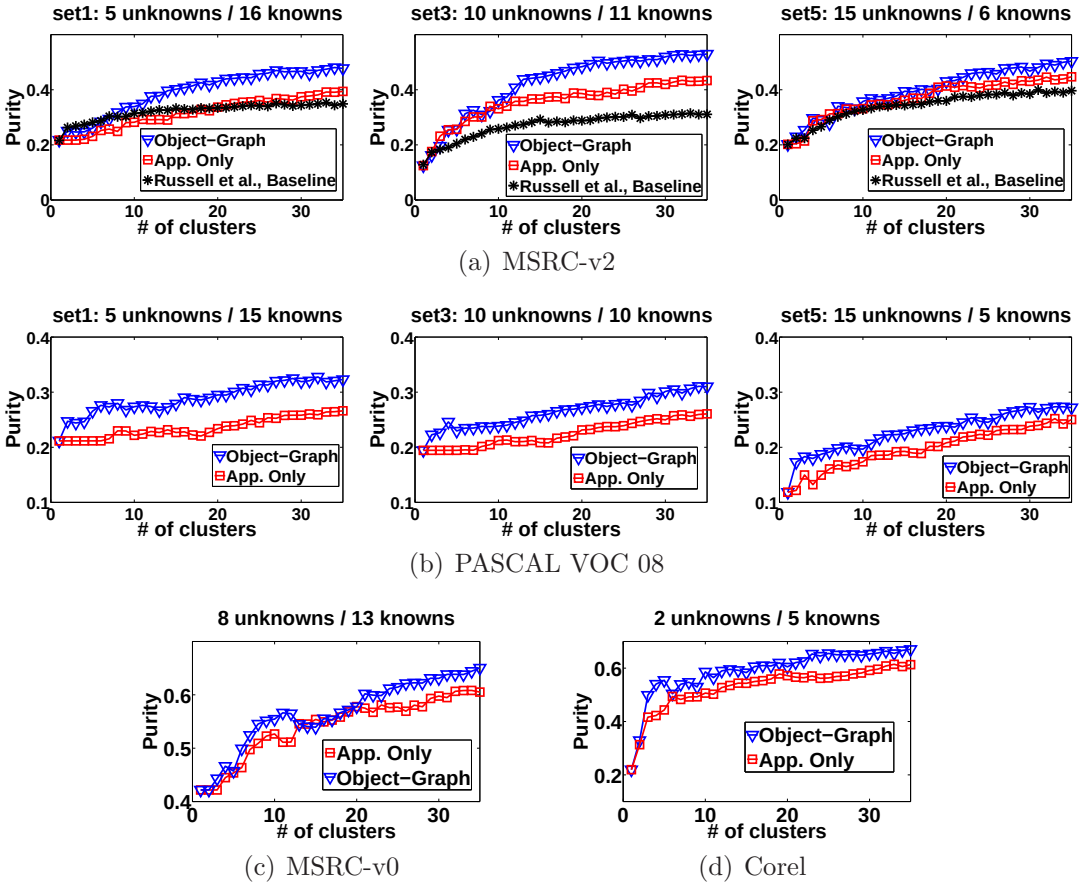
3.1.2.1 Object Discovery Accuracy

To support the main claim that the detection of familiar objects should aid in category discovery, I first evaluate how much accuracy improves when we form groups using appearance together with the object-graph, versus when we form groups using appearance alone. I thus generate two separate curves for purity scores: (1) an appearance-only baseline where we cluster unknown regions using only appearance features (App. only), and (2) our approach, where we cluster using both appearance and contextual information (Object-Graph).

Since our evaluation scenario necessarily differs from earlier work in unsupervised discovery, it is not possible to directly compare the output of our method with previously reported numbers: our method assumes some background knowledge about a subset of the classes, whereas existing discovery methods assume none. However, our appearance-only baseline shows the limits of what can be discovered using conventional approaches for this data, since previous unsupervised methods all rely solely on appearance [55, 72, 86, 127]. Furthermore, we also generate comparisons with the state-of-the-art LDA-based discovery method of Russell et al. [127] using the authors' publicly available code. This method assigns regions to clusters using the LDA topic model with SIFT appearance features. To our knowledge, theirs is the only other current unsupervised method that tests with datasets containing multiple objects per image, making it the most suitable method for comparison. In all results, our method and the baselines are applied to the same pool of segments (i.e., those our method identifies as unknown).

Figure 3.8 shows the results using the 2D object-graph on four datasets. Our model significantly outperforms the appearance-only baselines. These results confirm that the appearance and object-level contextual information complement each other to produce high quality clusters. Note the consistency in our method's improvement over the baselines with respect to the number of clusters.

Upon examining the relative performance on different known/unknown splits, we found that discovery performance depends to a limited extent on which categories are known, and how many. For example, both our method and the baseline have stronger discovery performance on MSRC-v2 set2 than on set1. This can be attributed to the fact that the unknowns in set2 are *grass,*



(a) MSRC-v2

(b) PASCAL VOC 08

(c) MSRC-v0

(d) Corel

Figure 3.8: Discovery accuracy results given by purity rates for all four 2D object-graph datasets as a function of k. Higher curves are better. We compare our 2D object-graph approach (Object-Graph) with appearance-only baselines. The discovered categories are more accurate using the proposed approach, as the familiar objects nearby help us to detect region similarity even when their appearance features may only partially agree.

	Building	Tree	Cow	Airplane	Bicycle
Our full model	**0.32**	**0.36**	**0.41**	**0.36**	0.21
App. only	0.27	0.33	0.20	0.21	0.10
Obj-Graph only	**0.32**	0.27	0.37	0.32	**0.24**

Table 3.1: Mean Average Precision (mAP) on MSRC-v2 set1 unknowns.

sky, water, road, and *dog*, which have strong appearance features and can be discovered reliably without much contextual information. When the ratio between the number of unknown categories to known categories increases (from left to right in Figure 3.8 (a) and (b)), there is a decrease in the information provided by the known object-level context, and consequently we find that our improvements over the baseline eventually have a smaller margin (see rightmost curves in (a) and (b), where only 5 or 6 objects are known). Overall, however, we find that the improvements are quite stable: across the 12 random splits tested for the MSRC and PASCAL, our method never detracts from the accuracy of the baseline.

To directly evaluate how accurately our 2D object-graph affinities compare the regions, we analyze the mean Average Precision (see Table 3.1). Our full model noticeably outperforms the appearance-only baseline in all categories. In fact, the object-graph descriptor alone (with no appearance information) performs almost as well as our full model. For bicycles, the affinities obtained using only appearance information are weak, and thus the full model actually performs slightly worse than the object-graph descriptor in isolation. We also see that our model's largest improvement occurs for the cow class (high appearance variance), whereas it is smaller for trees (low appearance variance). This makes sense because context is more helpful when grouping instances from a category with high appearance variation.

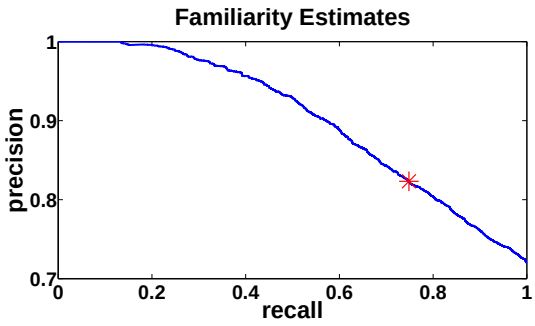


Figure 3.9: Precision-recall curve for known versus unknown decisions on the MSRC-v2 set1; the star denotes the cutoff (half of the maximum possible entropy value).

3.1.2.2 Impact of Known/Unknown Decisions

Next, I evaluate how well my method predicts known versus unknown regions. Figure 3.9 shows the precision-recall curve for our known-unknown decisions on the MSRC-v2. For this, we treat the known classes as positive, and the unknown classes as negative, and sort the regions by their entropy scores. The red star indicates the precision-recall value at $\frac{1}{2}\max E(s)$. With this (arbitrary) threshold, the regions considered for discovery are almost all true unknowns (and vice versa), at some expense of misclassifying unknown and known regions. Adjusting the "knob" on the threshold produces a tradeoff between the number of true unknowns considered for discovery versus the number of true knowns treated as unknowns. Learning the optimal threshold depends on the application, and for our problem setting, $\frac{1}{2}\max E(s)$ suffices.

As discussed in Chapter 2, novelty detection is a difficult problem. Our use of multiple segmentations provides some robustness to this issue in that it allows us to choose the regions that are least likely to be claimed by any known model. We will see the lowest entropy for the regions that are from known

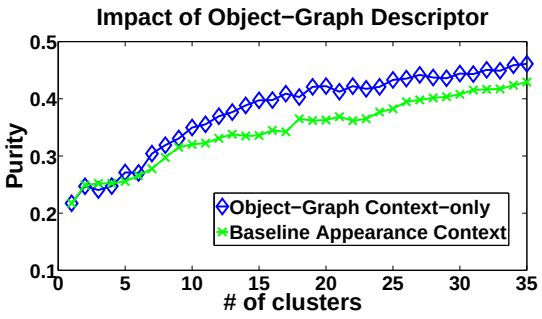


Figure 3.10: Comparison of the 2D object-graph descriptor to a "raw" appearance-based context descriptor.

objects, higher values on regions containing a mix of known and unknown objects, and the highest entropy for regions comprised entirely of unknown objects, as discussed in Section 3.1.1.1.

3.1.2.3 Impact of the Object-Graph Descriptor vs. Raw Appearance

I next evaluate how our 2D object-graph descriptor compares to a simpler alternative that directly encodes the surrounding appearance features. Since part of our descriptor's novelty rests on its use of object-level information, this is an important distinction to study empirically. We substitute class probability counts in the object-graph with raw feature histogram counts. Figure 3.10 shows the result on the MSRC-v2. Our object-graph performs noticeably better than the baseline, confirming that directly modeling class-interactions instead of surrounding appearance cues can improve discovery.

In addition to improved accuracy, our descriptor also has the advantage of lower dimensionality. The object-graph requires only $R \cdot 2N$-dimensional vectors for each unknown region, whereas the appearance baseline requires $R \cdot 2Q$-dimensional vectors, for Q texton + color + pHOG bins. In this case,

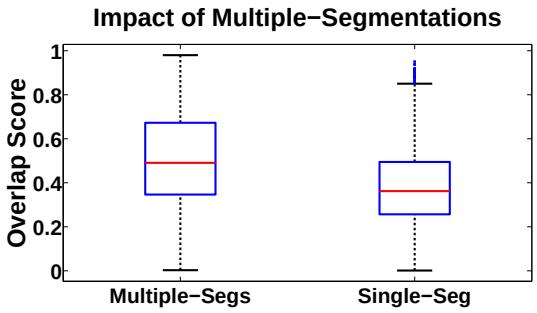


Figure 3.11: Maximal segmentation accuracy attainable per object using multiple segmentations versus a single segmentation.

our object-graph is about 70 times more compact.

3.1.2.4 Impact of Multiple Segmentations

I next study the impact that multiple segmentations have on providing candidate object regions that agree well with true boundaries. For each object in the image, we take the region from the pool of bottom-up multiple segmentations that has the highest overlap score with its ground-truth segmentation to compute the maximum overlap score [7]. We compare against taking regions from a single segmentation baseline that generates seven segments per image (the average number of regions per segmentation in the set of multiple segmentations).

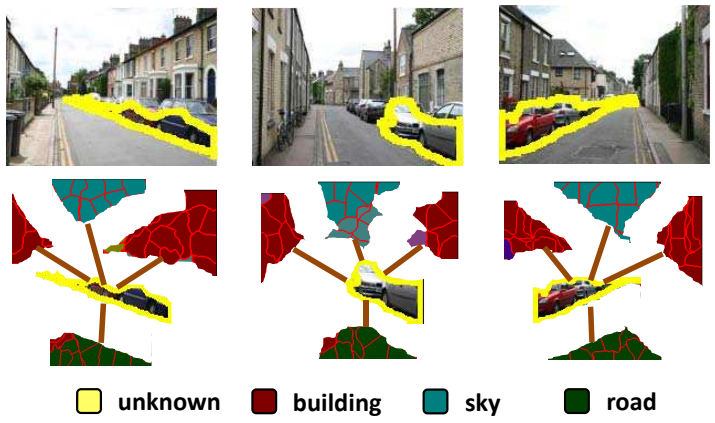
Figure 3.11 shows the result on the MSRC-v2. The regions in the pool of multiple segmentations provide significantly better candidates for representing objects than those in the pool of the single segmentation baseline. The median score for the multiple segmentation regions is about 0.5, which indicates that the best candidates have high overlap with true object regions. This result corroborates the findings in [127].

While the result highlights the importance of generating multiple segmentations, it also reveals the limitations of bottom-up segmentations for discovery since there is clearly room for improvement in segmentation quality. In Chapter 4, I explore how discovered top-down patterns in the unlabeled image collection can be used to refine the initial segmentations, so that we are not restricted to discovering patterns among the bottom-up segments.

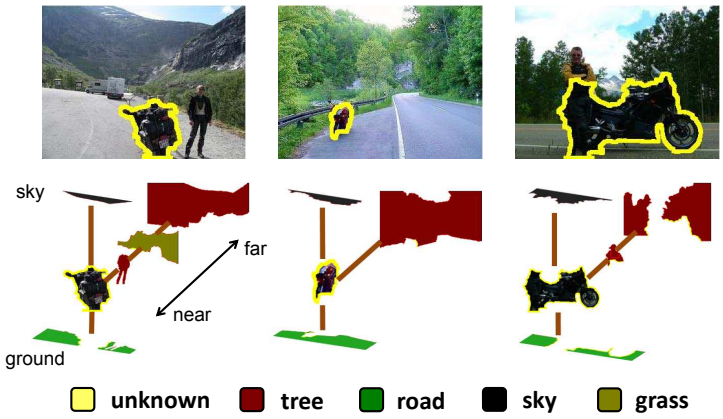
3.1.2.5 Example Object-Graphs

Thus far, I have established that the object-graph can boost discovery performance. It is also interesting to look more closely at what the graphs are actually capturing. Figure 3.12 (a) and (b) show examples of 2D and 3D object-graphs generated using our approach, respectively. The 2D object-graphs are generated on the MSRC-v0 dataset with *building, grass, sky, road, mountain, water, flower,* and *leaf* as knowns, and the 3D object-graphs are generated on the Gould 2009 dataset with *sky, tree, road, grass, water, building,* and *mountain* as knowns.

Our method correctly identifies the car and motorbike regions as unknowns (those with yellow boundaries), and produces accurate descriptions of the surrounding familiar object-level context. To visualize the familiar category posterior distributions in each surrounding region node, we label each node with the category that produces the *maximum* posterior probability. Furthermore, for the 2D object-graph (Figure 3.12 (a)), we group the nodes according to their predicted labels. However, note that for the actual implementation, we compute the object-graphs by taking the full posterior distributions, and connect each superpixel node to the central unknown region. Our method produces very similar object-graphs for the unknown regions, which



(a) 2D Object-Graphs



(b) 3D Object-Graphs

Figure 3.12: Examples of 2D and 3D object-graphs generated by our method. Our method correctly identifies the car and motorbike regions, in **(a)** and **(b)** respectively, as unknowns (regions with yellow borders), and produces accurate descriptions of the surrounding familiar object-level context. Our method groups the unknown regions, despite their variable appearance, due to their strong agreement in object-graphs. Note that the surrounding regions that do not belong to a familiar category cannot be classified correctly (e.g., the person regions in **(b)**); however, the *distribution* of their known-category posterior probabilities still provide meaningful appearance information that lead to accurate object-graph descriptions.

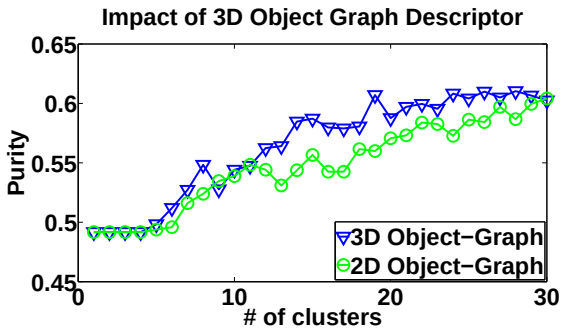


Figure 3.13: Comparison of the 3D object-graph descriptor to the 2D object-graph descriptor on the Gould 2009 dataset. The 3D descriptor is able to exploit the scene geometry prevalent in the data, to produce more accurate descriptions of spatial context.

enables them to be grouped despite their heterogeneous appearance.

3.1.2.6 Modeling Scene Depth with 3D Object-Graphs

I next evaluate the impact that the 3D object-graph has on discovery. We evaluate our method on the Gould 2009 dataset, since it has previously been tested for computing depth estimates and is appropriate for modeling 3D scene structure from single views. The other datasets contain some images of close-up objects, which the method of [64] does not handle as well. While we choose to test on single images to demonstrate the flexibility of our approach, stereo data would also be amenable when available, since their disparity maps provide depth information.

As before, we perform discovery on the regions that are deemed to be unknown. In addition, we remove any misclassified regions, i.e., true known regions misclassified as unknown, in order to isolate our analysis on the 2D versus 3D scene context description without any side effects caused by those errors. We consider in total seven neighboring regions: one region from the

sky plane, one region from the ground plane, and five neighboring regions in the vertical plane; empirically, we find that the regions generated from the occlusion boundary segmentation algorithm [64] tend to correspond well with the true number of objects in the image.

Figure 3.13 shows the results, compared against the 2D object-graph descriptor on the same set of unknown regions. The 3D object-graph outperforms the 2D object-graph. This can be attributed to the fact that the dataset is mostly composed of natural scene images, where 3D geometry estimates are more reliably computed by [64]. Furthermore, the 3D object-graph strictly matches regions that belong to the same geometric plane (e.g., sky regions are only compared against each other). In this way, the scene structure is retained in the comparison, providing matching scores that are more robust to camera pose variations. Nonetheless, the 2D object-graph still performs quite well, which indicates that modeling spatial layout in the 2D image plane is often sufficient to provide reliable object-level context descriptions.

3.1.2.7 Discovered Categories: Qualitative Results

Finally, I provide qualitative image examples of what our method discovers. Figures 3.14 and 3.15 shows examples of discovered categories from the 3457 MSRC-v0 images using our 2D object-graph approach, for $k = 30$. We show two sets of qualitative results using different methods to generate the candidate object regions: one using Normalized Cuts and the other using the hierarchical segmentation engine of [6]. This lets us analyze the influence that higher quality segmentations have on qualitative discovery accuracy. The cluster images are sorted by their degree (top left is highest, bottom right is lowest) as computed by the affinity matrix: $D(s_m) = \sum_{l \in L} K(s_m, s_l)$, where

L denotes the cluster containing segment s_m. We show the top 20-30 regions for each cluster, removing overlapping regions and limiting to only one region per image.

The resulting groups show good semantic consistency (here, we see windows, cars, bicycles, trees, chimneys, sheep, and cows). Notably, our clusters tend to be more inclusive of intra-class appearance variation than those that could be found with methods that rely only on appearance, such as [55, 72, 86, 127]. For example, note the presence of side and frontal/rear views in the sheep, car, and cow clusters (see first row in Figure 3.14 and second to fourth rows in Figure 3.15), and the distinct types of windows that get grouped together (see third row in Figure 3.14 and last row in Figure 3.15). Our algorithm also discovers cars and buildings as a single category, which often co-occur and are segmented together (see fifth row in Figure 3.15). This makes sense since their regions have similar appearance and similar surrounding context (i.e., road below). The segmentation quality of the discoveries made using the regions from [6] is better than those made using Normalized Cuts, which shows that better candidate object regions lead to higher quality discoveries. Overall, these results indicate that boosting affinities using both appearance and object-level context lead to semantically coherent discoveries.

So far, I have described how to discover a fixed number of categories at once in a batch setting. In the next section, I will show how to progressively discover categories in order of predicted "easiness", which will lead to more accurate category groupings.

Figure 3.14: Examples of discovered categories for the MSRC-v0 using Normalized Cuts regions. Our clusters show good semantic consistency and tend to be more inclusive of intra-class appearance variation than those found using appearance alone. For example, note the presence of side and frontal/rear views in the car clusters, and the distinct types of windows that get grouped together. When clustering with appearance alone, it would not be possible to realize the consistency across such varying viewpoints.

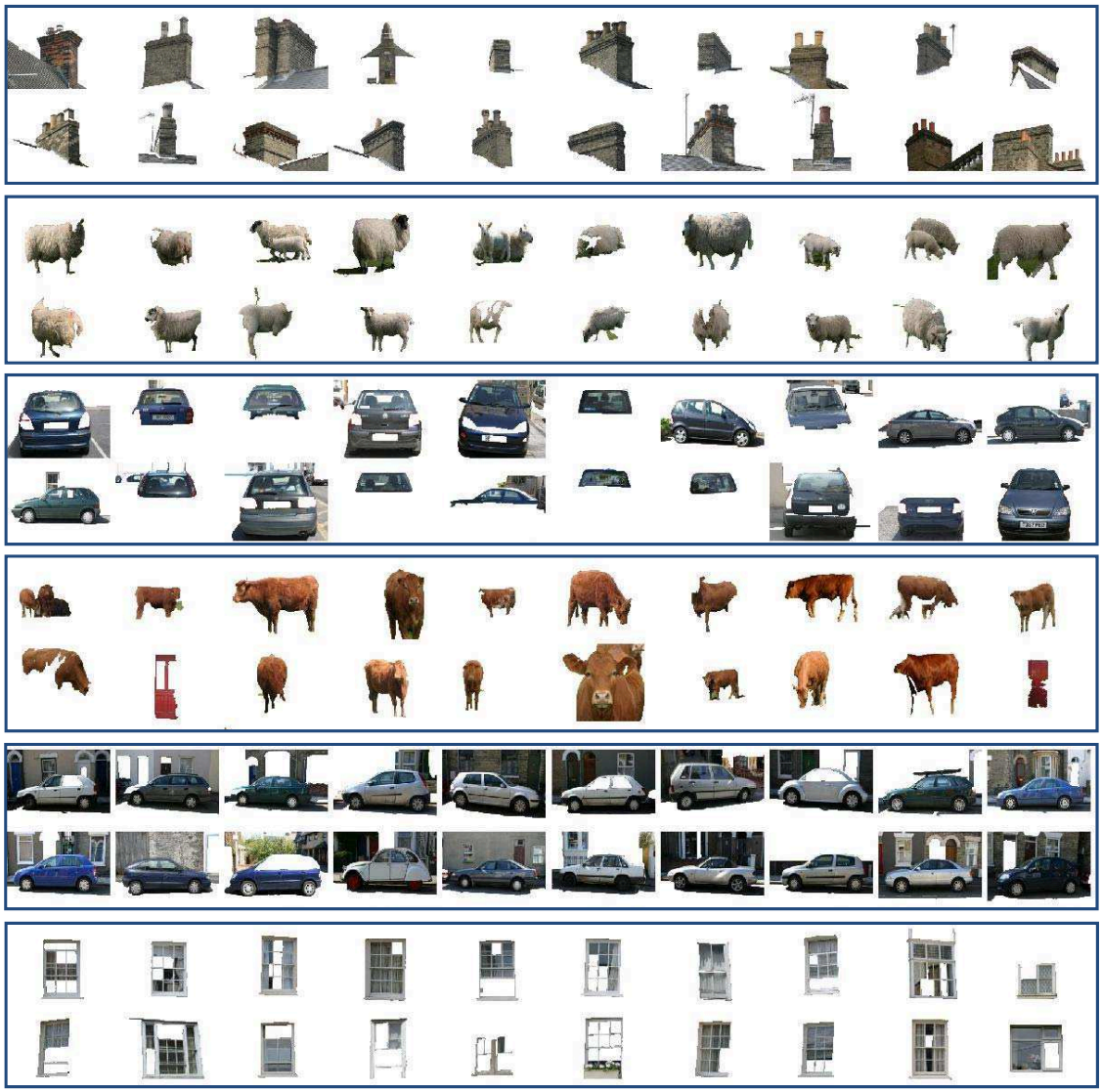


Figure 3.15: Examples of discovered categories for the MSRC-v0 using Oriented WaterShed Transform - Ultrametric Contour Map [6] regions. Note the presence of side and frontal/rear views in the car, cow, and sheep clusters.

3.2 Learning the Easy Things First: Self-Paced Visual Category Discovery

Existing discovery methods, including the approach described in the previous section, treat object category discovery as a one-pass "batch" procedure: the input is a set of unlabeled images and the output is a set of k discovered categories found via clustering. They implicitly assume that all categories are of similar complexity, and that all information relevant to learning is available at once. However, paying equal attention to *all* instances makes the grouping sensitive to outliers, and can skew the resulting models unpredictably. Furthermore, it denies the possibility of exploiting inter-object context cues during discovery; one cannot detect the typical relationships between objects if models for the component objects are themselves not yet formed.

In this section, I propose a *self-paced* approach to context-aware visual discovery.[4] The goal is to focus on the "easier" instances first, and gradually discover new models of increasing complexity. What makes some image regions easier than others? And why should it matter in what order objects are discovered? Intuitively, regions spanning a single object exhibit more regularity in their appearance than those spanning multiple objects or parts thereof, making them more apparent for a clustering algorithm to group. At the same time, regions surrounded by familiar objects have stronger context that can also make a grouping more apparent. For example, if the system discovers models for desks and computer monitors first, it is then better equipped to discover keyboards in their midst. In contrast, if it can currently only recognize kitchen objects, keyboards are less likely to emerge as an apparent cluster.

[4] I published the work described in this section in [91].

In human learning, it is common that easier concepts help shape the understanding of more difficult (but related) concepts. In math, one learns addition before multiplication; in CS, linked lists before binary trees. We aim to capture a similar strategy for visual discovery. However, a critical distinction is that my approach must accumulate its discoveries without any such prescribed curriculum. That is, it must self-select which aspects to discover first.

To implement this idea, I first introduce a measure of easiness that uses two criteria *automatically* estimated from the unlabeled data: (1) the likelihood that the region represents a single object from any generic category—its "objectness"; and (2) the likelihood that its surrounding image regions are instances of familiar categories for which we have trained models (i.e., the familiarity of the surrounding regions)—its "context-awareness".

In Section 3.1, I demonstrated the positive impact of modeling familiar object context for category discovery. Therefore, I initialize the system with models of "stuff" categories (grass, sky, etc.). Then, given an unlabeled image collection, it proceeds to discover "things" (objects) a *single* category at a time, in order of predicted easiness. After each discovery, it updates the set of familiar categories by training a detector for the newly found object class, which allows it to produce a richer context model for each remaining (harder) unfamiliar instance. Similarly, it revises the easiness estimates on all data, and loosen the easiness criterion for the next round of discovery. Thus, in contrast to a one-pass k-way partitioning, my approach gradually accumulates models for larger portions of the data. The process continues until all data is either accounted for, or else fails to meet the least selective easiness criterion. See Figure 3.16.

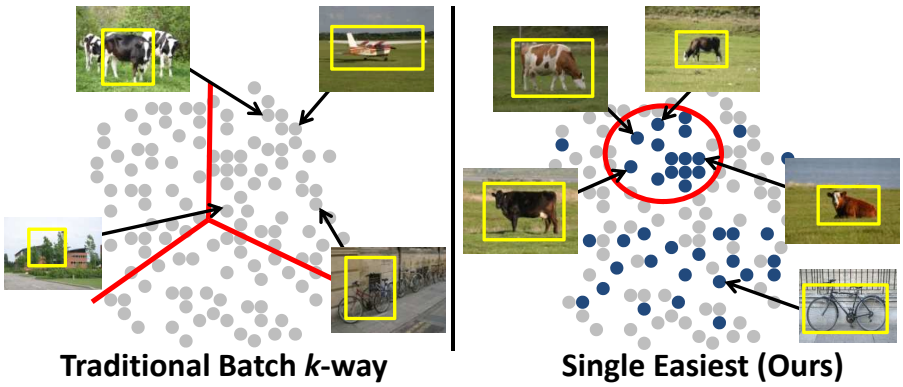


Figure 3.16: In contrast to traditional k-way batch clustering approaches (left), I propose to discover "easier" objects first. At each cycle of discovery, a measure of easiness isolates instances more amenable to grouping (darker dots on right).

The main contribution in this part of my thesis is the idea of visual discovery through a self-paced curriculum. It directly builds on my context-aware approach from Section 3.1, but now the system leverages familiar object context to discover a single category at each iteration. I validate all aspects of my approach on realistic natural images, and show clear advantages for summarization compared to conventional batch clustering and state-of-the-art discovery algorithms. Further, I show that we can train models to predict instances in novel images in an interactive setting where a human annotator names each discovered category. In this way, my approach achieves competitive results to fully supervised baselines at a fraction of the required human labeling cost.

3.2.1 Approach

As above, the goal is to discover visual categories from an unlabeled image collection by grouping image regions with similar appearance and context.[5] Throughout the discovery process, we maintain two disjoint sets of image subwindows: $\mathcal{D}$, the discovered windows that have been assigned to a cluster, and $\mathcal{U}$, the undiscovered windows that remain in the general unlabeled pool. In addition, we maintain an evolving set of familiar categories $\mathcal{C}_t = \{c_1, \ldots, c_{N_t}\}$, where N_t is the category count at iteration t. Initially $\mathcal{D}$ is empty.

My approach iterates over four main steps: (1) identifying the easy instances among the image regions in $\mathcal{U}$; (2) discovering the next prominent group of easy regions; (3) training a model with the discovered category to detect harder instances in the data, moving them to $\mathcal{D}$; and (4) revising the object-level context for all regions in $\mathcal{U}$ according to the most recent discovery. I first explain how we represent a cluster (Section 3.2.1.1), and how we initialize the set of familiar categories (Section 3.2.1.2). I then describe each of the four main steps in turn (Secs. 3.2.1.2 to 3.2.1.5).

3.2.1.1 Exemplar-based Category Models

We use a simple exemplar-based model to represent familiar classes, i.e., those the system has discovered thus far. Each region or window is represented by T types of texture/color descriptors (to be defined in Section 3.2.2). The likelihood of region $r \in \mathcal{U}$ given class $c_j \in \mathcal{C}_t$ is defined by its mean affinity to

[5] We use "region", "subwindow", and "window" interchangeably.

all instances that were grouped together to form class c_j:

$$P(r|c_j) \propto \frac{1}{T}\sum_{m=1}^{T}\frac{1}{|c_j|}\sum_{l\in c_j} K_m(r,l), \tag{3.4}$$

for $j = 1,\ldots,N_t$, where l indexes the exemplars in category j, and each K_m is a χ^2 kernel computed on the m-th feature type. These likelihood values are used below to capture how familiar regions appear to be.

3.2.1.2 Initializing the Pool of Familiar Categories

We initialize the familiar set $\mathcal{C}_0$ with classifiers for "stuff" categories, which are materials with regular fine-scale features, but no specific spatial shape, e.g., *grass, sky, water, road, leaves*. Stuff classes can be classified quite accurately, and are typically widespread in natural scenes. We therefore choose to use them as initial context, and allow the approach to immediately focus on discovering "things"—categories with well-defined shape that often appear amongst the stuff. Thus we populate $\{c_1,\ldots,c_{N_0}\}$ with true instances of the N_0 stuff classes. Given a novel image in the unlabeled collection, we generate its bottom-up segmentation, and can compute each region's likelihoods as defined in Eqn. 3.4.

3.2.1.3 Identifying Easy Objects

Next we proceed to identify the easiest instances among $\mathcal{U}$ according to both low-level image properties and the current familiar classes in $\mathcal{C}_t$. We define an "easiness" function

$$ES(w,\mathcal{C}_t) = Obj(w) + CA(w,\mathcal{C}_t) \tag{3.5}$$

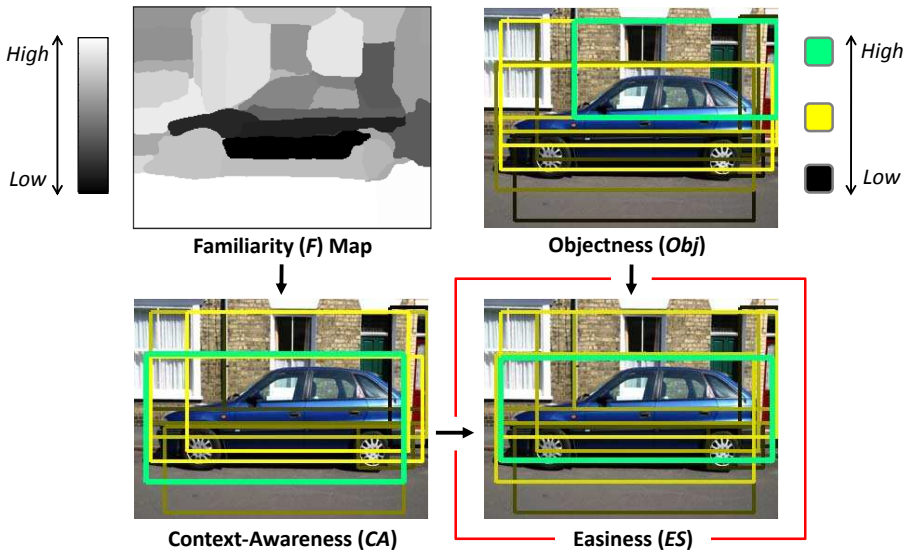


Figure 3.17: Objectness and context-awareness both influence the "Easiness" estimates. Context-awareness favors subwindows surrounded by familiar things, while objectness favors windows surrounding a thing appearing well-separated from background.

that scores a window w based on how likely it is to contain an object ("objectness", Obj) and to what extent it is surrounded by *familiar* objects ("context-awareness", CA).

We compute "objectness" to capture how well an image region appears to contain an object of *any* generic class. Note, this measure does *not* care about class familiarity, it reflects only the generic object-like properties of the window (saliency, apparent separation from background, etc.) We generate candidate regions using the measure developed in [3]. It uses a Bayesian classifier based on multiscale-saliency, color contrast, and superpixel straddling cues to compute the probability that a window contains any object, and is

trained using unrelated image data.[6] For each image, we sample 10,000 windows uniformly across the image at multiple scales, and compute the objectness score $Obj(w)$ for each window. We then sample 50 windows according to the resulting objectness distribution (see Figure 3.17, top right).

We compute "context-awareness" to capture how closely an image window's surrounding regions resemble familiar categories. We first compute the likelihoods defined in Section 3.2.1.1 for each image region; we average the values at any pixels covered by multiple partially overlapping regions. Using those probabilities, we compute a superpixel *familiarity map*, where the familiarity of superpixel s is:

$$F(s, \mathcal{C}_t) = \max_{c_j \in \mathcal{C}_t} P(s|c_j), \tag{3.6}$$

where the max reflects we care only about the degree to which s belongs to *any* familiar category (see Figure 3.17, top left).[7]

Let $s_1(w), \ldots s_R(w)$ denote the R spatially nearest superpixels surrounding window w, in order of proximity. The final context-awareness score is a spatially weighted average of their familiarity scores:

$$CA(w, \mathcal{C}_t) = \sum_{j=1}^{R} \mathrm{w}_j F\big(s_j(w), \mathcal{C}_t\big) \tag{3.7}$$

where $\mathrm{w}_j = R - j + 1$ serves to give regions nearest to the window the most influence. Note that our context-awareness score is similar to our entropy-based measure from Section 3.1.1.1, except now we care specifically about a

[6] We use the authors' code, which was built with INRIA Person, Pascal 06, and Caltech 101 images [3].

[7] The role of the superpixels is simply to summarize measurements coherently within local regions in the image, and ensure we cover regular regions around each window; however, note that the original likelihoods were computed from regions with larger spatial extents as in Section 3.1.1.2.

(a) Easy Instances (b) Hard Instances

Figure 3.18: Randomly selected examples among the easiest and hardest instances chosen by my algorithm. My method is able to bypass the hard or noisy instances, and focus on the easiest ones first. Note that a region with high objectness can yield low easiness if its context is yet unfamiliar (e.g., see red boxes around images in (b)).

region's probability of belonging to any known class rather than its overall uncertainty. Before combining the component $Obj(\cdot)$ and $CA(\cdot)$ terms, we rescale by mapping their distributions to standard Gaussians.

We sort all unclustered instances in decreasing order of easiness (Eqn. 3.5); see Figure 3.18 for examples. Then, we perform discovery on only the easiest instances, as determined by a threshold computed from the data: $\theta_t = 2\sigma - 0.1t$, where σ denotes the standard deviation of all easiness scores in $\mathcal{U}$ and t is the iteration of discovery. Since $ES(\cdot)$ has a standard Gaussian distribution, larger portions of its right tail are considered to be "easy" over the iterations. Due to our choice of the easiness criterion, in practice, our system considers a similar number of total instances to be "easy" at each iteration; this allows our system to produce meaningful candidate clusters that are not too small or large, among which it selects a single prominent category.

3.2.1.4 Single Prominent Category Discovery

Thus far we have a way to model familiar discovered objects and to identify the easiest instances. Now I overview how we represent each easy instance, and then how we extract a single prominent cluster among them.

Representation for each instance Given a candidate easy window $w \in \mathcal{U}$ at iteration t, we form an appearance $A(w)$ and context $G_t(w)$ descriptor. We use standard descriptors for appearance (e.g., pHOG; see Section 3.2.2), and a variant of the 2D object-graph for context. As described in Section 3.1.1.2, the object-graph pools the familiar category likelihoods for the window's spatially nearest superpixels, recording the values according to their relative layout. The resulting descriptor is a series of histograms:

$$G_t(w) = [H_1(w), \ldots, H_R(w)], \tag{3.8}$$

where for $i = 1, \ldots, R$ each component histogram

$$\begin{aligned} H_i(w) \;=\; & [\sum_{j=1}^{i} P(s_{j_a}(w)|\, c_1), \ldots, \sum_{j=1}^{i} P(s_{j_a}(w)|\, c_{N_t}) \\ & \sum_{j=1}^{i} P(s_{j_b}(w)|\, c_1), \ldots, \sum_{j=1}^{i} P(s_{j_b}(w)|\, c_{N_t})]. \end{aligned}$$

accumulates the likelihoods for the N_t familiar classes over the nearest i superpixels, where $s_{j_a}(w)$ denotes the j-th nearest superpixel above the window w, and $s_{j_b}(w)$ denotes the j-th nearest one below it. Nearness is determined based on region centroids. (See Figure 3.19.)

To compute the similarity between two windows w_i and w_j, we use the combined kernel:

$$K(w_i, w_j) = K_{\chi^2}(A(w_i), A(w_j)) + K_{\chi^2}(G_t(w_i), G_t(w_j)), \tag{3.9}$$

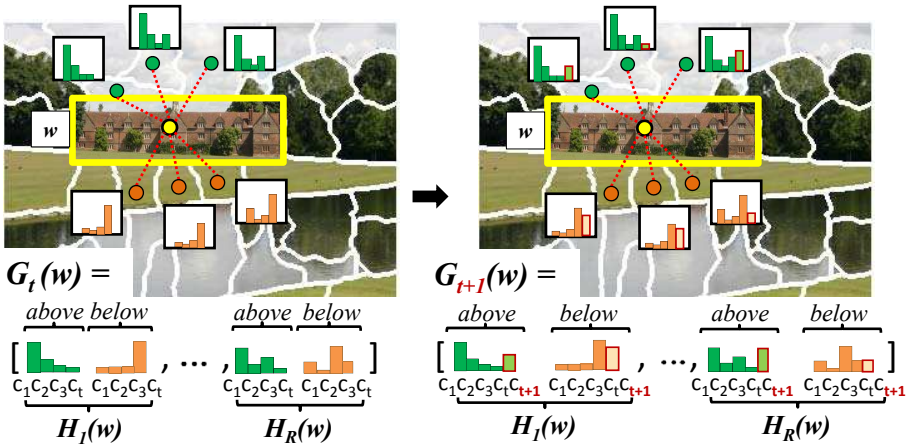


Figure 3.19: **(left)** The 2D object-graph descriptor for window w at iteration t. Each histogram $H_i(w)$ accumulates the likelihoods for the N_t familiar classes $(c_1, \ldots, c_t)$ over the nearest i superpixels, up to $i = R$. **(right)** The descriptor at iteration $t + 1$. Note how it has expanded to reflect the most recent discovered category: c_{t+1}.

where K_{χ^2} denotes a χ^2 kernel. Under this kernel, easy instances with both similar appearance and context are most likely to be grouped together. This is conceptually the same metric described in Section 3.1.1.4 and Equation 3.3, except now we represent objects with windows rather than regions.

Prominent category discovery Given the current easy windows and the combined kernel, at each iteration we want to expand the pool of discovered categories with a single prominent cluster. Recall, the easiest instances already serve to focus the algorithm on those regions with consistent representations. In particular, our context-awareness criterion is directly linked to the data representation during clustering: the easiest instances are surrounded by familiar regions with relatively high likelihoods (see Eqns. 3.6 and 3.7), which makes comparisons between their object-graphs meaningful. Thus, by seeking a single new cluster, we can conservatively identify the most obvious new group;

further, we can incrementally refine the context model most quickly for future discoveries.

To discover the most prominent category, we first partition the data into candidate groups, and then refine the most distinctive one. Specifically, we perform complete-link agglomerative clustering over the easy instances using the kernel in Eqn. 3.9, which offers robustness to outliers (i.e., windows that are poorly localized or contain rare objects) and allows us to target a cluster size rather than a cluster number. We stop merging when the distance between the most similar (yet-unclustered) instances becomes too large—specifically, greater than one standard deviation beyond the mean distance between all instances—and automatically select the tightest cluster with the highest silhouette coefficient [142] among the candidate groups. We then refine the selected instances with Single-Cluster Spectral Graph Partitioning (SCSGP) [112, 114], which maximizes the average consensus. This step reduces possible outliers in the discovered group from agglomerative clustering.

We found this procedure to perform much better in practice than simply directly applying a "single-cluster" algorithm (e.g., Min Cut or SCSGP alone). This is likely due to the latter's sensitivity to a small number of outlying points, and the presence of overlapping clusters.

3.2.1.5 Discovered Category Knowledge Expansion

Each newfound discovery—a single prominent cluster identified among the easiest instances—serves to benefit later discoveries; this is a key property of my self-paced curriculum learning approach. In particular, it helps at both the *intra-category* and *inter-category* levels, as I explain next.

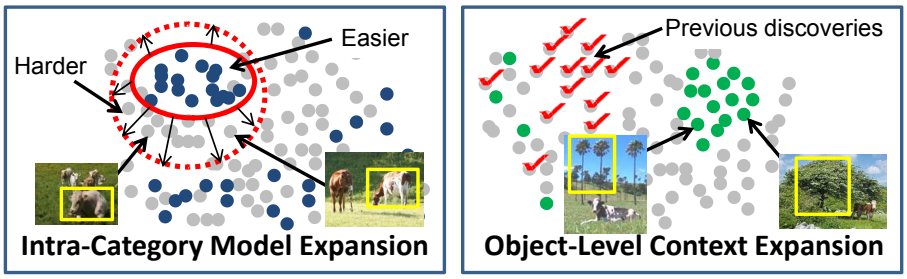


Figure 3.20: Discovered category knowledge expansion. A model built from the discovery allows harder related instances to be recovered (left). The new familiar objects serve as richer context for the next round of discovery (right).

Intra-category model expansion First, the initially discovered easier instances yield a model that can be used to detect the harder instances, which would not have clustered well due to their appearance or different context. We use instances in the newly discovered category to train a one-class SVM based on their appearance representation (no context). Then, we apply the classifier to all remaining windows in $\mathcal{U}$, merge the positively classified instances with the discovered category, and move them to $\mathcal{D}$.

While object-level context helps the discovery algorithm group the *easier* instances, we intentionally exclude context from the classifier's feature space for this stage. The goal is to be more inclusive and identify the harder instances of the class. For example, we might first discover cows in grass as the easy case, and then use the corresponding cow model to find other more challenging instances of cows that are partially occluded or surrounded by other animals (see Figure 3.20, left; darker dots denote easier instances).

Object-level context expansion Second, the expansion of the context model based on the discovered categories can help to discover certain harder ones. With each discovery, $\mathcal{C}_t$ expands. Thus, for every window remaining in

$\mathcal{U}$, we revise its object-graph $G_t(\cdot)$ to form $G_{t+1}(\cdot)$, augmenting it with class affinities for the discovered category, per spatial component (see Figure 3.19). This enriches the object-level context, altering both the feature space and the easiness scores. In effect, while we have weaker context models when detecting the easiest objects, we have richer context models when considering harder instances at later iterations. For example, having detected the "stuff" regions (grass, roads, sky), the system may discover cows in the simple meadow scenes, and then exploit its expanded context to later discover diverse-looking trees that appear in the context of both grass and cows (see Figure 3.20, right).

I validate the impact of both the intra-category model expansion and object-level context expansion on category discovery in Section 3.2.2, Figures 3.21 and 3.24, respectively.

3.2.1.6 Iterative Discovery Loop

Finally, having augmented $\mathcal{C}_t$ with the newly discovered category, we proceed to discover the next easiest category. Note that the easiness scores evolve at each iteration of the discovery loop as more objects become familiar. Further, the annealing of the threshold defined in Section 3.2.1.2 essentially loosens the "easiest" criterion over time, allowing the algorithm to discover harder categories in later iterations, when context models are potentially richer. As the method iterates, it accounts for more instances.

We iterate the process until the remaining instances in $\mathcal{U}$ are too hard: this makes the system robust to noisy and rare instances that do not belong to any cluster. Algorithm 2 summarizes the steps of my algorithm.

Input: Unlabeled images; stuff models $c_1, \ldots, c_{N_0}$.
Initialize $\mathcal{U}$ with all regions from unlabeled inputs; $\mathcal{D} = \emptyset$;
$\mathcal{C}_0 = \{c_1, \ldots, c_{N_0}\}$; $t \leftarrow 1$.
while *Easy instances remain in* $\mathcal{U}$*:* **do**
1. Identify easy instances $ES(w, \mathcal{C}_t) > \theta_t$ in $\mathcal{U}$. (Section 3.2.1.3)
2. Discover single prominent category among them. (Section 3.2.1.4)
3. Detect harder intra-class instances with one-class classifier; move instances to $\mathcal{D}$, add new category to $\mathcal{C}_t$. (Section 3.2.1.5)
4. Expand object-graph descriptor for each instance in $\mathcal{U}$. (Section 3.2.1.5)
5. Revise familiarity map; recompute easiness. (Section 3.2.1.3)
6. Loosen easiness criterion; $\theta_t = 2\sigma - 0.1t$. (Section 3.2.1.6)
$t \leftarrow t + 1$
end
Output: Set of t discovered categories in $\mathcal{D}$.

Algorithm 2: Algorithm recap

3.2.2 Results

My experiments quantify the proposed method's clustering and segmentation accuracy using standard metrics from previous work [72, 86, 89, 127], and I additionally demonstrate classification performance on novel images using models learned with the discovered categories.

Baselines I compare my approach to several baselines: (1) a side-by-side implementation of batch clustering, (2) a baseline that focuses on the hardest instances first (those with lowest easiness) but otherwise follows our pipeline, (3) my batch discovery method from Section 3.1, and 4) an existing state-of-the-art discovery method [127].

Dataset We use the MSRC-v0 dataset, which consists of 3,457 natural scenes with 21 object classes, and was studied in the previous section (see Figure 3.7). The wide variety of categories allows us to properly evaluate the impact of both

easiness selection and context refinement. We learn stuff classes on 40% of the data, and run discovery on the other 60%.[8] With 50 sampled windows per image, this makes 60,000 instances in the unlabeled pool.

Implementation details We use [6] to obtain candidate stuff regions. We combine texture, color, and shape features to form $A(w)$ for window w. To describe texture, we compute SIFT bag-of-words histograms for the regions and Spatial Pyramid histograms for the windows; we densely sample 8-pixel wide SIFT patches at every pixel. To describe color, we use Lab color space histograms, with 23 bins per channel. To describe shape, we compute pHOG descriptors with 3 levels and 8 bins. For the object-graphs, we generate an over-segmentation with roughly 50 superpixels per image, and fix $R = 20$, following [89]. We normalize all histograms to sum to 1. We set $\nu = 0.1$ for the one-class SVM.

Evaluation metrics To quantify discovery accuracy, we again use *purity* [142], which is the percentage of correctly labeled instances, where all instances in a cluster are assigned to its majority class's true label. To score a window, we take its true label to be that to which the majority of its pixels belong. To quantify the segmentation accuracy of a window w, we use the pixel-level *overlap score*, $OS = \frac{|GT \cap w|}{|GT \cup w|}$, where GT is the ground-truth object segmentation, i.e., the tightest bounding box covering the full object region associated with w's majority pixel label.

[8] In all experiments we treat "stuff" classes as initial context, as explained in Section 3.2.1.2. While in principle one could also use our framework with "things" as initial known classes, the implementation is not straightforward with the cues we chose (regions for stuff, windows for things). See Section 3.1.2.1 for results analyzing the impact of which classes serve as initial context for discovery.

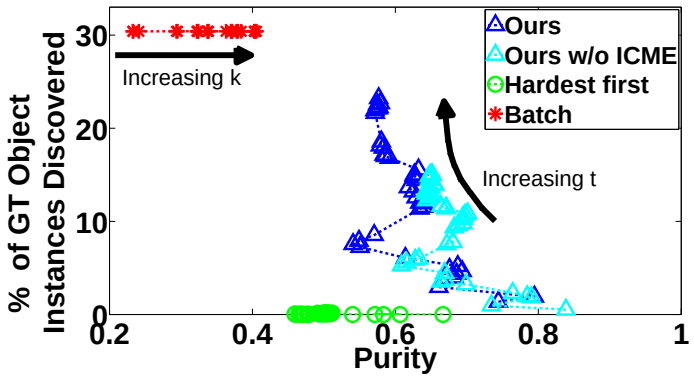


Figure 3.21: Discovery accuracy as a function of the percentage of unique object instances discovered. Our approach produces significantly more accurate clusters than either baseline, while selectively ignoring instances that cannot be grouped well.

3.2.2.1 Object Discovery Accuracy

I first analyze the quality of our discovered clusters, compared to both the batch and "hardest first" baselines. All methods use the same features and agglomerative clustering algorithm. The batch baseline is meant to show the limitations of existing methods, all of which determine k models in one pass over all the data. To ensure the batch baseline is competitive, we give it the non-overlapping windows with the highest objectness score per image as input.

Figure 3.21 shows the results. We plot purity as a function of the *percentage of ground-truth object instances discovered* in order to analyze the quality of the discovered groups *and* quantify the recall rate for the true objects found. We count true objects as windows with at least 50% overlap with ground truth; if multiple windows overlap a ground-truth object, we score only one of them. Each point shows the result for a given number of clusters,

for $k = t = [1, 40]$. At each iteration, our method finds about 5-15% of the instances to be "easy".

My approach provides significantly more accurate discoveries than either baseline. Note that purity increases with k for the batch method, since the k-way clusters computed over all windows get smaller, which by definition generates higher purity. In contrast, my method accounts for *more* windows as t increases, and purity gradually declines as the easiness criterion is relaxed. This difference highlights the core concept behind my approach: rather than force k splits, it steadily and selectively increases its pool of discovered objects. It purposely does not integrate all possible object instances (ignoring harder or poorly grouped ones), and yields accuracy more than twice as good as the batch approach. (In Figure 3.26, I show the impact that this has on generalization performance.) For reference, the upper bound on instances we could discover is 53%, which is the portion of true objects present in the initial 50 windows per image. Most of the missed objects (for any method) are small object parts, e.g., windows or doors on cars, or objects that are not well-represented with windows, e.g., walls that are labeled as "building" in the ground truth. Sampling more windows would likely increase recall of the objects, but could also decrease purity rates due to more noisy regions.

Our substantial improvement over the "hardest-first" baseline validates our claim that considering the easiest instances per iteration leads to more accurate models. It also indicates that the easiest instances are indeed those that best capture true object regions. Note that while the hardest-first baseline technically has higher purity than batch, it discovers almost no objects—most windows it chooses to group overlap multiple objects or object parts. This result shows the importance of evaluating the quality of the segmentations

(i.e., image windows) in addition to cluster quality. It also reveals the inherent difficulty in evaluating discovery methods, which I discuss in more detail in Section 3.4 and Chapter 6.

Finally, the plot also reveals the impact of our intra-category model expansion. By using models discovered on easier examples to classify harder instances of the same object, we successfully discover a larger percentage of the instances in the data, with only a slight reduction in purity. (Compare "Ours" to "Ours w/o ICME" in Figure 3.21.)

Figure 3.22 shows representative example discoveries, sorted by iteration. I display the top 10 regions for each category, as determined by their silhouette scores. Note that the easiest categories (trees and bicycles) have high objectness and context-awareness scores, as well as strong texture, color, and context consistency, causing them to be discovered early on. The harder chimney and sheep objects are not discovered until later. There are some failure cases as well (see $t = 3, 8$), such as re-discovering a familiar category (trees) or merging different categories due to similar appearance (cars and windows).

3.2.2.2 Object Segmentation Accuracy

Since the images contain multiple objects, my algorithm must properly segment each object in order to obtain clusters that agree with semantic categories. Thus, I next compare the overlap accuracy for the object instances we discover in 40 categories to (1) the initial 50 windows sampled per image according to their objectness scores, and (2) 50 *randomly* sampled windows per image.

Figure 3.23 shows the results. The windows sampled according to objectness are already significantly better than the random baseline, showing the

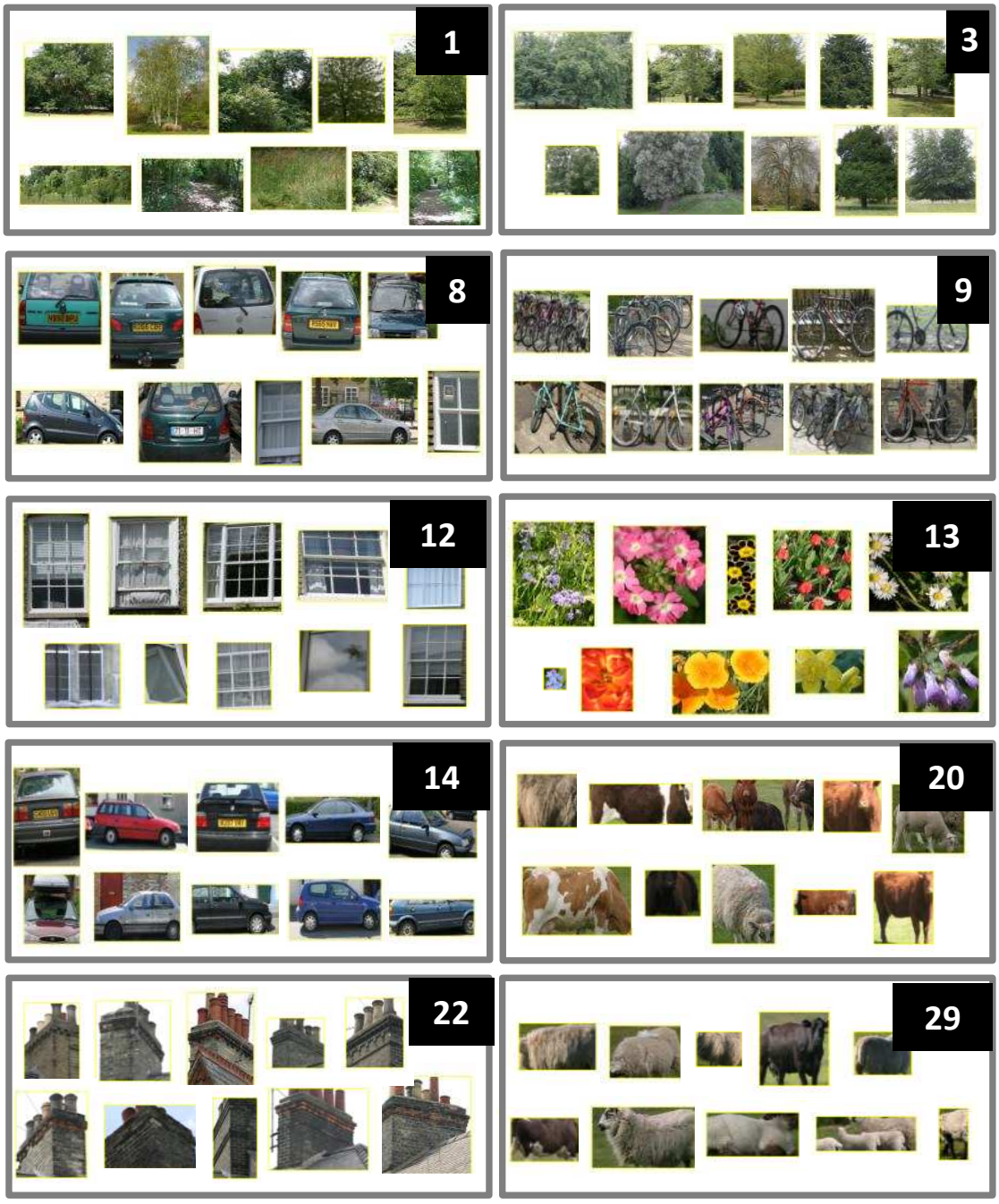


Figure 3.22: Examples of discovered categories; numbers indicate the iteration when that discovery was made. See text for details.

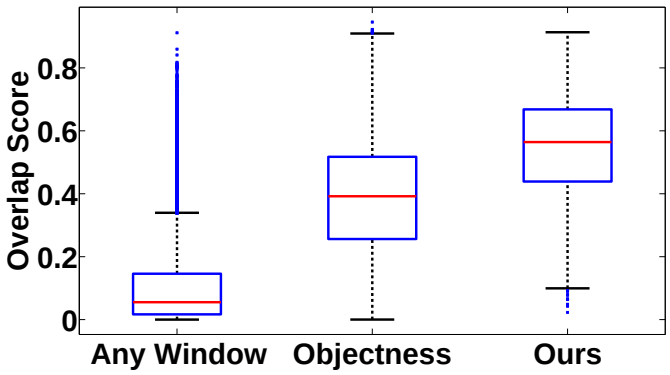


Figure 3.23: Object segmentation accuracy for random image windows (left), windows sampled by objectness alone (center), and those discovered by our approach (right). Higher values are better.

contribution we get from the method of [3]. However, my method produces even stronger segmentations, showing the impact of the proposed context-awareness and easiness scoring.

3.2.2.3 Impact of Expanding Models of Object Context

Next I evaluate the impact of object-level context expansion. To isolate this aspect, I compare against a baseline that follows the same pipeline as my method, but uses familiar models for only the initial stuff categories; it does not update its context model after each discovery.

Figure 3.24 shows the results, in terms of purity as a function of the number of discovered categories. As expected, the cluster quality is similar in the first few iterations, but then quickly degrades for the baseline. The first few discoveries consist of easy categories with familiar "stuff" surrounding them, and so the baseline performs similarly to our method. However, without any updates to the context model, it cannot accurately group the harder instances

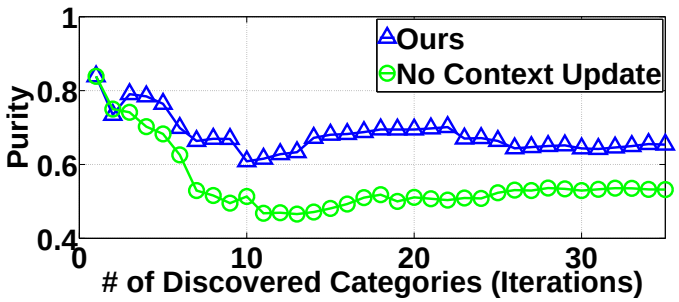


Figure 3.24: Impact of expanding the object-level context.

(e.g., cars, buildings). In contrast, by revising the object-level context with new discoveries, we obtain better results.

3.2.2.4 Comparison to State-of-the-Art

I next compare against two batch discovery algorithms: my context-aware discovery method from Section 3.1 and the state-of-the-art Latent Dirichlet Allocation topic model method of Russell et al. [127]. These are the most relevant methods, since both perform discovery on images with multiple objects (other techniques generally assume a single object per image). We run all methods on the same MSRC data, and use publicly available source code, which includes feature extraction. To quantify how well each method summarizes the same data, we use the F-measure: $\frac{2 \cdot P \cdot R}{P+R}$, where P denotes precision and R denotes recall.[9] Since we do not know the optimal k value for any method, we generate results for a range of values and show the distribution (we consider $k = [10, 40]$, since the data contains 21 total objects). Figure 3.25

[9]We evaluate recall with respect to each method's output discoveries, since the target categories are slightly different. Our object-graph and self-paced discovery methods attempt to discover only the "things", while the topic model method attempts to discover all categories.

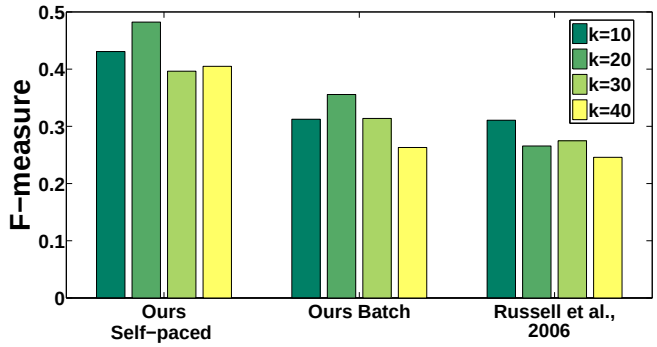


Figure 3.25: Comparison to state-of-the-art discovery methods. Our self-paced approach summarizes the data more accurately than both our batch approach (defined in Section 3.1) and a batch appearance-based clustering baseline [127].

shows that our method produces the most reliable summary of the unlabeled image data.

3.2.2.5 Predicting Instances in Novel Images

Finally, I test whether the discovered categories generalize to novel images outside of the discovery pool. The goal is to test how well the system can reduce human effort in preparing data for supervised classifier construction. The discovery system presents its clusters to a human annotator for labels, then uses that newly labeled data to train models for the named object categories. Given a novel image region, it predicts the object label.

We train one-vs-one SVM classifiers (with $C = 1$) for all discovered categories using the appearance kernels. To simulate obtaining labels from a human annotator, we label all instances in a cluster according to the ground-truth majority instance. In addition to the baselines from above, we compare to two "upper bounds" in which the ground truth labels on all instances are used to train a nearest-neighbor (NN) and SVM classifier. We test on the 40%

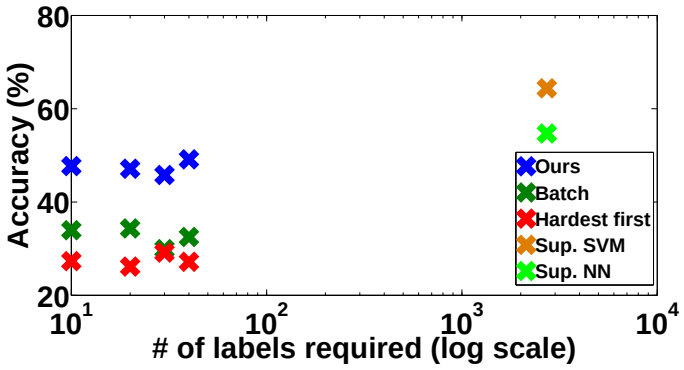


Figure 3.26: Classification results on novel images, where discovered categories are interactively labeled. Our method outperforms the other discovery baselines. The single-instance baseline produces the best result, but at the expense of requiring 2721 labels (one for each window); ours only requires k (one for each discovered category).

split that trained the stuff models (which is fine, since the test set used here consists only of objects), totaling 2,836 test windows from 16 object categories.

Figure 3.26 shows the results, for a range of iterations. Alongside test accuracy, we show the number of manually-provided labels required by each method. As expected, the fully supervised methods provide the highest accuracy, yet at the cost of significant human effort (one label per training window). On the other hand, my method requires a small fraction of the labels (one per discovered category), yet still achieves accuracy fairly competitive with the supervised methods, and substantially better than either the batch or hardest-first baselines.

This result suggests a very practical application for discovery, since it shows that we can greatly reduce human annotation costs and still obtain reliable category models. Building on this, in the next section, we will see how to save annotation costs for tagging faces in consumer photo collections.

3.3 Face Discovery with Social Context

The previous two sections showed how to discover novel objects in natural image collections using object-level context from familiar object categories. In this section, I adapt the proposed algorithm specifically to discover novel *faces* in untagged photo collections by leveraging the social context of co-occurring people.[10] The goal is to perform unsupervised clustering on faces detected in the images, in order to come up with a batch of photos likely of the same individual, so that the user can efficiently tag or prune them with minimal effort. In contrast to previous face clustering algorithms (e.g., [13, 136, 144]), my approach allows us to expand the representation of the detected faces to include not just their appearance, but also their *social context*. Specifically, this lets us use cues from co-occurring people in the same image in order to produce more reliable groups.

Why do co-occurrence cues help? New (yet unlearned) faces in a collection appear with some strong social context, as users' photos tend to dwell within different cliques of people: families, friends, co-workers, etc. This means the context of "familiar people" can both help disambiguate people with similar appearance, and help the system realize that instances of faces in different poses or expression are actually of the same person (see Figure 3.27).

I design a novel social context descriptor to capture the predictions of previously trained face models, and show that this "face-level" cue is more reliable than simply using the appearance of nearby faces as context. A system using the proposed approach frees the user from manually identifying each new face. Instead, it discovers novel recurring faces—and, critically, discovers them

[10] I published the work described in this section in [90].

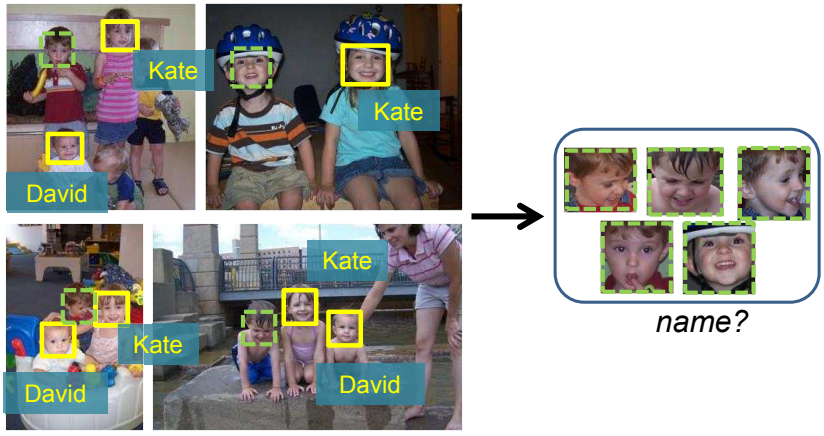


Figure 3.27: Main idea of my approach to unsupervised face discovery in personal photo collections. For any unfamiliar face not recognized by the system (in dotted green), we use co-occurrence cues from familiar faces nearby (in solid yellow) to produce more reliable groups. In this example, an appearance-based grouping method that clusters the unfamiliar faces would likely fail to recognize the many instances of the boy, given their variability. In contrast, by also representing the *social context* of people appearing near each unfamiliar face, my approach computes more reliable clusters. Having discovered a novel face, the system would present the images to a user for name-tagging.

more accurately by modeling the social context surrounding them. It can then present its discoveries (a cluster of photos) to the user, and he/she can confirm with tags (or reject). While related context cues have been explored to a limited extent for traditional supervised learning pipelines [47, 140, 157, 173], I am the first to consider unsupervised face discovery using social context. I demonstrate my approach for mining novel faces on a dataset drawn from multiple domains and two large personal photo collections that exhibit natural social context.

3.3.1 Approach

The goal is to discover novel faces from untagged image collections by exploiting the social nature of consumer photographs. In particular, we aim to use the co-occurrence information from *familiar* people to better discover faces of new people.

We follow a similar pipeline to Section 3.1 for object category discovery in natural scenes, but adapt it specifically for the face discovery setting for consumer photo collections, and show that it captures a very relevant form of social context that allows better unsupervised clustering in this domain. Given the central importance of face tagging for everyday consumer photo applications, this setting is particularly interesting to consider.

Given a pool of unlabeled photos, we first detect any faces in each image. We then identify novel faces that do not resemble any person for which we have trained models (Section 3.3.1.2). After isolating the unfamiliar faces, we form new people "categories" by grouping faces that have similar appearance *and* similar social networks (Section 3.3.1.3).

See Figure 3.28 for an overview of my system. In the following, I describe the main steps.

3.3.1.1 Learning Models for Tagged Faces

For each face region r found with a face detector, we extract texture features to serve as the appearance descriptor $A(r)$. We use pyramid of HOG (pHOG) [15] or Local Ternary Patterns (LTP) [143]. We train SVM classifiers for N initial people, $\{c_1, \ldots, c_N\}$, for whom we have tagged face images. These classifiers will allow us to identify the instances of each initial familiar person

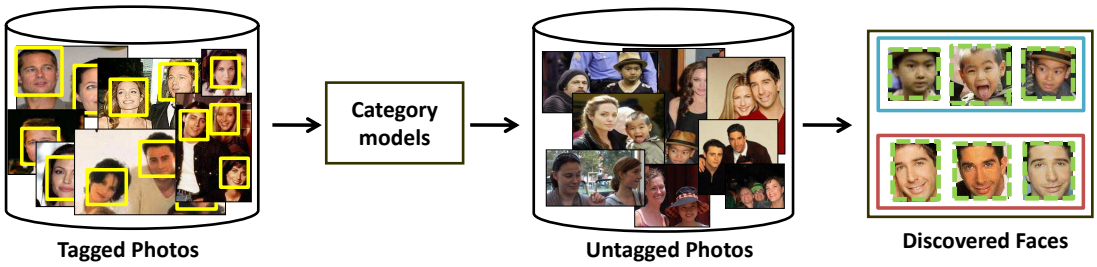


Figure 3.28: System overview. Given a photo collection with tagged faces, we train models for each person. Given a novel set of face images (that do not have any tags), we detect instances of familiar people in each image, and use their context to discover novel faces.

in novel images. We will use those predictions to describe the social context for each *unfamiliar* face, as we describe in more detail in Section 3.3.1.3.

3.3.1.2 Identifying Unfamiliar Faces

For any unlabeled photo, we would like to detect the people in it, and determine whether any of them resembles a *familiar person*. Doing so will allow us to isolate the unknown faces, and to build social context descriptors that portray the co-occurring familiar people.

For all unlabeled images, we run a face detector [155] to extract candidate faces. To compute the known/unknown decision for a face region r, we apply the N trained classifiers from Section 3.3.1.1 to the face to obtain its class membership posteriors $P(c_i|r)$, for $i = 1, \ldots, N$, where c_i denotes the i-th person class. Faces that resemble a known person c_i will produce a high value for $P(c_i|r)$, and low values for $P(c_j|r)$, $\forall j \neq i$. Faces that do not resemble any familiar person will have more evenly distributed posteriors.

Thus, to distinguish which faces should be considered to be unknown, we compute the entropy: $E(r) = -\sum_{i=1}^{N} P(c_i|r) \log P(c_i|r)$. Then, similar to

our strategy above, faces with low entropy values will likely belong to familiar people, while those with high values will likely be unfamiliar. We select a cutoff threshold t equal to one-quarter of the maximum possible entropy value, and treat faces with values above it as unknown. Our intentionally selective criterion allows us to compute accurate estimates on familiar people, and at the same time include as many unfamiliar faces as possible. We validate the impact of our conservative known/unknown decisions on discovery in Section 3.3.2.

3.3.1.3 Social Context Descriptors

For each unfamiliar face, we want to build a description that reflects that person's co-occurring familiar people, at least among those that we can already identify. Having such a description allows us to group faces that look similar (i.e., have similar appearance) and often appear among the same familiar people (i.e., have similar social context).

Suppose an image has T total faces: $r_1, \ldots, r_T$. We define the social context descriptor $S(r)$ as an N-dimensional vector that captures the distribution of familiar people that appear in the same image:

$$S(r) = \left[\sum_{j=1}^{T} P(c_1|r_j), \ldots, \sum_{j=1}^{T} P(c_N|r_j) \right]. \tag{3.10}$$

If our class predictions were perfect, with posteriors equal to 1 or 0, this descriptor would be an indicator vector telling which other people appear in the image. When surrounding faces do belong to previously learned people, we will get a "peakier" vector with reliable context cues, whereas when they do not appear to be a previously learned person the classifier outputs will simply summarize the surrounding appearance.

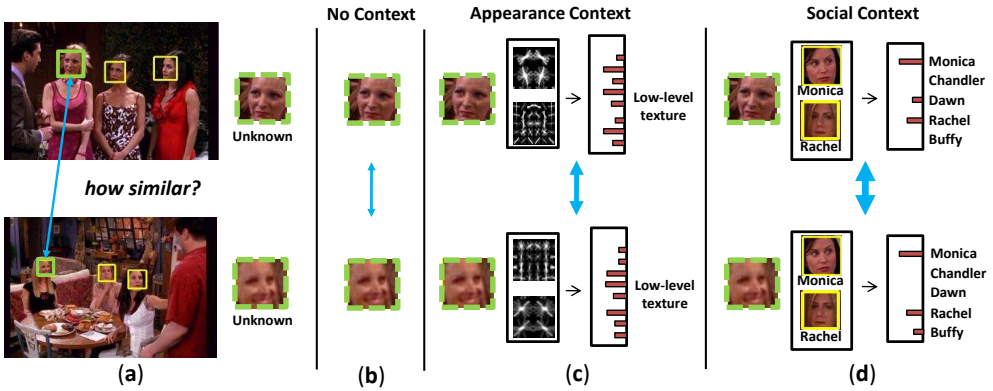


Figure 3.29: An example illustrating the impact of social context for discovery. The blue double-headed arrows indicate strength in affinity between the unknown regions. (a) Two images, where the unfamiliar faces are outlined in green. (b) Appearance information alone can be insufficient to deal with large pose or expression variations. (c) Modeling the context surrounding the face of interest can provide more reliable similarity estimates, but a context descriptor using raw *appearance* is limiting since it can only describe nearby faces with texture or color. (d) By modeling the *social* context using learned models of familiar people, we can obtain accurate matches between faces belonging to the same person.

Note that unlike the object-graph descriptor from Section 3.1 that considers the spatial layout of the objects, we do not encode the spatial relationships between people. This is because we do not expect high regularity in how certain individuals arrange themselves (though this can be useful for broader traits like gender and age [49, 157]).

Alternatively, one can imagine forming a context description using the raw appearance of co-occurring faces—for example, by recording the pHOG or LTP descriptors of the other faces detected in the image. However, context in the form of low-level appearance information may be insufficient to provide reliable grouping cues, since the appearance variabilities of the same person (due

to pose, expression changes, etc.) would not be accurately modeled (see Figure 3.29). By modeling social context using learned models of familiar people, we obtain more descriptive and compact representations. In Section 3.3.2, we directly evaluate the impact that the social context descriptor has on discovery over a baseline that uses low-level appearance features as context.

3.3.1.4 Discovering New Faces

Finally, we cluster all faces that were deemed to be unknown. We consider two clustering algorithms: (1) spectral clustering [111], and (2) complete-link agglomerative clustering. Spectral clustering provides flexibility in the choice of the affinity measure and is able to detect clusters of irregular shape. However, it requires the number of clusters as input, which is not always available for the discovery scenario. Agglomerative clustering offers more flexibility in this regard, since the size rather than the number of clusters can be targeted. Each clustering method takes as input a matrix of the pairwise affinities between all current unknown faces.

We want the discovered groups to be influenced both by the appearance of the face regions themselves, as well as their surrounding context. Therefore, given two face regions r_m and r_n, we evaluate a kernel function K that combines their appearance similarity and context similarity:

$$K(r_m, r_n) = \alpha \cdot K_{\chi^2}\left(S(r_m), S(r_n)\right) + (1-\alpha) \cdot K_{\chi^2}\left(A(r_m), A(r_n)\right), \quad (3.11)$$

where α weights the contribution of social context versus appearance (recall $A(r)$ is a pHOG or LTP descriptor). Each K_{χ^2} is a χ^2 kernel function for histogram inputs x and y:

$$K_{\chi^2}(x, y) = \exp\left(-\frac{1}{2\Omega}\left(\sum_j \frac{(x_j - y_j)^2}{x_j + y_j}\right)\right), \quad (3.12)$$

where j indexes the histogram bins, and Ω is a data-dependent scaling factor, which we set as the average χ^2 distance between all face regions. This is conceptually the same kernel described in Section 3.1.1.4 and Equation 3.3, except we replace the object-graph with our social context descriptor to encode context.

By considering both the appearance of the faces as well as their social context, we expect to be able to discover faces with occlusion (i.e., due to sunglasses or a hat) or large pose variations. For example, if the system knows what Monica and Chandler look like, it gets richer context descriptors to discover their pal Rachel, even in difficult cases such as when she is wearing sunglasses. Analyzing the facial appearance alone could have been inadequate to group the different instances of Rachel with and without sunglasses.

3.3.2 Results

In this section, I evaluate my method's face discovery performance.

Baselines I compare my method to two baselines: (1) a **no-context** baseline that simply clusters the face regions' appearance descriptors, and (2) an **appearance-context** discovery method that uses the appearance of surrounding faces as context (rather than the predicted categories). The second baseline substitutes the summed appearance descriptors of co-occurring faces for $S(r)$. These are important baselines to show that we would not be as well off simply looking at a model of appearance using image features, and to show the impact of social context analysis versus a low-level appearance context description for discovery.

Dataset We validate on three datasets. The first dataset (**Mixture**) is a compilation from three sources: The Gallagher Collection Person Dataset [48], an episode of *Buffy the Vampire Slayer* [36], and an episode of *Friends*. We chose these three since they contain natural cliques of people (family members, characters that appear in scenes together). There are a total of 12,542 images, 8,452 detected faces, and 23 unique people.

The second and third datasets are from [157], which are collected from real family photo albums from two different people. The second dataset (**Wang1**) has 1,125 images, 2,769 faces, and 47 people; the third dataset (**Wang2**) has 1,117 images, 3,282 faces, and 152 people. These datasets contain images encompassing real social relationships and thus are perfect testbeds for evaluating our method.[11] See Figure 3.30 for image examples.

We partition each dataset into two random subsets. The first is used to train N classifiers for the initial "knowns". These faces represent the set of people for which the system already has some tagged examples. On the second subset, we perform discovery using the N categories as context to obtain our set of discovered categories. To demonstrate that our method's improvements are robust with respect to N and which categories are chosen to be known, we test on four splits of the Mixture collection: two splits have 8 unknown people (489 and 540 face instances, respectively), the other two have 15 (1138 and 1044 face instances, respectively), all selected randomly. For the Wang1 and Wang2, we select as known the top 25% of the most frequently appearing people; the datasets have 16 and 104 unknown people (143 and 373 face instances), respectively. This reflects that the owner of the collection and

[11] While the data from [157] is relevant to our task, their supervised labeling application is distinct from ours and so not relevant for comparison.

Figure 3.30: Examples of photos from the Mixture and Wang 1 datasets.

his/her closest family members and friends would likely be labeled prior to those who appear less frequently.

Implementation details We use OpenCV for the face detector [155] and work only with true-positive detections. For the Mixture dataset, we use pHOG with two pyramid levels and eight bins to describe face appearance, and spectral clustering [111] to group the faces. For the Wang1 and Wang2 datasets, we use the Local Ternary Patterns (LTP) descriptor, which is a histogram of local intensity differences surrounding each pixel that encodes texture, and use publicly available code by the authors [143] and default parameters. We use agglomerative clustering for grouping. We worked with the pHOG descriptor in early experiments but later substituted it with the LTP descriptor due to it being more suitable for describing face patches.

To compute class probabilities, we use one-vs-one SVM classifiers, and obtain posteriors using pairwise coupling [165]. We normalize the context descriptors to sum to 1. We set α to 0.5 for the Mixture dataset and 0.2 for Wang1 and Wang2 datasets. Due to the larger number of people and their varying frequencies in the Wang datasets, increasing the weight on appearance produces better clusters. In general, α could be determined interactively by observing qualitative examples of the clusters. Training the known classifiers, building the context descriptors, computing kernels, and clustering the unknowns takes 1-5 minutes with a Matlab implementation.

Evaluation metrics We use the *F-measure* to quantify discovery accuracy. The F-measure reflects the coherency (precision P) of the clusters, while taking into account the recall R of the same-category instances: $F = \frac{2 \cdot P \cdot R}{P+R}$. We set the number of clusters to discover to be equal to the number of true un-

	# Unknowns	Ours	No-Context	App-Context
Mixture*	15	**0.30**	0.26	0.28
Wang1	16	**0.25**	0.20	0.21
Wang2	104	**0.24**	0.23	0.21

(a) Accuracy of discovery per dataset

	# Unknowns	Ours	No-Context	App-Context
split1	8	**0.34** (0.00)	0.24 (0.01)	0.26 (0.01)
split2	8	**0.32** (0.01)	0.23 (0.01)	0.29 (0.01)
split3	15	**0.30** (0.01)	0.26 (0.03)	0.28 (0.01)
split4	15	**0.33** (0.01)	0.28 (0.01)	0.30 (0.01)

(b) Impact of who is known ("splits")

Figure 3.31: Face discovery on the three datasets (a) and the different splits of the Mixture dataset (b) as judged by the F-measure. We compare our approach (Ours) with an appearance-context baseline (App-Context), and a baseline clustering only with the region descriptors (No-Context). Numbers in parentheses show range over 10 runs. Higher values are better. Our method outperforms both baselines in all cases, showing the impact of modeling the co-occurrence information of surrounding familiar people for discovery. **We take split3 to represent Mixture in (a), since it roughly corresponds to 25% of the people being known, parallel to the other datasets.*

familiar faces in the image collection, to meaningfully evaluate our method's discovery performance. To evaluate auto-tagging accuracy on novel images, we use standard multi-class recognition accuracy.

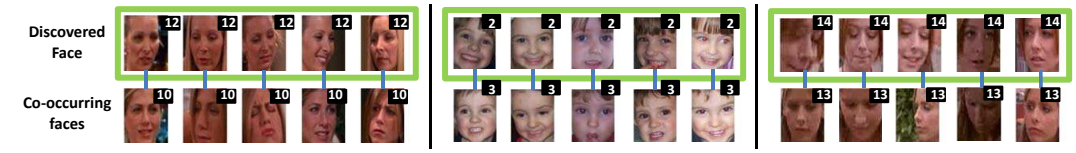
3.3.2.1 Face Discovery Accuracy

Figure 3.31 shows discovery results. My method significantly outperforms the baselines on all datasets, validating my claim that social context leads to better face discovery. In most cases, the appearance-context outperforms the no-context baseline, indicating that context can be useful even when described with low-level appearance features. However, our substan-

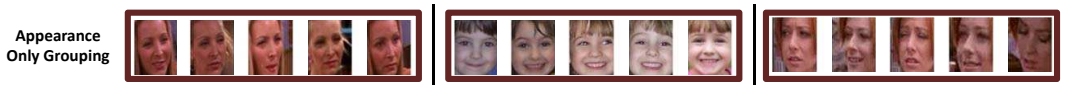
tial improvement over the appearance-context baseline shows the importance of representing context with models of familiar people. The absolute performance on the more challenging Wang1 and Wang2 datasets is slightly lower than that of the Mixture dataset. This is likely because of the larger number of unique people in those datasets. Still, my method performs well, showing practical results for real personal photo collections. Furthermore, discovery succeeds just as well when the number of unknown people is increased (top to bottom in Figure 3.31 (b)).

We also explored taking the *least* frequent people to be known on the Wang datasets. In this case, my method attains similar clustering performance to the baselines. This is due to those people appearing in only one or two photos in the collection. Thus, meaningful models cannot be learned, which results in unreliable social context descriptors. Although this is a failure mode of my method, it is reasonable to assume that the most frequently appearing people, as opposed to those that seldom appear, would likely be tagged. In future work, I would like to consider how the system could even suggest which faces a user should tag as initially familiar, so as to maximize discovery performance. For example, the system could select face clusters that are large and tight.

Figure 3.32 (a) shows qualitative results. The representative faces of each discovered person exhibit a wide range of pose and/or illumination variations, and would not have been grouped if only facial appearance were considered. By leveraging the context from familiar people, we successfully group faces belonging to the same person. In contrast, when forming groups using only appearance cues, the discovered faces exhibit limited variability in pose or expression (see Figure 3.32 (b)). I show the impact of these differences on



(a) Discovery using social context



(b) Discovery using only appearance

Figure 3.32: Face discovery examples. (a) The first row shows representative faces of the dominant person for a discovered face, with their respective co-occurring faces below. The second row faces belong to a known person—their social context helps to group the diverse faces of the same person in the first row. The numbers indicate the ground-truth face ID. (b) Limitations of appearance-based grouping. The images show representative faces of the dominant person for a discovered face using only appearance features. Notice the limited variability in pose and expression of each grouped person, as compared to our discoveries in (a).

predicting novel tags with the discovered face models at the end of this section.

3.3.2.2 Impact of Known/Unknown Decisions

I next evaluate how accurately we predict novel instances to be familiar or unfamiliar. For this, we compute precision-recall curves, treating the known instances as positive and the unknowns as negative. See Figure 3.33. Our choice of the known/unknown cutoff point (indicated by the red star) leads to accurate classification for the true knowns (among the ones we determine to be known) at the cost of including some of them in the pool of unknowns. This result is especially relevant for the face tagging scenario, since the system should provide the user with a wide variety of unfamiliar (i.e., untagged) people to tag.

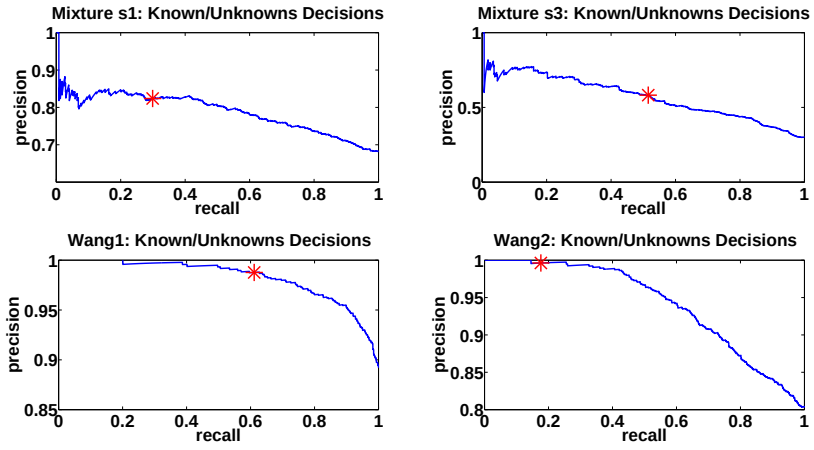


Figure 3.33: Precision-recall curves showing the known/unknown estimates.

While we fix the selection criterion to make all known/unknown decisions in Figure 3.31, in order to further test our method's robustness to those predictions we measure discovery accuracy while varying the entropy cutoff value. When setting the maximum entropy value at which a face is unknown as $t = \{0.2, 0.3, \ldots, 0.6\}$, we observe consistent improvement (0.01 to 0.09 points) over the baselines.

3.3.2.3 Face Recognition in Novel Images

Finally, I evaluate how our discovered faces can be used to predict tags in novel photos. This experiment simulates an interactive face-tagging application, where the user is presented a cluster of faces that the system discovers, and the human tags it with the appropriate name. The system can then automatically tag other instances of that person given new images (for example, when the user uploads new batches of photos to her online photo collection). For this task, we use the Mixture dataset since it has a more balanced distribution in frequency counts of people in the data, providing

	Mixture split1			Mixture split2		
	Ours	No-Context	App-Context	Ours	No-Context	App-Context
k=10	0.22	0.23	**0.29**	**0.28**	0.16	0.20
k=20	**0.30**	0.17	0.25	**0.21**	0.14	0.16
k=30	**0.27**	0.18	0.24	**0.19**	0.12	0.16
	Mixture split3			Mixture split4		
	Ours	No-Context	App-Context	Ours	No-Context	App-Context
k=10	**0.25**	0.18	0.22	**0.22**	0.18	0.21
k=20	**0.22**	0.17	0.20	**0.22**	0.14	0.19
k=30	**0.22**	0.14	0.17	**0.18**	0.10	0.13

Table 3.2: Face prediction on novel images with discovered faces on the Mixture dataset, as measured by classification accuracy. Note that the number of discovered clusters, k, is equivalent to the cost of human tagging effort required to map the discovered faces to predictive models. The models learned from faces discovered using social context generalize better than the baselines on novel face instances. The results show that our approach can serve to save human tagging effort.

a better testbed to evaluate prediction accuracy. The Wang datasets have heavy-tailed distributions in which a handful of people occur very frequently while the remaining people appear in only a few photos.

We classify the unknown instances in a third subset of the image data that is disjoint from both the subset on which we learned the initial familiar people models and the subset on which we performed discovery. There are 510, 600, 1152, and 1043 test instances for each split (1-4), respectively.

We train one-vs-one SVM classifiers for the discovered faces using the appearance descriptors. We label each discovered face cluster with its majority instance ground-truth tag. For this experiment, we vary the number of face clusters k that the system discovers in order to analyze the tradeoff between manual tagging effort and recognition accuracy.

Table 3.2 shows the result. For almost all k on each split, we consistently classify novel instances of discovered people much better than either baseline (the App-Context baseline performs the best on split1, $k = 10$). This result shows that the models learned from faces discovered using social context generalize better on novel face instances than those learned from faces discovered using appearance alone, and is evidence that my approach can indeed serve to save human tagging effort.

3.4 Discussion

In this chapter, I presented a novel context-aware category discovery framework for unsupervised learning of categories. In particular, I showed how to model the interaction between familiar categories and unknown regions to discover novel categories in unlabeled images. I evaluated my approach for two applications—object category discovery in natural images and face discovery in consumer photo collections—and showed significant improvements over traditional appearance-based discovery approaches. The results clearly demonstrated the value of breaking the stark division between the supervised and unsupervised learning paradigms for visual category discovery.

Unlike existing discovery frameworks that assume no prior category knowledge and focus only on the appearance of the image regions, my approach assumes that it is given a set of categories for which it has trained models, and uses those models as object-level context to describe an unfamiliar region. As a result, the clusters tend to be more inclusive of intra-class appearance variation than those that could be found with methods that rely only on appearance, which leads to significant improvements in discovery accuracy.

I also showed how to target the easier categories first through a self-

paced curriculum, in which the system discovers a single category at a time, in order of predicted easiness. To this end, I introduced a measure of easiness that uses two criteria automatically estimated from the unlabeled data: "objectness" and "context-awareness". In contrast to a one-pass k-way partitioning, my approach gradually accumulates models for larger portions of the data and is more robust to outliers in the data.

What are the main assumptions of my approach? For any object or face to be discovered, its visual pattern must be recurring, and its surrounding familiar objects should belong to a similar set of categories and share similar configurations or co-occurrence patterns across the image collection. This means that co-occurring objects that are often segmented together, such as bicycles and bicycle racks, can be discovered as a single category. Note however that in my experiments, I evaluated discovery given human labeled categories, in which case bicycles and bicycle racks are treated as separate categories; i.e., we will be penalized for grouping them together. Furthermore, in many applications, it is natural to assume that the data will have some repeating categories. For example, personal photo collections contain many recurring faces of the same people. Nonetheless, for any arbitrary dataset, this assumption may not hold. My self-paced framework is most suitable for these datasets, since it will only discover categories among the easiest instances, and ignore harder instances that cannot be grouped well.

Among the many challenges of visual discovery, we learned that evaluation can be particularly challenging. We learned that it is important to evaluate both the quality of the grouping as well as the segmentation quality of their cluster instances. I revisit this issue in more detail in Chapter 6.

My self-paced approach, being a sequential learning algorithm, can

be susceptible to error propagation from earlier mis-predictions. Although it greatly mitigates those errors by targeting the easiest instances at each time step, we observed that redundant categories can be discovered and outliers can be included in some discoveries. This suggests that a human-in-the-loop could enhance discovery performance. For example, the system could present each discovered category to a human and ask him or her to merge it with an existing category or to prune it. The challenge would be to devise the most cost effective form for obtaining human assistance at each discovery cycle.

While throughout we have fixed the relative weighting between the appearance and context terms in the kernel used for region grouping, we could also learn the weights on a held-out validation set. Ideally, we would learn category-specific weights, since the weighting should depend on the category (e.g., appearance is more important to group grass patches, whereas to group car regions, context could be more important). Multiple kernel clustering [171] in my self-paced discovery framework could be a solution, where at each iteration we would apply a different set of appearance and context weights to maximize the separation between the easiest category and the remaining harder instances.

We found that our clusters can be imperfect, which shows the difficulty of simultaneously producing accurate segmentations and correct grouping of regions in natural image collections. However, my main message that we do not need a stark division between the supervised and unsupervised learning paradigms was confirmed through extensive experiments and is independent of discovery quality. Admittedly, known/unknown detection, or more generally "novelty detection", is a very difficult problem. It would be interesting to investigate ways to provide more robust known/unknown decisions, either

avoiding it all together by directly encoding the known/unknown confidences into the clustering, or by using constraints and/or input from human interactions.

Finally, in Sections 3.2.2.5 and 3.3.2.3, we simulated a human labeling the discovered categories to demonstrate a practical application for category discovery. In future work, one could consider how to best add real human supervision. The method could display a summary of each discovery (e.g., the most confident instances) to the human, who would then label it for the system to learn a model for automatic prediction in novel images.

So far, I have shown how to discover novel object categories in natural image collections, using a fixed set of candidate regions computed using multiple segmentations. However, a limitation of a fixed set of regions is that there could be some objects, especially those that are non-homogeneous in appearance, that are not accurately represented. In the next chapter, I will show how to extend the current framework to use the discovered categories as top-down cues to perform object segmentation in images and videos.

Chapter 4

Segmentation with Discovered Top-Down Cues

In this chapter, I show how to use discovered top-down cues to perform object segmentation in images and videos. Specifically, for object category discovery, rather than commit to a pool of candidate bottom-up segments, my approach allows any initially discovered shared appearances in the image collection to influence segmentation boundaries, and then in turn, lets the refined regions influence the category-level grouping. For video segmentation, I take the same idea and show how to use automatically discovered top-down object-level cues to segment the foreground objects without any human supervision.

In the ensuing sections, I first apply this idea for collective image segmentation (Section 4.1), and then show how to apply it for video object segmentation (Section 4.2).[1]

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

In Sections 3.1 and 3.2, I presented a framework to discover novel object categories from unlabeled image collections. However, there is still an unresolved problem. Namely, we assumed that each semantic object will have a corresponding sub-image in the pool of multiple segmentations of bottom-up

[1] I first presented the ideas in this chapter in [88, 93].

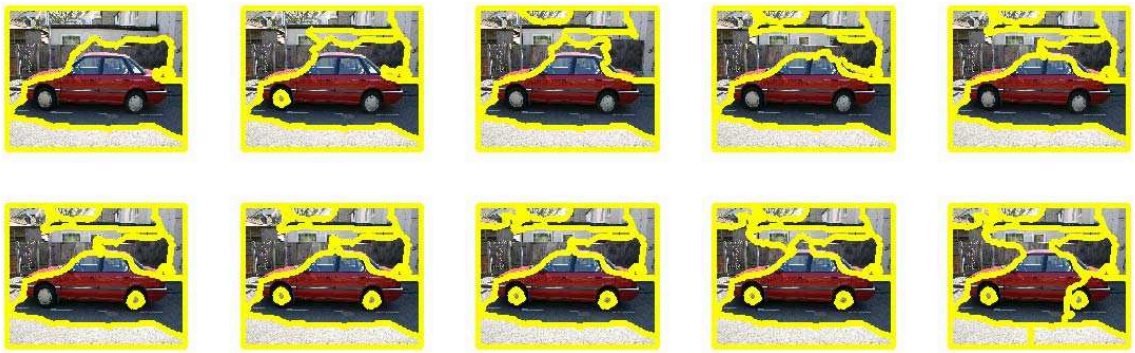


Figure 4.1: Bottom-up segmentation methods can only group regions with similar color or texture. Even with multiple segmentations, the car cannot be correctly segmented.

regions or objectness windows.

While using multiple segmentations helps safeguard against missing "good" regions, there is still a risk of omitting meaningful segments from the pool, and thus the system may never have 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 true object boundaries (see Figure 4.1). In fact, recent studies [103] 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 a typical natural image contains only about 10 objects. While the objectness measure helps to generate more object-like regions, since it lacks category-specific top-down knowledge, it still faces the same challenges as bottom-up segmentation methods, albeit to a lesser degree. With multiple objects present within a single image, a discovery 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.

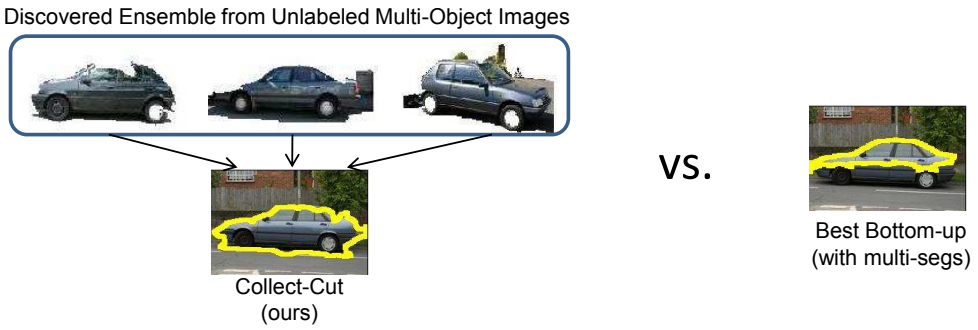


Figure 4.2: I devise an algorithm to discover and exploit the shared structure in a collection of unlabeled images in order to segment the objects more accurately than is possible with bottom-up methods.

This implies a notable computational burden for existing discovery 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.

My 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 (see Figure 4.2).[2] Rather than commit to a pool of candidate segments, my 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

[2]I published the work described in this section in [88].

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. I design an energy function amenable to graph cuts 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., [125, 126]), my 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 the use of an ensemble foreground model, as well as the integration of my context-aware discovery algorithm to find the initial groups. Finally, by favoring cuts that separate familiar and unfamiliar regions, my 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.

I illustrate the proposed 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, I illustrate the positive impact the refined segmentations have on unsupervised category discovery, thus enhancing the impact of the context-aware discovery approach from Chapter 3.

4.1.1 Approach

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. I call the method "Collect-Cut" since it uses the image *collection* to estimate the graph cut-based segmentation.

The proposed method works as follows: given a pool of unlabeled images, we decompose each into multiple segmentations. After clustering the segments from all images, for each group the method chooses representative instances to act as an *ensemble* of possible appearance models. 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. I formulate these preferences in an energy function amenable to graph cuts algorithms. Finally, having refined each region, we recompute a clustering on all regions. The final output is a set of segmented discovered objects.

In the following, I first briefly describe how we obtain the initial region groups among the multiple segmentations (Section 4.1.1.1); from each of those we extract a set of representative exemplars, as explained in Section 4.1.1.2. Then, in Section 4.1.1.3, I introduce the energy function that will express how every region from the original segmentation should be adapted to align with the preferences described above. Finally, we reform the discovered categories based on the refined regions in Section 4.1.1.5.

4.1.1.1 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 [89] that I described in Section 3.1. As we have seen, 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. 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., [127, 134]), as I will illustrate in the experiments.

4.1.1.2 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. 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., [132]), 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 (see Eqn. 3.3). 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 Figure 4.3 (c) for an example). When refining a region's boundaries, the idea is to treat resemblance to *any one* of the representative ensemble regions as support for the object of interest, as described in the following section.

4.1.1.3 Collective Graph-Cut Segment Refinement

Given the discovered ensemble models, we take each initial "seed" region and refine its segmentation using graph cuts [17, 18, 78]. 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.

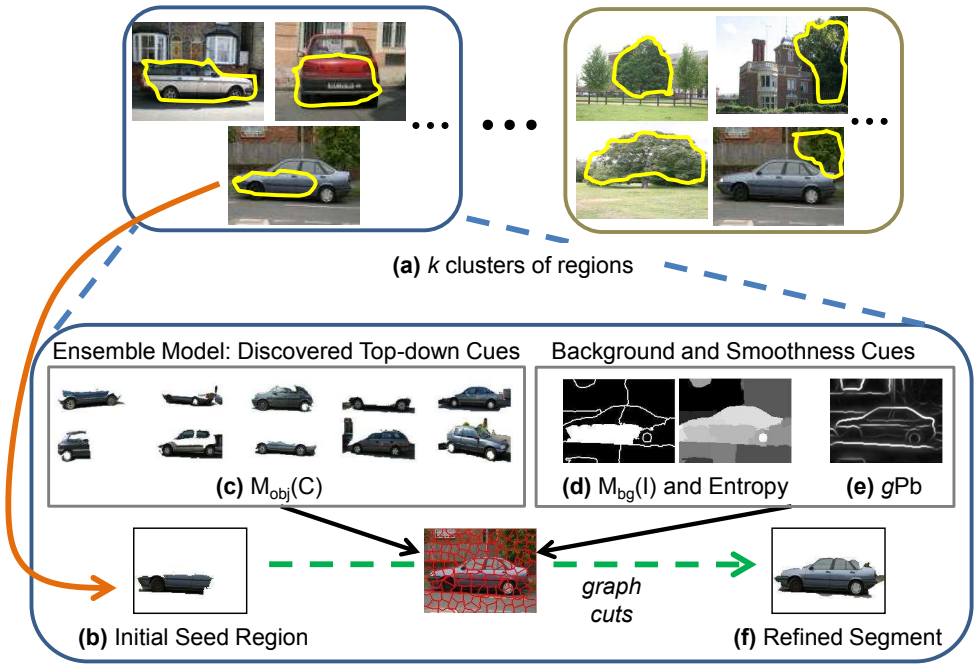


Figure 4.3: Overview of the proposed method. We use graph cuts to minimize an energy function that encodes top-down object cues, entropy-based background cues, and neighborhood smoothness constraints. In this example, suppose the familiar object categories are *building* and *road*. (a) A set of k clusters 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 [102] 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.

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

$$\chi^2(s_1, s_2) = \frac{1}{2}(\chi^2_{color}(s_1, s_2) + \chi^2_{texton}(s_1, s_2)). \quad (4.1)$$

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.[3] 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:

$$E(f, s_{seed}) = \sum_{i \in \mathcal{S}} D_i(f_i) + \sum_{i,j \in \mathcal{N}} V_{i,j}(f_i, f_j), \quad (4.2)$$

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

[3] 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.

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. 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:

$$D_i(f_i) = \begin{cases} \exp\left(-d(p_i, M_{obj}(C))\right), & \text{if } f_i = 0; \\ \exp\left(-d(p_i, M_{bg}(I))\right), & \text{if } f_i = 1. \end{cases} \tag{4.3}$$

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 either when a superpixel is labeled as background but has a low distance to the ensemble model, or when it is labeled as object but has a low distance to the 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 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 the distances between superpixel p_i and each model

as:

$$
\begin{aligned}
d(p_i, M_{obj}(C)) &= \min_j \chi^2(p_i, s_{C_j}), \text{ for } s_{C_j} \in M_{obj}(C), \\
d(p_i, M_{bg}(I)) &= \chi^2(p_i, s_{I_k^*}), \text{where} \\
k^* &= \operatorname*{argmin}_k \; w_k \; \chi^2(p_i, s_{I_k}),
\end{aligned} \tag{4.4}
$$

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:

$$
\begin{aligned}
w_k &= (-\log H(s_{I_k}))^{-1}, \text{where} \\
H(s_{I_k}) &= -\frac{1}{\log_2 N} \sum_{n=1}^{N} P(o_n|s_{I_k}) \log_2 P(o_n|s_{I_k}),
\end{aligned} \tag{4.5}
$$

and $o_1, \ldots, o_N$ are the N familiar object models used by our context-aware discovery method. 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 Figure 4.3 (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. (If there are no previously learned category models, we simply replace this entropy-based term with a spatial distance term; see below.)

Finally, we define the smoothness term in Eqn. 4.2 as:

$$V_{i,j}(f_i, f_j) = |f_i - f_j| \cdot \exp(-\beta \cdot z(p_i, p_j)), \tag{4.6}$$

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 [102] detector (see Figure 4.3 (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. 4.2 with graph cuts [18], and use the resulting label assignment as the refined segmentation for region s_{seed} (see Figure 4.3 (f)).

4.1.1.4 Fully Unsupervised Variant

While we are most interested in the case where we have access to some previously learned familiar object category models, the Collect-Cut segmentation technique is also applicable in the fully unsupervised setting with some minor changes. To apply our framework in the fully unsupervised setting, we replace the context-aware clustering from Section 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 [127], which uses Latent Dirichlet Allocation (LDA) 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 $H(s_{I_k})$

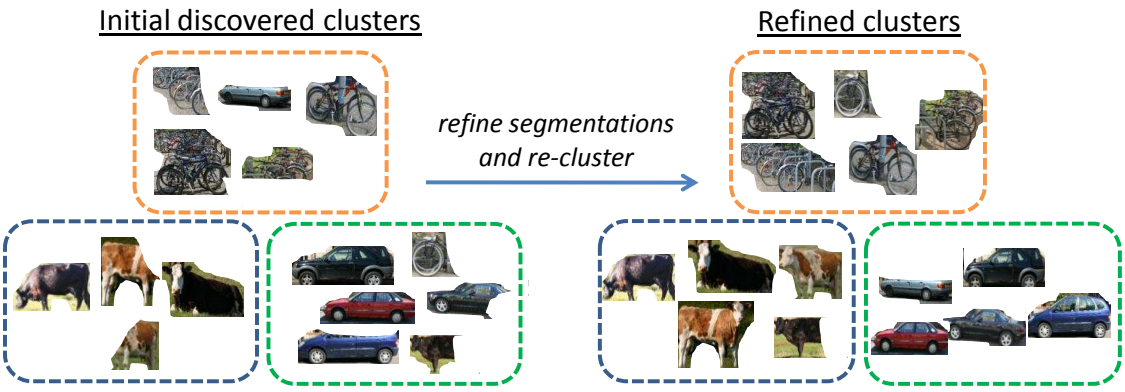


Figure 4.4: Illustrative example showing the discovery iteration process. Compared to the initial discovery outputs, we can form better groups by clustering the refined segmentations from our collect-cut algorithm.

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.

4.1.1.5 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 graph-cut, the discovery procedure can (potentially) form better groups than were possible at the previous iteration. See Figure 4.4.

4.1.2 Results

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

Datasets We use the MSRC v0 and v2 datasets from Section 3.1.2 in the previous chapter. 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. For MSRC-v2, we take splits $s1$ ={*building, tree, cow, airplane, bicycle*} or $s2$ ={*tree, sheep, car, bicycle, sign*}, as the unknown classes. For the MSRC-v0, we take splits $s1$ ={*cow, sheep, airplane, car, bicycle*} or $s2$ ={*tree, sheep, chimney, door, window*}, as the unknown classes. When evaluating the method without using familiar objects as context, all 21 classes are considered as unknown. See Figure 3.7 in the previous chapter for image examples.

Implementation details We use Normalized Cuts [132] 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 [102]. 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. For the method of [127], we use affine-covariant regions described with SIFT descriptors. For each segment, we build a BOF histogram with quantized SIFT features. We normalize each histogram to sum to one. We fix $\beta = 10$ for the smoothness term after examining a few image examples. We

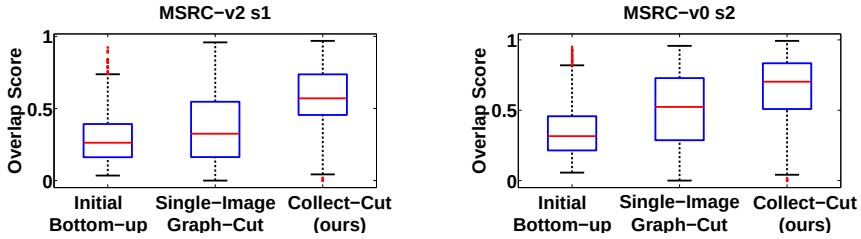
set $r = 5, 10$ for the MSRC-v2 and v0, respectively. We present results only for $k = 30$, which is approximately the number of object types in the datasets.

4.1.2.1 Object Segmentation Accuracy

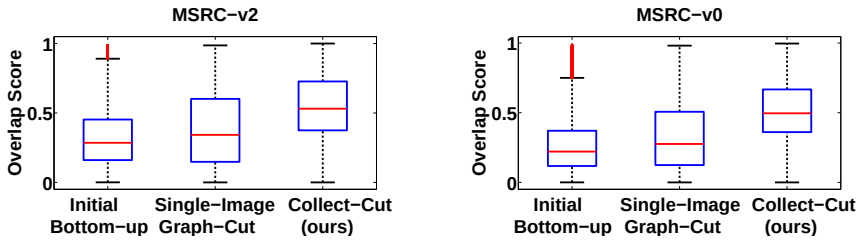
I first evaluate my method's segmentation results. To quantify accuracy, we use the pixel-level segmentation *overlap score*, OS. The quality of segmented region R with respect to the ground-truth object segmentation GT is measured as: $OS = \frac{|GT \cap R|}{|GT \cup R|}$, where we take as GT the full 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.

I compare my **Collect-Cut** method against two baselines: (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 loosely modeled this baseline after a model devised in [168], and it represents the best we could do if trying to refine the segmentation independently for each image.

Figure 4.5 shows the results. I evaluate my 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



(a) Semi-supervised via familiar-object context



(b) Fully unsupervised

Figure 4.5: 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).

regions/categories that are scored. 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, my method produces significantly better segmentations than either baseline.

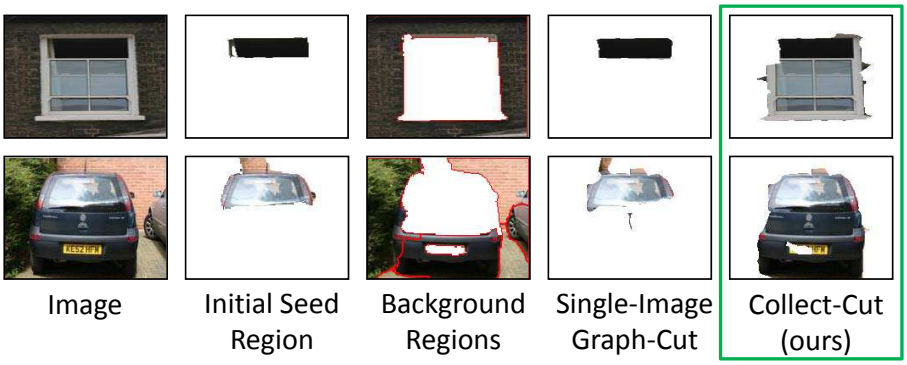


Figure 4.6: Two illustrative results comparing my Collect-Cut to the single-image graph-cuts baseline. If the initial seed region captures only a single part of a multi-part object (i.e., heterogeneous appearance), a method restricted to using only the single image for segmentation may fail. In contrast, by integrating the ensemble of discovered shared cues found in the collection, our approach can more fully recover the object's segmentation.

We noticed that the single-image baseline has particular difficulty in refining segmentations for objects with heterogeneous appearance. For example, an initial seed region may capture the windshield of a car in a cluster comprised mostly of cars and car parts (which is possible due to our context-aware discovery). The single-image baseline resists joining the windshield to the other car parts due to their contrasting appearance, whereas our ensemble model captures multiple aspects of the car, and can allow them to be labeled as "object". See Figure 4.6 for examples.

Table 4.1 shows my 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.

Figure 4.7 shows representative (good and bad) qualitative segmenta-

	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 4.1: 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.

tion examples, where we compare against the *best* segment from the initial pool of multiple-segmentations.

Figure 4.8 shows good multi-object segmentation examples, where we aggregate our method's refined object regions into a single image-level segmentation. Specifically, after refining each region in an image from the initial multiple segmentations, we will have multiple, potentially overlapping, regions in the image. We solve a new multi-label problem that enforces each superpixel in the image to select its best overlapping (larger) region. We devise an energy function in which the data term measures the cost of a superpixel being assigned to an overlapping region, and the smoothness term measures the cost of assigning different labels to neighboring superpixels. We minimize the energy using graph-cuts to obtain a single segmentation for the image.

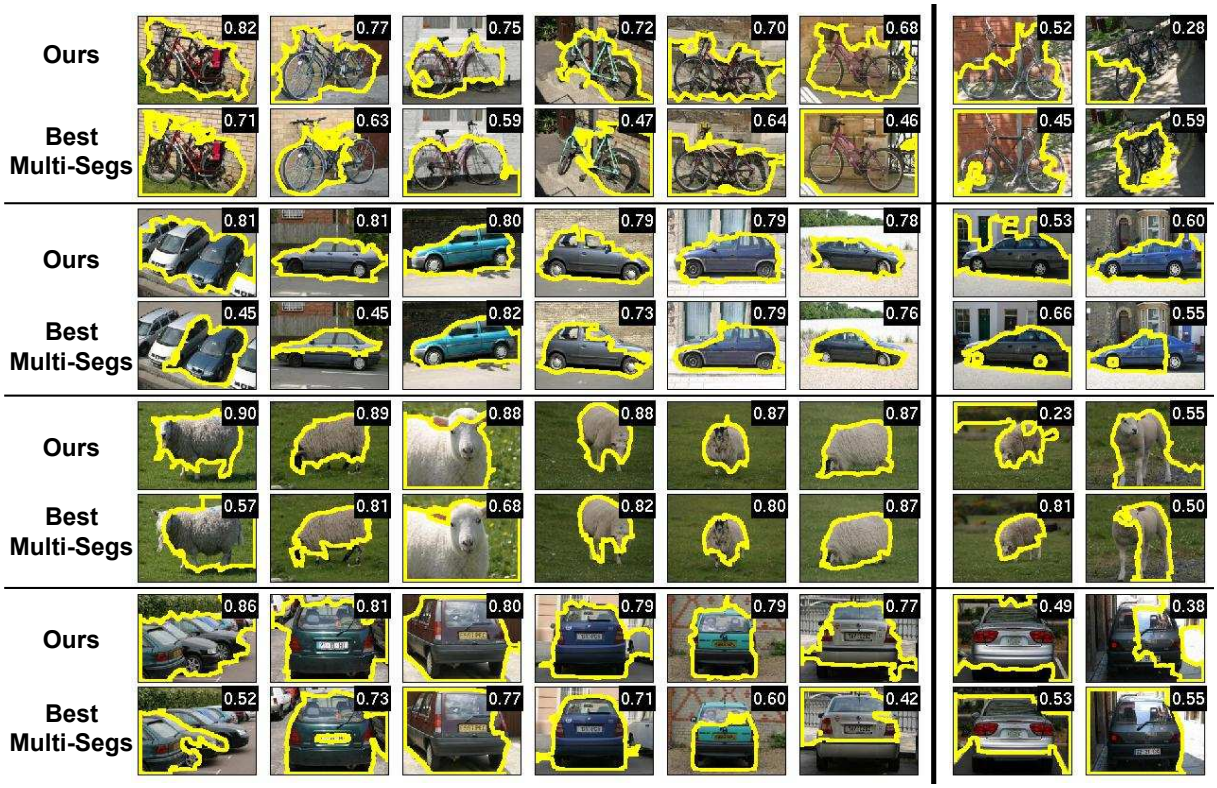


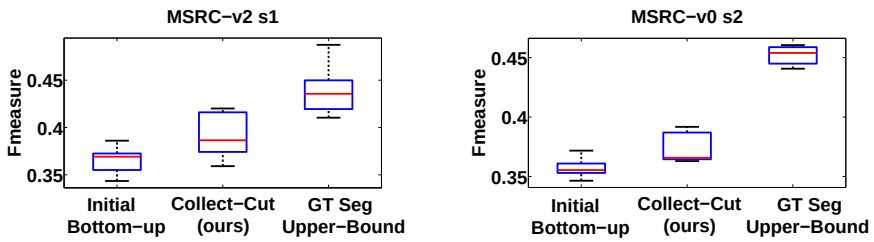
Figure 4.7: Qualitative comparison: our results vs. the best corresponding segment available in the pool of multiple-segmentations. Numbers above denote overlap scores. The **first 6 columns** are examples where our method performs well, extracting the true object boundaries much more closely than the 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.

Figure 4.8: Examples of high quality multi-object segmentation results. We aggregate our refined segmentations into a single segmentation of the image.

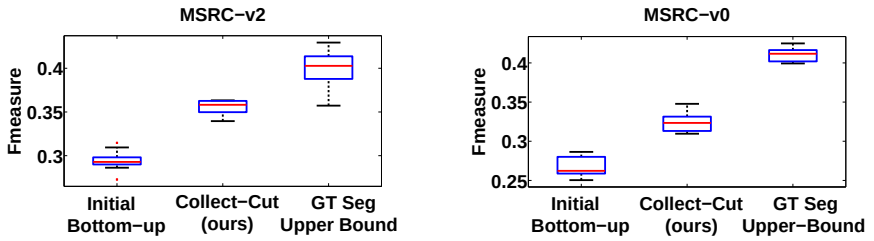
4.1.2.2 Category Discovery Accuracy

Having established that Collect-Cut segmentations can more faithfully follow true object boundaries, I 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.

Figure 4.9 shows the results. I compare three variants: (1) running discovery with the initial bottom-up multiple segmentations pool, (2) running discovery with our method's results, and (3) running discovery with the ground truth object segments, which provides an upper bound on accuracy. My 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 clus-



(a) Semi-supervised via familiar-object context



(b) Fully unsupervised

Figure 4.9: Impact of collective segmentation on discovery accuracy, as evaluated by the F-measure (higher values are better). For discovery, we plug in both (a) my context-aware clustering algorithm [89], and (b) an appearance-only discovery method [127]. In both cases, using my Collect-Cut algorithm to refine the original bottom-up segments yields more accurate grouping.

tered. The upper bound on accuracy in this experiment is imperfect—showing the limits of clustering multiple generic object categories. Note that these results bring together findings from both context-aware discovery and collective segmentation; we see that segmentation accuracy can be improved with familiar object-context, and that discovery accuracy can be improved with high quality object segmentations.

In the next section, I show how we can apply this idea of discovering top-down segmentation cues from a collection of images to perform unsupervised video object segmentation.

4.2 Key-Segments for Video Object Segmentation

The previous section showed how to segment objects in unlabeled image collections using top-down cues from discovered categories. In this section, I extend this idea to segmenting the foreground objects in unlabeled videos by using cues from a set of discovered key object segments in the data. Like an image collection, a video has an assortment of images (frames); however, unlike image collections, a single video is certain to contain repeating object *instances*. So the key challenge is how to focus on the key foreground objects, while ignoring the irrelevant background clutter.

As discussed in Chapter 2, existing unsupervised video segmentation methods explore tracking regions or keypoints over time [19, 21, 152] or formulate clustering objectives to group pixels from all frames using appearance and motion cues [56, 65]. Aside from the well-known challenges associated with tracking (drift, occlusion, and initialization) and clustering (model selection and computational complexity), these methods lack an explicit notion of *what a foreground object should look like* in video data. Consequently, similar to what we observed earlier for static images, the low-level grouping of pixels usually results in a so-called "over-segmentation".

Instead, I propose an approach that automatically discovers a set of *key-segments* to explicitly model likely foreground regions for video object segmentation. The main idea is to leverage both static and dynamic cues to detect persistent object-like regions, and then estimate a complete segmentation of the video using those regions and a novel localization prior that uses their partial shape matches across the sequence.[4] See Figure 4.10.

[4] I published the work described in this section in [93].

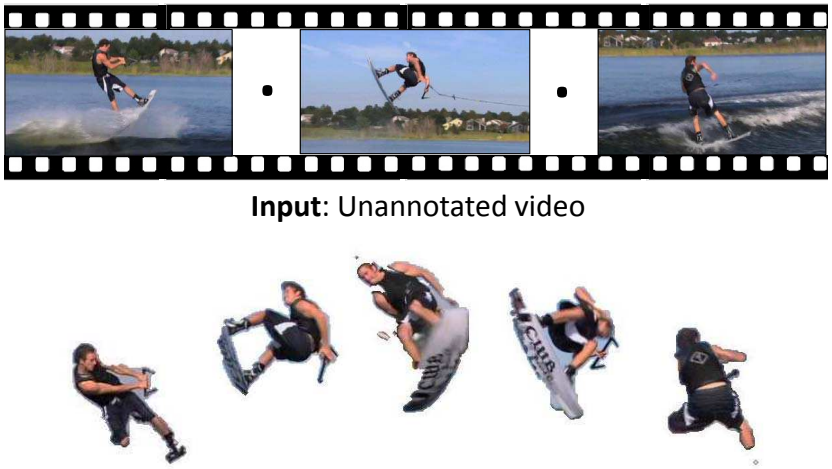


Figure 4.10: My idea is to discover a set of key-segments to automatically generate a foreground object segmentation of the video.

To implement this idea, I first introduce a measure that reflects a region's likelihood of belonging to a foreground object. To capture *object-like motion and persistence*, we use dynamic inter-frame properties such as motion difference from surroundings and recurrence. Intuitively, a region that moves differently from its surroundings and appears frequently throughout the video will likely be among the main objects of interest. Conversely, one that seldom occurs is more likely to be an uninteresting, background object. To capture *object-like appearance and shape*, we use static properties such as a well-defined closed boundary in space and clear separation from surroundings, as recently explored in static images [3, 24, 35]. We use both aspects to group the key-segments, estimating multiple inlier/outlier partitions of the candidate regions. Each ranked partition automatically defines a foreground and background model, with which we solve for a pixel-wise segmentation using graph cuts on a space-time Markov Random Field (MRF). The rank reflects

the corresponding object's centrality to the scene.

How does key-segment discovery help video object segmentation? The key-segments are a reliable source for learning the appearance of a foreground object, since they were determined to be both object-like and frequently occurring in the video. Furthermore, key-segments detected across the sequence imply probability distributions for the location and scale of the object in other frames, which we show how to capture through a novel partial shape matching localization prior. What is the advantage of having a *group* of key-segments? An ensemble alleviates imprecise segmentations on any individual key-segment and captures background diversity in the video, since the background visible in each key-segment's frame can vary. In practical terms, my approach substantially reduces annotator effort; rather than outlining an object of interest, one can simply use (or peruse) the suggested foreground object(s).

To my knowledge, no prior work explores category-independent foreground segmentation for videos where simple background subtraction is insufficient. Towards this goal, important novel components of my technique include (1) a new motion-based measure of object-like regions in video that complements existing image-based cues, (2) a localization prior using partial shape matches in video, and (3) a space-time graph segmentation that accommodates the key-segments. I apply my unsupervised method to challenging benchmark videos, analyze its components in detail, and show state-of-the-art results compared to existing unsupervised and supervised methods.

Related work Video object segmentation is often performed in an interactive or supervised way. Interactive methods require a user to annotate object boundaries in some key frames, which are then propagated to other frames while a user stands by to adjust errors [10, 117, 169]. Tracking-based methods

attempt to reduce the supervision to a manual segmentation on only the first frame (e.g., [124, 148]). However, all such methods demand user input drawing regions of interest, and may suffer from sensitivity to a user's annotation expertise.

Bottom-up approaches can segment videos in a fully automatic manner, based on cues like motion and appearance similarity. Motion segmentation methods (e.g., [131]) cluster pixels in video using bottom-up motion cues. Recent methods either perform pixel-level segmentation in a spatio-temporal video volume from scratch [56], begin with an image segmentation per frame and then match segments across nearby frames, e.g., [19, 65, 152], or use dense flow to cluster long-term motion trajectories [21]. Without any top-down notion of objects, however, such methods tend to over-segment, yielding regions that taken alone may lack semantic meaning.

4.2.1 Approach

The goal is to discover object-like *key-segments* in an unlabeled video, and learn appearance and shape models from them to automatically segment the foreground objects. This is directly building on my "Collect-Cut" approach from the previous section, except now we focus on segmenting the recurring foreground objects instance in a single video.

There are three main steps to my approach: (1) scoring each image region using appearance and motion cues to determine how likely it is to belong to a foreground object; (2) clustering the regions to discover key-segments that represent a single object, and ranking those clusters according to their region scores; and (3) segmenting each foreground object in the video using its model learned from the corresponding key-segments. The final output is a ranked set

of foreground object segmentations. I now describe each step in turn.

4.2.1.1 Finding Object-like Regions in Video

In order to segment a foreground object in the video, we first need a representation of that object. Since we assume no prior knowledge on its size, location, shape, or appearance, we initially generate a diverse set of object "proposals" in each frame using the static region-ranking method of [35]. The proposals are guided by models learned from true segmentations of arbitrary objects[5], and have been shown to better align with object boundaries than traditional bottom-up segments do. For each frame in the video, we generate roughly 1000 regions.

To find "object-like" regions among the proposals, we look for regions that have (1) appearance cues typical to objects in general, and (2) differences in motion patterns relative to their surroundings. These properties are well-suited for defining objects in video; any region that is salient in terms of both appearance and motion may correspond to a true object. This is tied to our "easiness" criterion in Chapter 3, which looks for regions that are object-like in appearance, and are surrounded by familiar objects in static images. Specifically, we define a function:

$$S(r) = A(r) + M(r), \tag{4.7}$$

that scores a region r according to its static intra-frame appearance score $A(r)$ and dynamic inter-frame motion score $M(r)$. See Figure 4.11.

[5]Note those exemplars are *disjoint* from the objects appearing in the videos we process; specifically, the region proposal function of [35] was trained with Berkeley Segmentation data.

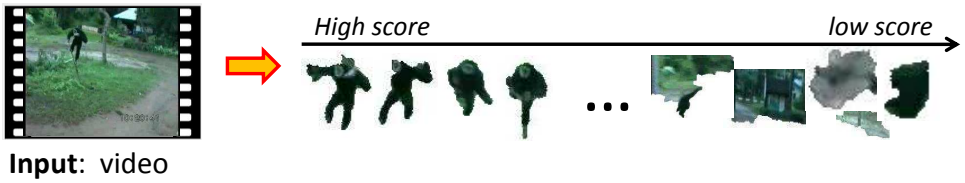


Figure 4.11: We score each region according to their object-like appearance and motion scores.

We compute $A(r)$ using [35]. It reflects cues indicative of a generic object, such as the probability of a surrounding occlusion boundary, color differences with nearby pixels, and the probability of belonging to a vertical surface. Note this measure only looks at the appearance of the region within each frame, and does not care about the motion.

We compute $M(r)$ to measure the confidence that region r corresponds to a coherently moving object in the video. We compute optical flow histograms for the region r and the pixels $\bar{r}$ around it within a loosely fit bounding box, and then score r as:

$$M(r) = 1 - \exp(-\chi^2_{flow}(r, \bar{r})), \tag{4.8}$$

where $\chi^2_{flow}(r, \bar{r})$ is the χ^2-distance between L_1-normalized optical flow histograms. Note that this cue is not simply looking for large motions or appearance changes from background (e.g., as one would in background subtraction). Rather, we are describing how the motion of the proposal region differs from its closest surrounding regions; this allows us to forgo assumptions about camera motion, and also to be sensitive to different magnitudes of motion. Furthermore, the region r itself is a product of an object-like ranking, not an arbitrary bottom-up segment.

Before combining $A(r)$ and $M(r)$, we standardize each to zero-mean unit-variance using the distribution of scores across all regions in the video.

4.2.1.2 Discovering Key-Segments Across Frames

Given the scored regions, we next identify *groups* of *key-segments* that may represent a foreground object in the video. For each frame, we take the top N highest-scoring regions to form a candidate pool $\mathcal{C}$ spanning the entire sequence. Many regions belonging to a foreground object should be present in $\mathcal{C}$ (as they were predicted to be most "object-like"), but there may also be noisy segments. Thus, we specifically treat this stage as gathering multiple hypotheses among the highly ranked object-like regions, computing multiple partitions of $\mathcal{C}$. In Section 4.2.1.3, I will explain how to use them to segment the entire video.

To extract the groups, we first define similarity between two regions r_m and r_n:

$$K(r_m, r_n) = \exp(-\frac{1}{\Omega}\chi^2_{color}(r_m, r_n)), \tag{4.9}$$

where $\chi^2_{color}(r_m, r_n)$ is the χ^2-distance between unnormalized color histograms of r_m and r_n, and Ω denotes the mean of the χ^2-distances among all regions. This measure gives high affinity to regions that have similar color and similar size. We compute the pairwise affinities between all regions m,$n \in \mathcal{C}$, to obtain the affinity matrix $K_{\mathcal{C}}$.

We next perform a form of spectral clustering [112, 114] with $K_{\mathcal{C}}$ to produce multiple binary inlier/outlier partitions of the data, with the objective of maximizing the inliers' intra-cluster affinity (normalized by the number of inliers). Each eigenvector of $K_{\mathcal{C}}$ produces a partitioning of the data; we binarize

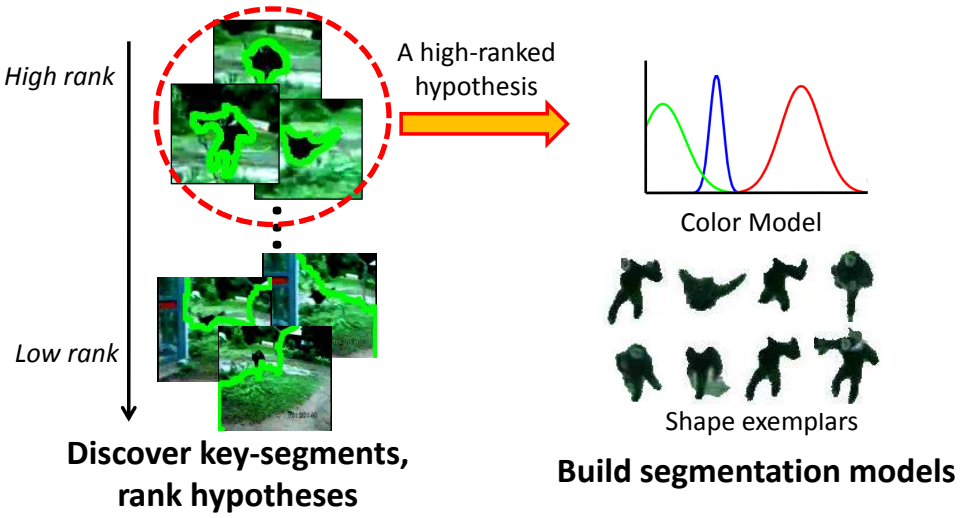


Figure 4.12: We rank the discovered key-segment groups according to their average object-like score, and build segmentation models for each object hypothesis.

the continuous eigenvector to form an indicator vector that denotes the inlier set, using the technique in [112].

Each cluster (inlier set) is a hypothesis h of a foreground object's key-segments. We automatically rank the clusters based on the average object-like score $S(r)$ of its member regions. If that scoring is successful, the clusters among the highest ranks will correspond to the primary foreground object(s), since they are likely to contain frequently appearing object-like regions (as we confirm in Figure 4.18 below).

4.2.1.3 Foreground Object Segmentation

Each ranked partition ("key-segment hypothesis") automatically defines a foreground and background model. For now, suppose we extract a

color distribution and set of shape exemplars for each hypothesis (see Figure 4.12). We next devise a space-time Markov Random Field (MRF) model that uses these models to guide a pixel-wise segmentation for the entire video. In practice, we process the hypotheses in rank order, exploiting the quality of the object-like ranking discussed above.

Importantly, a top-ranked hypothesis helps form models of both the object itself *and* the remaining background objects, for two reasons. First, the foreground features common to the selected key-segments are more pronounced, while unique or isolated features are discounted. Second, the diversity in background appearance is captured through the (potentially) different backgrounds present in each key-segment's frame. For example, as the camera pans to follow a primary object of interest, the surrounding background can change substantially; so long as a key-segment hypothesis spans frames from various backgrounds, it will help propagate the figure-ground labeling accordingly.

Space-time graph definition We define a graph over each frame's pixels: a node corresponds to a pixel, and an edge between two nodes corresponds to the cost of a cut between two pixels. The energy function we minimize for hypothesis h takes a familiar form, similar to Eqn 4.2 from the previous section:

$$E(f,h) = \sum_{i \in \mathcal{S}} D_i^h(f_i) + \gamma \sum_{i,j \in \mathcal{N}} V_{i,j}(f_i, f_j), \tag{4.10}$$

where f is a labeling of the pixel nodes, $\mathcal{S} = \{p_1, \ldots, p_n\}$ is the set of n pixels in the video, $\mathcal{N}$ consists of neighboring pixels, and i and j index the pixels. Each pixel p_i is assigned to $f_i \in \{0, 1\}$, where 0 corresponds to background and 1 corresponds to foreground. The pixel neighborhood $\mathcal{N}$ consists of four

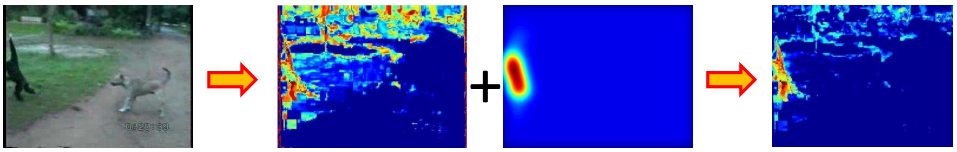


Figure 4.13: We compute the foreground likelihood in each frame of the video using the discovered color and shape models.

spatially neighboring pixels in the same frame, and two temporally neighboring pixels in adjacent frames. We assign a pixel's temporal neighbor in the next frame by its optical flow vector displacement. Related space-time graphs are defined in [148, 152].

The neighborhood term $V_{i,j}$ encourages label smoothness in space and time. We use a standard contrast-dependent function defined in [125], which favors assigning the same label to neighboring pixels that have similar color.

The data term D_i^h defines the cost of labeling pixel i with label f_i, given key-segments in h. Specifically,

$$D_i^h(f_i) = -\log\left(\alpha \cdot U_i^c(f_i, h) + (1-\alpha) \cdot U_i^l(f_i, h)\right), \tag{4.11}$$

where $U_i^c(\cdot)$ is the color-induced cost, and $U_i^l(\cdot)$ is the local shape match-induced cost. Both terms are depicted in Figure 4.13, and explained in detail next.

Appearance-based models To model the foreground and background appearance, we estimate two Gaussian Mixture Models (GMM) in RGB colorspace: (1) a GMM fg^{color} for pixels in h's key-segments; and (2) a GMM bg^{color} for pixels in the complement of h's key-segments, among all frames in h. We set $U_i^c(f_i, h)$ to be the pixel-likelihoods computed from each GMM. A

pixel that has similar color to the foreground (background) object will have high cost if labeled as background (foreground).[6]

Location priors via partial shape matching Beyond simple appearance terms, for video segmentation, we also want to exploit the consistency of recurring foreground objects viewed over time. In particular, we have a strong localization prior from one frame to the next. Our use of optical flow to define neighbors partially captures this via label smoothness, but is closely tied to appearance agreement and can fail when the foreground and background GMMs share similar color components. Thus, as the final component of our model, we introduce a novel technique to prime the location and scale of the foreground object in a frame using key-segment shapes.

The main idea is to use the key-segments detected across the sequence, projecting their shapes into other frames via local shape matching. The spatial extent of that projected shape then serves as a location and scale prior in which we prefer to label pixels as foreground. Since we have multiple key-segments and many possible local shape matches, many such projected shapes are aggregated together, essentially "voting" for the location/scale likelihoods. See Figure 4.14.

More specifically, we project the key-segments onto each frame in the video by matching Boundary Preserving Local Regions (BPLR) [75]. A BPLR is a densely-extracted local feature that preserves object boundaries and partial shape.[7] For each video frame, we generate BPLRs and retain for shape matching those that produce better (lower distance) matches to the BPLRs of

[6]Note the $-\log(\cdot)$ in Eqn. 4.11.

[7]Other detectors are feasible, but we specifically choose BPLR due to its robustness when matching deformable objects.

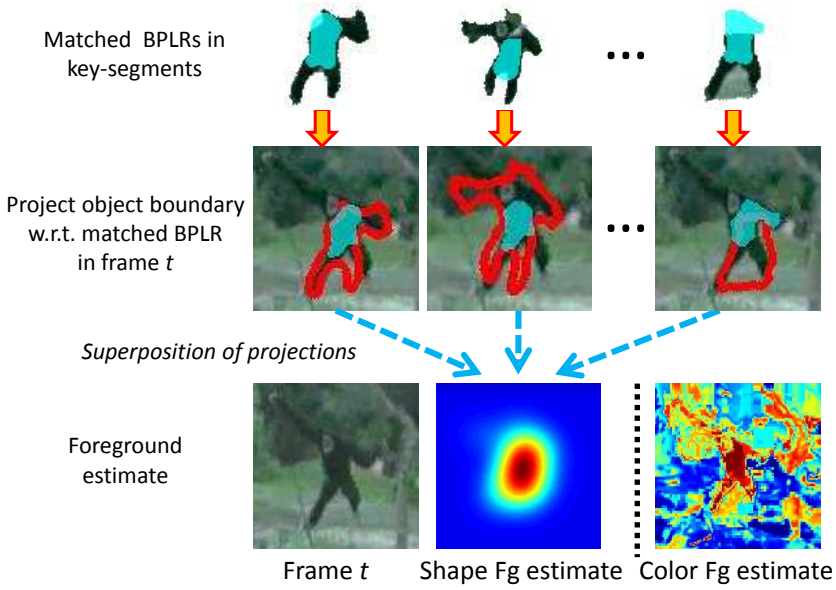


Figure 4.14: Foreground location and scale estimates with BPLR matches.

the key-segments than to the BPLRs of their image complements. We create a vote space that has the same size as the frame, and project the matched key-segment onto the frame after aligning the locations and scales of the matched BPLRs. We weight the votes according to the match similarity. This process is repeated for all retained BPLRs, and we normalize the vote space such that the maximum value is one.

Then, the vote value at p_i gives its foreground location likelihood:

$$U_i^l(f_i) = \begin{cases} P(p_i|bg^{shape}(h)), & \text{if } f_i = 0; \\ P(p_i|fg^{shape}(h)), & \text{if } f_i = 1, \end{cases} \tag{4.12}$$

and the background location likelihood is its complement. $U_i^l(f_i)$ measures whether a pixel lies in a projected region of the key-frames. Pixels that are part of a commonly projected region will have high probability of being labeled as foreground. See "Shape Fg estimate" in Figure 4.14.

When is this most useful? By using partial (local) shape feature matches to drive each shape projection, we intend to account for deformations and articulations that the foreground object may exhibit. For example, a running monkey's global shape can vary significantly from frame-to-frame. However, its arms and legs will only undergo small changes in shape. Thus, a local match (e.g., at the arm or leg) derived from a key-segment can usefully map in the rough global shape prior, despite the change in pose.

In addition, this likelihood helps disambiguate labels when there are similar colors in both the foreground and background models, or if there is a background object that did not appear in any of the key-segments' frames. Note that the key-segment color models only capture cues *within* their own frames. This means that the background objects that appear in the non-key-segment frames are not modeled, and may easily be mislabeled as foreground. For example, if a tree with brown leaves appears behind a brown monkey (the foreground object), the tree could otherwise be mislabeled as foreground. Table 4.3 in the results specifically validates the impact of the term $U_i^l(f_i)$.

Minimization procedure for video labeling We minimize Eqn. 4.10 with binary graph cuts [18], and use the resulting label assignment as the foreground object segmentation of the video for hypothesis h. See Figure 4.15.

For efficiency, rather than segment the entire video at once, we sequentially label each frame in turn, using a space-time graph of three frames that connects its two adjacent frames. In addition, for better accuracy, rather than simply pass through the frames in sequential order, we proceed in a greedy ordering from the most confident frames that contain key-segments. That is, we start by labeling and fixing the key-segments' frames, and then solve others in their order of temporal proximity. This more effectively propagates the

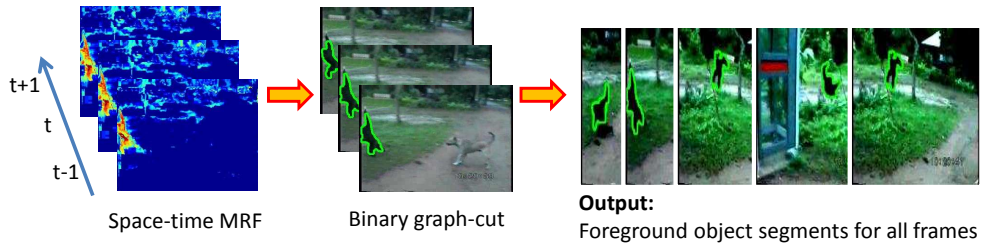


Figure 4.15: We solve for a pixel-wise segmentation using graph cuts on a space-time Markov Random Field (MRF) to segment the object across the entire video.

foreground/background labels of one frame to the next through optical flow connections.

4.2.1.4 Summary of the Approach

To recap, my method takes an unlabeled video, and produces foreground-background segmentations ranked by the object's expected centrality to the scene. The main steps are: (1) extract proposal regions from all frames, (2) score all regions by $S(r)$, (3) take top-ranked regions, and partition into inlier/outlier hypotheses. For each hypothesis, (4) extract foreground model and local shape features from all its key-segments, (5) match shape features across all frames to create shape-based foreground likelihood maps, (6) minimize Eqn. 4.10 using graph cuts with series of space-time graphs, (7) return binary pixel-wise segmentation.

Since my method ranks the foreground results by confidence, one can use it in a completely unsupervised manner to define the primary foreground objects (e.g., for summarization applications as I will discuss in the subsequent chapters). Alternatively, if a user is in the loop, s/he can select the desired foreground object.

4.2.2 Results

The main questions in my experiments are (1) to what extent are object-like regions better identified by using motion cues unique to video, (2) how well does my method rank each hypothesis, and (3) how accurate is my method's foreground object segmentation?

Datasets We test on two datasets: [148] and [56], eight videos in total. We use the SegTrack dataset [148], which contains six videos (*monkeydog, bird, girl, birdfall, parachute, penguin*) and pixel-level ground-truth (GT) for the primary foreground object. The videos span a wide degree of difficulty with challenges such as foreground/background color overlap, large shape deformation, and large camera motion. To my knowledge, it is the largest publicly available pixel-labeled video dataset. We do not provide in-depth quantitative results on the *penguin* video, since it lacks the GT to properly evaluate our algorithm; only a single penguin is labeled as the foreground object amidst a group of penguins.

In addition, we generate qualitative results on two videos from the dataset of [56]; note that it lacks pixel-level ground-truth needed for quantitative analysis. See Figure 4.16 for example frames of the datasets.

Implementation details We use [35] to generate regions. To describe color, we use Lab space histograms, with 23 bins per channel, and $K = 5$ component GMMs. To describe motion, we use optical flow histograms with 61 bins per x and y direction, using [97]; we dilate a region's bounding box by 30 pixels when computing the background histograms. We extract BPLRs[8] every 6

[8] http://vision.cs.utexas.edu/projects/bplr/

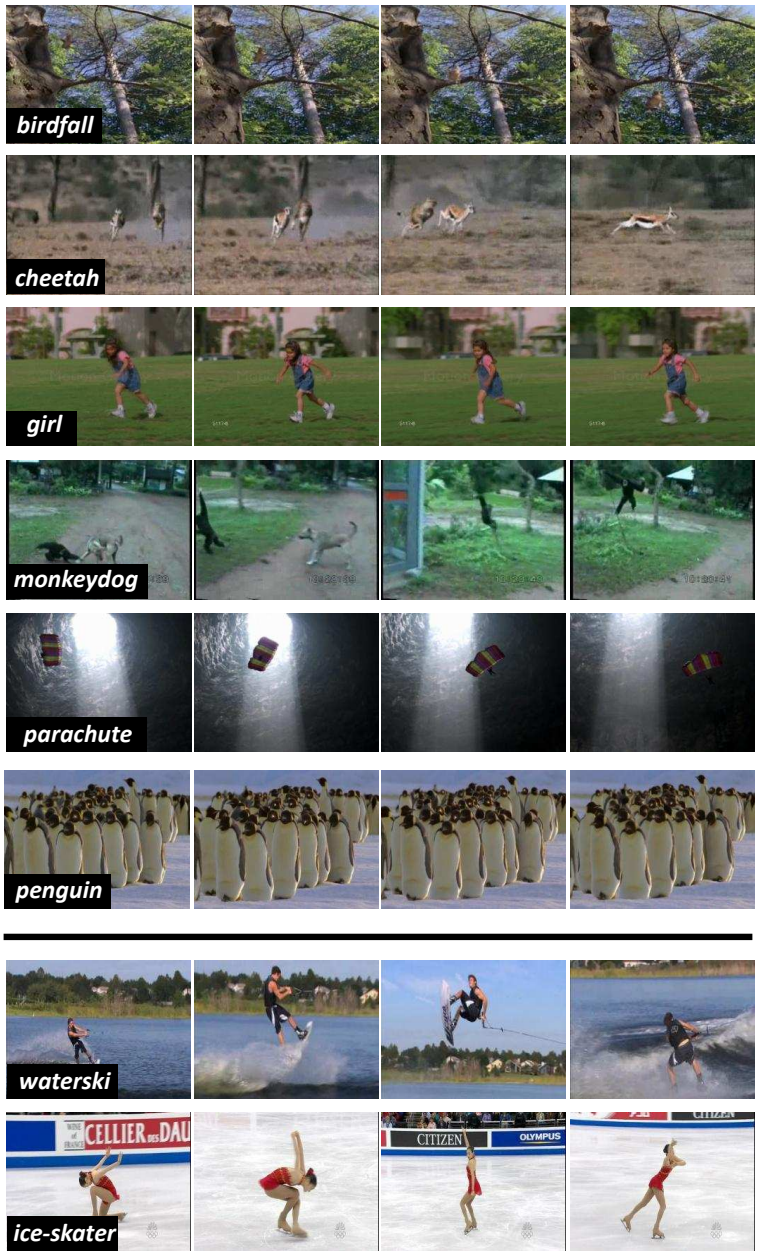


Figure 4.16: Example frames of the datasets used for video object segmentation. The first four rows show frames from SegTrack [148] and the bottom two rows show frames from [56].

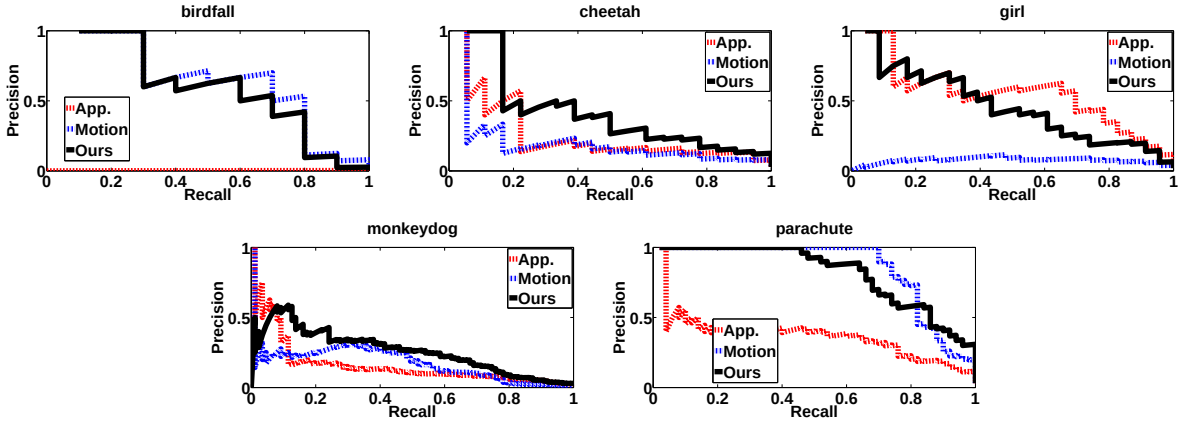


Figure 4.17: Precision-Recall curves for foreground object prediction. We analyze the different components of our video object-like scoring function. (**Ours**): full model; (**App.**): appearance-based region scoring; (**Motion**): motion-based region scoring. Higher curves are better.

pixels. We set $N = 10$.

For the graph-cuts minimization, we set $\alpha = 0.5$ and $\gamma = 4$ for the smoothness term. These parameters are fixed for scoring all videos. We smooth the partial shape match vote space with a Gaussian kernel to be robust to minor alignment errors and shape deformations.

Generating regions takes about 3 minutes / frame, computing GMMs takes about 2 minutes, and segmentation takes about 1 second / frame with a Matlab implementation.

4.2.2.1 Object Prediction Accuracy

I first evaluate my method's ability to predict object-like regions, and compare: (1) the static appearance component [35] that computes $A(r)$, (2) the dynamic motion component $M(r)$, and (3) my full model $S(r)$ that uses both.

Figure 4.17 shows precision-recall curves for the three variants on all regions in each video. A region r is considered to be a true positive (i.e., foreground object), if its *overlap score* $= \frac{|GT \cap r|}{|GT \cup r|}$ is greater than 0.5, following PASCAL convention [37].

The results clearly demonstrate that motion plays a significant role in identifying foreground object regions in video. This is particularly true for the *birdfall* and *parachute* sequences, in which the foreground object has large motion patterns compared to its surroundings. Static appearance is important as well, as can be seen for the *girl, cheetah*, and *monkey* videos. In the *girl* video, the foreground object exhibits articulated motions in which one part (e.g., arm) has substantially larger motion compared to another part (e.g., torso), which explains the low precision of the motion-only component. By accounting for both motion and appearance, our full model produces the best predictions overall.

4.2.2.2 Object Hypothesis Rank Accuracy

I next evaluate my method's hypothesis ranking. Figure 4.18 shows the mean ground-truth region overlap score for each of the ranked hypotheses. High rank hypotheses have high mean overlap-scores, while low rank hypotheses have low mean overlap-scores. This shows our automatically generated ranking is highly indicative of how well each hypothesis represents the primary object of interest. Among all videos, only the *monkeydog* sequence lacks a strong hypothesis among the top three ranks. This is due to an artifact of the data: each frame contains black margins, which artificially produce high motion scores (since their motion is constant, while the remaining objects are moving or appear to be moving due to camera motion); the top-three

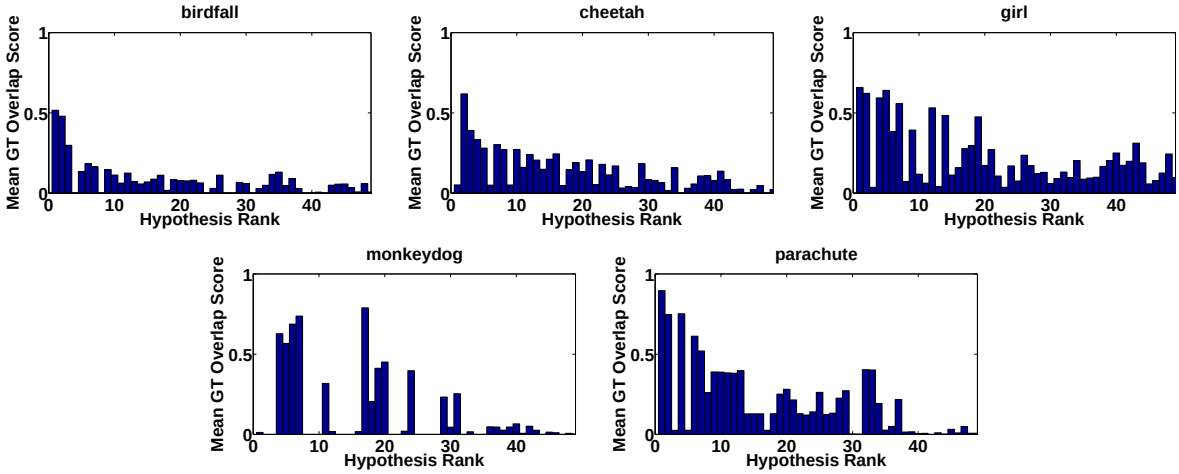


Figure 4.18: Our method's automatically ranked hypotheses and their mean ground-truth overlap scores. Higher bars at lower ranks reflect better foreground object prediction and ranking. Our ranking focuses attention to primary foreground objects.

hypotheses predict these to be the foreground object. However, the fourth ranked hypothesis correctly predicts the monkey to be the primary object.

What do the hypotheses and their key-segments look like? Figure 4.19 shows key-segments of the highest-ranked hypothesis that corresponds to the primary object. The number in parentheses indicate its rank. On six of the eight videos, our very top-ranked hypothesis corresponds to the primary foreground object. If desired, one could easily re-rank the hypotheses to enforce diversity by penalizing pixel overlap with higher ranked key-segments.

It is evident that the key-segments are representative exemplars of the foreground object. This allows my method to learn reliable color and shape models for segmenting out the object in all frames, including those that did not produce any key-segments, as I show next.

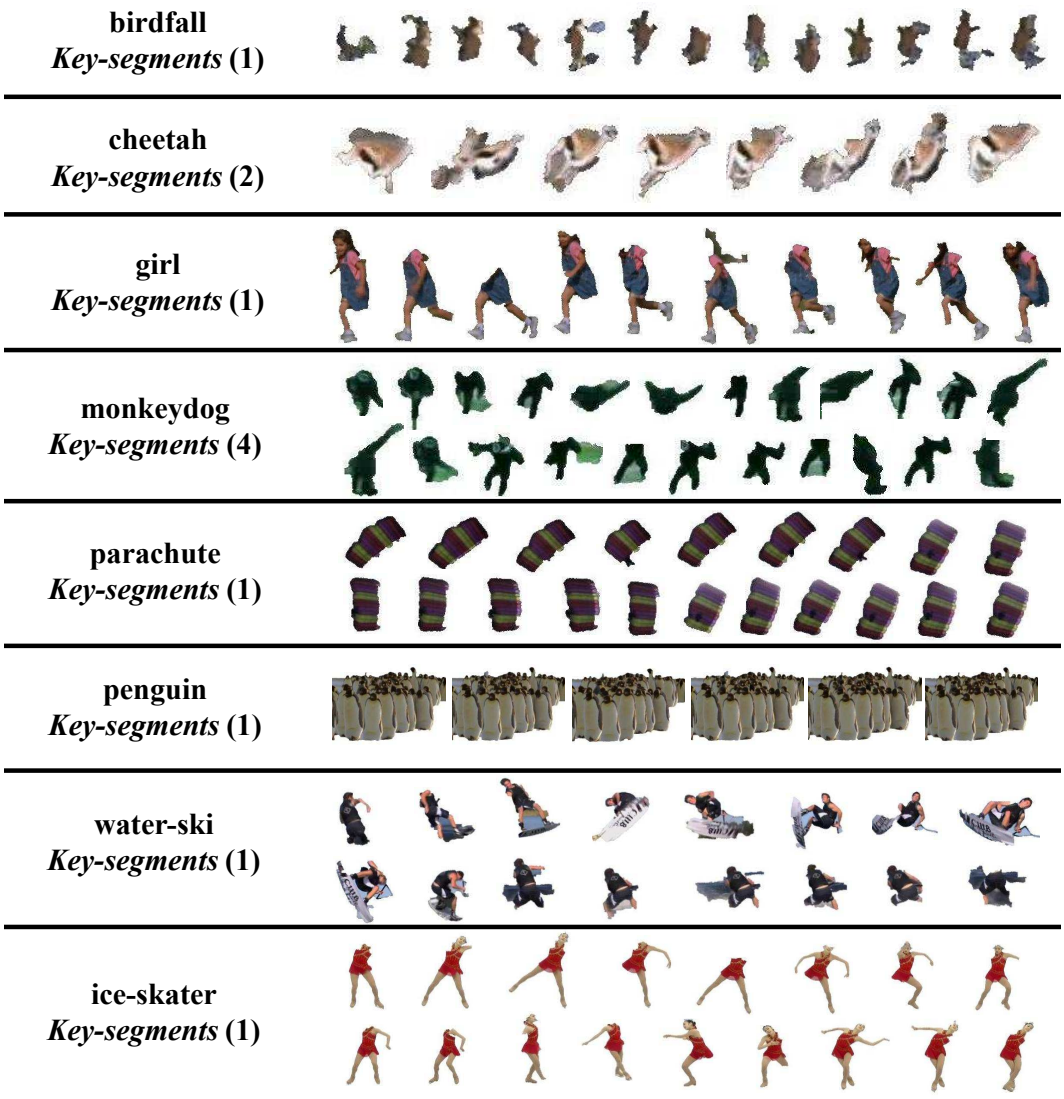


Figure 4.19: The discovered key-segments. The numbers indicate the rank of each hypothesis. The hypothesis corresponding to the primary object has high rank, and its key-segments have high overlap with true object boundaries. The first six rows show results on SegTrack [148] videos. The last two rows show results on videos from [56]. Best viewed on pdf.

	Ours	[148]	[26]	Top $A(r)$ region	Bg Sub
birdfall	288	**252**	454	26156	7435
cheetah	**905**	1142	1217	27728	28763
girl	1785	**1304**	1755	10236	45019
monkeydog	**521**	563	683	38083	31099
parachute	**201**	235	502	75168	27242
penguin	136285(*)	**1705**	6627	147686	61089
Manual seg?	No	Yes	Yes	No	No

Table 4.2: Segmentation error as measured by the average number of incorrect pixels per frame. Lower values are better. We compare our method (**Ours**) with two state-of-the-art methods ([148] and [26]), which require the first frame to be annotated. $*$*See text about penguin ground-truth.*

4.2.2.3 Object Segmentation Accuracy

In this section, I evaluate my method's final segmentation results. We compare against two state-of-the-art methods: (1) the motion coherence segmentation method of [148], and (2) the level set-based tracker of [26]. These methods require human labeling of the object boundary in the first frame. In contrast, my method requires no hand drawn supervision to guide the segmentation. (One may choose among my method's ranked segmentation proposals, but this does not change segmentation quality.)

Table 4.2 shows the results. To quantify segmentation accuracy, we use the average per-frame pixel error rate [148], $\epsilon(S) = \frac{|\mathbf{XOR}(S,GT)|}{F}$, where S is each method's segmentation, GT is the ground-truth segmentation, and F is the total number of frames. I evaluate my method with the segmentation of the hypothesis that corresponds to the object with ground-truth annotation.

My method produces the best results on three of the five videos (*cheetah, monkeydog, parachute*), and produces the second best result on the *birdfall* video. Our higher error on the *girl* video is caused by an over-segmentation

of the key-segments. This is primarily due to some inaccurate initial region proposals from [35], which is reasonable since the object exhibits large appearance variation. For the *penguin* video, our top-ranked hypothesis corresponds to the group of penguins, whereas the ground-truth annotates only a single penguin. Since the group of penguins are so close and similar, it is not clear whether one or all penguins makes a better foreground estimate.

The last two columns in Table 4.2 show error rates when taking the region with the highest appearance-based score $A(r)$ per frame and when performing standard background subtraction [139], respectively. Clearly, $A(r)$ alone is insufficient to predict the primary object in the video. Background subtraction completely falls apart, since it cannot handle large camera motions. By taking into account both motion and persistence to discover the key-segments, we obtain significantly better foreground segmentations.

Figure 4.20 shows qualitative segmentation examples. My method produces high quality segmentations of the primary foreground object. There are some failure cases as well, such as when the object is mislabeled due to low contrast with its surrounding regions (see last column of *bird* video), and when parts of the object are missed (see the second and third columns of *girl* video).

The last two rows show comparisons to the unsupervised method of [56]. My method produces a figure-ground segmentation at the object-level by automatically finding its key-segments. In contrast, [56] relies only on bottom-up pixel-level motion and appearance cues, which sometimes results in an over-segmentation of an object.

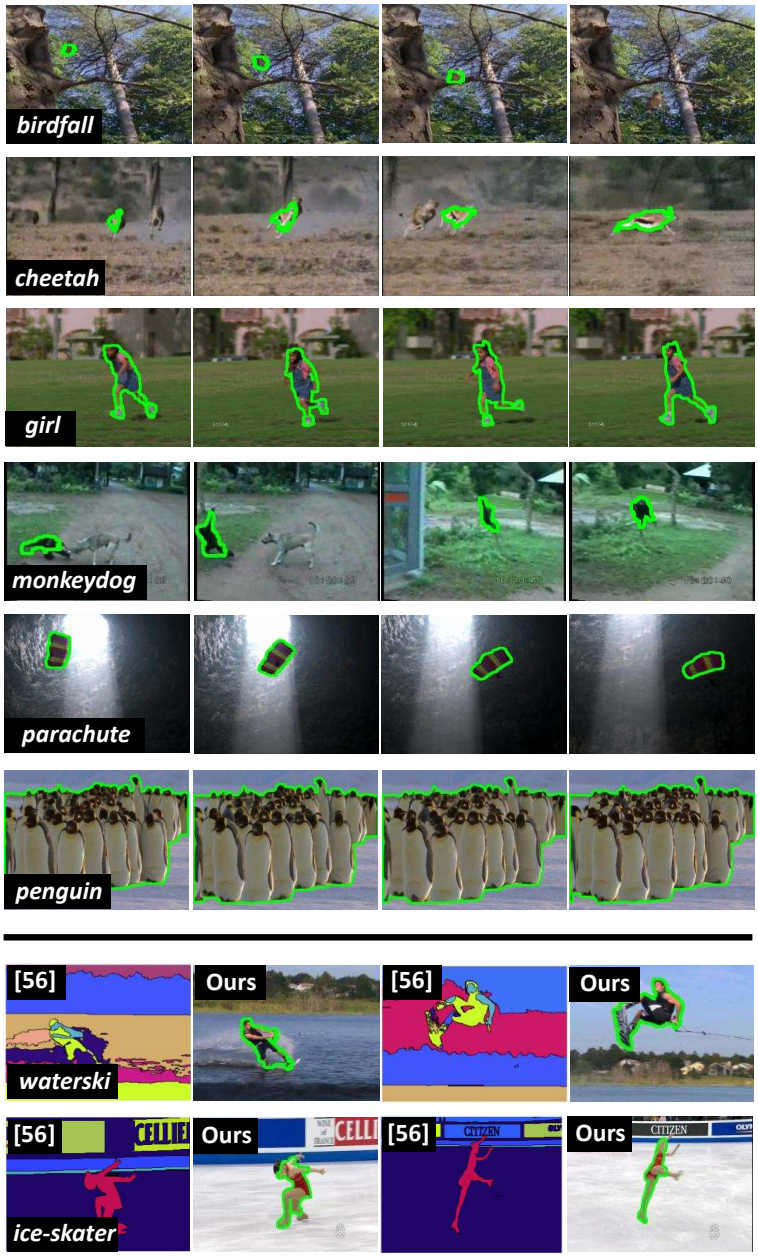


Figure 4.20: Segmentation results. The first six rows show results on SegTrack [148] videos. The last two rows compare our results to [56]. Best viewed on pdf.

	Ours	Ours w/o partial shape match
birdfall	**288**	414
cheetah	**905**	1024
girl	1785	**1534**
monkeydog	**521**	1261
parachute	201	**188**

Table 4.3: Segmentation error. Lower values are better. We compare our full method (**Ours**) with a baseline that only models color information (**Ours w/o partial shape match**). Our partial shape matching improves segmentation quality.

4.2.2.4 Impact of Partial Shape Matching

Finally, I study the impact of our partial shape matching location prior. We compare against a baseline that only models color, but otherwise follows the same pipeline as our full method. For this baseline, we set $\gamma = 50$ as in [125] to adjust the scales of the cost values between the methods. Table 4.3 shows the results. The partial shape matching improves segmentation accuracy in most videos. As discussed earlier, some of the key-segments of the *girl* video are over-segmented, which means that the projected shape can miss the articulated body parts (e.g., arms); increasing the color term helps in this case. Overall, we find a substantial advantage from the partial shape match.

4.3 Discussion

In this chapter, I presented a novel framework to perform object segmentation in images and videos using discovered top-down cues, building on the context-aware discovery idea from Chapter 3. I showed how to discover and group candidate image segments that belong to the same category, rank the grouped segments by predicted informativeness, and encode the discov-

ered top-down cues in energy functions that are amenable to existing graph-cuts algorithms to perform fast and accurate segmentation. I evaluated the framework for two applications—multi-object segmentation in natural images and foreground object segmentation in videos—and obtained similar or higher quality segmentations than previous bottom-up approaches and even state-of-the-art supervised methods.

We can also apply the same idea to discover meaningful features in unlabeled image collections. My work on foreground feature discovery shows that the mutual reinforcement of object-level and feature-level similarity improves unsupervised image clustering and foreground detection in unlabeled images [86]. Similarly, my work on shape discovery shows how to discover objects characterized by shape [87], where local appearance matches serve to anchor the surrounding edge fragments, yielding a more reliable affinity function for images that accounts for both shape and appearance. This allows discovery of the foreground object contours in each image, and summarization of the prototypical shapes for each category. For space, I focused on collective segmentation in this thesis; I refer the interested reader to the published papers for the foreground feature and shape discovery [86, 87].

Overall my results in this chapter illustrate the proposed method's advantage of discovering shared structure in the unlabeled set of images or video frames when computing segmentations. I demonstrated the value of introducing knowledge about previously learned categories directly using the formulations developed in Chapter 3. The results indicate that when some recurring objects are present in the image collection or video, exploiting their repetition leads to high quality segmentations that better capture full objects, which are not possible to obtain with bottom-up methods. I also introduced a novel par-

tial shape match location prior that primes the foreground object's location and scale in each image frame.

What are the assumptions of my approach? First of all, we assume that the image collection contains one or more repeating objects. As mentioned in Section 3.4, this may not always hold. If the data lacks repeating objects, the resulting clusters will be very noisy and will lead to inaccurate segmentations. To be robust to these errors, we could measure the reliability of each cluster. If a cluster is considered to be noisy, it should not be used to build top-down models for segmentation. My key-segments framework is more widely applicable, since repeating object instances in videos is almost guaranteed.

Admittedly, the initial discovered clusters for building the top-down object models may not fully represent all objects in the image collection, mainly due to the initial bottom-up segmentation step. Small or heterogenous objects tend to be missed or oversegmented by most bottom-up segmentation methods. Category independent segmentation methods [3, 35] work better, since they learn to segment generic objects with labeled training images. One reason for the success of my key-segments approach is due to the method of [35] combined with the proposed object-like motion cue, which generates more reliable candidate object segmentations compared to bottom-up segmentation methods. However, there can still be noisy object hypotheses (containing object fragments, or good segments that belong to multiple objects) that would result in inaccurate segmentations.

To overcome some of these limitations, we could have a human-in-the-loop. The system could present its discoveries to the human and request annotations in various forms, such as pruning any of the object hypotheses or merging two or more of them if they belong to the same object. Furthermore,

for video object segmentation, we could generate the initial bottom-up regions using motion cues. In the current framework, we generate regions using static image cues, and then select those that are object-like using dynamic motion cues. While in practice this produces many object-like regions, we could do even better by exploiting motion for the region generation step, which would be especially helpful when an object and its surrounding background regions share similar appearance patterns. I would like to explore these directions in future work.

Chapter 5

Discovering Important People and Objects for Egocentric Video Summarization

The previous chapters showed how to continuously discover novel categories amidst familiar objects through a self-paced curriculum, and how to build models from the discoveries to perform unsupervised segmentation. Building on many of the techniques that I have already introduced, in this chapter I explore the value of discovery and segmentation for automatic *summarization* of visual data. In particular, I show how to produce important object-driven summaries for first-person videos captured from a wearable camera.

The goal of video summarization is to produce a compact visual summary that encapsulates the key components of a video. Its main value is in turning hours of video into a short summary that can be interpreted by a human viewer in a matter of seconds. Automatic video summarization methods would be useful for a number of practical applications, such as analyzing surveillance data, video browsing, action recognition, or creating a visual diary.

As discussed in Chapter 2, existing methods extract keyframes [52, 164, 170], create montages of still images [5, 25], or generate compact dynamic summaries [118, 122]. Despite promising results, they assume a static background or rely on low-level appearance and motion cues to select what will go into the final summary. However, in many interesting settings, such as egocentric

videos, YouTube style videos, or feature films, the background is moving and changing. More critically, a system that lacks high-level information on *which objects matter* may produce a summary that consists of irrelevant frames or regions. In other words, existing methods do not perform *object-driven* summarization and are indifferent to the impact that each object has on generating the "story" of the video.

An interesting and practical domain for video summarization is wearable (i.e., egocentric) camera data. An egocentric video offers a first-person view of the world that cannot be captured from environmental cameras. For example, we can often see the camera wearer's hands, or find the object of interest centered in the frame. Essentially, a wearable camera focuses on the user's activities, social interactions, and interests. I aim to exploit these properties for egocentric video summarization.

Good summaries for egocentric data would have wide potential uses. Not only would recreational users (including "life-loggers") find it useful as a video diary, but there are also higher-impact applications in law enforcement, elder and child care, and mental health. For example, the summaries could facilitate police officers in reviewing important evidence, suspects, and witnesses, or aid patients with memory problems to remember specific events, objects, and people [61]. Furthermore, the egocentric view translates naturally to robotics applications—suggesting, for example, that a robot could summarize what it encounters while navigating unexplored territory, for later human viewing.

Motivated by these problems, I propose an approach that learns category-independent *importance* cues designed explicitly to target the *key objects and people* in the video. The main idea is to leverage novel egocentric and high-

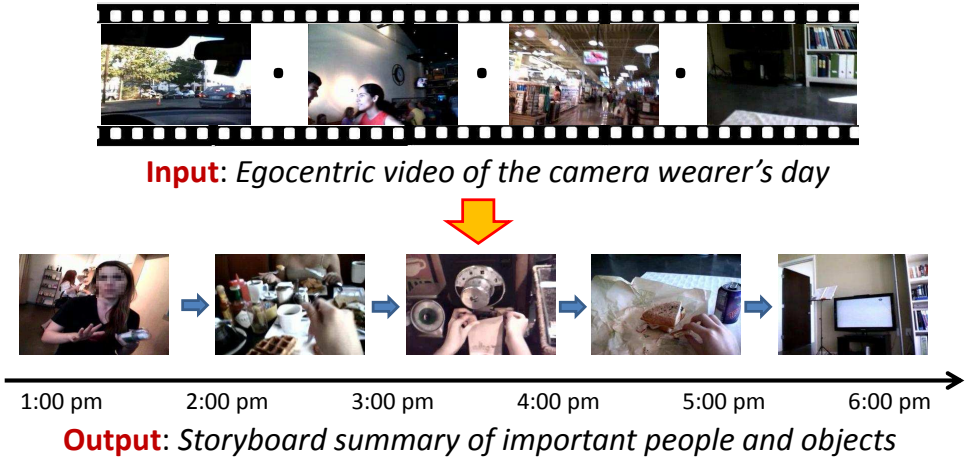


Figure 5.1: My approach takes as input an unannotated egocentric video, and produces a compact storyboard visual summary that focuses on the key people and objects in the video.

level saliency features to train a model that can predict important regions in the video, and then to produce a concise visual summary that is driven by those regions (see Figure 5.1). By learning to predict important regions, the system can focus the visual summary on the main people and objects, and ignore irrelevant or redundant information.

I emphasize that we are not aiming to predict importance for any specific category (e.g., cars). Instead, we learn a general model that can predict the importance of any *object instance*, irrespective of its category. This category-independence avoids the need to train importance predictors specific to a given camera wearer, and allows the system to recognize as important something it has never seen before. In addition, it means that objects from the same category can be predicted to be (un)important depending on their role in the story of the video. For example, if the camera wearer has lunch with his friend Jill, she would be considered important, whereas people in the

same restaurant sitting around them could be unimportant. Then, if they later attend a party but chat with different friends, Jill may no longer be considered important in that context.

The main contribution of this part of my thesis is a novel egocentric video summarization approach that is driven by predicted important people and objects. I apply my method to challenging real-world videos captured by users in uncontrolled environments, and process a total of 17 hours of video—orders of magnitude more data than previous work in egocentric analysis. Evaluating the predicted importance estimates and summaries, I find my approach outperforms state-of-the-art saliency measures for this task, and produces significantly more informative summaries than traditional methods unable to focus on the important people or objects.

Related work on egocentric visual analysis Vision researchers have only recently begun to explore egocentric visual analysis. Early work with wearable cameras segments visual and audio data into events [28]. Recent methods explore activity recognition [39, 137, 172], handled object recognition [123], novelty detection [2], or activity discovery for non-visual sensory data [66]. Unsupervised algorithms are developed to discover scenes [67] or actions [76] based on low-level visual features extracted from egocentric data. In contrast, we aim to build a visual summary, and model high-level importance of the objects present. To my knowledge, we are the first to perform visual summarization for egocentric data.

5.1 Approach

The goal is to create a storyboard summary of a person's day that is driven by the important people and objects. The video is captured using a wearable camera that continuously records what the user sees. I define *importance* in the scope of egocentric video: important things are those with which the camera wearer has significant interaction.

There are four main steps to my approach: (1) using novel egocentric saliency cues to train a category-independent regression model that predicts how likely an image region belongs to an important person or object; (2) partitioning the video into temporal events. For each event, (3) scoring each region's importance using the regressor; and (4) selecting representative keyframes for the storyboard based on the predicted important people and objects.

I first describe how we collect the video data and ground-truth annotations needed to train our model. I then describe each of the main steps in turn.

5.1.1 Egocentric Video Data Collection

We use the Looxcie wearable camera[1], which captures video at 15 fps at 320 x 480 resolution. It is worn around the ear and looks out at the world at roughly eye-level. We collected 10 videos, each of three to five hours in length (the maximum Looxcie battery life), for a total of 37 hours of video.

Four subjects wore the camera for us: one undergraduate student, two grad students, and one office worker, ranging in age from early to late 20s and both genders. The different backgrounds of the subjects ensure diversity in

[1] http://looxcie.com/

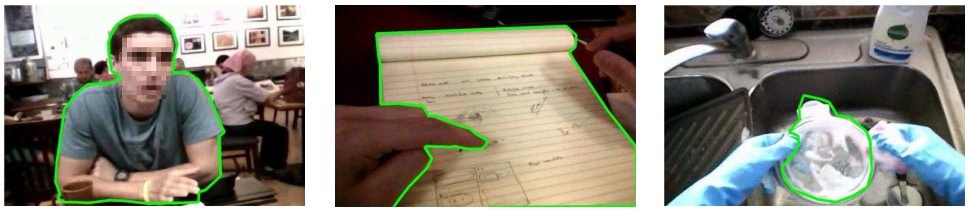


Figure 5.2: Example annotations obtained using Mechanical Turk.

the data—not everyone's day is the same—and is critical for validating the category-independence of our approach. We asked the subjects to record their natural daily activities, and explicitly instructed them not to stage anything for this purpose. The videos capture a variety of activities such as eating, shopping, attending a lecture, driving, cooking, and working on the computer.

5.1.2 Annotating Important Regions in Training Video

To train the importance predictor, we first need ground-truth training examples. In general, determining whether an object is important or not can be highly subjective. Fortunately, an egocentric video provides many constraints that are suggestive of an object's importance.

In order to learn meaningful egocentric properties without overfitting to any particular category, we crowd-source large amounts of annotations using Amazon's Mechanical Turk (MTurk). For egocentric videos, an object's degree of importance will highly depend on what the camera wearer is doing before, while, and after the object or person appears. In other words, the object must be seen in the context of the camera wearer's activity to properly gauge its importance.

We carefully design two annotation tasks to capture this aspect. In the

first task, we ask workers to watch a three minute accelerated video (equivalent to 10 minutes of original video) and to describe in text what they perceive to be essential people or objects necessary to create a summary of the video. In the second task, we display uniformly sampled frames from the video and their corresponding text descriptions *obtained from the first task*, and ask workers to draw polygons around any described person or object. If none of the described objects are present in a frame, the annotator is given the option to skip it. See Figure 5.2 for example annotations.

We found this two-step process more effective than a single task in which the same worker both watches the video and then annotates the regions s/he deems important, likely due to the time required to complete both tasks. Critically, the two-step process also helps us avoid bias: a single annotator asked to complete both tasks at once may be biased to pick easier things to annotate rather than those s/he finds to be most important. Our setup makes it easy for the first worker to freely describe the objects without bias, since s/he only has to enter text. We found the resulting annotations quite consistent, and only manually pruned those where the region outlined did not agree with the first worker's description. For a 3-5 hour video, we obtain roughly 35 text descriptions and 700 object segmentations.

5.1.3 Learning Region Importance in Egocentric Video

I now discuss the procedure to train a general purpose category-independent model that will predict important regions in any egocentric video, independent of the camera wearer. Given a video, we first generate candidate regions for each frame using the segmentation method of [24]. We purposefully represent objects at the frame-level, since our uncontrolled setting usually prohibits

reliable space-time object segmentation due to frequent and rapid head movements by the camera wearer.[2] We generate roughly 800 regions per frame.

For each region, we compute a set of candidate features that could be useful to describe its importance. Since the video is captured by an active participant, we specifically want to exploit egocentric properties such as whether the object/person is interacting with the camera wearer, whether it is the focus of the wearer's gaze, and whether it frequently appears. In addition, we aim to capture high-level saliency cues—such as an object's motion and appearance, or the likelihood of being a human face—and generic region properties shared across categories, such as size or location. I describe each feature in detail below.

Egocentric features Figure 5.3 illustrates the three proposed egocentric features. To model **interaction**, we compute the Euclidean distance of the region's centroid to the closest detected hand in the frame. Given a frame in the test video, we first classify each pixel as (non-)skin using color likelihoods and a Naive Bayes classifier [68] trained with ground-truth hand annotations on disjoint data. We then classify any superpixel (computed using [40]) as hand if more than 25% of its pixels are skin. While simple, we find this hand detector is sufficient for our application. More sophisticated methods (e.g., [79]) would certainly be possible as well.

To model **gaze**, we compute the Euclidean distance of the region's centroid to the frame center. Since the camera moves with the wearer's head, this is a coarse estimate of how likely the region is being focused upon.

[2]Indeed, we found KLT tracks to last only a few frames on our data.

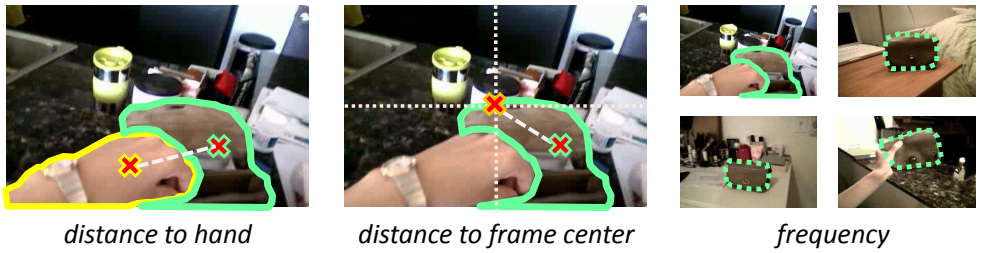


Figure 5.3: Illustration of our egocentric features.

To model **frequency**, we record the number of times an object instance is detected within a short temporal segment of the video. We create two frequency features: one based on matching regions, the other based on matching points. For the first, we compute the color dissimilarity between a region r and each region r_n in its surrounding frames, and accumulate the total number of positive matches:

$$c_{region}(r) = \sum_{f \in \mathcal{W}} [(\min_n \chi^2(r, r_n^f)) \leq \theta_r], \tag{5.1}$$

where f indexes the set of frames $\mathcal{W}$ surrounding region r's frame, $\chi^2(r, r_n)$ is the χ^2-distance between color histograms of r and r_n, θ_r is the distance threshold to determine a positive match, and $[\cdot]$ denotes the indicator function. The value of c_{region} will be high/low when r produces many/few matches (i.e., is frequent/infrequent).

The second frequency feature is computed by matching DoG+SIFT interest points. For a detected point p in region r, we match it to all detected points in each frame $f \in \mathcal{W}$, and count as positive those that pass the ratio test [101]. We repeat this process for each point in region r, and record their

average number of positive matches:

$$c_{point}(r) = \frac{1}{P}\sum_{i=1}^{P}\sum_{f\in\mathcal{W}}\left[\frac{d(p_i, p^f_{1^*})}{d(p_i, p^f_{2^*})} \leq \theta_p\right], \tag{5.2}$$

where i indexes all detected points in region r, $d(p_i, p^f_{1^*})$ and $d(p_i, p^f_{2^*})$ measure the Euclidean distance between p_i and its best matching point $p^f_{1^*}$ and second best matching point $p^f_{2^*}$ in frame f, respectively, and θ_p is Lowe's ratio test threshold for non-ambiguous matches [101]. The value of c_{point} will be high/low when the SIFT points in r produce many/few matches. For both frequencies, we set $\mathcal{W}$ to span a 10 minute temporal window.

Object features In addition to the egocentric-specific features, we include three high-level (i.e., object-based) saliency cues. To model **object-like appearance**, we use the learned region ranking function of [24], which is a similar variant of the object-like appearance features we used in Sections 3.2 and 4.2. It reflects Gestalt cues indicative of *any* object, such as the sum of affinities along the region's boundary, its perimeter, and texture difference with nearby pixels. (Note that the authors trained their measure on PASCAL data, which is disjoint from ours.) We stress that this feature estimates how "object-like" a region is, and *not its importance*. It is useful for identifying full object segments, as opposed to fragments.

To model **object-like motion**, we use the feature defined in Section 4.2. It looks at the difference in motion patterns of a region relative to its closest surrounding regions. Similar to the appearance feature above, it is useful for selecting object-like regions that "stand-out" from their surroundings.

To model the **likelihood of a person's face**, we compute the maximum overlap score $\frac{|q \cap r|}{|q \cup r|}$ between the region r and any detected frontal face q in the frame, using [155].

Region features Finally, we compute the region's **size**, **centroid**, **bounding box centroid**, **bounding box width**, and **bounding box height**. They reflect category-independent importance cues and are blind to the region's appearance or motion. We expect that important people and objects will occur at non-random scales and locations in the frame, due to social and environmental factors that constrain their relative positioning to the camera wearer (e.g., sitting across a table from someone when having lunch, or handling cooking utensils at arm's length). Our region features capture these statistics.

Altogether, these cues form a 14-dimensional feature space to describe each candidate region (4 egocentric, 3 object, and 7 region feature dimensions).

Regressor to predict region importance Using the features defined above, we next train a model that can predict a region's importance. The model should be able to learn and predict a region's *degree* of importance instead of whether it is simply "important" or "not important", so that we can meaningfully adjust the compactness of the final summary (as we demonstrate in Section 5.2). Thus, we opt to train a regressor rather than a classifier.

While the features defined above can be individually meaningful, we also expect significant interactions between the features. For example, a region that is near the camera wearer's hand might be important only if it is also object-like in appearance. Therefore, we train a linear regression model with

pair-wise interaction terms to predict a region r's *importance score*:

$$I(r) = \beta_0 + \sum_{i=1}^{N} \beta_i x_i(r) + \sum_{i=1}^{N} \sum_{j=i+1}^{N} \beta_{i,j} x_i(r) x_j(r), \tag{5.3}$$

where the β's are the learned parameters, $x_i(r)$ is the ith feature value, and $N = 14$ is the total number of features.

For training, we define a region r's target importance score by its maximum overlap $\frac{|GT \cap r|}{|GT \cup r|}$ with any ground-truth region GT in a training video obtained from Section 5.1.2. We standardize the features to zero-mean and unit-variance, and solve for the β's using least-squares. For testing, our model takes as input a region r's features (the x_i's) and predicts its importance score $I(r)$.

5.1.4 Segmenting the Video into Temporal Events

Given a new video, we first partition the video temporally into events, and then isolate the important people and objects in each event. Events allow the final summary to include multiple instances of an object/person that is central in multiple contexts in the video (e.g., the dog at home in the morning, and then the dog at the park at night).

To detect egocentric events, we cluster scenes in such a way that frames with similar global appearance can be grouped together even when there are a few unrelated frames ("gaps") between them.[3] Let $\mathcal{V}$ denote the set of all video frames. We compute a pairwise distance matrix $D_{\mathcal{V}}$ between all frames

[3]Traditional shot detection is impractical for wearable camera data; it oversegments events due to frequent head movements.

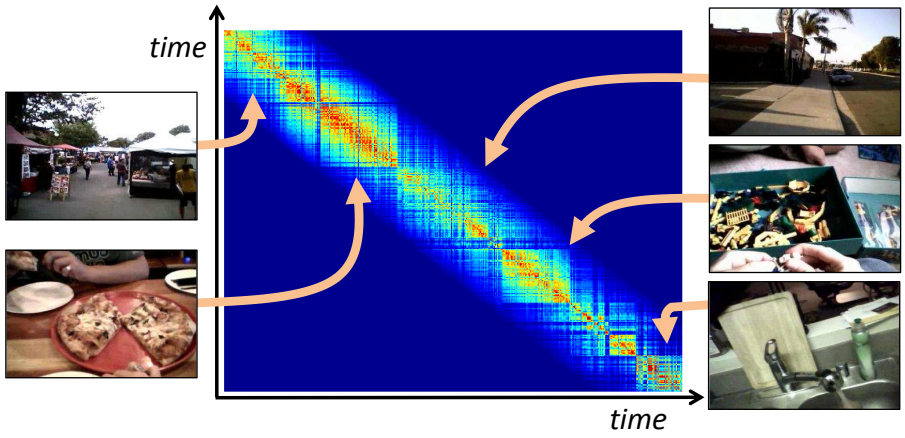


Figure 5.4: Distance matrix that measures global color dissimilarity between all frames. (Blue/red reflects high/low distance.) The images show representative frames of each discovered event.

$f_m, f_n \in \mathcal{V}$, using the distance:

$$D(f_m, f_n) = 1 - w^t_{m,n} \exp(-\frac{1}{\Omega}\chi^2(f_m, f_n)), \tag{5.4}$$

where $w^t_{m,n} = \frac{1}{t}\max(0, t - |m - n|)$, t is the size of the temporal window surrounding frame f_m, $\chi^2(f_m, f_n)$ is the χ^2-distance between color histograms of f_m and f_n, and Ω denotes the mean of the χ^2-distances among all frames. Thus, frames similar in color receive a low distance, subject to a weight that discourages frames too distant in time from being grouped.

We next discover events using a variant of the clustering technique from Section 3.2 for finding the easiest object category: we generate candidate groups and retain the most prominent ones. Specifically, we perform complete-link agglomerative clustering with $D_{\mathcal{V}}$, grouping frames until the smallest maximum inter-frame distance is larger than two standard deviations beyond Ω. The first and last frames in a cluster determine the start and end frames of an event, respectively. Since events can overlap, we retain (almost) disjoint events

by eliminating those with greater than θ_{event} overlap with events with higher silhouette-coefficients [142] in a greedy manner. Higher/lower θ_{event} leads to more/fewer events in the final summary. See Figure 5.4 for the distance matrix computed from one of our subject's day, and the representative frames for each discovered event.

One could further augment the distance in Eqn. 5.4 with GPS locations, when available (though GPS alone would be insufficient to discriminate multiple indoor positions in the same building).

5.1.5 Discovering an Event's Key People and Objects

For each event, we aim to select the important people and objects that will go into the final summary, while avoiding redundancy. Given an event, we first score each bottom-up segment in each frame using our regressor. We take the highest-scored regions (where "high" depends on a user-specified summary compactness criterion, see below) and group instances of the same person or object together. Since we do not know a priori how many important things an event contains, we generate a candidate pool of clusters from the set $\mathcal{C}$ of high-scoring regions, and then remove any redundant clusters, as follows.

To extract the candidate groups, we directly apply the key-segment discovery approach from the previous chapter. We first compute an affinity matrix $K_{\mathcal{C}}$ over all pairs of regions $r_m, r_n \in \mathcal{C}$, where affinity is determined by color similarity: $K_{\mathcal{C}}(r_m, r_n) = \exp(-\frac{1}{\Gamma}\chi^2(r_m, r_n))$, where Γ denotes the mean χ^2-distance among all pairs in $\mathcal{C}$. We next partition $K_{\mathcal{C}}$ into multiple (possibly overlapping) inlier/outlier clusters using a factorization approach [114]. The method finds tight sub-graphs within the input affinity graph while resisting the influence of outliers. Each resulting sub-graph consists of a candidate

important object's instances. To reduce redundancy, we sort the sub-graph clusters by the average $I(r)$ of their member regions, and remove those with high affinity to a higher-ranked cluster. Finally, for each remaining cluster, we select the region with the highest importance score as its representative. Note that this grouping step reinforces the egocentric frequency cue described in Section 5.1.3.

5.1.6 Generating a Storyboard Summary

Finally, we create a storyboard visual summary of the video. We display the event boundaries and frames of the selected important people and objects (see Figure 5.9). Each event can display a varying number of frames, depending on how many unique important things our method discovers. We automatically adjust the *compactness* of the summary with selection criteria on the region importance scores and event overlaps, as we illustrate in our results.

In addition to being a compact video diary of one's day, our storyboard summary can be considered as a *visual index* to help a user peruse specific parts of the video. This would be useful when one wants to relive a specific moment or search for less important people or objects that occurred with those found by our method.

5.2 Results

In this section, I analyze (1) the performance of my method's important region prediction, (2) my egocentric features, and (3) the accuracy and compactness of my storyboard summaries.

Dataset We collected 10 videos from four subjects, each 3-5 hours long. Each person contributed one video, except one who contributed seven. The videos are challenging due to frequent camera viewpoint/illumination changes and motion blur. For evaluation, we use four data splits: for each split we train with data from three users and test on one video from the remaining user. Hence, the camera wearers in any given training set are disjoint from those in the test set, ensuring we do not learn user- or object-specific cues.

Implementation details We use Lab space color histograms, with 23 bins per channel, and optical flow histograms with 61 bins per direction. We set $t = 27000$, i.e., a 60 minute temporal window. We set $\theta_r = 10000$ and $\theta_p = 0.7$ after visually examining a few examples. We fix all parameters for all results. For efficiency, we process every 15th frame (i.e., 1 fps).

5.2.1 Important Region Prediction Accuracy

I first evaluate my method's ability to predict important regions, compared to three state-of-the-art high- and low-level saliency methods: (1) the object-like score of [24], (2) the object-like score of [35], and (3) the bottom-up saliency detector of [156]. The first two are learned functions that predict a region's likelihood of overlapping a true object, whereas the low-level detector aims to find regions that "stand-out". Since the baselines are all general-purpose metrics (not tailored to egocentric data), they allow us to gauge the impact of the proposed egocentric cues for finding important objects in video.

We use the annotations obtained on MTurk as ground truth (GT) (see Section 5.1.2). Some frames contain more than one important region, and some contain none, simply depending on what the annotators deemed important. On average, each video contains 680 annotated frames and 280,000 test

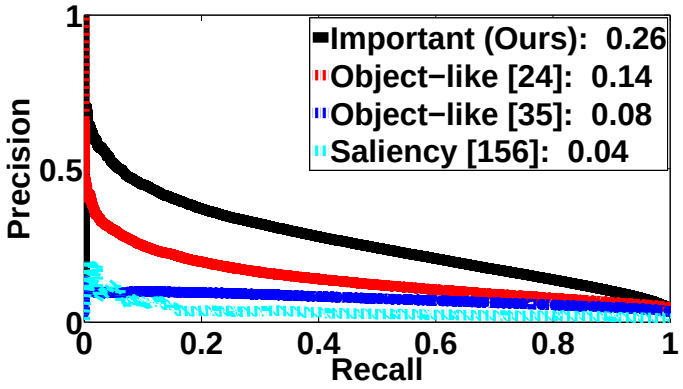


Figure 5.5: Precision-Recall for important object prediction across all splits. Numbers in the legends denote average precision. Compared to state-of-the-art high-level [24, 35] and low-level [156] saliency methods, our egocentric approach more accurately discovers the important regions.

regions. A region r is considered to be a true positive (i.e., important object), if its $overlapscore = \frac{|GT \cap r|}{|GT \cup r|}$ with any GT region is greater than 0.5, following PASCAL convention.

Figure 5.5 shows precision-recall curves on all test regions across all train/test splits. Our approach predicts important regions significantly better than all three existing methods. The two high-level methods can successfully find prominent object-like regions, and so they noticeably outperform the low-level saliency detector. However, by focusing on detecting *any* prominent object, unlike our approach they are unable to distinguish those that may be important to a camera wearer.

Figure 5.6 shows example important regions detected by each method. The top four rows show examples of correct predictions made by our method. The high-level saliency detection methods [24, 35] aim to detect any prominent object. Therefore, they are unable to predict objects that may be important

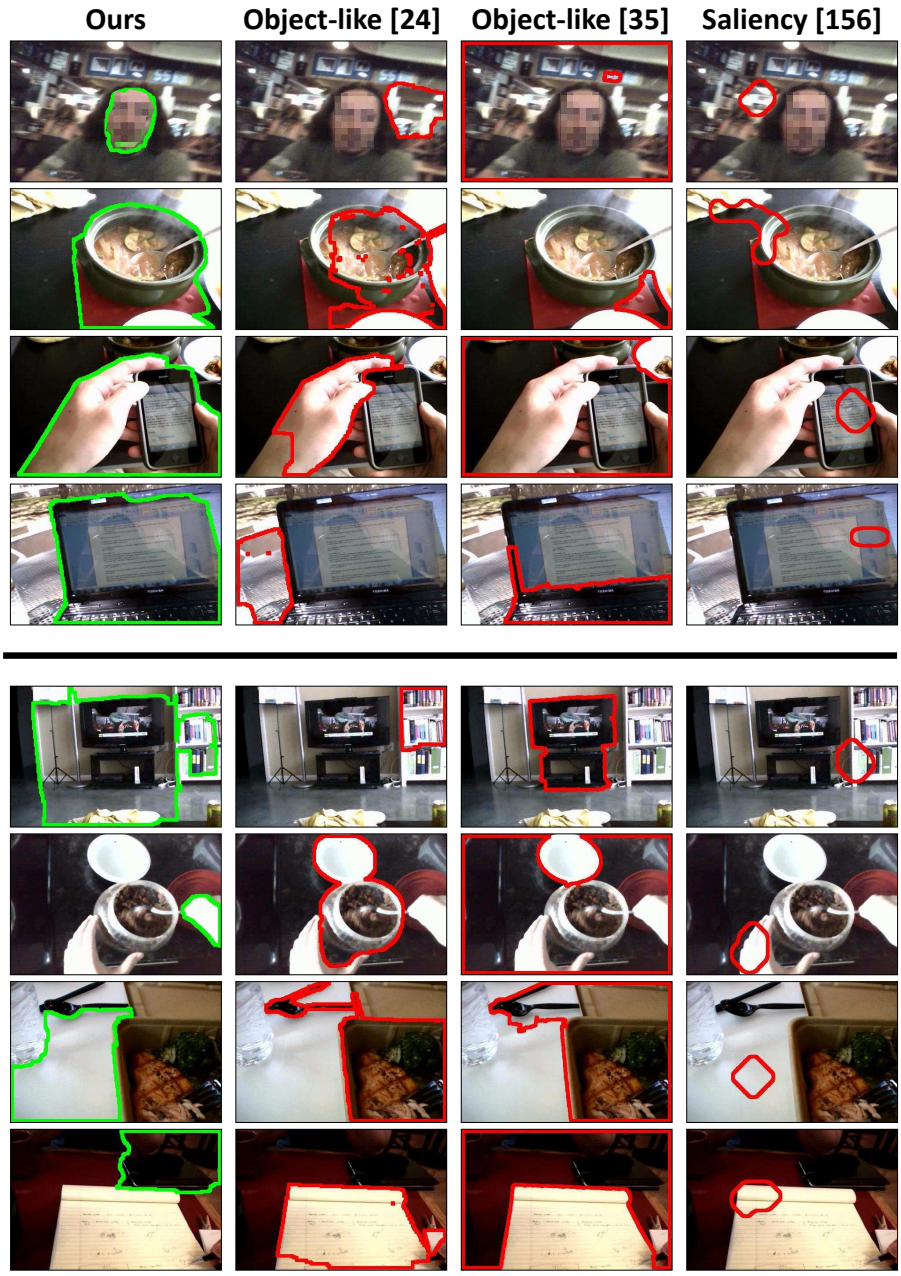


Figure 5.6: Example selected regions/frames. The first four rows show examples of correct predictions made by our approach, and the bottom four rows show failure cases in which the high-level saliency methods [24, 35] outperform our approach.

to a camera wearer. The low-level saliency detection method [156] fails to find object-like regions, and instead produces local estimates of saliency. The bottom four rows show examples of incorrect predictions made by our method. The high-level saliency detection methods [24, 35] produce better predictions for these examples. In the first example, our method produces an under-segmentation of the important object and includes regions surrounding the television. In the second example, our method incorrectly detects the users hand to be important, while in the third and fourth examples, it determines background regions to be important due to their high frequency. Overall, we find our egocentric approach more accurately discovers the important regions.

5.2.2 Which Cues Matter Most for Predicting Importance?

I next evaluate which features matter most for predicting important objects in egocentric videos. Figure 5.7 shows the top 28 out of 105 ($= 14 + \binom{14}{2}$) features that receive the highest learned weights (i.e., β magnitudes). Region size is the highest weighted cue, which is reasonable since an important person/object is likely to appear roughly at a fixed distance from the camera wearer. Among the egocentric features, gaze and frequency have the highest weights. Frontal face overlap is also highly weighted; intuitively, an important person would likely be facing and conversing with the camera wearer.

Some highly weighted pair-wise interaction terms are also quite interesting. The feature measuring a region's face overlap *and* y-position has more impact on importance than face overlap alone. This suggests that an important person usually appears at a fixed height relative to the camera wearer. Similarly, the feature for object-like appearance *and* y-position has high weight, suggesting that a camera wearer often adjusts his ego-frame of reference to

1. *size*	8. *height*	15. *obj app.*	22. *bbox x + reg freq.*
2. *size + height*	9. *pt freq.*	16. *x*	23. *x + reg freq.*
3. *y + face*	10. *size + reg freq.*	17. *size + x*	24. *obj app. + size*
4. *size + pt freq.*	11. *gaze*	18. *gaze + x*	25. *y + interaction*
5. *bbox y + face*	12. *face*	19. *obj app. + y*	26. *width + height*
6. *width*	13. *y*	20. *x + bbox x*	27. *gaze + bbox x*
7. *size + gaze*	14. *size + width*	21. *y + bbox x*	28. *bbox y + interaction*

Figure 5.7: Top 28 features with highest learned weights.

view an important object at a particular height.

Surprisingly, the pairing of the interaction (distance to hand) and frequency cues receives the lowest weight. A plausible explanation is that the *frequency* of a handled object highly depends on the camera wearer's activity. For example, when eating, the camera wearer's hand will be visible and the food will appear frequently. On the other hand, when grocery shopping, the important item s/he grabs from the shelf will (likely) be seen for only a short time. These conflicting signals would lead to this pair-wise term having low weight. Another paired term with low weight is an "object-like" region that is frequent; this is likely due to unimportant background objects (e.g., the lamp behind the camera wearer's companion). This suggests that higher-order terms could yield even more informative features.

5.2.3 Egocentric Video Summarization Accuracy

Next I evaluate my method's summarization results. We compare against two baselines: (1) uniform keyframe sampling, and (2) event-based adaptive keyframe sampling. The latter computes events using the same procedure as our method (Section 5.1.4), and then divides its keyframes evenly across events. These are natural baselines modeled after classic keyframe and

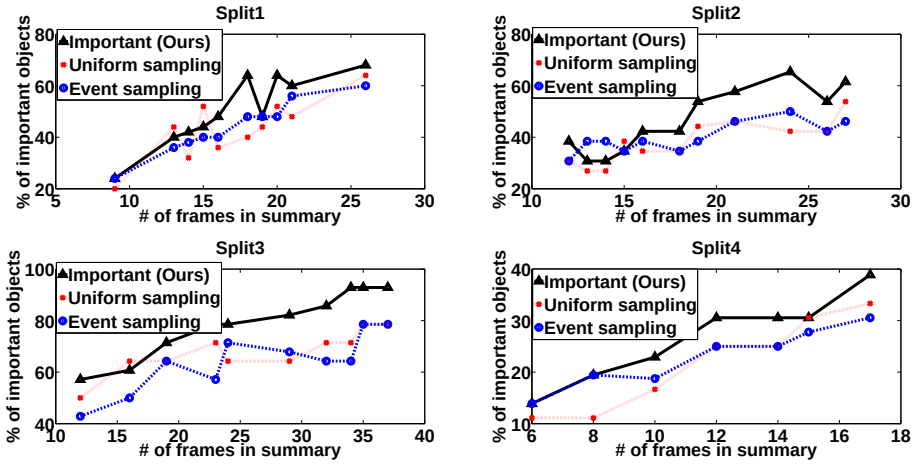


Figure 5.8: Comparison to alternative summarization strategies, in terms of important object recall rate as a function of summary compactness.

event detection methods [164, 170], and both select keyframes that are "spread-out" across the video.

Figure 5.8 shows the results. We plot the *percentage of important objects found* as a function of the *number of frames in the summary*, in order to analyze both the recall rate of the important objects as well as the compactness of the summaries. Each point on the curve shows the result for a different summary of the required length. To vary compactness, our method varies both its selection criterion on $I(r)$ over $\{0, 0.1, \ldots, 0.5\}$ and the number of events by setting $\theta_{event} = \{0.2, 0.5\}$, for 12 summaries in total.[4] We create summaries for the baselines with the same number of frames as those 12. If a frame contains multiple important objects, we score only the main one. Likewise, if a summary contains multiple instances of the same GT object, it gets credit only once. Note that this measure is very favorable to the baselines,

[4]Among these parameter combinations, some summaries may end up being the same length, in which case we average their recall rates.

since it does not consider object *prominence* in the frame. (We measure object prominence in the next section.) For example, we give credit for the tv in the last frame in Figure 5.9 (top), bottom row, even though it is only partially captured. Furthermore, by definition, the uniform and event-based baselines are likely to get many hits for the most frequent objects. These make the baselines very strong and meaningful comparisons.

Overall, our summaries include more important people/objects with fewer frames. For example, in Split 2, our method finds 54% of important objects in 19 frames, whereas the uniform keyframe method requires 27 frames. With very short summaries, all methods perform similarly; the selected keyframes are more spread-out, so they have higher chance of including unique people/objects. With longer summaries, our method always outperforms the baselines, since they tend to include redundant frames repeating the same important person/object. On average, we find 9.13 events/video and 2.05 people/objects per event (ranging in $[4, 13]$ and $[0, 6]$, respectively).

The two baselines perform fairly similarly to one another, though the event-based keyframe selector has a slight edge by doing "smarter" temporal segmentation. Still, both are indifferent to objects' importance in creating the story of the video; their summaries contain unimportant or redundant frames as a result.

Figure 5.9 shows example full summaries from our method and the uniform baseline. The colored blocks for ours indicate the automatically discovered events. We see that our summary not only has better recall of important objects, but it also selects views in which they are prominent in the frame. Note that our summaries can sometimes include redundant frames that capture the same object if there are errors in the event segmentation (see the man captured

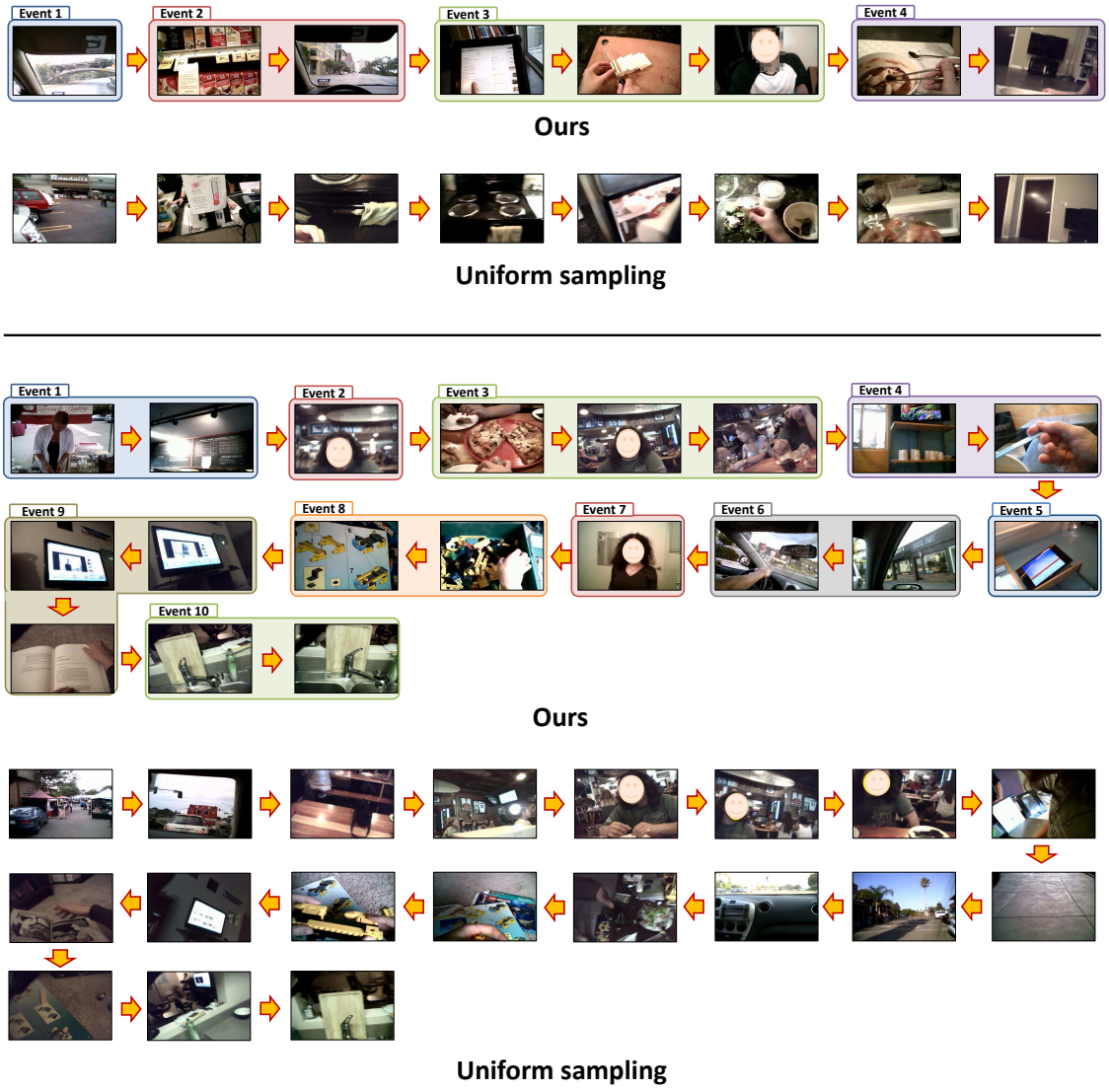


Figure 5.9: Our summary versus uniform sampling. Our summary focuses on the important people and objects.

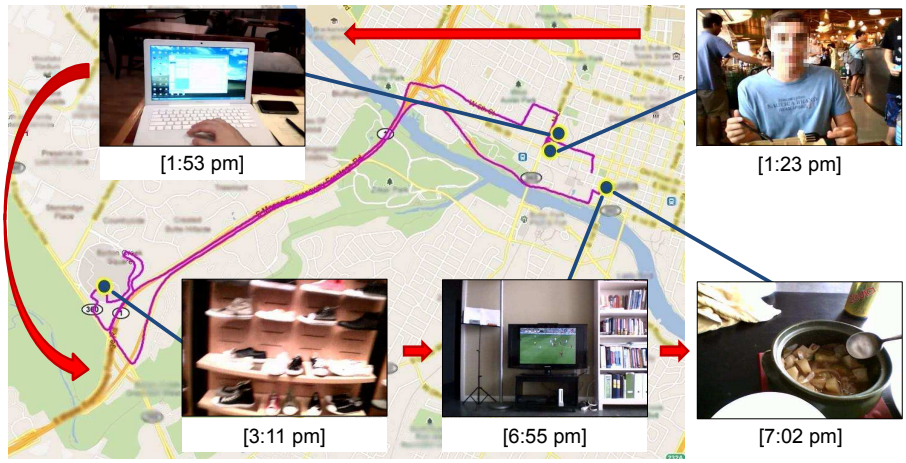


Figure 5.10: An application of our approach that shows the GPS tracks of the camera wearer, the important people and objects that s/he interacted with, and their timeline.

in both Event 2 and Event 3) or candidate important object clustering (the sink being captured twice in Event 10). Overall, we find our summary more clearly reveals the story compared to that of the baseline. For instance, for the top example: *selecting an item at the supermarket* $\rightarrow$ *driving home* $\rightarrow$ *cooking* $\rightarrow$ *eating and watching tv*. We provide original video clips and more summaries at the project webpage: http://vision.cs.utexas.edu/projects/wearable/.

Figure 5.10 shows another example; we track the camera wearer's location with a GPS receiver, and display our method's keyframes on a map with the tracks (purple trajectory) and timeline. This result suggests a novel multi-media application of our visual summarization algorithm.

5.2.4 How Prominent are the Selected Important Objects?

The evaluation above considers the recall rate of the important objects, but does not measure the *prominence* of the objects in the selected frames.

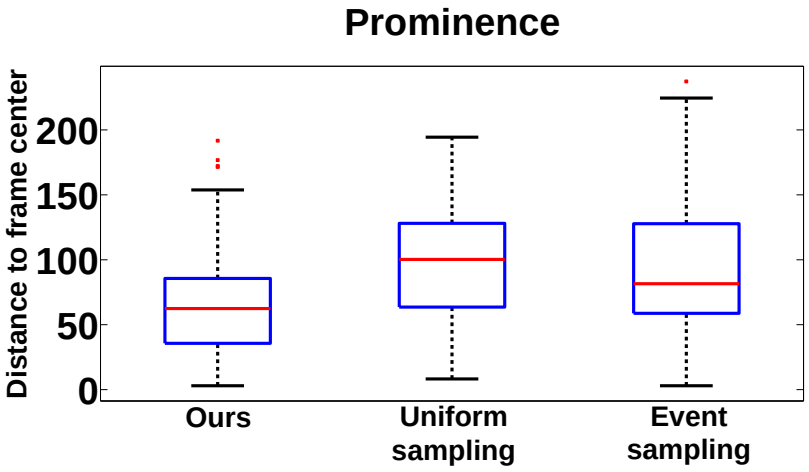


Figure 5.11: Comparison to alternative summarization strategies, in terms of object prominence measured by the distance of the selected important region to the frame center.

An informative summary should have high recall of the important objects and should contain frames in which the important objects are displayed prominently (i.e., large and centered).

To measure object prominence, we compute the Euclidean distance of the selected important region's centroid to the frame center. Figure 5.11 shows the distribution of object prominence scores for our summaries and the uniform and event-based sampling baselines' summaries. We see that our summaries more prominently display the important objects than those of the baselines. The baselines lack a notion of object importance, and therefore tend to produce summaries in which the important object is not the main focus of the frame.

5.2.5 User Studies to Evaluate Summaries

To quantify the *perceived* quality of our summaries, we ask the camera wearers to compare our method's summaries to those generated by uniform

	Much better	Better	Similar	Worse	Much worse
Imp. captured	31.25%	37.5%	18.75%	12.5%	0%
Overall quality	25%	43.75%	18.75%	12.5%	0%

Table 5.1: User study results. Numbers indicate percentage of responses for each question, always comparing our method to the baseline (i.e., highest values in "much better" are ideal).

keyframe sampling (event-based sampling performs similarly). The camera wearers are the best judges, since they know the full extent of their day that we are attempting to summarize.

We generate four pairs of summaries, each of different length. We ask the subjects to view our summary and the baseline's (in some random order unknown to the subject, and different for each pair), and answer two questions: (1) *Which summary captures the important people/objects of your day better?* and (2) *Which provides a better overall summary?* The first specifically isolates how well each method finds important, prominent objects, and the second addresses the overall quality and story of the summary.

Table 5.1 shows the results. In short, out of 16 total comparisons, our summaries were found to be better 68.75% of the time. Overall, these results are a promising indication that discovering important people/objects leads to higher quality summaries for egocentric video.

5.3 Discussion

In this chapter, I presented an egocentric video summarization approach, which produces a compact visual summary that focuses on the discovered important people and objects. The framework builds on many of the techniques described in the previous chapters. I introduced novel egocentric

features that could be indicative of important people and objects to the camera wearer. I showed how to train a regression model using egocentric, object, and region features, how to perform temporal segmentation of the video into events, and how to produce a story-board summary that is driven by the discovered important people and objects. I evaluated my approach on challenging real-world videos captured by users in uncontrolled environments, and showed that it produces significantly more informative summaries than traditional methods that are unable to focus on the important people or objects. I also showed that my approach outperforms state-of-the-art saliency measures for predicting important objects in egocentric videos.

What are the assumptions of my approach? I defined as important the things with which the camera wearer has significant interactions, and assumed that the corresponding important cues can be learned and shared across users. I believe my importance definition is valid for the general egocentric setting, since the camera wearer is likely to engage in social activities with friends, co-workers, etc., that involve interactions with objects such as food, coffee, computer, etc. These are things that the camera wearer will typically want to remember, as confirmed by our user study experiments for evaluating the summaries. However, what is truly important to the wearer can still be highly subjective; depending on the user, a person or object that s/he has significant interactions with may or may not be considered important. For example, suppose that the camera wearer is sitting in front of a computer all day. One user could consider the computer to be important, while another user could consider the computer to be unimportant because it is something that s/he sees everyday (i.e., too frequent).

To overcome the subjectivity issue of what is important, we could learn

a user-specific model that is in tune with what the user considers to be important. The challenge of training such a model would be in devising a way to efficiently obtain annotations from the user. We could start with our current user-independent model, and then have the user annotate its output predictions. The annotations would be used to retrain the model, and the process could be iterated until the user agrees with the model's predictions.

In general, evaluating summaries (or any output of an unsupervised discovery method) is a challenging task. In my experiments, I quantitatively measured the recall rate of the important objects versus the compactness of the summaries generated by my method versus those of the baselines. To quantify the perceived quality, I asked the camera wearers to compare the summaries. While these are meaningful ways for evaluating the summaries, there are some limitations. Since there is no "ground-truth" visual summary, I could not compare any method's summary against a gold-standard. I therefore was unable to measure how good any summary is on an absolute scale. Furthermore, only the camera wearer could provide a meaningful evaluation of the summary, since s/he is the only person who is aware of all the key happenings of the day. Thus, with the current protocol, there are scalability issues for evaluating a method's strengths and weaknesses, which is necessary to improve the system.

To allow independent judges (who have not seen the video) to evaluate a visual summary's informativeness, we could instead compare verbal summaries of the data. Specifically, the camera wearer could first summarize in words the key happenings of her day (i.e., who she met and what she did). Then, an independent judge could describe in words what she thinks happened that day given each method's visual summary. We could then compare the camera

wearer's verbal summary to the judge's verbal summary, and find the closest match. We could also accumulate the responses from multiple judges to obtain more reliable estimates of the informativeness of each method's summary.

Finally, we found that our event segmentations can be imperfect, which shows the difficulty of grouping frames according to low-level scene statistics. This can sometimes lead to redundant keyframes showing the same object. One way to overcome this issue is to use a GPS receiver and generate event clusters using both location and scene appearance. This would provide better separation of events, especially when the scene appearance between two neighboring events is similar. However, GPS alone would not suffice, since it cannot receive signals in indoor environments. In the future, I am interested in combining non-visual sensor signals with wearable camera data for multimedia applications of my summary work.

Chapter 6

Future Work

There are several avenues for further research prompted by this thesis, which I believe to be essential in supporting my vision of producing a system that can discover visual categories with minimal human supervision.

Evaluation for visual category discovery methods is a challenging task. Unlike the supervised scenario in which the system learns specific object properties from labeled training data and identifies (with some generalization) similar instances in novel data, in the unsupervised scenario, there is no explicit human guidance on what the system should learn. Furthermore, there could be multiple plausible solutions. For example, when computing similarities between scenes for clustering, are two images with the same objects, albeit with different layouts, scales, and orientations, more similar to each other than two images that have the same global scene structure but with objects belonging to different categories? As another example, given a dataset of animal images, is it okay to group cats and dogs together? How about if cats and dogs are the only animals in the dataset, and the remaining images contain man-made objects such as cars, televisions, and buildings?

In this thesis, I evaluated my context-aware discovery system with human-annotated category labels and segmentations in a fair and meaningful manner. However, there were some limitations—for example, a cow's head and legs are labeled simply as "cow" and different types of flowers are all

labeled as "flower". Most often, supervised methods are able to learn the properties and biases of the labeled data provided by the annotator. Without any training, we cannot expect an unsupervised learner to conform to those rules. I addressed this issue for the wearable camera discovery setting, by obtaining ground-truth labels *after* the discovery or summarization had taken place.

However, a broader issue still remains. Specifically, without any supervision, it is unclear what *types* of objects should be discovered. For the first-person discovery setting, the system was able to train with human-labeled annotations indicating what kinds of objects are considered to be important. However, there are scenarios in which the system may not have access to any meaningful labeled data. For example, if a robot is navigating an unexplored territory, we cannot predetermine what kinds of objects will appear, and what characteristics (e.g., frequency, appearance or motion patterns) they will have. In some applications, things that frequently appear are considered important. In others, such as surveillance videos, objects or actions that rarely occur are more important. Thus, a key challenge is to figure out what to target for discovery, without having observed the data first.

Alternatively, in a practical system one could determine the unsupervised learning objective online with a human-in-the-loop as the system processes the data. For example, for video summarization, the system could provide several summaries of the data, each with different target objectives, and the human could score each summary to indicate its importance based on what s/he has observed up to that point. This iterative loop could stop once the machine's and human's goals are matched.

Chapter 7

Conclusion

My thesis presented a visual category discovery framework that automatically focuses on the prevalent objects in images and videos, and learns models from them for category grouping, segmentation, and summarization with minimal human supervision.

I first described a context-aware category discovery approach that discovers novel categories that occur amidst known objects within unannotated images. Unlike the traditional discovery framework, which assumes no prior category knowledge and uses only appearance information of the image regions, my approach assumes that it is given a set of categories for which it has trained models, and uses those models as object-level context to describe an unfamiliar region. I demonstrated the approach on generic natural scenes as well as the specific case of faces in consumer photo collections, and showed that this leads to the discovered categories being more accurate and inclusive of intra-class appearance variation than those that could be found with methods that rely only on appearance.

I further showed that context-aware category discovery can be considered as a self-paced, continuing process. My approach focuses on the easier objects first, and gradually discovers new models of increasing complexity. After each discovery, it updates the set of familiar categories by training a detector for the newly found object class, which allows it to produce a richer

context model for each remaining harder, unfamiliar instance. I validated my approach on realistic natural images, and showed clear advantages compared to conventional state-of-the-art batch clustering algorithms. Overall, my results show how these new methods can (1) discover novel object categories in a realistic scenario in which there are a mix of known and unknown objects in the unlabeled visual data, (2) continuously discover categories by exploiting the variable complexity of objects, and (3) play a role in auto-tagging applications and reduce human effort for training recognition systems.

I then explained how to go further by not only discovering what categories exist, but also discovering how to segment their object instances. To overcome the chicken-and-egg problem of simultaneously estimating both the proper object segmentations and correct category groupings, I proposed to discover the shared representative instances for each category in the unlabeled visual collection. My method takes the discovered recurring structures, and builds top-down segmentation models from them to segment the corresponding objects in the entire collection. I applied the approach for image and video segmentation, and showed that the segmentations computed jointly on the collection agree more closely with true object boundaries, when compared to bottom-up baselines that rely solely on low-level appearance and motion features or can only access cues from a single image. Furthermore, for category discovery in natural images, I showed that the refined segmentations produce even more accurate clusters when provided to the context-aware discovery framework I discussed above.

Finally, building on many of the techniques from above, I described a novel egocentric video summarization approach that is driven by the discovered important people and objects. Existing summarization techniques lack high-

level information on which objects matter, and tend to produce summaries that consist of irrelevant frames or regions. In other words, they are indifferent to the impact that each object has on generating the "story" of the video. Instead, I showed how to learn category-independent importance cues designed explicitly to target the key objects and people in the video. Evaluating the predicted importance estimates and summaries on hours of challenging real-world videos captured by users in uncontrolled environments, I showed that my approach outperforms state-of-the-art saliency measures for this task, and produces significantly more informative summaries than traditional methods unable to focus on the important people or objects. This part of my thesis opens some interesting future directions for using discovery to supply practical summaries for efficient visual browsing.

In summary, the main impact of my thesis is that it shows how to build large-scale visual discovery systems that can automatically discover visual concepts with minimal human supervision. Specifically, I discussed the need for visual discovery approaches, presented the key challenges and effective techniques to address them, and explored several real-world applications. I believe that my thesis has opened the door to many interesting problems in visual discovery for object recognition and summarization.

Bibliography

[1] A. Agarwal and B. Triggs. Hyperfeatures Multilevel Local Coding for Visual Recognition. In *Proceedings of the European Conference on Computer Vision (ECCV)*, 2006.

[2] O. Aghazadeh, J. Sullivan, and S. Carlsson. Novelty Detection from an Egocentric Perspective. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2011.

[3] B. Alexe, T. Deselaers, and V. Ferrari. What is an Object? In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2010.

[4] Y. Amit and A. Trouve. POP: Patchwork of Parts Models for Object Recognition. *International Journal of Computer Vision (IJCV)*, 75(2), 2007.

[5] A. Aner and J. R. Kender. Video Summaries through Mosaic-Based Shot and Scene Clustering. In *Proceedings of the European Conference on Computer Vision (ECCV)*, 2002.

[6] P. Arbelaez, M. Maire, C. Fowlkes, and J. Malik. From Contours to Regions: An Empirical Evaluation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2009.

[7] P. Arbelaez, M. Maire, C. Fowlkes, and J. Malik. Contour Detection and Hierarchical Image Segmentation. *IEEE Transactions on Pattern*

Analysis and Machine Intelligence (TPAMI), 2011.

[8] H. Arora, N. Loeff, D. Forsyth, and N. Ahuja. Unsupervised Segmentation of Objects using Efficient Learning. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2007.

[9] F. Bach, G. Lanckriet, and M. Jordan. Multiple Kernel Learning, Conic Duality, and the SMO Algorithm. In *Proceedings of International Conference on Machine Learning (ICML)*, 2004.

[10] X. Bai, J. Wang, D. Simons, and G. Sapiro. Video Snapcut: Robust Video Object Cutout using Localized Classifiers. In *ACM Transactions on Graphics (Proceedings of SIGGRAPH)*, 2009.

[11] A. Basharat, A. Gritai, and M. Shah. Learning Object Motion Patterns for Anomaly Detection and Improved Object Detection. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2008.

[12] D. Batra, A. Kowdle, D. Parikh, J. Luo, and T. Chen. Interactively co-segmentating topically related images with intelligent scribble guidance. *International Journal of Computer Vision (IJCV)*, 93(3):273–292, 2011.

[13] T. Berg, A. Berg, J. Edwards, M. Maire, R. White, Y. Teh, E. Learned-Miller, and D. Forsyth. Names and Faces in the News. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2004.

[14] E. Borenstein, E. Sharon, and S. Ullman. Combining Top-down and Bottom-up segmentation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2004.

[15] A. Bosch, A. Zisserman, and X. Munoz. Representing Shape with a Spatial Pyramid Kernel. In *Proceedings of the ACM International Conference on Image and Video Retrieval (CIVR)*, 2007.

[16] Y-L. Boureau, F. Bach, Y. LeCun, and J. Ponce. Learning Mid-Level Features for Recognition. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2010.

[17] Y. Boykov and V. Kolmogorov. An Experimental Comparison of Min-Cut/Max-Flow Algorithms for Energy Minimization in Vision. *IEEE Transactions on Pattern Analysis and Machine Intelligence (TPAMI)*, 26(9):1124–1137, 2004.

[18] Y. Boykov, O. Veksler, and R. Zabih. Efficient Approximate Energy Minimization via Graph Cuts. *IEEE Transactions on Pattern Analysis and Machine Intelligence (TPAMI)*, 20(12):1222–1239, 2001.

[19] W. Brendel and S. Todorovic. Video Object Segmentation by Tracking Regions. In *Proceedings of the IEEE International Conference on Computer Vision (ICCV)*, 2009.

[20] S. Brin and L. Page. The Anatomy of a Large-Scale Hypertextual Web Search Engine. In *International World Wide Web Conference (WWW)*, 1998.

[21] T. Brox and J. Malik. Object Segmentation by Long Term Analysis of Point Trajectories. In *Proceedings of the European Conference on Computer Vision (ECCV)*, 2010.

[22] Caltech 101 Image Database, L. Fei-Fei, R. Fergus, and P. Perona.

[23] Caltech 256 Image Database, G. Griffin and A. Holub and P. Perona. *http://www.vision.caltech.edu/ImageDatasets/Caltech256/*.

[24] J. Carreira and C. Sminchisescu. Constrained Parametric Min-Cuts for Automatic Object Segmentation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2010.

[25] Y. Caspi, A. Axelrod, Y. Matsushita, and A. Gamliel. Dynamic Stills and Clip Trailer. In *The Visual Computer*, 2006.

[26] P. Chockalingam, N. Pradeep, and S. Birchfield. Adaptive Fragments-Based Tracking of Non-Rigid Objects Using Level Sets. In *Proceedings of the IEEE International Conference on Computer Vision (ICCV)*, 2009.

[27] O. Chum, M. Perdoch, and J. Matas. Geometric min-Hashing: Finding a (Thick) Needle in a Haystack. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2009.

[28] B. Clarkson and A. Pentland. Unsupervised Clustering of Ambulatory Audio and Video. In *Proceedings of the IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP)*, 1999.

[29] D. Comaniciu and P. Meer. Mean Shift: A Robust Approach Toward Feature Space Analysis. *IEEE Transactions on Pattern Analysis and Machine Intelligence (TPAMI)*, 24(5):603–619, 2002.

[30] T. F. Cootes, C. J. Taylor, D. H. Cooper, and J. Graham. Active shape models - their training and application. *Computer Vision and Image Understanding (CVIU)*, 61(1):38–59, January 1995.

[31] D. Crandall, P. Felzenszwalb, and D. Huttenlocher. Spatial Priors for Part-based Recognition using Statistical Models. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2005.

[32] G. Csurka, C. Bray, C. Dance, and L. Fan. Visual Categorization with Bags of Keypoints. In *Proceedings of the European Conference on Computer Vision (ECCV)*, 2004.

[33] A. Dempster, N. Laird, and D. Rubin. Maximum likelihood from incomplete data via the em algorithm. *Journal of the Royal Statistical Society, Series B*, 39(1):1–38, 1977.

[34] D. Dueck and B. Frey. Non-metric Affinity Propagation for Unsupervised Image Categorization. In *Proceedings of the IEEE International Conference on Computer Vision (ICCV)*, 2007.

[35] I. Endres and D. Hoiem. Category Independent Object Proposals. In *Proceedings of the European Conference on Computer Vision (ECCV)*, 2010.

[36] M. Everingham, J. Sivic, and A. Zisserman. Hello! My Name is... Buffy - Automatic Naming of Characters in TV Video. In *Proceedings of the British Machine Vision Conference (BMVC)*, 2006.

[37] M. Everingham, L. Van Gool, C. K. I. Williams, J. Winn, and A. Zisserman. The PASCAL Visual Object Classes Challenge 2011 Results. http://www.pascal-network.org/challenges/VOC/voc2011/workshop/index.html.

[38] M. Everingham, A. Zisserman, C. K. I. Williams, and L. Van Gool. The PASCAL Visual Object Classes Challenge 2006 Results. http://www.pascal-network.org/challenges/VOC/voc2006/results.pdf.

[39] A. Fathi, A. Farhadi, and J. Rehg. Understanding Egocentric Activities. In *Proceedings of the IEEE International Conference on Computer Vision (ICCV)*, 2011.

[40] P. Felzenszwalb and D. Huttenlocher. Efficient Graph-Based Image Segmentation. *International Journal of Computer Vision (IJCV)*, 59(2), 2004.

[41] P. Felzenszwalb and D. Huttenlocher. Pictorial Structures for Object Recognition. *International Journal of Computer Vision (IJCV)*, 61(1), 2005.

[42] P. Felzenszwalb and D. Huttenlocherz. Efficient Matching of Pictorial Structures. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2000.

[43] P. Felzenszwalb, D. McAllester, and D. Ramanan. A Discriminatively Trained, Multiscale, Deformable Part Model. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2008.

[44] R. Fergus, L. Fei-Fei, P. Perona, and A. Zisserman. Learning Object Categories from Google's Image Search. In *Proceedings of the IEEE International Conference on Computer Vision (ICCV)*, 2005.

[45] R. Fergus, P. Perona, and A. Zisserman. Object Class Recognition by Unsupervised Scale-Invariant Learning. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2003.

[46] W. Freeman and H. Zhang. Shape-Time Photography. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2003.

[47] A. Gallagher and T. Chen. Using Group Prior to Identify People in Consumer Images. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2007.

[48] A. Gallagher and T. Chen. Clothing Cosegmentation for Recognizing People. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2008.

[49] A. Gallagher and T. Chen. Understanding Images of Groups of People. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2009.

[50] C. Galleguillos, A. Rabinovich, and S. Belongie. Object Categorization using Co-Occurrence, Location and Appearance. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2008.

[51] T. Gartner, P. Flach, and S. Wrobel. On Graph Kernels: Hardness Results and Efficient Alternatives. In *Proceedings of the Conference on Computational Learning Theory (COLT)*, 2003.

[52] D. Goldman, B. Curless, D. Salesin, and S. Seitz. Schematic Storyboarding for Video Visualization and Editing. In *ACM Transactions on Graphics (Proceedings of SIGGRAPH)*, 2006.

[53] S. Gould, R. Fulton, and D. Koller. Decomposing a Scene into Geometric and Semantically Consistent Regions. In *Proceedings of the IEEE International Conference on Computer Vision (ICCV)*, 2009.

[54] K. Grauman and T. Darrell. The Pyramid Match Kernel: Discriminative Classification with Sets of Image Features. In *Proceedings of the IEEE International Conference on Computer Vision (ICCV)*, 2005.

[55] K. Grauman and T. Darrell. Unsupervised Learning of Categories from Sets of Partially Matching Image Features. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2006.

[56] M. Grundmann, V. Kwatra, M. Han, and I. Essa. Efficient Hierarchical Graph Based Video Segmentation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2010.

[57] C. Gu, J. J. Lim, P. Arbelaez, and J. Malik. Recognition using Regions. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2009.

[58] R. Hamid, A. Johnson, S. Batta, A. Bobick, C. Isbell, and G. Coleman. Detection and Explanation of Anomalous Activities: Representing Activities as Bags of Event n-Grams. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2005.

[59] X. He, R. Zemel, and M. Carreira-Perpinan. Multiscale Conditional Random Fields for Image Labeling. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2004.

[60] G. Heitz and D. Koller. Learning Spatial Context: Using Stuff to Find Things. In *Proceedings of the European Conference on Computer Vision (ECCV)*, 2008.

[61] S. Hodges, E. Berry, and K. Wood. Sensecam: A Wearable Camera which Stimulates and Rehabilitates Autobiographical Memory. *Memory*, 2011.

[62] D. Hoiem, A. Efros, and M. Hebert. Geometric Context from a Single Image. In *Proceedings of the IEEE International Conference on Computer Vision (ICCV)*, 2005.

[63] D. Hoiem, A. Efros, and M. Hebert. Putting Objects in Perspective. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2006.

[64] D. Hoiem, A. N. Stein, A. Efros, and M. Hebert. Recovering Occlusion Boundaries from a Single Image. In *Proceedings of the IEEE International Conference on Computer Vision (ICCV)*, 2007.

[65] Y. Huang, Q. Liu, and D. Metaxas. Video Object Segmentation by Hypergraph Cut. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2009.

[66] T. Huynh, M. Fritz, and B. Schiele. Discovery of Activity Patterns using Topic Models. 2008.

[67] N. Jojic, A. Perina, and V. Murino. Structural Epitome: A Way to Summarize One's Visual Experience. In *Advances in Neural Information Processing Systems (NIPS)*, 2010.

[68] M. Jones and J. Rehg. Statistical Color Models with Application to Skin Detection. *International Journal of Computer Vision (IJCV)*, 46(1), 2002.

[69] A. Kaplan and G. Murphy. The Acquisition of Category Structure in Unsupervised Learning. *Memory & Cognition*, 27:699–712, 1999.

[70] A. Kapoor, K. Grauman, R. Urtasun, and T. Darrell. Gaussian Processes for Object Categorization. *International Journal of Computer Vision (IJCV)*, 88(2):169–188, 2009.

[71] H. Kashima, K. Tsuda, and A. Inokuchi. Kernels on Graphs. *Kernels and Bioinformatics*, 2004.

[72] G. Kim, C. Faloutsos, and M. Hebert. Unsupervised Modeling of Object Categories Using Link Analysis Techniques. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2008.

[73] G. Kim and A. Torralba. Unsupervised Detection of Regions of Interest using Iterative Link Analysis. In *Advances in Neural Information Processing Systems (NIPS)*, 2009.

[74] J. Kim and K. Grauman. Observe Locally, Infer Globally: a Space-Time MRF for Detecting Abnormal Activities with Incremental Updates. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2009.

[75] J. Kim and K. Grauman. Boundary-Preserving Dense Local Regions. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2011.

[76] K. Kitani, T. Okabe, Y. Sato, and A. Sugimoto. Fast Unsupervised Ego-Action Learning for First-Person Sports Video. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2011.

[77] P. Kohli, L. Ladicky, and P. Torr. Robust Higher Order Potentials for Enforcing Label Consistency. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2008.

[78] V. Kolmogorov and R. Zabih. What Energy Functions can be Minimized via Graph Cuts? *IEEE Transactions on Pattern Analysis and Machine Intelligence (TPAMI)*, 26(2):147–159, 2004.

[79] M. Kolsch and M. Turk. Robust Hand Detection. In *Proceedings of the Int. Conf. on Automatic Face & Gesture Recognition*, 2004.

[80] M. P. Kumar, P. H. S. Torr, and A. Zisserman. Obj cut. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2005.

[81] G.R.G. Lanckriet, L. El Ghaoui, and M.I. Jordan. Robust Novelty Detection with Single-Class MPM. In *Advances in Neural Information Processing Systems (NIPS)*, 2002.

[82] J. Laurikkala, M. Juhola, and E. Kentala. Informal Identification of Outliers in Medical Data. In *Intelligent Data Analysis in Medicine and Pharmacology (IDAMAP)*, 2000.

[83] S. Lazebnik and M. Raginsky. An Empirical Bayes Approach to Contextual Region Classification. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2009.

[84] S. Lazebnik, C. Schmid, and J. Ponce. Semi-Local Affine Parts for Object Recognition. In *Proceedings of the British Machine Vision Conference (BMVC)*, 2004.

[85] S. Lazebnik, C. Schmid, and J. Ponce. Beyond Bags of Features: Spatial Pyramid Matching for Recognizing Natural Scene Categories. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2006.

[86] Y. J. Lee and K. Grauman. Foreground Focus: Unsupervised Learning From Partially Matching Images. *International Journal of Computer Vision (IJCV)*, 85, 2009.

[87] Y. J. Lee and K. Grauman. Shape Discovery from Unlabeled Image Collections. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2009.

[88] Y. J. Lee and K. Grauman. Collect-Cut: Segmentation with Top-Down Cues Discovered in Multi-Object Images. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2010.

[89] Y. J. Lee and K. Grauman. Object-Graphs for Context-Aware Category Discovery. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2010.

[90] Y. J. Lee and K. Grauman. Face Discovery with Social Context. In *Proceedings of the British Machine Vision Conference (BMVC)*, 2011.

[91] Y. J. Lee and K. Grauman. Learning the Easy Things First: Self-Paced Visual Category Discovery. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2011.

[92] Y. J. Lee and K. Grauman. Object-Graphs for Context-Aware Visual Category Discovery. In *IEEE Transactions on Pattern Analysis and Machine Intelligence (TPAMI)*, 2011.

[93] Y. J. Lee, J. Kim, and K. Grauman. Key-Segments for Video Object Segmentation. In *Proceedings of the IEEE International Conference on Computer Vision (ICCV)*, 2011.

[94] B. Leibe, A. Leonardis, and B. Schiele. Combined Object Categorization and Segmentation with an Implicit Shape Model. In *Wkshp on Statistical Learning in Computer Vision*, 2004.

[95] B. Leibe, A. Leonardis, and B. Schiele. Robust Object Detection with Interleaved Categorization and Segmentation. *International Journal of Computer Vision (IJCV)*, 77(1), 2008.

[96] H. Ling and S. Soatto. Proximity Distribution Kernel for Geometric Context in Recognition. In *Proceedings of the IEEE International Conference on Computer Vision (ICCV)*, 2007.

[97] C. Liu. *Beyond Pixels: Exploring New Representations and Applications for Motion Analysis*. PhD thesis, MIT, 2009.

[98] D. Liu and T. Chen. Background Cutout with Automatic Object Discovery. In *Proceedings of the IEEE International Conference on Image Processing (ICIP)*, 2007.

[99] D. Liu and T. Chen. Unsupervised Image Categorization and Object Localization using Topic Models and Correspondences between Images. In *Proceedings of the IEEE International Conference on Computer Vision (ICCV)*, 2007.

[100] T. Liu, J. Sun, N. Zheng, X. Tang, and H. Shum. Learning to Detect a Salient Object. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2007.

[101] D. Lowe. Distinctive Image Features from Scale-Invariant Keypoints. *International Journal of Computer Vision (IJCV)*, 60(2), 2004.

[102] M. Maire, P. Arbelaez, C. Fowlkes, and J. Malik. Using Contours to Detect and Localize Junctions in Natural Images. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2008.

[103] T. Malisiewicz and A. Efros. Improving Spatial Support for Objects via Multiple Segmentations. In *Proceedings of the British Machine Vision Conference (BMVC)*, 2007.

[104] T. Malisiewicz and A. Efros. Recognition by Association via Learning Per-exemplar Distances. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2008.

[105] T. Malisiewicz and A. Efros. Beyond Categories: The Visual Memex Model for Reasoning About Object Relationships. In *Advances in Neural Information Processing Systems (NIPS)*, 2009.

[106] G. Manson, G. Pierce, and K. Worden. On the Long-Term Stability of Normal Condition for Damage Detection in a Composite Panel. In *International Conference on Damage Assessment of Structures*, 2001.

[107] G. Manson, G. Pierce, K. Worden, T. Monnier, P. Guy, and K. Atherton. Long Term Stability of Normal Condition Data for Novelty Detection. In *International Symposium on Smart Structures and Materials*, 2000.

[108] M. Markou and S. Singh. Novelty Detection: a Review - Part 1: Statistical Approaches. *Signal Processing*, 83(12):2481–2497, 2003.

[109] M. Markou and S. Singh. Novelty Detection: a Review - Part 2: Neural Network Based Approaches. *Signal Processing*, 83(12):2499–2521, 2003.

[110] L. Mukherjee, V. Singh, and C. R. Dyer. Half-Integrality Based Algorithms for Cosegmentation of Images. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2009.

[111] A. Ng, M. Jordan, and Y. Weiss. On Spectral Clustering: Analysis and an Algorithm. In *Advances in Neural Information Processing Systems (NIPS)*, 2001.

[112] E. Olson, M. Walter, J. Leonard, and S. Teller. Single Cluster Graph Partitioning for Robotics Applications. In *Proceedings of Robotics: Science and Systems (RSS)*, 2005.

[113] D. Parikh, C. L. Zitnick, and T. Chen. From Appearance to Context-Based Recognition: Dense Labeling in Small Images. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2008.

[114] P. Perona and W. Freeman. A Factorization Approach to Grouping. In *Proceedings of the European Conference on Computer Vision (ECCV)*, 1998.

[115] J. Philbin and A. Zisserman. Object Mining using a Matching Graph on Very Large Image Collections. In *Proceedings of the Indian Conference on Computer Vision, Graphics and Image processing (ICVGIP)*, 2008.

[116] J. Platt. *Advances in Large Margin Classifiers*, chapter Probabilistic Outputs for Support Vector Machines and Comparisons to Regularized Likelihood Methods. MIT Press, 1999.

[117] B. Price, B. Morse, and S. Cohen. Livecut: Learning-based Interactive Video Segmentation by Evaluation of Multiple Propagated Cues. In *Proceedings of the IEEE International Conference on Computer Vision (ICCV)*, 2009.

[118] Y. Pritch, A. Rav-Acha, A. Gutman, and S. Peleg. Webcam Synopsis: Peeking Around the World. In *Proceedings of the IEEE International Conference on Computer Vision (ICCV)*, 2007.

[119] T. Quack, V. Ferrari, B. Leibe, and L. Van Gool. Efficient Mining of Frequent and Distinctive Feature Configurations. In *Proceedings of the IEEE International Conference on Computer Vision (ICCV)*, 2007.

[120] P. Quelhas, F. Monay, J.-M. Odobez, D. Gatica-Perez, T. Tuytelaars, and L. Van Gool. Modeling Scenes with Local Descriptors and Latent Aspects. In *Proceedings of the IEEE International Conference on Computer Vision (ICCV)*, 2005.

[121] A. Rabinovich, A. Vedaldi, C. Galleguillos, E. Wiewiora, and S. Belongie. Objects in Context. In *Proceedings of the IEEE International Conference on Computer Vision (ICCV)*, 2007.

[122] A. Rav-Acha, Y. Pritch, and S. Peleg. Making a Long Video Short. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2006.

[123] X. Ren and C. Gu. Figure-Ground Segmentation Improves Handled Object Recognition in Egocentric Video. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2010.

[124] X. Ren and J. Malik. Tracking as Repeated Figure/Ground Segmentation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2007.

[125] C. Rother, V. Kolmogorov, and A. Blake. Grabcut. In *ACM Transactions on Graphics (Proceedings of SIGGRAPH)*, 2004.

[126] C. Rother, V. Kolmogorov, T. Minka, and A. Blake. Cosegmentation of Image Pairs by Histogram Matching - Incorporating a Global Constraint into MRFs. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2006.

[127] B. Russell, A. Efros, J. Sivic, W. Freeman, and A. Zisserman. Using Multiple Segmentations to Discover Objects and their Extent in Image Collections. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2006.

[128] B. Russell, A. Efros, J. Sivic, W. Freeman, and A. Zisserman. Segmenting Scenes by Matching Image Composites. In *Advances in Neural Information Processing Systems (NIPS)*, 2009.

[129] B. Scholkopf, R. Williamson, A. Smola, J. Shawe-Taylor, and J. Platt. Support Vector Method for Novelty Detection. In *Advances in Neural Information Processing Systems (NIPS)*, 2000.

[130] C. Schuldt, I. Laptev, and B. Caputo. Recognizing Human Actions: A Local SVM Approach. In *Proceedings of the International Conference on Pattern Recognition (ICPR)*, 2004.

[131] J. Shi and J. Malik. Motion Segmentation and Tracking Using Normalized Cuts. In *Proceedings of the IEEE International Conference on Computer Vision (ICCV)*, 1998.

[132] J. Shi and J. Malik. Normalized Cuts and Image Segmentation. *IEEE Transactions on Pattern Analysis and Machine Intelligence (TPAMI)*, 22(8):888–905, August 2000.

[133] J. Shotton, J. Winn, C. Rother, and A. Criminisi. Textonboost: Joint Appearance, Shape and Context Modeling for Multi-Class Object Recognition and Segmentation. In *Proceedings of the European Conference on Computer Vision (ECCV)*, 2006.

[134] J. Sivic, B. Russell, A. Efros, A. Zisserman, and W. Freeman. Discovering Object Categories in Image Collections. In *Proceedings of the IEEE International Conference on Computer Vision (ICCV)*, 2005.

[135] J. Sivic and A. Zisserman. Video Google: A Text Retrieval Approach to Object Matching in Videos. In *Proceedings of the IEEE International Conference on Computer Vision (ICCV)*, 2003.

[136] Y. Song and T. Leung. Context-Aided Human Recognition Clustering. In *Proceedings of the European Conference on Computer Vision (ECCV)*, 2006.

[137] E. Spriggs, F. De la Torre, and M. Hebert. Temporal segmentation and activity classification from first-person sensing. In *CVPR Workshop on Egocentric Vision*, 2009.

[138] C. Stauffer and E. Grimson. Learning patterns of activity using real-time tracking. *IEEE Transactions on Pattern Analysis and Machine Intelligence (TPAMI)*, 22, August 2000.

[139] C. Stauffer and W.E.L. Grimson. Adaptive background mixture modelsfor real-time tracking. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 1999.

[140] Z. Stone, T. Zickler, and T. Darrell. Autotagging Facebook: Social Network Context Improves Photo Annotation. In *First IEEE Workshop on Internet Vision*, 2008.

[141] M. Swain and D. Ballard. Color Indexing. *International Journal of Computer Vision (IJCV)*, 7(1):11–32, 1991.

[142] P. Tan, M. Steinbach, and V. Kumar. *Introduction to Data Mining*. 2005.

[143] X. Tan and B. Triggs. Enhanced Local Texture Feature Sets for Face Recognition under Difficult Lighting Conditions. *IEEE Transactions on Image Processing*, 19(6):1635–1650, June 2010.

[144] Y. Tian, W. Liu, R. Xiao, F. Wen, and X. Tang. A Face Annotation Framework with Partial Clustering and Interactive Labeling. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2007.

[145] S. Todorovic and N. Ahuja. Extracting Subimages of an Unknown Category from a Set of Images. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2006.

[146] A. Torralba. Contextual Priming for Object Detection. *International Journal of Computer Vision (IJCV)*, 2003.

[147] A. Torralba and A. Efros. Unbiased Look at Dataset Bias. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2011.

[148] D. Tsai, M. Flagg, and J. Rehg. Motion Coherent Tracking with Multi-Label MRF Optimization. In *Proceedings of the British Machine Vision Conference (BMVC)*, 2010.

[149] Z. Tu. Auto-context and Its Application to High-level Vision Tasks. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2008.

[150] Z. Tu, X. Chen, A. L. Yuille, and S.-C. Chen. Image Parsing: Unifying Segmentation, Detection, and Recognition. In *Proceedings of the IEEE International Conference on Computer Vision (ICCV)*, 2003.

[151] T. Tuytelaars, C. Lampert, M. Blaschko, and W. Buntine. Unsupervised Object Discovery: A Comparison. *International Journal of Computer Vision (IJCV)*, 2010.

[152] A. Vazquez-Reina, S. Avidan, H. Pfister, and E. Miller. Multiple Hypothesis Video Segmentation from Superpixel Flows. In *Proceedings of the European Conference on Computer Vision (ECCV)*, 2010.

[153] A. Vedaldi and S. Soatto. Relaxed Matching Kernels for Object Recognition. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2008.

[154] S. Vicente, V. Kolmogorov, and C. Rother. Cosegmentation Revisited: Models and Optimization. In *Proceedings of the European Conference on Computer Vision (ECCV)*, 2010.

[155] P. Viola and M. Jones. Rapid Object Detection using a Boosted Cascade of Simple Features. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2001.

[156] D. Walther and C. Koch. Modeling Attention to Salient Proto-Objects. *Neural Networks*, 19:1395–1407, 2006.

[157] G. Wang, A. Gallagher, J. Luo, and D. Forsyth. Seeing People in Social Context: Recognizing People and Social Relationships. In *Proceedings of the European Conference on Computer Vision (ECCV)*, 2010.

[158] J. Wang, J. Yang, K. Yu, F. Lv, T. Huang, and Y. Gong. Locality-constrained linear coding for image classification. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2010.

[159] X. Wang, X. Ma, and E. Grimson. Unsupervised Activity Perception by Hierarchical Bayesian Models. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2007.

[160] M. Weber, M. Welling, and P. Perona. Unsupervised Learning of Models for Recognition. In *Proceedings of the European Conference on Computer Vision (ECCV)*, 2000.

[161] D. Weinshall, H. Hermansky, A. Zweig, J. Luo, H. Jimison, F. Ohl, and M. Pavel. Beyond Novelty Detection: Incongruent Events, when General and Specific Classifiers Disagree. In *Advances in Neural Information Processing Systems (NIPS)*, 2008.

[162] R. Weischedel. Adaptive Natural Language Processing. In *Proceedings of the Annual Meeting of the Association for Computational Linguistics (ACL)*, 1990.

[163] J. Winn and N. Jojic. LOCUS: Learning Object Classes with Unsupervised Segmentation. In *Proceedings of the IEEE International Conference on Computer Vision (ICCV)*, 2005.

[164] W. Wolf. Key Frame Selection by Motion Analysis. In *Proceedings of the IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP)*, 1996.

[165] T.-F. Wu, C.-J. Lin, and R. C. Weng. Probability Estimates for Multi-Class Classification by Pairwise Coupling. *Journal of Machine Learning Research (JMLR)*, 5:975–1005, August 2004.

[166] T. Xiang and S. Gong. Incremental and adaptive abnormal behaviour detection. *Computer Vision and Image Understanding (CVIU)*, 111:59–73, June 2008.

[167] J. Yang, K. Yu, Y. Gong, and T. Huang. Linear spatial pyramid matching sparse coding for image classification. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2009.

[168] S. Yu and J. Shi. Segmentation given partial grouping constraints. *IEEE Transactions on Pattern Analysis and Machine Intelligence (TPAMI)*, 26(2):173–183, 2004.

[169] J. Yuen, B. Russell, C. Liu, and A. Torralba. Labelme Video: Building a Video Database with Human Annotations. In *Proceedings of the IEEE International Conference on Computer Vision (ICCV)*, 2009.

[170] H. J. Zhang, J. Wu, D. Zhong, and S. Smoliar. An Integrated System for Content-Based Video Retrieval and Browsing. In *Pattern Recognition*, 1997.

[171] B. Zhao, J. Kwok, and C. Zhang. Multiple Kernel Clustering. In *Proceedings of the IEEE International Conference on Data Mining (ICDM)*, 2009.

[172] L. Zhao, G. Sukthankar, and R. Sukthankar. Robust Active Learning using Crowdsourced Annotations for Activity Recognition. In *AAAI workshop on Human Computation*, 2011.

[173] M. Zhao, Y. Teo, S. Liu, T.-S. Chua, and R. Jain. Automatic Person Annotation of Family Photo Album. In *Proceedings of the ACM International Conference on Image and Video Retrieval (CIVR)*, 2006.

[174] H. Zhong, J. Shi, and M. Visontai. Detecting unusual activity in video. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2004.

Vita

Yong Jae Lee was born on February 22nd, 1984, the son of Sang Pal Lee and Young Hee Lee. After spending his childhood years in Ethiopia, South Korea, New Jersey, Thailand, and Belgium, he received his high school diploma from the Hong Kong International School, Hong Kong, in 2002. He received the Bachelor of Science degree in Electrical Engineering from the University of Illinois at Urbana-Champaign in 2006, and joined the University of Texas at Austin Computer Vision Group headed by Prof. Kristen Grauman in June 2007 as a graduate research assistant.

Permanent address: Hanshin 2-cha 103-506, Jamwon-Dong,
Seocho-Gu, Seoul, South Korea

This dissertation was typeset with LaTeX[†] by the author.

[†]LaTeX is a document preparation system developed by Leslie Lamport as a special version of Donald Knuth's TeX Program.