Addressing Ambiguity In Object Instance Detection

Edward Hsiao

CMU-RI-TR-13-16

Submitted in partial fulfillment of the requirements for the degree of Doctor of Philosophy in Robotics.

The Robotics Institute
Carnegie Mellon University
Pittsburgh, PA 15213

June 2013

Thesis Committee
Martial Hebert, Chair
Alexei Efros
Takeo Kanade
Andrew Zisserman, *University of Oxford*

Abstract

In this thesis, we study the topic of ambiguity when detecting object instances in scenes with severe clutter and occlusions. Our work focuses on the three key areas: (1) objects that have ambiguous features, (2) objects where discriminative point-based features cannot be reliably extracted, and (3) occlusions.

Current approaches for object instance detection rely heavily on matching discriminative point-based features such as SIFT. While one-to-one correspondences between an image and an object can often be generated, these correspondences cannot be obtained when objects have ambiguous features due to similar and repeated patterns. We present the Discriminative Hierarchical Matching (DHM) method which preserves feature ambiguity at the matching stage until hypothesis testing by vector quantization. We demonstrate that combining our quantization framework with Simulated Affine features can significantly improve the performance of 3D point-based recognition systems.

While discriminative point-based features work well for many objects, they cannot be stably extracted on smooth objects which have large uniform regions. To represent these feature-poor objects, we first present Gradient Networks, a framework for robust shape matching without extracting edges. Our approach incorporates connectivity directly on low-level gradients and significantly outperforms approaches which use only local information or coarse gradient statistics. Next, we present the Boundary and Region Template (BaRT) framework which incorporates an explicit boundary representation with the interior appearance of the object. We show that the lack of texture in the object interior is actually informative and that an explicit representation of the boundary performs better than a coarse representation.

While many approaches work well when objects are entirely visible, their performance decrease rapidly with occlusions. We introduce two methods for increasing the robustness of object detection in these challenging scenarios. First, we present a framework for capturing the occlusion structure under arbitrary object viewpoint by modeling the Occlusion Conditional Likelihood that a point on the object is visible given the visibility labelings of all other points. Second, we propose a method to predict the occluding region and score a probabilistic matching pattern by searching for a set of valid occluders. We demonstrate significant increase in detection performance under severe occlusions.

Contents

Abstract i

Acknowledgments vii

List of Figures ix

List of Tables xiii

Part I. Introduction 1

Chapter 1. Overview 3

1.1 Detecting Object Instances 4

1.2 Summary of Contributions 6

1.3 Roadmap 6

1.4 Publication Note 7

Chapter 2. Background 9

2.1 Recognition Using Discriminative Features 9

2.2 Shape Matching 14

2.3 Occlusion Reasoning 19

Part II. Ambiguity of Discriminative Features 21

Chapter 3. Addressing Ambiguous Features 23

3.1 Quantization Framework 25

3.2 Viewpoint Variations 29

3.3 Evaluation 30

3.4 Discussion 38

Part III. Representing Objects Without Discriminative Features 39

Chapter 4. Shape Matching Using Gradient Networks 41
4.1 Formulation . . . 43
4.2 Local Shape Potential . . . 44
4.3 Shape Matching . . . 47
4.4 Probability Calibration . . . 48
4.5 Soft Shape Model . . . 49
4.6 Evaluation . . . 49
4.7 Discussion . . . 58

Chapter 5. Combining Boundary and Region Information 59
5.1 Boundary Representation . . . 59
5.2 Region Representation . . . 60
5.3 Implementation Details . . . 63
5.4 Evaluation . . . 64
5.5 Discussion . . . 69

Part IV. Occlusion Reasoning 71

Chapter 6. Representation under Arbitrary Viewpoint 73
6.1 Occlusion Model . . . 75
6.2 Combining with Object Detection . . . 82
6.3 Evaluation . . . 84
6.4 Discussion . . . 93

Chapter 7. Coherent Reasoning through Efficient Search 95
7.1 Occlusion Model . . . 95
7.2 Combining with Object Detection . . . 100
7.3 Evaluation . . . 102
7.4 Discussion . . . 107

Part V. Conclusion 109

Chapter 8. Contributions 111

Chapter 9. Future Directions 113
9.1 Fine-grained Verification . . . 113
9.2 Scalable Representation . . . 114

9.3 Incorporating Depth Information 115

Chapter 10. Closing Thoughts **117**

Appendices **119**

Appendix A. Datasets **121**

A.1 CMU Grocery Dataset (CMU10_3D) 122

A.2 CMU Kitchen Occlusion Dataset (CMU_KO8) 123

Appendix B. Computing Occlusion Distributions **125**

B.1 Probability Density of $\hat{w}$ 125

B.2 Occlusion Prior 126

B.3 Occlusion Conditional Likelihood 127

B.4 Computing Area for Silhouette 129

References **131**

Acknowledgments

First and foremost, I would like to thank my advisor Martial Hebert for his guidance and support in developing this thesis. Thank you for giving me the freedom to explore the field and for pushing me forward when I needed it. The skills and knowledge I have learned from you will continue to shape me in the future.

I would also like to thank my thesis committee members, Alexei Efros, Takeo Kanade and Andrew Zisserman who have offered invaluable comments and suggestions.

I am also grateful to Pietro Perona, my undergraduate thesis advisor at the California Institute of Technology, who got me interested in the field of Computer Vision and who continues to be a mentor. He contributed greatly to my decision to pursue a PhD.

In addition, I am grateful for having the opportunity to work with many great researchers during my internship at Microsoft Research: Richard Szeliski, Larry Zitnick, Krishnan Ramnath, Sudipta Sinha and Simon Baker. It was a pleasure and an honor to work with them.

Thanks also to my office mates, Scott Satkin, Daniel Munoz, Stéphane Ross, Yuandong Tian, and Carl Doersch for years of random discussions and good times. Life at CMU would not have been the same without you guys. I would also like to thank all my friends who have gotten me out of the office to play volleyball, racquet sports and golf, and to all those who have joined me in exploring the Pittsburgh food scene.

Last but not least, special thanks to my parents and my sister for their many years of love and encouragement. I would not be here today without your support.

List of Figures

1.1 Feature-rich vs. feature-poor. 4
1.2 Ambiguous object viewpoints. 5

2.1 Discriminative SIFT matching on feature-poor objects. 12
2.2 Levels of feature richness. 13
2.3 Invariant methods consider properties of shape primitives that are invariant across viewpoint. 15
2.4 Non-invariant methods create a template for each viewpoint of the object. . . 17
2.5 Sensitivity of edge extraction. 18
2.6 Limitations of HOG. 18

3.1 Example of keypoint locations with similar local appearance. 24
3.2 Examples where large portions of an object are repeated. 24
3.3 Quantization Framework. 25
3.4 Discriminative Hierarchical Matching (DHM). 26
3.5 Example of quantized matching. 27
3.6 Example of tomato soup can recognized at a viewpoint significantly different from the closest model view. 28
3.7 3D model of the tomato soup can from 25 images. 30
3.8 Example detections on CMU10_3D using the Quantization Framework. 31
3.9 Effect of different quantization bandwidths on the Average Precision. 32
3.10 Averaged Precision/Recall plots on CMU10_3D using the Quantization Framework. 35
3.11 Examples of misdetection with Collet *et al.* and SA+Q. 36

4.1 Example of shape matching under heavy occlusion using Gradient Networks. 42
4.2 Failure of sparse edge point methods in clutter. 42
4.3 Gradient Network (GN). 44
4.4 Illustration of shape matching algorithm using GN. 45
4.5 Computation of the color potential. 46
4.6 Shape similarity for different number of message passing iterations. 47

4.7 Probability calibration of the shape similarity using the Extreme Value Theory. 48
4.8 Results of shape matching using GN. 53
4.9 Typical example of when GN performs better than rL2D, OCM and HOG. . 53
4.10 FPPI/DR results using GN for single view on CMU_KO8. 55
4.11 FPPI/DR results using GN for multiple view on CMU_KO8. 56
4.12 Response maps for detecting a cup. 57
4.13 Detection rate under different occlusion levels using GN. 57
4.14 False positives of GN. 58
4.15 GN and junctions 58

5.1 Example of false positives when using shape only. 60
5.2 Boundary and Region Templates (BaRT). 60
5.3 Effect of strong gradients on the object boundary. 61
5.4 Grid optimization for region template. 62
5.5 Modification of HOG to handle uniform regions. 63
5.6 Comparison of HOG with and without uniform handling. 63
5.7 Example detections using BaRT in cluttered household environments. 65
5.8 Precision/Recall curves using BaRT for single view of CMU_KO8. 67
5.9 Precision/Recall curves using BaRT for multiple view of CMU_KO8. 68
5.10 Example false positives of BaRT. 69

6.1 Example detections under severe occlusions after occlusion reasoning. 74
6.2 Occlusion model under arbitrary viewpoint. 76
6.3 Computation of occlusion prior. 78
6.4 Example of occlusion distributions. 79
6.5 The occlusion prior and conditional distribution under different camera viewpoints for a pitcher. 80
6.6 The occlusion prior and conditional distribution under different camera viewpoints for a cap. 80
6.7 Using an arbitrary object silhouette. 81
6.8 Examples of occlusion hypotheses. 82
6.9 Distribution of height, length and width of occluders in household environments. 84
6.10 Validity of occlusion model. 85
6.11 Example detection results under severe occlusions in cluttered household environments. 85
6.12 Precision/Recall plots using OPP and OCLP for single view on CMU_KO8 for all templates. 88
6.13 Precision/Recall plots using OPP and OCLP for multiple views on CMU_KO8 for all templates. 89

6.14 Performance under different occlusion levels on CMU_KO8 using OPP and OCLP. 90
6.15 A typical case where OCLP performs better than OPP. 91
6.16 Typical failure cases of OCLP. 91
6.17 Visualization of occlusion distributions learned using different amounts of data. 92
6.18 Learning the occlusion prior and conditional likelihood from data. 92
6.19 Parameter sensitivity of OPP and OCLP. 93

7.1 Example of OCLP's sensitivity to misclassifications. 96
7.2 Example occlusion predictions using Occlusion Efficient Subwindow Search (OESS). 97
7.3 Illustration of the upper bound on occlusion quality. 99
7.4 Example detections and occlusion reasoning using OESS on CMU_KO8. . . . 101
7.5 Occlusion prediction performance using OESS on CMU_KO8. 103
7.6 Example failure cases using OESS for occlusion segmentation. 103
7.7 Precision/Recall plots using OESS on CMU_KO8 for single view for all templates. 105
7.8 Precision/Recall plots using OESS on CMU_KO8 for multiple views for all templates. 106
7.9 Images containing the hardest true positives to detect. 107
7.10 Highest scoring false positives. 108

9.1 Fine-grained discrimination of mugs. 114

A.1 CMU Grocery Dataset (CMU10_3D). 122
A.2 CMU Kitchen Occlusion Dataset (CMU_KO8). 123

B.1 Detailed illustration of how to compute the area A_{V_i,O_c}. 127
B.2 Detailed illustration of how to compute the area A_{V_i,V_j,O_c}. 128

List of Tables

3.1 Average Precision on CMU10_3D using the Quantization Framework. 34
3.2 Detections (%) within 5 cm and 22.5 degrees of the true pose on CMU10_3D. 37
3.3 Translation error in cm (top) and rotation error in degrees (bottom) for the correct detections on CMU10_3D. 37

4.1 F-measure characterizing the shape matching using GN on CMU_KO8. 52
4.2 Detection rate using GN at 1.0 FPPI on CMU_KO8. 52
4.3 Average detection rate using GN at 1.0 FPPI on CMU_KO8. 54

5.1 Object detection using BaRT: Mean Average Precision. 66

6.1 Object detection performance using OPP and OCLP: Mean Average Precision. 87

7.1 OESS vs. brute force speed. 99
7.2 Object detection using OESS on CMU_KO8: Mean Average Precision. 104

Part I

Introduction

Chapter 1

Overview

As humans, we interact with objects perpetually throughout the day, from pouring coffee into a mug in the morning to switching on the TV with a remote at night. We often take for granted the seemingly simple task of identifying the objects around us. For example, we can find a mug on a cluttered kitchen counter or a TV remote on the coffee table without a second thought. Even when objects are severely occluded and only a small portion of an object is visible, we are often still able to identify them with high accuracy. However for computers, this is not the case. Even though researchers have been working on the problem of object detection for decades, detecting objects automatically in natural scenes still proves to be a significant challenge.

At the beginning of the millennium, the development of SIFT [70] and other discriminative keypoint features [79] produced significant advances in recognizing objects which have unique patterns and textures. These objects, such as cereal boxes and paintings (Figure 1.1a), have many corners within the object interior where keypoints can be stably and repeatably extracted. Researchers have exploited the unique appearance of these objects by creating highly discriminative descriptors to represent the local statistics around each keypoint. By matching only a small number of keypoints, these systems [22, 48] can recognize and estimate the 6D pose of an object given only a single image. In this thesis, we refer to objects where keypoint features can be reliably extracted as *feature-rich*.

While matching discriminative features works for the majority of feature-rich objects, many man-made objects contain repeated patterns from logos, text and printed graphics. Features extracted from these regions will have similar descriptors, and in the extreme case, they may be exactly the same. Current algorithms discard ambiguous matches because they are assumed to arise from background clutter. However, ignoring these matches often results in insufficient correspondences for reliable recognition.

Feature-rich objects also only comprise a small portion of the objects that we interact with everyday. In daily living environments, for example, many objects lack texture [94] such as cups, glasses and staplers (Figure 1.1b). These objects are of particular

(a) Feature-rich (b) Feature-poor

Figure 1.1: Feature-rich vs. feature poor. (a) The cereal box and painting are examples of feature-rich objects where many keypoints can be stably extracted. (b) The cup and stapler are examples of feature-poor objects where keypoint features are unreliable.

importance for personal household robotic systems such as HERB [106] and PR2 [20] which need to manipulate them, as well as for applications in augmented reality and visual search (e.g., Google Goggles [1]). For these objects, keypoint features can not be extracted reliably and we refer to them as *feature-poor*. Research has shown that feature-poor objects are among the most difficult to recognize [3, 44, 115].

So what makes detecting feature-poor objects particularly challenging? Without stable keypoints, feature-poor objects are primarily defined by their contour structure and shape. While keypoint features are highly discriminative, contour fragments are locally just smooth varying curves with very little distinctiveness. A straight edge, for example, can be found anywhere in the image. To increase the distinctiveness, longer contour fragments are often used, but extracting them reliably is in itself very difficult.

The problem of detecting feature-poor objects is further exacerbated by occlusions, which are common in natural scenes with large amounts of clutter. Occlusions break up long contour fragments and in general reduce the overall information available for object detection. Since many contour fragments can easily align locally to background clutter, this results in false detections with higher score than true detections that are occluded. Many shape-based systems work well when objects are entirely visible, but degrade rapidly in the presence of occlusions.

1.1 Detecting Object Instances

In this thesis, we investigate how to address ambiguity for detecting object instances in scenes with severe clutter and occlusions. But how do we define an object instance? Ideally, this would refer to the exact same object, but does a small hairline scratch on an object make it a different instance? How about many scratches? In addition,

Figure 1.2: Ambiguous object viewpoints. From the side view, these two soup cans are clearly different. However from the top view, they look exactly the same.

the majority of man-made objects in the world are not unique. Thousands of copies of a particular model of laptop or cell phone exist in the world. Requiring an object instance to be the exact same object is too restrictive and unrealistic to achieve even for humans. Thus, we define an object instance to be a set of objects encompassing all those manufactured in the same way and all those visually indistinguishable to a human being. By this definition, for example, all Apple iPhone 5s are the same object instance.

The problem of detecting object instances is significantly different from the problem of detecting object categories. The primary difference between the two lies in the amount and type of variation that needs to be handled. For object instances, there is in principle no physical variation, but only variations in the image domain due to changes in viewpoint and lighting. On the other hand, for categories, the goal is to capture the variations within human defined groups which can be from physical differences as well as variations in the image domain.

So, if all we have to address are viewpoint and lighting variations when detecting object instances, can we be sure that an object instance has been correctly detected? Regrettably, the answer is that even for humans, there are cases where we can not be a hundred percent certain. Many objects in the real world have very similar physical appearance. Under certain camera viewpoints, two objects can look identical as shown in Figure 1.2. If an object is occluded, the level of ambiguity increases further. Given only a single image, the best we can do is generate a reasonable hypothesis and a confidence measure of how likely it corresponds to the object of interest. On a physical system such as a robot, the hypothesis may be verified and disambiguated by moving the robot or manipulating the object to obtain more viewpoints. In this thesis, we focus on instance detection from a single image.

1.2 Summary of Contributions

The main contribution of this thesis are as follows:

- Addressing discriminative feature matching in the presence of ambiguous features.
  - Discriminative Hierarchical Matching (DHM) - framework for preserving feature ambiguity at the matching stage until hypothesis testing to address similar features.
  - Simulated Affine features - additional features extracted from affine transformed model images to detect objects with viewpoints significantly different from the model images.
- More robust representation of feature-poor objects.
  - Gradient Networks - method for explicit shape matching on low-level gradients without extracting edges.
  - Boundary and Region Templates - framework to capture explicit boundary and region information.
- Methods to increase robustness of object detection under severe occlusions.
  - Occlusion reasoning under arbitrary viewpoint - analytical model which incorporates environmental statistics with simple 3D reasoning.
  - Occlusion Efficient Subwindow Search (OESS) - method to coherently reason probabilistically about occlusions on boundary and region together.
- Datasets (Appendix A)
  - CMU Grocery Dataset (CMU10_3D) of 10 feature-rich grocery items in cluttered household scenes.
  - CMU Kitchen Occlusion Dataset (CMU_KO8) of 8 feature-poor kitchen objects in cluttered scenes with severe occlusions.

1.3 Roadmap

In the first part of the thesis, we begin by addressing feature ambiguity and similar features when using discriminative keypoints for detection (Chapter 3). In the second part, we discuss methods for better representing objects when discriminative features can not be stably extracted. We introduce our Gradient Networks (Chapter 4) method for matching shape explicitly without extracting contours. Then we present our Boundary and Region Template method (Chapter 5) which captures both the explicit shape of the

object and the interior appearance. In the next part, we introduce methods to exploit the structure of occlusions for increasing the robustness of object detection in cluttered scenes. We present a model for reasoning about occlusions under arbitrary viewpoint (Chapter 6) and a formulation of occlusion reasoning as efficient search (Chapter 7). Finally, we conclude the thesis with a discussion of our contributions and future directions.

1.4 Publication Note

The publications which comprise this thesis are listed below:

- E. Hsiao, A. Collet and M. Hebert. Making specific features less discriminative to improve point-based 3D object recognition. In *Proceedings of IEEE Conference on Computer Vision (CVPR)*, 2010. [48]
- E. Hsiao and M. Hebert. Occlusion reasoning for object detection under arbitrary viewpoint. In *Proceedings of IEEE Conference on Computer Vision (CVPR)*, 2012. [49]
- E. Hsiao and M. Hebert. Shape-based instance detection under arbitrary viewpoint. Book chapter in *Shape Perception in Human and Computer Vision: An Interdisciplinary Perspective*, Springer, 2013. [51]
- E. Hsiao and M. Hebert. Gradient networks: Explicit shape matching without extracting edges. In *Proceedings of AAAI Conference on Artificial Intelligence (AAAI)*, 2013. [50]

Chapter 2

Background

Significant research has gone into recognizing object instances from a single image. In this chapter, we provide a general overview of the approaches in this area. We begin by reviewing methods which use discriminative features and discuss the types of objects that they work on. Next, we review the literature on shape-based methods which are commonly used when keypoint features cannot be reliably extracted. Finally, we summarize occlusion reasoning approaches that have been used in the context of object detection.

2.1 Recognition Using Discriminative Features

A popular approach in the literature for recognizing object instances is to use discriminative keypoint features. The general paradigm of these approaches is to extract features on a set of model images and then match them discriminatively to those extracted on a test image. Extensive research has been dedicated to designing repeatable methods for extracting keypoints, making them scale and rotation invariant, and creating highly discriminative descriptors. Given a set of candidate feature matches, geometric constraints are typically enforced to recognize the object and estimate its pose. In the following, we summarize the research for each of these components and empirically evaluate the types of objects discriminative features work on.

2.1.1 Keypoint Detection

The goal of keypoint detection is to extract locations on the object which are repeatable, well localized and informative. On smooth surfaces and contours, localization is difficult and the local neighborhood in these regions is usually not very distinctive. Thus, the majority of keypoint detection methods focus on finding corners. A comprehensive review of historical work is presented by Mikolajczyk and Schmid [80].

One of the first interest point detection techniques was proposed by Schmid and Mohr [102] and used local extremas of the Harris [42] operator. While the Harris operator is invariant to image rotations, the scale invariance is obtained by extracting features at multiple scales. The Hessian operator [80] has also been used similarly and has been shown to be more robust in certain experiments. To find stable keypoints more efficiently across scale, Lowe [70] used local extremas of Difference-of-Gaussian (DOG) filtered images. Mikolayczyk and Schmid in [76] detect keypoints at multiple scales and then select characteristic points using scale space selection [67]. More recently, FAST [96] was proposed to use a segment test heuristic and machine learning to detect corners at frame rate.

While most keypoint detectors find corners, others try to find stable regions. The Maximally Stable Extremal Regions (MSER) [74] method detects stable blobs by finding regions which do not change over a large range of binarization thresholds. The method has been shown to outperform many corner-based approaches [80]. While MSER works well when objects have many small regions, they are sensitive to occlusions when objects have large regions.

Most keypoint detection methods are scale and rotation invariant. However, when performing matching across different viewpoints, this is often not enough. More recent approaches try to extend the range that keypoints can be matched by being invariant to affine transformations as well. Tuytelaars and Van Gool [117] fit affine invariant regions to image intensities. The Harris-Affine and Hessian-Affine [77] detectors adapt the Harris and Hessian detectors by fitting an affine covariant region iteratively around each keypoint based on the second moment matrix. The ASIFT [82] method extracts features from affine transformations of both model and test images and considers matching between all pairs of transformed images. Incorporating affine invariance has been shown to significantly increase the robustness of feature matching.

2.1.2 Feature Description

Once a keypoint is detected, local invariance to scale, rotation, and affine transformations is obtained by normalizing the neighborhood region. SIFT [70] estimates the dominant orientation in a local neighborhood of the keypoint and warps the surrounding patch to a canonical orientation. Harris-Affine and Hessian-Affine [77] methods warp the estimated affine elliptical region into a circle.

Given a normalized region, the common philosophy for object instance detection is to design descriptors that are as unique as possible. The SIFT feature histograms the gradients in the region using a 4×4 grid with 8 gradient orientation bins and weights their contribution using their gradient magnitude. This descriptor has shown to be extremely discriminative, especially when combined with the ratio test [70]. The

Gradient Location and Orientation Histogram (GLOH) [79] extends SIFT by using a log-polar binning instead of a square grid and is shown to be more robust. To speed up the computation time, the Speeded Up Robust Features (SURF) [9] use Haar wavelets and Integral Images [118] to compute derivatives quickly. Other extensions of SIFT, such as DAISY [113] have been proposed for dense matching. For real time applications, BRIEF [17] uses a sequence of binary tests to describe a keypoint. A comparison of popular feature descriptors is presented by Mikolajczyk and Schmid [79].

2.1.3 Feature Matching

Given a set of model features and a set of test features, the general paradigm is to generate one-to-one correspondences. The simplest approach is to take the nearest neighbor using the Euclidean distance, however, some descriptors are close to everything, especially in high dimensional space [93] making this approach very brittle. Lowe proposed the ratio test [70] which considers the ratio of the distance to the first nearest neighbor with the distance to the second nearest neighbor. If the ratio is less than a certain threshold, it is a good match.

Other approaches use machine learning techniques to match the features. Lepetit and Fua used Random Decision Trees [64] to classify keypoints. The FERNS [86] approach formulates feature point recognition in the Naive Bayes classification framework. Nister and Stwenius [84] build a vocabulary tree for efficient keypoint classification.

2.1.4 Feature-richness of Objects

Feature-based methods work really well on book covers, paintings and cereal boxes, but are unable to obtain good matches with smooth objects as shown in Figure 2.1. Yet, for many of these smooth objects, it is not because there is a lack of keypoints. From the figure, there are many keypoints on the thermos. So what is the difference between keypoints on book covers and those on smooth objects?

The main difference is that the majority of keypoints on smooth objects, such as the thermos, are on the boundary at the intersection of background clutter with the object or at corners due to specularities and lighting effects. These types of keypoints are not repeatable, since they will not fire at the same locations if the background or lighting is slightly different. The only repeatable keypoints are corners of the object.

However, these corners are also not very informative. These corners are on the object boundary, resulting in a large portion of the descriptor capturing the background clutter. Since descriptors are designed to be highly discriminative, the same feature extracted on different backgrounds will have very different values. Thus, they will not be matched unless the background statistics around the keypoint are very similar as well, which is often not the case.

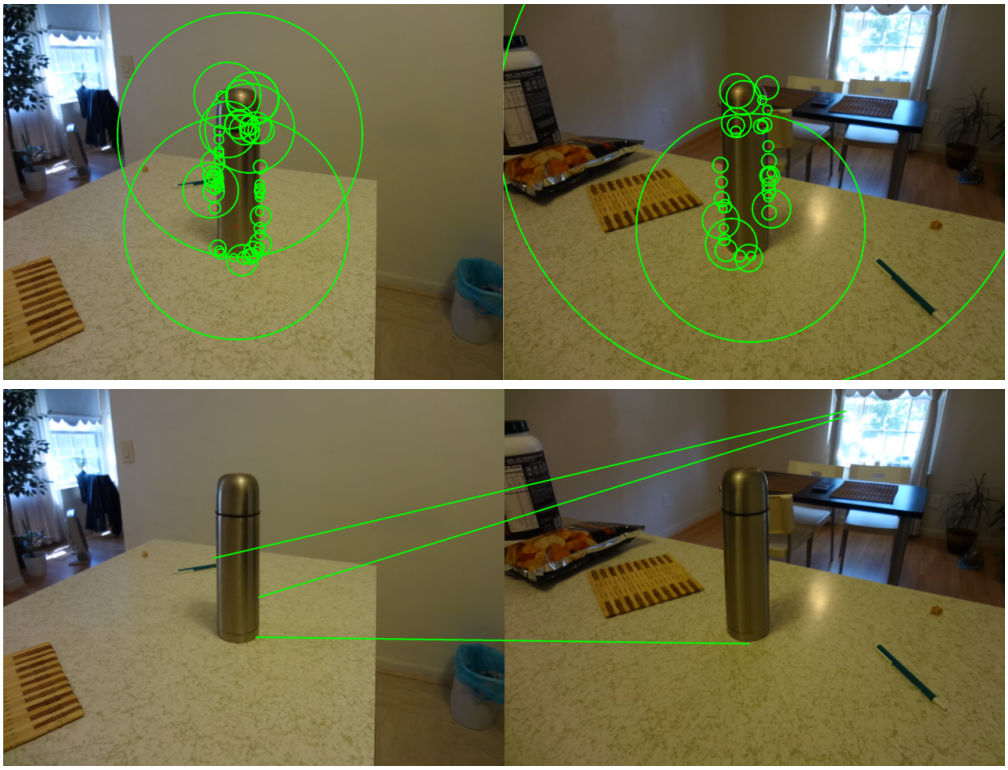


Figure 2.1: Discriminative SIFT matching on feature-poor objects. (top) SIFT keypoints extracted on a thermos for two images. (bottom) SIFT matches between the images using the ratio test. Even though both images contain many features on the thermos, they still cannot be matched.

Stochastic textures are also a problem for discriminative features. These textures have the same statistical properties between objects, but do not have the same exact appearance. Many objects have stochastic textures such as a wooden bowl or a fur coat. While many keypoints can be extracted on these textures, these features are usually not useful. For the same exact model of the object, the textures will not be exactly the same for two physical objects, leading to different descriptors. In addition, the keypoints are often difficult to localize in these regions.

However, feature-based matching can be used on texture instances. A texture instance is the same exact physical object with the same stochastic texture. Since the appearance of the texture does not change, the descriptors extracted do not change as well, and thus can be matched.

This leads to a simple metric which we empirically show is a good indicator of which objects feature-based methods will work on. Given the above observations, we define an *informative keypoint* to be one where the local neighborhood region used to compute its descriptor is contained entirely within the object. These keypoints can be stably extracted and their descriptors are not affected by background clutter. The number of

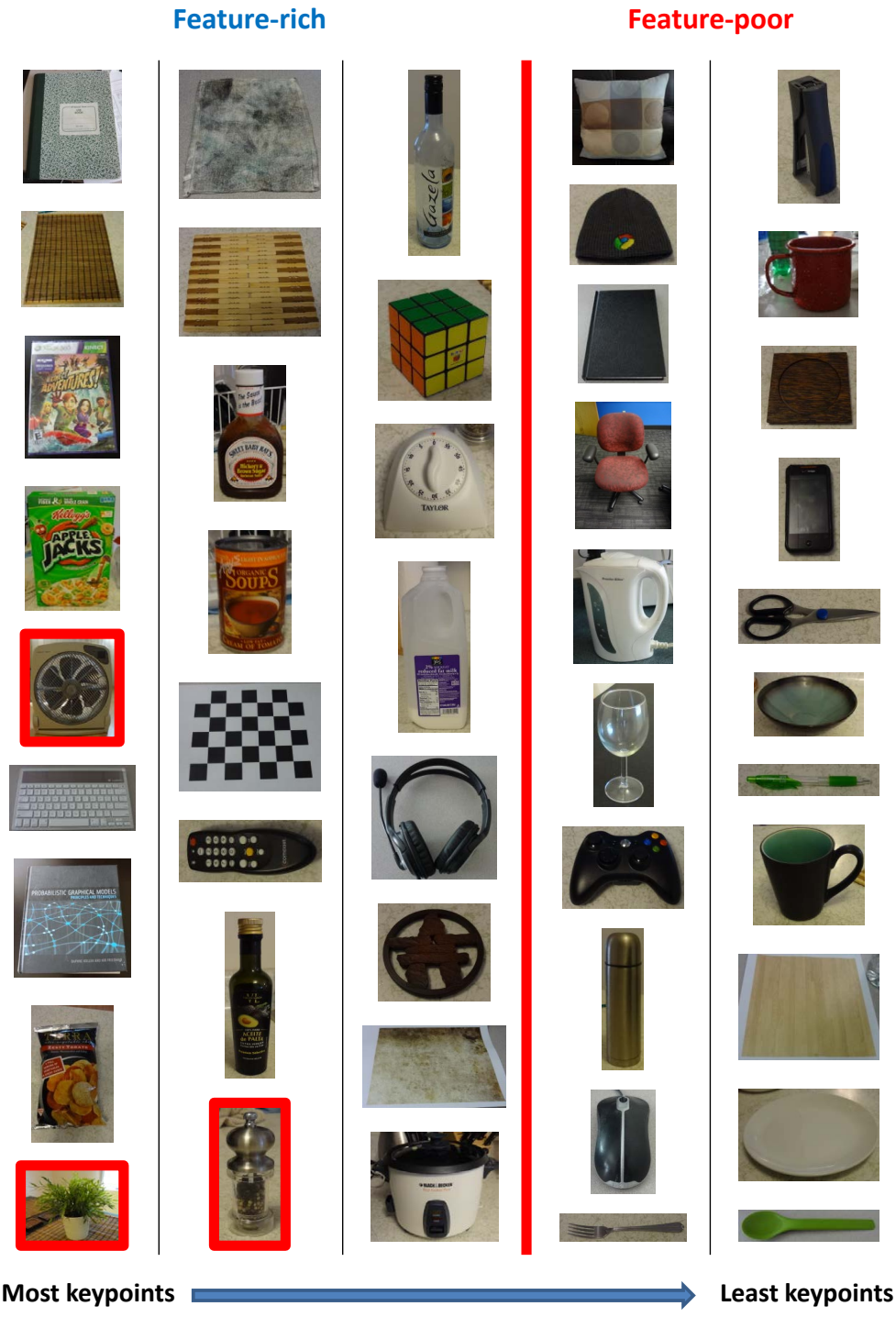


Figure 2.2: Levels of feature richness. From left to right, and top to bottom, we show the ordering of objects with the most keypoints that are contained entirely within the object interior. Those objects to the left of the red line can be matched reliably, while those on the right can not. The outliers with red borders can not be matched.

informative keypoints thus corresponds to the feature-richness of the object and is a good indicator of when approaches such as SIFT will work.

Figure 2.2 shows the ranking of feature-richness using this simple metric. We performed a simple experiment by taking three images of 46 common household objects on different backgrounds. If more than five features were matched for all pairs of object images, we considered recognition to work. The red line divides the feature-poor objects from the feature-rich. We can see that there is a fairly clear boundary with only a few outliers (i.e., fan, plant, pepper shaker) shown by the red boxes.

Each of these outliers contains an interior region which appears significantly different with a slightly change in viewpoint. These areas are essentially a stochastic texture, and thus are difficult to match even though there are many keypoints. From a single image, it is impossible to determine if a region is a stochastic texture and our metric is the best that we can do. Given multiple images, the number of correct matches between all pairs of images can be used.

2.2 Shape Matching

In the previous section, we described keypoint-based approaches for detecting feature-rich objects. While these approaches work well for many objects, there exist many feature-poor objects (Figure 2.2) where they do not work. These feature-poor objects are primarily defined by their contour structure and approaches for recognizing them focus on shape matching [31, 44, 53, 115]. However, many object shapes are very simple, comprising of only a small number of curves and junctions. Even when considering a single viewpoint, these curves and junctions are often locally ambiguous as they can be observed on many different objects. The collection of curves and junctions in a global configuration defines the shape and is what makes it more discriminative.

Object instance detection, however, requires detecting objects under arbitrary viewpoint. Introducing viewpoint variations further compounds shape ambiguity as the additional curve variations can match more background clutter. Much research has gone into representing shape variation across viewpoint. In general, current models can be divided roughly into two main paradigms: *invariant* and *non-invariant* models. On one hand, *invariant* models create a unified object representation across viewpoint by explicitly modeling the structural relationships of high level shape primitives (e.g., curves and lines). On the other hand, *non-invariant* models use view-based templates and capture viewpoint variations by sampling the view space and matching each template independently. In the following, we summarize these two classes of shape matching approaches.

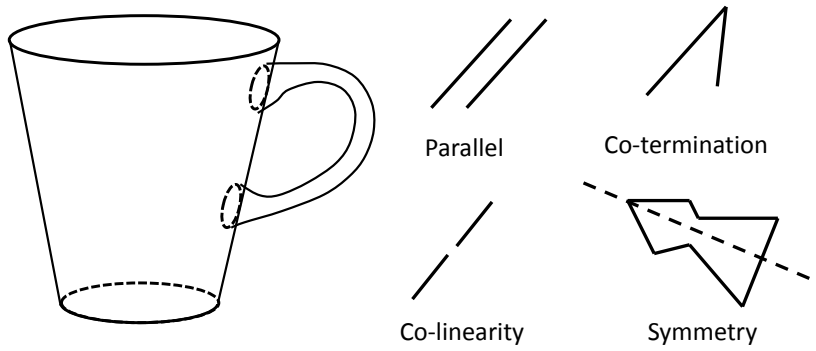


Figure 2.3: Invariant methods consider properties of shape primitives that are invariant across viewpoint. Common invariant properties that are used are parallelism, co-termination, co-linearity and symmetry.

2.2.1 Invariant methods

Invariant methods are based on representing structural relationships between view-invariant shape primitives [14, 41]. Typically, these methods represent an object in 3D and reduce the problem of object detection to generating correspondences between a 2D image and a 3D model. To facilitate generating these correspondences, significant work has gone into designing shape primitives [13] that can be differentiated and detected solely from their perceptual properties in 2D while being relatively independent of viewing direction. Research in perceptual organization [68] and non-accidental properties (NAPs) [124] have demonstrated that certain properties of edges in 2D are invariant across viewpoint and unlikely to be produced by accidental alignments of viewpoint and image features. These properties provide a way to group edges into shape primitives and are used to distinguish them from each other and from the background. Example of such properties are collinearity, symmetry, parallelism and co-termination as illustrated in Figure 2.3. After generating candidate correspondences between 2D image and 3D model using these properties, the position and pose of the object can then be simultaneously computed.

In earlier approaches, 3D CAD models [32, 53, 127] were extensively studied for view-invariant object recognition. For simple, polyhedral objects, CAD models consist of lines. However for complex, non-polyhedral objects, curves, surfaces and volumetric models [55] are used. In general, obtaining a compact representation of arbitrary 3D surfaces for recognition is very challenging. Biederman's Recognition-by-Components (RBC) [13] method decomposes objects into simple geometric primitives (e.g., blocks and cylinders) called *geons*. By using geons, structural relationships based on NAPs can be formulated for view-invariant detection.

Given geometric constraints from NAPs and an object model, the recognition prob-

lem reduces to determining if there exists a valid object transformation that aligns the model features with the image features. This correspondence problem is classically formulated as search, and approaches such as interpretation trees [40, 41], Generalized Hough Transforms [41] and alignment [18, 52] are used.

Interpretation trees [40, 41] consider correspondences as nodes in a tree and sequentially identify nodes such that the feature correspondences are consistent with the geometric constraints. If a node does not satisfy all the geometric constraints, the subtree below that node is abandoned. Generalized Hough Transforms (GHT) [41], on the other hand, cluster evidence using a discretized pose space. Each pair of model and image feature votes for all possible transformations that would align them together. Geometric constraints are combined with the voting scheme to restrict the search of feasible transformations. Finally, alignment-based techniques [18, 52] start with just enough correspondences to estimate a *hypothesis* transformation. *Verification* is then used to search for additional model features satisfying the geometric constraints. The hypothesis with the most consistent interpretation is chosen.

While CAD models and geons have been shown to work well in a number of scenarios, automatically learning 3D models is a considerable challenge [15, 40]. In addition, geons are unable to approximate many complex objects. To address these issues, recent approaches [62, 87] try to learn view-invariant features and non-accidental properties directly from 2D data. A common paradigm is to align and cluster primitives that have similar appearance across viewpoint. For example, the Implicit Shape Model (ISM) [62] considers images patches as primitives and uses Hough voting for recognition. To determine view-invariant features, images patches from all viewpoints of the object are clustered. Each cluster corresponds to a locally view-invariant patch and is associated with a probabilistic set of object centers. A match to a cluster casts a probabilistic vote for its corresponding object positions.

The critical issue with allowing local deformations is that it is difficult to enforce global consistency of deformations without storing the constraints for each viewpoint individually. However, if the constraints are defined individually for each viewpoint, the view-invariance is lost and the approach is equivalent to matching each view independently (i.e., non-invariant).

Another common issue with invariant approaches is that they rely on stable extraction of shape primitives. This is a significant limitation since reliable curve extraction and grouping [68] still proves to be a considerable challenge. While there has been significant development in object boundary detection [4, 26], no single boundary detector is able to extract all relevant curves. The Global Probability of Boundary (gPb) detector [4], which is designed to ignore stochastic textures, often confuses interior contours with stochastic texture. These interior edges provide distinctiveness that is necessary for

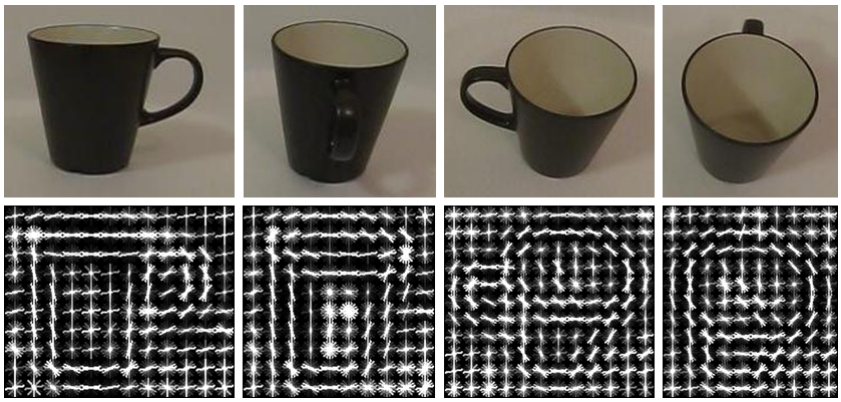


Figure 2.4: Non-invariant methods create a template for each viewpoint of the object.

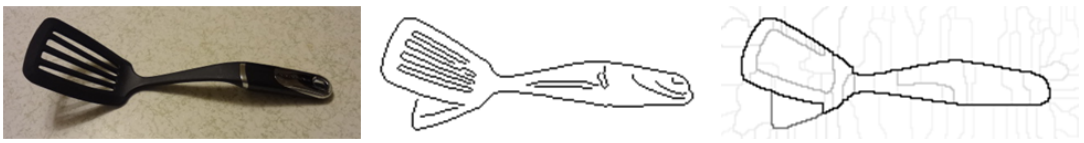
recognizing specific objects. Due to the challenges of creating 3D models, extracting shape primitives and learning geometric constraints from data, many recent approaches have moved away from using invariant shape primitives. In the next section, we discuss how non-invariant, view-based methods are able to address some of the above issues.

2.2.2 Non-invariant (view-based) methods

Non-invariant methods represent an object under multiple viewpoints by creating a "view-based" template [90] for each object view (Figure 2.4). Each template captures a specific viewpoint, only allowing slight deformation from noise and minor pose variation. Unlike invariant methods which define geometric constraints between pairs or sets of shape primitives, non-invariant methods directly fix both the local and global shape configurations. To combine the output of view-based templates, the scores from each view are normalized [73, 101] and non-maximal suppression is applied.

Non-invariant methods have a number of benefits over invariant ones. First, using view-based templates bypasses the 3D model generation and allows the algorithm to directly observe the exact projection of the object to be recognized. This has the benefit of not approximating the shape with volumetric primitives (e.g., geons), which can lose fine-grained details needed for recognizing specific objects. Secondly, template matching approaches can operate directly on low-level features and do not require extraction of high-level shape primitives. Finally, many non-invariant approaches achieve recognition performances on par or better than invariant ones, while being relatively simple and efficient to implement. Recent results show that they can be successfully applied to tasks such as robotic manipulation.

A number of methods exist for representing object shape from a single view. These range from using curves and lines [30, 31, 107] to sparse edge features [44, 63] and gradient histograms [23]. Methods which use curves and lines often employ 2D view-invariant



(a) Image (b) Canny edges (c) gPb

Figure 2.5: Sensitivity of edge extraction. Current state-of-the-art methods in boundary detection (gPb [5]) are unable to stably extract interior contours which are essential for recognizing specific objects. Canny, on the other hand, can detect these edges, but will also fire on spurious texture edges.

(a) ambiguous shape (b) background corruption

Figure 2.6: Limitations of HOG. (a) Ambiguous shape; two patches can have the same HOG descriptor. (b) Background corruption; gradients on the background can severely change the HOG descriptor.

techniques, similar to the approaches described in Section 2.2.1, to reduce the number of view samples needed. Interpretation trees [41], Generalized Hough Transforms [41] and alignment techniques [18] which are used for 3D view-invariance are similarly applied to 2D geometric constraints.

While some approaches use 2D view-invariance, others simply brute force match all the possible viewpoints. Lines [31] and contour fragments [85, 104], in the simplest form, are represented by a set of points [11, 44] and Chamfer matching [8] is used to find locations that align well in an edgemap. Local edge orientation is often incorporated [104] in the matching cost to increase robustness to clutter. These methods, however, consider each point independently and do not use edge connectivity. Other approaches capture connectivity by approximating curves as sequences of line segments or splines [128] instead of points. A common issue with these approaches, however, is the difficulty of breaking contours at repeatable locations due to noise in the edgemaps and object occlusions. A further limitation of these approaches is their reliance on stable edge detection (Figure 2.5), which still remains an open area of research [5]. To bypass

edge extraction, other methods represent the shape by using coarse gradient statistics. Histogram of Oriented Gradients (HOG) [23] bins gradient magnitudes into nine orientation bins. These methods, however, only provide a coarse match of shape (Figure 2.6a), losing many fine-grained details needed for instance detection. They are also sensitive to strong background gradients (Figure 2.6b).

An additional criticism of non-invariant methods is that they require a large number of templates to sample the view space. For example, LINE2D [44] requires 2000 templates per object. While this many templates may have resulted in prohibitive computation times in the past, advances in algorithms [44, 45] and processing power have demonstrated that template matching can be done very efficiently (e.g., LINE2D and DOT are able to match objects at 10 frames per second). To increase the scalability, template clustering and branch-and-bound [45] methods are commonly used. In addition, templates are easily scanned in parallel and many can be implemented efficiently on Graphics Processing Units (GPUs) [92].

2.3 Occlusion Reasoning

Occlusions are common in real world scenes and are a major obstacle to robust object detection. For feature-rich objects, discriminative keypoint features can be used to match unique local patterns on the object even under severe occlusions. For feature-poor objects, however, occlusions further increase the shape ambiguity. While many shape matching approaches work really well when objects are entirely visible, their performance decrease rapidly with occlusions. When objects are under heavy occlusions, the score of false positives begin to overwhelm the scores of true detections, resulting in the inability to recognize objects robustly in these scenarios.

In the past, occlusion reasoning for object detection has been extensively studied [41, 89, 108]. Occlusions are commonly modeled as regions that are inconsistent with object statistics. Girshick *et al.* [38] use an occluder part in their grammar model when all parts cannot be placed. Wang *et al.* [120] use the scores of individual HOG filter cells, while Meger *et al.* [75] use depth inconsistency from 3D sensor data to classify occlusions. Local coherency of occlusions are often enforced with a Markov Random Field [33] to reduce noise in these classifications. Li *et al.* [66] use RANSAC to generate a large set of hypotheses and hallucinate points at positions where there is high error. These approaches, however, assume that occlusions can happen randomly on an object. While this is true for some cases, in real world environments, objects are usually occluded by other objects resting on the same surface. It is thus often more likely for the bottom of an object to be occluded than the top of an object [27].

Recently, researchers have attempted to learn the structure of occlusions from data [36,

57]. With enough data, these methods can learn an accurate model of occlusions. However, obtaining a broad sampling of occluder objects is usually difficult, resulting in biases to the occlusions of a particular dataset. This becomes more problematic when considering object detection under arbitrary view [44, 109, 112]. Learning approaches need to learn a new model for each view of an object. This becomes intractable, especially when recent studies [44] have claimed that approximately 2000 views are needed to sample the view space of an object.

In general occlusion reasoning has primarily been used to separate regions which belong to the object from those that do not. This allows the detector to ignore occluded regions which would otherwise corrupt the overall score. Girshick *et al.* [38] uses an occluder part to ignore regions that are classified as occlusions, while Wang *et al.* [120] turn off deformable parts in these regions. Other methods use occlusions to determine the depth ordering [119, 125] of detections and remove parts that have already been explained.

Part II

Ambiguity of Discriminative Features

Chapter 3

Addressing Ambiguous Features

Discriminative keypoint features work really well for object detection when the features on the object are unique. Current state-of-the-art methods for instance detection with keypoint features employ 3D models, and their general paradigm [21, 39, 97] is to first generate correspondences between image features and model features, and then to use the 3D positions associated with the matched model features to estimate the pose of an object by enforcing geometric constraints. Given a set of perfect correspondences between 3D points and 2D projections, the problem of determining the pose of a calibrated camera has been extensively studied [2, 65]. The main problem that remains unsolved in 3D object recognition is the problem of automatically generating enough reliable correspondences. Even though techniques like RANSAC are able to deal with incorrect correspondences, often there are just not enough correspondences to begin with. If enough correspondences are provided, recovering the pose is essentially solved and we show that the various methods for recovering pose have very similar recognition performance.

One main issue that arises with manmade objects is that there are inevitably locations on the object that have similar local appearances. Features extracted from these locations have similar descriptors, and in the extreme case, the descriptors may be exactly the same. Most current algorithms perform matching discriminatively; ambiguous matches are often discarded because they are assumed to arise from background clutter. This is exemplified by the ratio test [70], which compares the distance to the closest neighbor with the distance to the second closest neighbor. Discriminative matching prevents features with similar descriptors from being matched, even though these features contain rich information about the pose of the object. The presence of similar features is an inherent issue in matching and no amount of tuning parameters or design of local features can circumvent this problem. Figure 3.1 shows examples where local patches around keypoints have very similar appearance, and Figure 3.2 shows objects where large portions of the object are repeated due to logos and images.

Figure 3.1: Example of keypoint locations with similar local appearance. Text regions on man-made objects often look very similar.

Figure 3.2: Examples where large portions of an object are repeated. (left) Two sides of the orange juice carton are almost exactly the same. (right) The logo of the cereal box is repeated at different scales.

In this chapter, we argue that matching is more robust if we do not commit initially to specific point-to-point correspondences. Instead, if a match is ambiguous, we claim that the image feature should be associated with a set of possible locations on the model, retaining the ambiguity of the correspondence until hypothesis testing. Figure 3.3 illustrates our proposed framework. Given a candidate pose of an object, the correspondence ambiguity can be resolved as the one which best fits the hypothesized pose. Finally, the candidate pose with the greatest evidence after considering multiple hypotheses is chosen as the pose of the object.

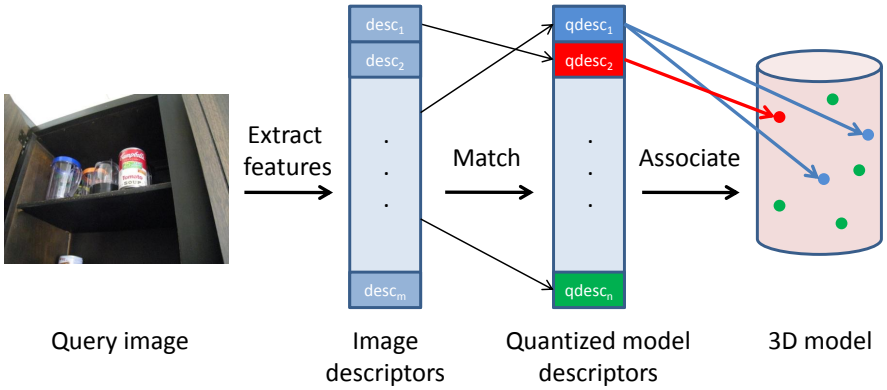


Figure 3.3: Quantization Framework. Features are extracted from a query image and then matched to a set of quantized model descriptors. Each quantized model descriptor is associated with all of its possible locations on the 3D model, which allows similar features to be matched.

3.1 Quantization Framework

We propose to maintain feature ambiguity by quantizing the features on a model. Each quantized feature is associated with a descriptor and all of its possible model locations. These quantized features are still matched discriminatively, but the quantization allows us to associate a feature on a query image with multiple locations on a model. Because retaining feature ambiguity increases the potential number of outliers, we demonstrate an efficient way to handle these additional correspondences.

Vector quantization of features has been used widely in the computer vision literature for categorization tasks such as scene recognition [95] and object categorization [123]. Many of the algorithms used for these tasks fall in the realm of the Bag of Words approach, where a dictionary of visual features is learned through clustering and new images are categorized by comparing histograms of quantized visual words. In these cases, quantization is used as a way to generalize and be robust to intra-category variations.

Most related to our work are methods that employ geometric reasoning on visual words [19, 112] for image retrieval and category recognition. However, for the task of specific 3D object recognition, the prevailing view is to use highly discriminative features. As a result, multiple features with similar appearance on a model are rarely matched.

We claim that these similar features are essential to obtain reliable 3D object recognition. We introduce ambiguity into the matching process by quantizing the model features and associating each quantized descriptor with potentially multiple locations on the model. When an image feature is matched to a quantized feature, it is associated with all the possible locations of that feature. During hypothesis testing, the most likely correspondence given the current pose can be then be determined. Our framework

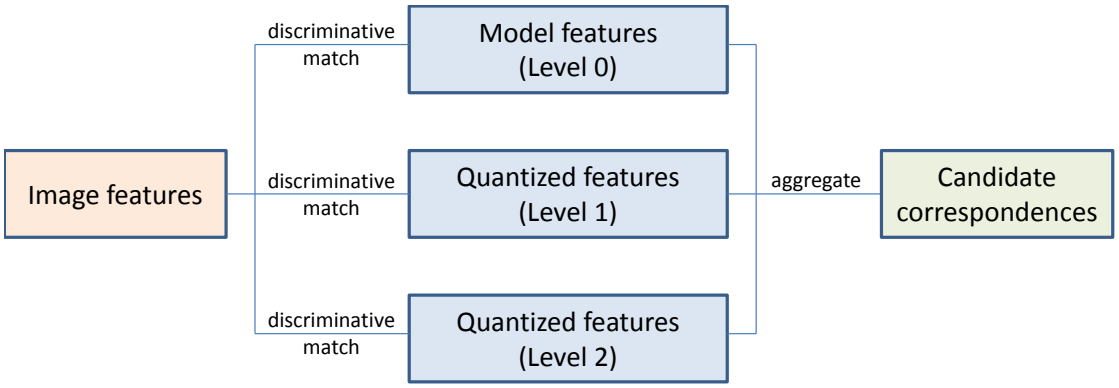


Figure 3.4: Discriminative Hierarchical Matching (DHM). We preserve ambiguity by matching the features at multiple levels of quantization and aggregating them to obtain the candidate correspondences.

allows us to choose the most likely hypothesis given what we have seen, and combines both ambiguous and unique features in a unified framework.

3.1.1 Hierarchical mean-shift quantization

In general, it is very difficult to choose the number of feature clusters *a priori* as different models have different number of features and degree of feature similarity. We choose the mean shift algorithm because it clusters features based on the similarity of the descriptors in feature space. The bandwidth parameter of mean shift is a rough indication of the desired intra-cluster variation and is more relevant to set than the number of clusters.

In our implementation, we use a dual-bandwidth approach where features are quantized in a hierarchical manner [84] using two levels of mean shift with bandwidths r_1 and r_2, such that $r_1 < r_2$. Clustering in this way allows matching to be more robust to the distribution of descriptors in feature space. Our quantization scheme results in three levels of quantized features, where the finest level $l = 0$ corresponds to the original features. Each quantized feature q_i^l at level l is then associated with a set of 3D positions on the model corresponding to all the features in that cluster.

3.1.2 Discriminative hierarchical matching (DHM)

For the task of image retrieval, a common technique is hierarchical matching on vocabulary trees [84] which assigns a visual word to every image feature. However, for the task of object recognition, most image features arise from background clutter. Assigning a visual word to every image feature can increase the number of outlier correspondences by orders of magnitude, making RANSAC intractable. We propose to perform discriminative matching on each level of the hierarchy to limit matches to background clutter.

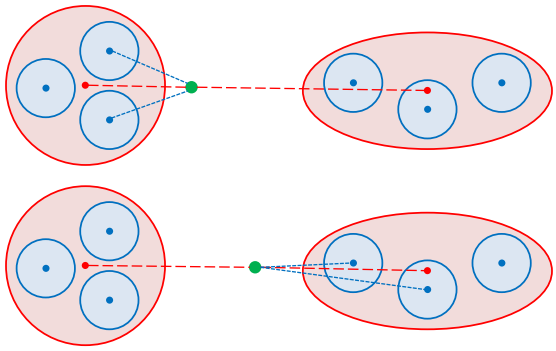


Figure 3.5: Example of quantized matching where (top) a query feature in green matches at a coarser level, but does not match at a finer level by the ratio test and on (bottom) the vice versa is true.

Candidate correspondences are obtained by independently matching the image features with each of the three levels of features. A feature is matched on a particular level if it satisfies the ratio test within that level. The matched image feature is then associated with all the possible 3D locations of its corresponding quantized model feature. The final set of correspondences is obtained by aggregating all candidate correspondences at all levels, removing any duplicate point correspondences.

Figure 3.5 illustrates the reason for our choice of hierarchical clustering. In Figure 3.5 (top), the query feature in green is equidistant to the centers of the two fine clusters (blue), but it is significantly closer to the coarse cluster (red) on the left than the coarse cluster on the right. At this stage, it is impossible to disambiguate the correspondences in the two fine clusters, so the quantized matching returns all the candidate locations of the coarse cluster on the left for later processing. Conversely in Figure 3.5 (bottom), the query feature is equidistant to the centers of the two coarse clusters, but will match at the fine cluster level. If there were only one level of clustering, one of these two situations would result in no correspondence.

3.1.3 View-constrained RANSAC

Quantized matching drastically increases the number of outliers as all potential locations on the model for a particular quantized feature that do not correspond to the actual location are incorrect. This is a significant issue as the number of iterations of RANSAC needed to guarantee a consistent set of inliers increases dramatically with the number of times a feature is repeated. If each feature is repeated α times, then approximately α^n times more iterations are needed to guarantee the same level of performance from RANSAC, where n is the sample size.

Prior to the advent of highly discriminative locally invariant features, such as SIFT,

Figure 3.6: Example of tomato soup can recognized (left) at a viewpoint significantly different from the closest model view (right).

local features were mostly shape-based and very ambiguous (e.g., corners, high curvature points, curve inflections). Given that one-to-one matching was infeasible, it was not uncommon for the co-visibility [89] of model features to be used as a constraint to reduce the search space. This constraint avoided attempting to estimate an object's pose from a set of features that were not simultaneously visible on the model. In early literature on the topic, methods such as interpretation trees [41], Hough transforms [41], alignment [52] and grouping [68] were used to address feature ambiguity.

Here, we introduce a modification of RANSAC, termed view-constrained RANSAC, to again exploit the co-visibility of model features. In practice, this is implemented by maintaining the set of views for which each point is visible when generating the 3D models. We will refer to the set of cameras for which a point P_i is visible in as its view set V_i. The view-constrained RANSAC algorithm begins by choosing a correspondence $C_{i,j}$ between an image point p_i and a model point P_j at random from the set of candidate matches, C. Only points P_k with a view set V_k that overlaps with the view set of the selected model point P_j are retained. The view-constrained set of correspondences $C_{vc(j)}$ for a model point P_j is defined as:

$$C_{vc(j)} = \{C_{i,k} : V_j \cap V_k \neq \emptyset \wedge k \neq j\}\,. \tag{3.1}$$

The remaining $n-1$ points needed to generate a pose hypothesis are then selected at random without replacement from the view-constrained set of points $C_{vc(j)}$. The process is repeated for a fixed number of iterations and the pose with the greatest consistent evidence is selected.

3.2 Viewpoint Variations

In unstructured environments, objects can appear in any orientation and position, often significantly different from the images used to generate the 3D models. Accounting for all possible viewpoints is infeasible, yet a 3D recognition system must still recover the object pose given a finite set of training images. A naïve solution to this problem is to incorporate images of the object from all possible viewpoints, although densely sampling the view space would require a very large number of images. In the following, we describe our approach for handling viewpoint variation more tractably and how the quantization framework can facilitate feature matching.

In the past, popular techniques to address this have been to use affine invariant features [78], affine invariant patches [97] and view clustering [69]. Other approaches have accounted for viewpoint change by simulating novel viewpoints using affine or perspective transformations of the model images [43, 64, 86]. Viewpoint simulation has been used to determine a keypoint's repeatability [43] and to model a keypoint's local appearance [64, 86]. Recently, Morel *et al.* [82] demonstrated that directly matching features extracted from these simulated viewpoints significantly outperformed the state-of-the-art affine invariant features [78] under large viewpoint change. Matching is performed by extracting features from a finite set of affine transformations of both model and query images and then comparing all sets of features.

Our approach is inspired by Morel *et al.* and incorporates features extracted from affine warped images onto the 3D models. One problem with this approach, however, is that the total number of features on the model may increase by an order of magnitude or more, with many features having similar descriptors. Pruning these features is very difficult because there is no clear metric as to when two features are similar enough to remove one of them. Our quantization framework facilitates matching to similar features and results in a seamless integration of the these features into a recognition system. We show in the evaluation that handling viewpoint in this way can significantly increase the performance of 3D object recognition.

An affine transformation A can be decomposed as:

$$A = \lambda R(\psi) \begin{bmatrix} t & 0 \\ 0 & 1 \end{bmatrix} R(\phi), \tag{3.2}$$

using Singular Value Decomposition (SVD), where $R(\psi)$, $R(\phi)$ are rotation matrices, $\lambda > 0$, and $t \geq 1$. In this decomposition, λ corresponds to the zoom and $R(\psi)$ corresponds to the planar rotation of the camera. For the case of SIFT-based systems, we can ignore these terms as SIFT features are both scale and rotation invariant. However, other types of features may require sampling the whole space of transformations. The remaining terms in the decomposition correspond to the camera viewpoint, where $t = \frac{1}{\cos(\theta)}$ is the

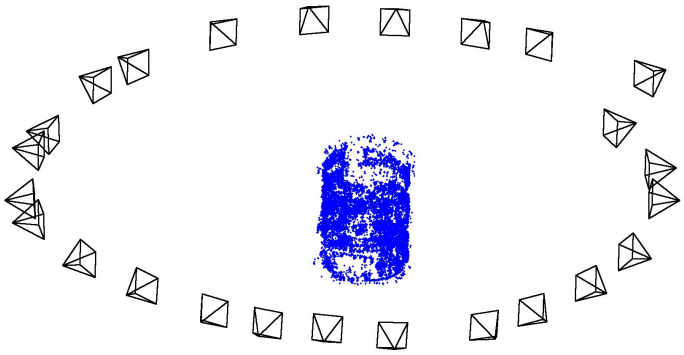


Figure 3.7: 3D model of the tomato soup can from 25 images.

tilt of the camera and ϕ is the longitude angle.

We consider tilts of $t = \{1, \sqrt{2}, 2\}$ corresponding to latitude angles of $\theta = \{0°, 45°, 60°\}$ in our implementation. For each t, we follow Morel *et al.* and sample the longitude angles ϕ by an arithmetic series $\phi = \{0, b/t, ..., kb/t\}$ for $b = 72°$ and $k = \lfloor 180° \cdot t/b \rfloor$. Each pair $\{t, \phi\}$ specifies an affine transformation $A_{t,\phi}$ which we use to transform a model image I:

$$I_{t,\phi}(x, y) = I(A_{t,\phi}(x, y)). \tag{3.3}$$

From the affine transformed image $I_{t,\phi}$, we extract SIFT features and compute the locations of each keypoint $p^i = A_{t,\phi}^{-1} p_{t,\phi}^i$ on the original image. We refer to these features as simulated affine (SA) features.

3.3 Evaluation

In order to validate the performance of our quantization framework and simulated affine features in feature-based object recognition, two sets of experiments were conducted. The first set evaluates our algorithm's ability to recognize objects in images, while the second set evaluates the algorithm's accuracy in recovering the full pose (3D position and orientation) of objects in images. Given that our methods can be easily used to extend any point-based 3D object recognition algorithm, we use three state-of-the-art algorithms (Gordon and Lowe [39], EPnP [65], and Collet *et al.* [21]) as our baseline systems. We incorporate the SA features and quantization separately (SA, Q) and together (SA+Q) to show their performance gains in complex scenes.

We evaluate our approach on the CMU Grocery Dataset which contains 10 common grocery items in cluttered household scenes. The dataset contains 620 images, 500 collected in natural environments and the other 120 collected in a calibrated setup. We use the 500 images to evaluate the recognition performance and the 120 to evaluate the accuracy of the pose estimation.

Figure 3.8: Example detections on CMU10_3D. The images were taken in cluttered environments with different lighting conditions, various object viewpoints and occlusions. The bottom two rows show the views used to generate the models for two objects.

3.3.1 Base systems

The 3D object recognition systems used as baselines in our evaluation are those of Gordon and Lowe [39], EPnP [65], and Collet *et al.* [21]. All of these systems use sparse 3D models of objects with SIFT features for recognition and share a common methodology which we summarize here. The goal of these systems is to estimate a transformation $M = [R, t]$ of a 3D model with respect to the camera frame for each object class instance in the image. This is accomplished by minimizing the sum of reprojection errors between the set of N projected 3D points $\mathbf{P}$ from the model and the set of N 2D points in the image, $\mathbf{p}$. The optimal transformation M^* is defined as:

$$M^* = \arg\min_M \sum_{i=1}^{N} d(\mathbf{p}_i, M\mathbf{P}_i)^2 \tag{3.4}$$

The 3D models used are created with a standard Structure from Motion [110] algorithm from 25 images taken at approximately equally spaced intervals in a circle around each object, as shown in Figure 3.7. Every 3D point on the model is associated with a corresponding SIFT descriptor. Finally, proper alignment and scale for each model are computed to match the real object dimensions.

When using SA features, we augment the basic 3D model by first extracting SA features from each of the model images. Then, using the estimated camera geometry, we search for correspondences of each SA feature along the epipolar lines in the nearby views. These correspondences are used to triangulate the SA features onto the 3D model.

When incorporating quantization, we use quantized descriptors (Section 3.1.1) and replace the ratio test with quantized matching (Section 3.1.2) and RANSAC with view-constrained RANSAC (Section 3.1.3).

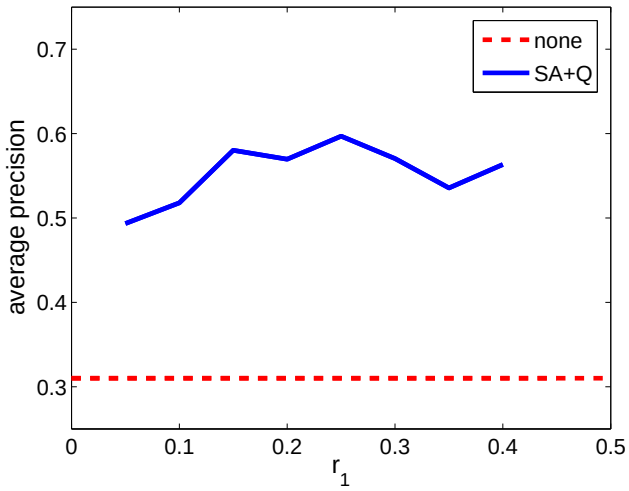


Figure 3.9: Effect of different quantization bandwidths on the Average Precision for the orange juice carton using the Collet *et al.* system. We vary the smaller bandwidth, r_1, and choose the larger bandwidth to be $r_2 = 1.5r_1$. There is no significant change in performance for $r_1 \in [0.15, 0.30]$, and even over the entire range, we obtain substantial improvement over the baseline system.

Gordon and Lowe [39]

Gordon and Lowe introduced a fast 3D scene recognition algorithm, which we modify to recognize objects. The algorithm extracts SIFT features from the input image and matches against each object model using the ratio test to obtain a set of candidate 3D-2D correspondences $\mathbf{P} \leftrightarrow \mathbf{p}$. Using RANSAC, a random subset of n points is chosen and used to estimate a pose hypothesis by minimizing the reprojection error with Levenberg-Marquardt. If the number of points consistent with the pose hypothesis is higher than a threshold, a new object instance is created and the pose is refined using all consistent points. This procedure is repeated until the number of unallocated points is lower than a threshold, or the maximum number of iterations has been exceeded.

Enhanced PnP [65]

Enhanced PnP is a non-iterative, $O(n)$ solution to the PnP problem which does not require any initialization and is much faster than standard iterative minimization techniques. The EPnP 3D recognition system we created is similar to that of Gordon and Lowe, but instead of using Levenberg-Marquardt, we use the EPnP algorithm.

Collet *et al.* [21]

The algorithm introduced by Collet *et al.* improves on the Gordon and Lowe method by combining RANSAC and mean shift clustering on the set of 3D-2D correspondences.

This combination allows for a real-time solution of the correspondence problem, even when there are many instances of the same object present. After extracting 3D-2D correspondences from a new image, the 2D locations $\mathbf{p}$ are clustered using the mean shift algorithm. Each cluster of points p^k is then processed independently by running the Gordon and Lowe pose estimation described in Section 3.3.1. Finally, all detected instances from different clusters with similar estimated pose are merged together, and the instances with the most consistent points survive.

Parameters

The parameters for our experiments were calibrated on images not in the dataset and were kept constant for every system and every object. The mean shift cluster bandwidths used for feature quantization were $r_1 = 0.2$ and $r_2 = 0.3$, although the exact choice has little impact on the overall performance of the system (Figure 3.9). For matching, we choose a ratio test threshold of 0.8. We also restrict image features to have at most 10 model correspondences in the view-constrained RANSAC to maintain tractability. The evaluation on this dataset was performed only once.

3.3.2 Object Detection

We first evaluated the performance of each system for object detection. For each detection in an image, we project all the points of the corresponding model onto the image using the recovered pose and calculate the region A inside the convex hull. We use the region overlap criterion [122]:

$$\frac{A \cap A_{gt}}{A \cup A_{gt}} > 0.5, \tag{3.5}$$

between the region A and the ground truth segmentation A_{gt} to determine if an object is correctly detected.

Figure 3.10 shows the averaged Precision/Recall plots for the three baseline systems. To summarize the performance of all the objects for each baseline system, we use the Average Precision corresponding to the area underneath the Precision/Recall curve. The results are shown in Table 3.1.

From the table, the performance of the baseline systems is very similar when none of our algorithms are incorporated. EPnP and the Gordon and Lowe system show similar performance gains when augmented with the proposed methods, suggesting that matching has a larger impact on the performance of 3D recognition than the particular choice of pose estimation algorithm. Collet *et al.*'s system, which combines RANSAC and mean shift clustering, shows further improvement once SA features and quantization are added. The use of mean shift clustering in conjunction with RANSAC reduces the outlier-inlier ratio in each cluster, and makes RANSAC more tractable with the significant

Gordon and Lowe	none	SA	Q	SA+Q
Clam chowder can	0.36	0.56	0.46	**0.79**
Diet coke can	0.09	0.07	0.04	**0.23**
Juice box	0.37	0.44	0.44	**0.71**
Orange juice carton	0.28	0.44	0.33	**0.53**
Pot roast soup	0.32	0.18	0.53	**0.79**
Rice pilaf box	0.63	0.81	0.56	**0.81**
Rice tuscan box	0.50	**0.66**	0.47	0.62
Soy milk can	0.07	0.05	0.14	**0.39**
Soy milk carton	0.44	0.46	0.44	**0.66**
Tomato soup can	0.48	0.48	0.45	**0.72**
Average	0.35	0.41	0.39	**0.62**

EPnP	none	SA	Q	SA+Q
Clam chowder can	0.36	0.52	0.49	**0.79**
Diet coke can	0.08	0.07	0.05	**0.23**
Juice box	0.27	0.35	0.43	**0.73**
Orange juice carton	0.27	0.30	0.27	**0.53**
Pot roast soup	0.32	0.19	0.54	**0.73**
Rice pilaf box	0.60	0.71	0.41	**0.81**
Rice tuscan box	0.45	0.56	0.40	**0.64**
Soy milk can	0.04	0.08	0.17	**0.39**
Soy milk carton	0.28	0.39	0.52	**0.64**
Tomato soup can	0.40	0.55	0.46	**0.75**
Average	0.31	0.37	0.37	**0.62**

Collet *et al.*	none	SA	Q	SA+Q
Clam chowder can	0.37	0.43	0.78	**0.92**
Diet coke can	0.12	0.04	0.28	**0.51**
Juice box	0.33	0.44	0.66	**0.87**
Orange juice carton	0.31	0.48	0.39	**0.61**
Pot roast soup	0.32	0.21	0.67	**0.81**
Rice pilaf box	0.61	0.76	0.71	**0.96**
Rice tuscan box	0.49	0.60	0.51	**0.80**
Soy milk can	0.06	0.03	0.27	**0.57**
Soy milk carton	0.36	0.46	0.63	**0.88**
Tomato soup can	0.45	0.47	0.76	**0.92**
Average	0.34	0.39	0.57	**0.78**

Table 3.1: Average Precision. We show the results by object for the three base systems: (top) Gordon and Lowe, (middle) EPnP, and (bottom) Collet *et al.* We demonstrate the improvements of simulated affine features (SA), quantization (Q), and the combination of the two (SA+Q).

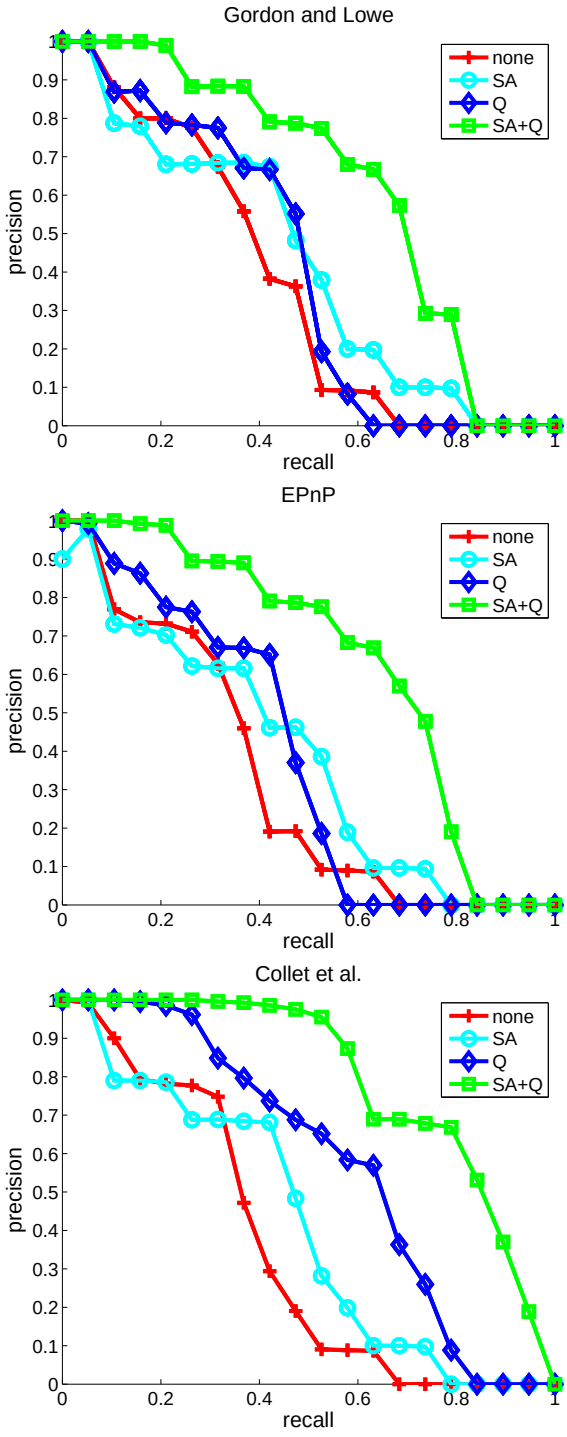


Figure 3.10: Averaged Precision/Recall plots: (left) Gordon and Lowe, (center) EPnP, and (right) Collet *et al.* For each plot, we show the improvements from simulated affine features (SA), quantization (Q) and the combination of the two (SA+Q).

Figure 3.11: Examples of misdetection with Collet *et al.* and SA+Q. In the first two images, the point matches are on only one side of the object, resulting in a planar pose ambiguity. For the third image, the system finds a repeated pattern on the wrong side of the object. In the last two images, the system does not find the objects due to significant lighting and viewpoint changes from the model training images.

increase in correspondences added by our algorithms. Some example detections are shown in Figure 3.8.

The objects which show the most improvement, as expected, are the objects with repeated patterns (e.g., diet coke can, soy milk can). Recognizing some of these objects is already very difficult, as they have particularly few features. With repeated patterns on them as well, most systems are unable to generate enough correspondences to estimate a reliable pose. Our improvements on the Collet *et al.* system increase the performance of the diet coke can by over four times and that of the soy milk can by over nine times.

The remaining objects which do not have repeated patterns also benefit significantly from the addition of SA features and quantization, doubling the performance for almost all the objects compared to the Collet *et al.* system. Objects such as the juice box and the pot roast soup have large regions where there is tiny text. Given that these regions look similar locally, most systems cannot find enough unique correspondences in these areas. Quantization addresses this issue and uses these features for pose estimation.

Figure 3.11 shows example failures of the system. The first two images are false positives due to a planar pose ambiguity described and addressed in [103]. In the center image, the system detects the repeated pattern on the object correctly, but chooses the wrong side of the object because it fails to incorporate matches from other sides of the object. Finally, the last two images show examples where the objects were not detected. In these images, the lighting conditions and viewpoints are too different from the images used to generate the model and there are not enough correct matches to estimate a pose.

3.3.3 Pose accuracy

An accurate pose is essential for object manipulation by a robot as well as for general understanding of a scene. In this section we evaluate the pose accuracy of the recognition systems. To conduct this experiment, we extrinsically calibrated a camera and ground truthed the objects in 12 poses each. For 8 of the object poses, we placed the object at 0.5 m from the camera and rotated it standing upright at intervals of 45 degrees. The

Correct detections (%)	none	SA	Q	SA+Q
Gordon and Lowe	63	80	69	88
EPnP	61	73	70	88
Collet *et al.*	65	75	74	92

Table 3.2: Detections (%) within 5 cm and 22.5 degrees of the true pose. SA+Q gives significant improvement over the baseline.

Translation error (cm)	none	SA	Q	SA+Q
Gordon and Lowe	1.17	1.31	1.10	1.12
EPnP	1.19	1.20	1.15	1.13
Collet *et al.*	1.22	1.30	1.12	1.18
Rotation error (degrees)	none	SA	Q	SA+Q
Gordon and Lowe	4.59	5.25	4.77	5.17
EPnP	4.73	5.66	4.47	5.18
Collet *et al.*	4.84	5.04	4.87	5.31

Table 3.3: Translation error in cm (top) and rotation error in degrees (bottom) for the correct detections. SA+Q approximately maintains the accuracy while improving the recognition rate.

remaining 4 poses were with the object lying on the table and were rotated at 90 degree intervals.

We evaluate the pose for both rotation and translation error. We compute the translation error as the Euclidean distance and the rotation error as the quaternion angle $2\arccos(q^T q_{gt})$ from the ground truth pose. For this set of experiments, we measure the translation error on the plane of the table and consider the error of objects which were detected within 5 cm and 22.5 degrees of the true pose.

Table 3.2 shows the percentage of correct detections for each of the systems. Out of 120 total experiments per system, the baseline systems retrieved less than two-thirds correctly. SA features and quantization boosted recognition rate to close to 90 percent for each of the systems. It is worth mentioning that some of the instances that were not detected correspond to poses where only the repeated pattern is visible; in these cases, it is impossible even for a human to disambiguate.

Table 3.3 shows the average translation error in cm and average rotation error in degrees. Despite the average rotation error being slightly higher with our proposed methods, this error of less than a degree is well within the uncertainty of the manual ground truth. Importantly, we were able to achieve a higher recognition rate while maintaining essentially equivalent pose accuracy.

3.4 Discussion

In this chapter, we showed that not committing to specific point-to-point correspondences until the hypothesis verification step can significantly improve the performance of recognition. We develop a framework in which features are quantized and matched in a hierarchical manner. To maintain the tractability of RANSAC, we propose a view-constrained RANSAC method to reduce the ratio of potential outliers to inliers. We show that incorporating features from affine transformed images is a way to address viewpoint change and that matching to these features is facilitated by the quantization framework. Our results on a difficult dataset demonstrate that quantization combined with SA features can significantly improve the performance of current state-of-the-art 3D recognition systems.

Part III

Representing Objects Without Discriminative Features

Chapter 4

Shape Matching Using Gradient Networks

Keypoint features cannot be used to detect objects that are feature-poor. These objects (e.g., hammer in Figure 4.1) are primarily defined by their contour structure, which are often just simple collections of curves and junctions. Even though many shape matching approaches work well when objects are un-occluded, their performance decrease rapidly in natural scenes where occlusions are common (see evaluation in Section 4.6.3). This sensitivity to occlusions arises because these methods are either heavily dependent on repeatable contour extraction or only consider information very locally. The main contribution of this chapter is to increase the robustness of shape matching under occlusions by formulating the problem as traversing paths in a *gradient network*.

In the past, significant research has been dedicated to representing and matching shape for object detection. A common representation is to use lines [31] and contour fragments [85, 104]. In the simplest form, contours are represented by a set of points [11, 44] and Chamfer matching [8] is used to find locations that align well in an edgemap. Local edge orientation is often incorporated [104] in the matching cost to increase robustness to clutter. These methods, however, consider each point independently and do not use edge connectivity.

To incorporate connectivity, some methods enforce the constraint that matched points are close together [111], but this still does not ensure that the matches belong to the same image contour. Other approaches capture connectivity by approximating curves as sequences of line segments or splines [128] instead of points. A common issue with these approaches, however, is the difficulty of breaking contours at repeatable locations due to noise in the edgemaps and object occlusions. To address this issue, many-to-one contour matching [107] pieces together image contours to match the object shape using Shape Context [10] features. The Contour Segment Network [31] method finds paths that match the shape through a network of extracted line segments. A ma-

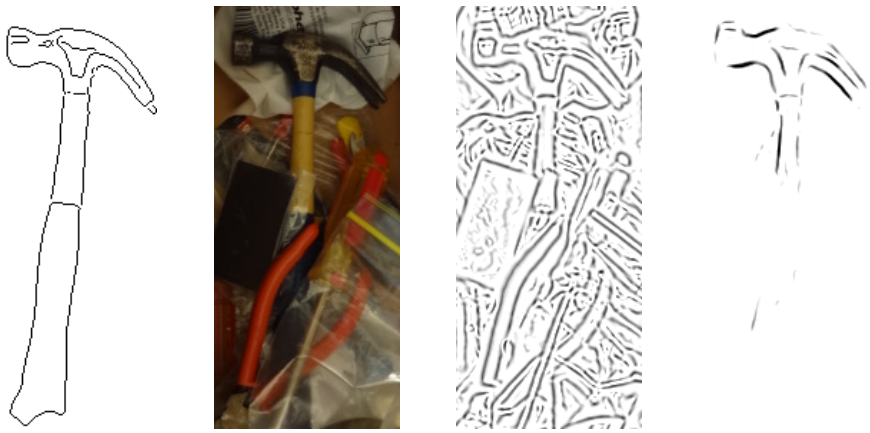


Figure 4.1: Example of shape matching under heavy occlusion. (left) Hammer template, (center-left) image window, (center-right) normalized gradient magnitudes, and (right) probability that each pixel matches the shape of the hammer.

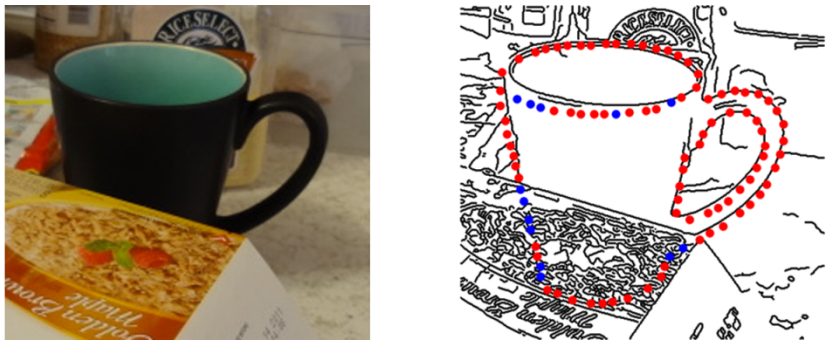


Figure 4.2: Failure of sparse edge point methods in clutter. (left) Input window. (right) Red points match the cup and blue points do not. In the textured region at the bottom of the cup, there are many matched points because it is easy to find gradients which match the cup template very locally.

jor limitation of these approaches is their reliance on stable edge detection, which still remains an open area of research [5].

To bypass edge extraction, some methods represent the shape by using coarse gradient statistics. Histogram of Oriented Gradients (HOG) [23] bins gradient magnitudes into nine orientation bins. These methods, however, only provide a coarse match of shape, losing many fine-grained details needed for instance detection. For example, a HOG cell with a single line and a HOG cell with multiple parallel lines have exactly the same descriptor. In addition, HOG cells on the object boundary are easily corrupted by strong background gradients and by object occlusions.

To capture the shape more explicitly without extracting edges, the LINE2D [44] method scans a template of sparse edge points across a gradient map. The rLINE2D [49] method increases the robustness of LINE2D by only considering points where the quan-

tized edge orientation matches exactly. These approaches, however, do not account for edge connectivity, resulting in spurious matches in clutter (Figure 4.2) and high-scoring false positives in these regions.

In a parallel line of research, there has been work on classifying edges which belong to a specific object category. The Boosted Edge Learning (BEL) detector [26] extends the Probabilistic Boosting Tree [116] algorithm to classify whether each location in the image belongs to the edge of the object. To speed up the classification, some approaches train local classifiers only at Canny edge points [91]. Sparse coding [72] has also been used to learn class-specific edges. However in all of these cases, the classification is done independently at each location, effectively losing connectivity and the global shape. They also require a large amount of labeled training data and for the background in the test images to be very similar to the training set.

In this chapter, we propose a shape matching approach which captures contour connectivity directly on low-level image gradients. For each image pixel, our algorithm estimates the probability (Figure 4.1) that it matches a template shape. The problem is formulated as traversing paths in a *gradient network* and is inspired by the edge extraction method of GradientShop [12]. Our results show significant improvement in shape matching and object detection on a difficult dataset of feature-poor objects in natural scenes with severe clutter and occlusions.

4.1 Formulation

For a template shape placed at a particular image location, our method returns for each image pixel, the probability that it matches the template. We begin by defining the *gradient network* of an image. Then, we formulate shape matching as finding paths in the network which have high local shape similarity. We describe the local shape potential for each node in the network, followed by the algorithm used for shape matching.

For each pixel p in an image, let $\nu(p)$ be the gradient magnitude and $\theta(p)$ be the gradient orientation computed using oriented filters [34]. Let Q_0^p be the set of four pixels at integer coordinates closest to the floating point coordinate calculated by translating the pixel p a distance of $\sqrt{2}$ in the direction of the tangent $\theta(p) + \pi/2$. Similarly, let Q_1^p be the set of four pixels in the direction of the tangent $\theta(p) - \pi/2$. A *gradient network* is then defined as a graph where each pixel p in the image is a node that is connected to the eight pixels $q \in \{Q_0^p, Q_1^p\}$ as shown in Figure 4.3. We define $\phi_\beta(p, q)$ to be the bilinear interpolation weight for each q with respect to its ideal floating point coordinate.

In addition, let $\mathbb{S}(\mathbf{z})$ be a template shape $\mathbb{S}$ placed at position $\mathbf{z}$ in the image. Initially, for simplicity of explanation, we define $\mathbb{S}(\mathbf{z})$ by N edge points $\mathcal{Y} = \{y_1, ..., y_N\}$, each with a gradient orientation ψ_i. Later, we extend the formulation to directly operate on model

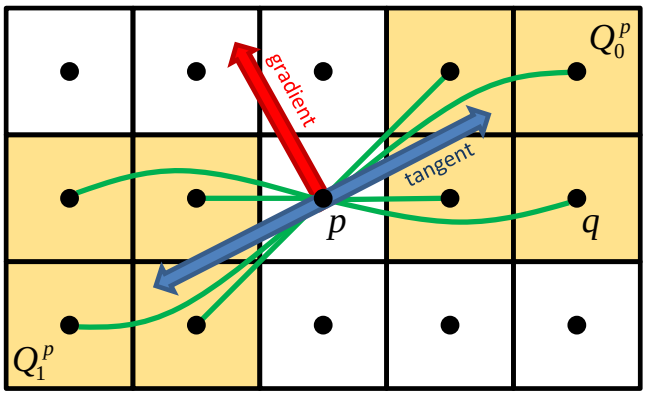


Figure 4.3: Gradient Network (GN). Each node is a pixel in the image. We create a network (green) by connecting each pixel p with its 8 neighbors in the direction of the local tangent.

edge strengths. For conciseness of notation, superscript $\mathbb{S}$ in the following derivation is implicitly $\mathbb{S}(\mathbf{z})$. Let $d^{\mathbb{S}}(p)$ be the distance from p to the nearest model edge point $y^* \in \mathcal{Y}$ and $\theta^{\mathbb{S}}(p) = \psi^*$ be the orientation of that edge point. Both values can be computed simultaneously using a Distance Transform [16]. The goal is then to find long connected paths in the gradient network which match the template $\mathbb{S}(\mathbf{z})$ well.

4.2 Local Shape Potential

We begin by defining the local shape potential, $\Phi^{\mathbb{S}}(p)$, which measures how well each node p in the gradient network matches $\mathbb{S}(\mathbf{z})$ locally. This potential is composed of three terms: 1) the region of influence $\phi^{\mathbb{S}}_{roi}$, 2) the local appearance $\phi^{\mathbb{S}}_{\mathcal{A}}$, and 3) the edge potential ϕ_E. It is given by:

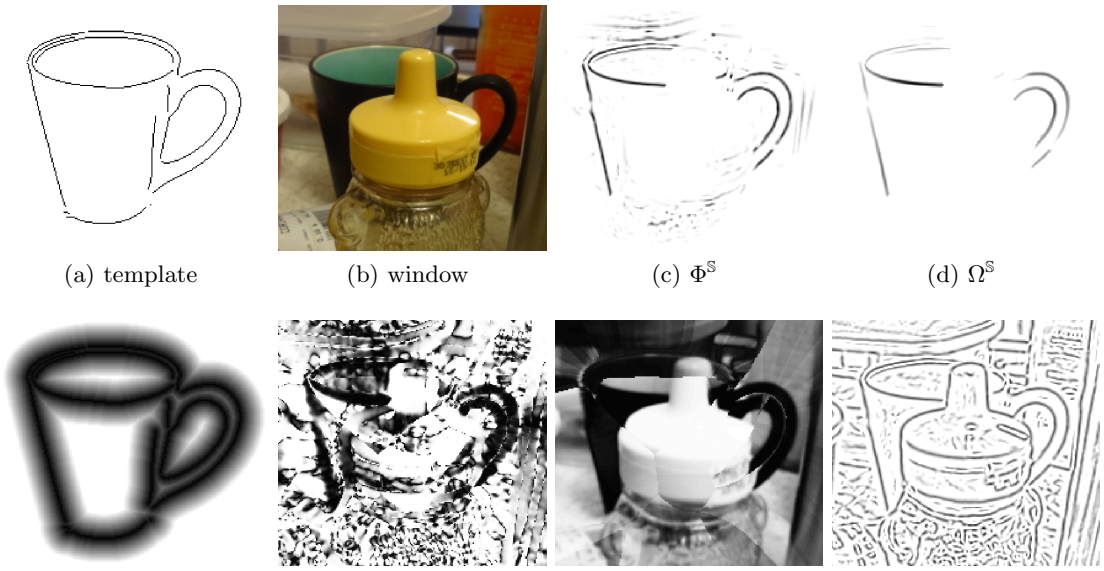
$$\Phi^{\mathbb{S}}(p) = \phi^{\mathbb{S}}_{roi}(p) \cdot \phi^{\mathbb{S}}_{\mathcal{A}}(p) \cdot \phi_E(p). \tag{4.1}$$

4.2.1 Region of Influence

Given $\mathbb{S}(\mathbf{z})$, we only want to consider pixels which are sufficiently close as candidates for matching while simultaneously allowing slight deformations of the template [6]. We employ a linear weighting scheme to define the region of influence as:

$$\phi^{\mathbb{S}}_{roi}(p) = \max\left[1 - \frac{d^{\mathbb{S}}(p)}{\tau_d}, 0\right], \tag{4.2}$$

where τ_d is the farthest distance from the shape that we want to consider. We set $\tau_d = 15$ to be the same as in Oriented Chamfer Matching [104].



(a) template (b) window (c) $\Phi^{\mathbb{S}}$ (d) $\Omega^{\mathbb{S}}$

(e) $\phi^{\mathbb{S}}_{roi}$ (f) $\phi^{\mathbb{S}}_{\theta}$ (g) $\phi^{\mathbb{S}}_{\mathcal{C}}$ (h) ϕ_E

Figure 4.4: Illustration of shape matching algorithm. Given (a) the template and (b) the image window, we compute (c) the local shape potential and apply the message passing algorithm to produce (d) the shape similarity. The local shape potential is composed of the (e) region of interest, (f) orientation, (g) color, and (h) edge potentials.

4.2.2 Local Appearance

This term describes how well each pixel matches the local appearance of $\mathbb{S}(\mathbf{z})$. Many types of information can be used, ranging from local gradient orientation to interior appearance of the object, such as color and texture. To illustrate our approach, we consider the effects of gradient orientation and color. The local appearance potential is defined as:

$$\phi^{\mathbb{S}}_{\mathcal{A}}(p) = \phi^{\mathbb{S}}_{\theta}(p) \cdot \phi^{\mathbb{S}}_{\mathcal{C}}(p), \tag{4.3}$$

where $\phi^{\mathbb{S}}_{\theta}$ is the orientation potential and $\phi^{\mathbb{S}}_{\mathcal{C}}$ is the color potential. We define the local orientation potential as:

$$\phi^{\mathbb{S}}_{\theta}(p) = \exp\left(-\frac{\left[\theta(p) - \theta^{\mathbb{S}}(p)\right]^2}{2\sigma_{\theta}^2}\right), \tag{4.4}$$

with $\sigma_{\theta} = \pi/8$ (i.e., the orientation bin size of LINE2D [44]).

The color potential captures the color around the object shape. Unlike BEL [26] and [91] which consider local patches centered on an edge, we only use information from the object interior to be more robust to background clutter. Let v_i be the unit-norm gradient vector at y_i pointing to the object interior and c_i be the L*u*v color of the object extracted a fixed distance in the direction of v_i for each model edge point y_i.

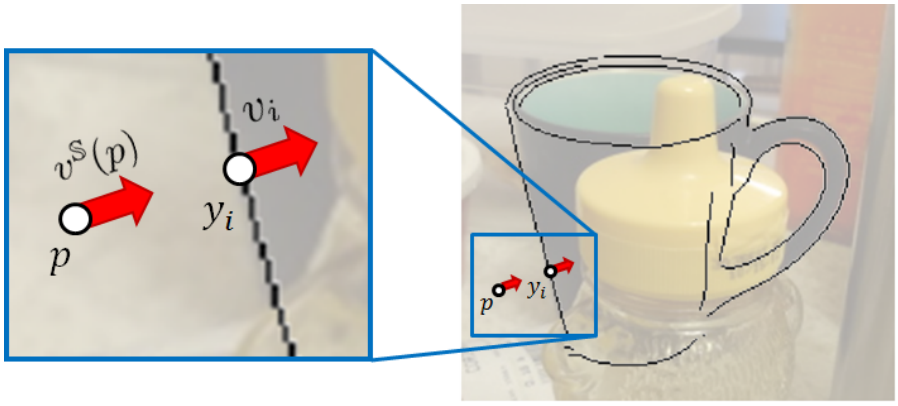


Figure 4.5: Computation of the color potential. For model edge point y_i, the unit-norm vector v_i points to the object interior. The color c_i is the object color extracted a fixed distance from y_i in the direction of v_i. For pixel p, the closest model point y^* is y_i, so $v^{\mathbb{S}}(p) = v_i$ and $\mathcal{C}^{\mathbb{S}}(p) = c_i$. From the image, we extract the color $\mathcal{C}(p)$ at the same fixed distance from p in the direction of $v^{\mathbb{S}}(p)$.

Then let $v^{\mathbb{S}}(p) = v^*$ and $\mathcal{C}^{\mathbb{S}}(p) = c^*$ correspond to the $y^* \in \mathcal{Y}$ closest to p. From the image, we extract the color $\mathcal{C}(p)$ at the same fixed distance from p in the direction of $v^{\mathbb{S}}(p)$. This corresponds to what the object interior would look like from this pixel if it is part of the shape. The local color potential is then defined as:

$$\phi_{\mathcal{C}}^{\mathbb{S}}(p) = \exp\left(-\frac{\left[\mathcal{C}(p) - \mathcal{C}^{\mathbb{S}}(p)\right]^2}{2\sigma_{\mathcal{C}}^2}\right). \tag{4.5}$$

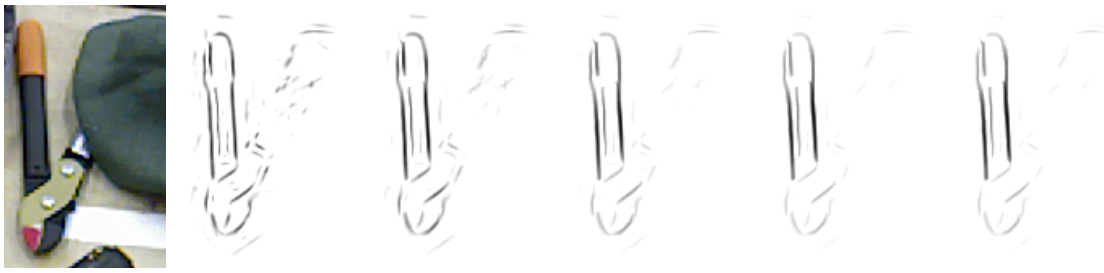
We set $\sigma_{\mathcal{C}}^2 = 1/15$ according to [71] for L*u*v color normalization. Figure 4.5 illustrates the computation of the color potential.

4.2.3 Edge Potential

The edge potential, ϕ_E, characterizes how likely a pixel belongs to an edge in the image. Many different metrics can be used. In the simplest form, the edge potential can be the raw gradient magnitude $\nu(p)$. In GradientShop, the authors normalize the magnitude of each pixel with respect to the magnitudes in a 5×5 neighborhood κ to be more robust to edge contrast. If μ_κ and σ_κ are the mean and standard deviation of the magnitudes in κ, then the normalized gradient magnitude is:

$$\hat{\nu}(p) = \frac{\nu(p) - \mu_\kappa}{\sigma_\kappa + \epsilon}. \tag{4.6}$$

More complicated edge potentials, such as the output of edge detectors like the Global Probability of Boundary (gPb) [4], can also be used instead. In the evaluation, we explore the effect of different edge potentials.



(a) window (b) $\Phi^{\mathbb{S}}$ (c) iter 5 (d) iter 10 (e) iter 25 (f) iter 100

Figure 4.6: Shape similarity for different number of message passing iterations.

4.3 Shape Matching

While the local shape potential can be used as a measure of shape similarity, it considers only a very limited scope when determining how well each pixel matches $\mathbb{S}(\mathbf{z})$. By itself, it is prone to incorrect similarities from accidental alignments in background clutter and occlusions (Figure 4.4c). Our key idea for obtaining a more robust shape similarity is to broaden the scope of each pixel, p, by traversing the path in the gradient network on which it is centered. The pixel matches the shape well if this path consists of a long contiguous set of pixels which all have high local shape potential.

We characterize the contiguity from pixel p to each $q \in \{Q_0^p, Q_1^p\}$ by the pairwise potential:

$$\Psi^{\mathbb{S}}(p,q) = \phi_\beta(p,q) \cdot \phi_\theta(p,q) \cdot \phi_{\mathcal{A}}^{\mathbb{S}}(q), \tag{4.7}$$

where $\phi_\beta(p,q)$ is the bilinear interpolation weight and $\phi_\theta(p,q) = \exp\left(-\frac{[\theta(p)-\theta(q)]^2}{2(\pi/5)^2}\right)$ is the edge smoothness [12]. The local appearance potential, $\phi_{\mathcal{A}}^{\mathbb{S}}(q)$, effectively breaks the contiguity when the shape of the neighbor q is improbable. We do not include the region of influence potential as we do not wish to overly penalize an imperfectly aligned template.

This formulation of shape matching is related to the edge extraction approach of GradientShop [12]. We adapt their message passing technique to estimate the shape similarity. The problem of estimating the shape similarity at p is broken into two sub-problems; one for estimating the similarity in the direction of Q_0^p and the other for estimating the similarity in the direction of Q_1^p. At each iteration t, the messages are computed as:

$$m_0^{\mathbb{S},t}(p) = \sum_{q \in Q_0^p} \Psi^{\mathbb{S}}(p,q) \cdot \left[\Phi^{\mathbb{S}}(q) + m_0^{\mathbb{S},t-1}(q)\right], \tag{4.8}$$

$$m_1^{\mathbb{S},t}(p) = \sum_{q \in Q_1^p} \Psi^{\mathbb{S}}(p,q) \cdot \left[\Phi^{\mathbb{S}}(q) + m_1^{\mathbb{S},t-1}(q)\right], \tag{4.9}$$

(a) uncalibrated (b) calibrated

Figure 4.7: Probability calibration of the shape similarity using the Extreme Value Theory. (a) Raw, uncalibrated shape similarity returned by the message passing algorithm. (b) Probabilistic similarity after calibration.

and the estimated shape similarity is:

$$\Omega^{\mathbb{S},t}(p) = m_0^{\mathbb{S},t}(p) + m_1^{\mathbb{S},t}(p) + \Phi^{\mathbb{S}}(p). \tag{4.10}$$

The messages are initialized to $m_0^{\mathbb{S},0}(p) = m_1^{\mathbb{S},0}(p) = 0$, and the message passing is iterated for a fixed number of iterations to produce the final shape similarity estimate $\Omega^{\mathbb{S}}$. Empirically, the message passing converges in 25 iterations and we use this for all of our experiments. Figure 4.6 shows the shape similarity for different number of message passing iterations.

4.4 Probability Calibration

The shape similarity $\Omega^{\mathbb{S}}$, computed in Equation 4.10, depends on the template shape $\mathbb{S}$. This makes it difficult to compare the similarity values of different templates, and thus difficult to choose the highest scoring template for object recognition. A method to calibrate these values is thus needed.

There exist many methods for performing score normalization. A common method is to fit Gaussian distributions [81] to the scores and select the maximum likelihood class. Exemplar SVMs [73] fit a sigmoid function to the positive and negative scores of each instance detector to convert the scores into probabilities. However, a limitation of these approaches is that they require a large amount of data to model the distributions accurately. Unlike category recognition where positive instances can easily be mined from the Internet, it is often much more difficult to obtain many images of the same object instance under multiple viewpoints. Negative data on the other hand is easily obtained by sampling background images.

We use the Extreme Value Theory [101] to calibrate the shape similarity since it only requires the distribution of similarity values on negative data. The method fits a Weibull function to the extreme values (i.e., tail) of the negative distribution and estimates the positive distribution as its conditional density. We sample random locations in background images and use all the $\Omega^{\mathbb{S}}$ within the region of influence as negative data. Figure 4.7 shows an example of the calibration.

4.5 Soft Shape Model

The above formulation defines the template $\mathbb{S}(\mathbf{z})$ as a discrete set of edge points. This discrete representation requires either a manually specified template or automatic edge extraction. Manual specification, however, is impractical for a large set of templates, and automatic edge extraction requires time consuming parameter tuning to obtain good edgemaps. We address this limitation by computing a soft shape model using the raw edge strength (e.g., edge potential) of every object pixel (i.e., object mask), instead of discrete edge points. In the following, we define a soft way to compute the distance $d^{\mathbb{S}}$ and orientation $\theta^{\mathbb{S}}$ which fully describe the relationship between $\mathbb{S}(\mathbf{z})$ and the image.

Let $\mathcal{Y} = \{y_1, ..., y_N\}$ be all the pixels representing $\mathbb{S}(\mathbf{z})$, each with an edge strength γ_i and gradient orientation ψ_i. We define the soft distance $d^{\mathbb{S}}$ as:

$$d^{\mathbb{S}}(p) = \min_i \left[D(p, y_i) + 1/\gamma_i - 1/\gamma^{max} \right], \tag{4.11}$$

where $D(p, y_i)$ is the Euclidean distance between p and y_i, and γ^{max} is the maximum edge strength. Then, the soft orientation is $\theta^{\mathbb{S}}(p) = \psi^*$ and corresponds to the $y^* \in \mathcal{Y}$ that minimizes Equation 4.11. Both values can be computed simultaneously using the Generalized Distance Transform [29]. If γ_i is binary, then the soft shape model reduces to the discrete case.

4.6 Evaluation

In order to validate the performance of Gradient Network in shape-based object instance detection, we performed two sets of experiments. The first evaluates the algorithm's accuracy in shape matching, while the second evaluates the algorithm's ability to detect objects. We compare the effects of using a hard model versus a soft model, as well as the effects of different edge and local appearance potentials. We evaluate our algorithm on the challenging CMU Kitchen Occlusion (CMU_KO8) dataset [49].

4.6.1 Algorithms

We compare our approach with a number of state-of-the-art methods for template-based shape matching. For fair comparison, we use the same M sampled model edge points x_i

(a subset of $\mathcal{Y}$) for all the methods. These points are specified relative to the template center. We give a brief description of the algorithms in our comparison below.

Gradient Network (GN) Our algorithm returns a shape similarity $\Omega^{\mathbb{S}}$ for each pixel given the template $S(\mathbf{z})$. For fair comparison, we apply a 7×7 max spatial filter (i.e., equivalent to LINE2D) to $\Omega^{\mathbb{S}}$ resulting in $\hat{\Omega}^{\mathbb{S}}$. The template score at $\mathbf{z}$ is then $\sum_{i=1}^{M} \hat{\Omega}^{\mathbb{S}}(x_i + \mathbf{z})$. We use the soft shape model with normalized gradient magnitudes for the edge potential, and both color and orientation for the appearance.

Our algorithm takes on average 2 ms per location $\mathbf{z}$ on a 3GHz Core2 Duo CPU. In practice, we run our algorithm only at the hypothesis detections of rLINE2D [49], which has been shown to have high recall. The combined computation time is about 1 second per image.

LINE2D (L2D) [44] This method quantizes all gradient orientations into 8 orientation bins. The similarity for point x_i is the cosine of the smallest quantized orientation difference, $\Delta\theta_i$, between its orientation and the image orientations in a 7×7 neighborhood of $x_i + \mathbf{z}$. The score of a window is $\sum_{i=1}^{M} \cos(\Delta\theta_i)$.

rLINE2D (rL2D) [49] This method binarizes LINE2D by only considering model edge points which have the same quantized orientation as the image. The algorithm is more robust than LINE2D in cluttered scenes with severe occlusions. The score of a window is $\sum_{i=1}^{M} \delta(\Delta\theta_i = 0)$ where $\delta(z) = 1$ if z is true.

Oriented Chamfer Matching (OCM) [104] This method extends Chamfer matching to include the cost of the orientation dissimilarity. Let DT be the distance transform of the image edgemap and DT_θ be the orientation of the nearest edge point, then the OCM score at position $\mathbf{z}$ is $\sum_{i=1}^{M} DT(x_i + \mathbf{z}) + \lambda \sum_{i=1}^{M} D_\theta \left[DT_\theta(x_i + \mathbf{z}), \psi_i\right]$. The parameter λ is learned for each shape independently.

Histogram of Oriented Gradients (HOG) [23, 73] This method represents an object as a grid of gradient histograms. An Exemplar SVM [73] is learned for each shape. We use the one hundred negative images in CMU_KO8, the same parameters as [73], and three hard negative mining iterations for training. The object is detected by convolving the learned template with the HOG of the image.

rL2D-gPb and GN-gPb To explore the use of more complex edge potentials, we extend rL2D and GN to use the output of gPb [4], a state-of-the-art edge detector that uses texture and color segmentations. gPb outputs the probability B that a pixel belongs to an object boundary. The **rL2D-gPb** algorithm applies a 7×7 max spatial filter to

B to produce $\hat{B}$ and computes the score at position $\mathbf{z}$ as $\sum_{i=1}^{M} \hat{B}(x_i + \mathbf{z}) \cdot \delta(\Delta\theta_i = 0)$. The **GN-gPb** algorithm uses $\phi_E = B$ as the edge potential.

4.6.2 Shape Matching

We first evaluate the performance in matching accuracy. Each algorithm, besides HOG, returns a similarity measure per model point. Ideally for an image window, points corresponding to visible object parts should have higher similarity than those that are occluded. Given the groundtruth occlusion labels for every image, we partition the similarity scores into visible and occluded scores, and report the F-measure (i.e., maximum geometric mean of precision and recall) in Table 4.1. We do not include HOG in this evaluation because it does not return point confidences, and thus cannot be compared fairly with the other methods. Figure 4.8 shows some qualitative results.

From the table, GN outperforms all the baseline algorithms. L2D, rL2D and OCM consider information very locally resulting in many incorrect point confidences. rL2D-gPb removes spurious texture responses by using gPb, but performs poorly because its similarity measure is not indicative of how well the shape matches (e.g., high contrast edges have high gPb probabilities irrespective of shape). By considering long connected paths in gPb which match the shape, GN-gPb performs significantly better than rL2D-gPb. However, it performs slightly worse than GN, because gPb often gives low probability to interior object edges, resulting in incorrect confidences in these areas. The table also shows, importantly, that both the orientation and color appearance potentials at edge points are informative for shape matching.

Figure 4.9 shows a scenario where GN performs better than rL2D, OCM and HOG. In this example, the template is a straight line and the image contains two fragmented edges of the same orientation. This is a typical scenario in clutter where it is easy to find fragmented edges which match the template. Ideally, we would want these fragmented matches to be down-weighted. However, since both rL2D and OCM look at only the orientation of the edge and its distance to the template, all the points on the template will match well, leading to high scoring false positives. Similarly, HOG will have a high score since it looks at coarse statistics of gradient orientations, which in this case, are all in the same orientation. On the other hand, since the GN method looks for long connected image gradients, it down-weights these fragmented matches.

In addition, we evaluate the effect of using a soft shape model (GN) versus a hard shape model (GN-hard). We tuned a Canny edge detector very carefully on the model images to obtain the best possible contours for GN-hard. Our results show that using a soft shape model, which does not require any parameter tuning, actually performs on par and even slightly better than a hard shape model.

	combined	orientation	color
naive (label all visible)	0.78	-	-
L2D	0.83	-	-
rL2D	0.83	-	-
rL2D-gPb	0.79	-	-
OCM	0.79	-	-
GN	0.87	0.85	0.83
GN-hard	0.86	0.85	0.83
GN-gPb	0.85	0.84	0.84

Table 4.1: F-measure characterizing the shape matching. For methods which use GN, we evaluate the effects of using orientation, color, and their combination for the local appearance potential, $\phi^{\mathbb{S}}_{\mathcal{A}}$. We also compare soft shape models (GN) with hard shape models (GN-hard).

Single	L2D	rL2D	rL2D-gPb	OCM	HOG	GN	GN-gPb
baking pan	0.46	0.68	0.41	0.66	0.69	**0.89**	0.86
colander	0.58	0.87	**1.00**	0.74	0.85	0.92	0.97
cup	0.45	0.80	0.93	0.71	0.86	**0.98**	0.96
pitcher	0.45	0.84	0.67	0.76	0.77	0.85	**0.89**
saucepan	0.49	0.82	0.71	0.70	0.69	0.99	**1.00**
scissors	0.29	0.62	0.27	0.53	0.75	**0.87**	0.86
shaker	0.29	0.68	0.91	0.49	0.72	0.84	**0.93**
thermos	0.57	0.80	0.50	0.71	0.80	**0.94**	0.94
Mean	0.45	0.76	0.68	0.66	0.77	0.91	**0.93**
Multiple	L2D	rL2D	rL2D-gPb	OCM	HOG	GN	GN-gPb
baking pan	0.32	0.41	0.19	0.45	0.65	**0.97**	0.90
colander	0.53	0.81	**0.95**	0.31	0.82	0.93	0.94
cup	0.34	0.67	0.78	0.42	0.90	**0.97**	0.97
pitcher	0.43	0.65	0.11	0.28	0.68	**0.86**	0.83
saucepan	0.41	0.76	0.64	0.59	0.82	**0.99**	0.98
scissors	0.37	0.60	0.07	0.17	0.64	**0.93**	0.80
shaker	0.34	0.61	0.50	0.18	0.59	0.84	**0.89**
thermos	0.38	0.75	0.40	0.36	0.85	0.93	**0.95**
Mean	0.39	0.66	0.45	0.35	0.74	**0.93**	0.91

Table 4.2: Detection rate at 1.0 FPPI on CMU_KO8.

4.6.3 Object Detection

Next we evaluate the performance for object detection. An object is correctly detected if the intersection-over-union (IoU) of the predicted bounding box and the groundtruth bounding box is greater than 0.5. The CMU_KO8 dataset is split into two parts: 800 images for single viewpoint and 800 images for multiple viewpoints. Figure 4.10 and 4.11

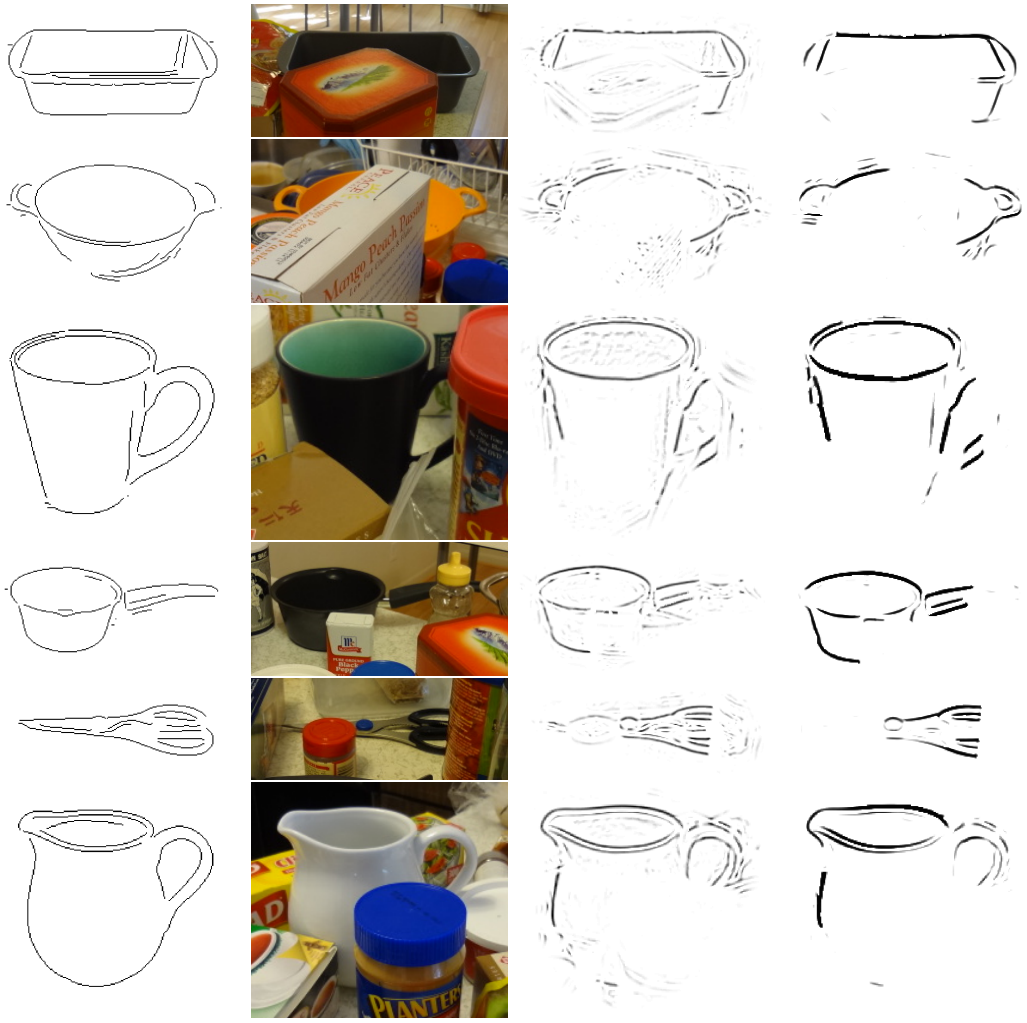


Figure 4.8: Results of shape matching using GN. From left to right, we show: 1) template, 2) window, 3) $\Phi^{\mathbb{S}}$, and 4) probability that each pixel matches the template.

Figure 4.9: Typical example of when GN performs better than rL2D, OCM and HOG. The fragmented image edges in black, which are common in cluttered background, should not match well to the template shape in blue. However, since the image edges have the same orientation as the template and they are sufficiently close to it, both rL2D and OCM will have a high matching score. Similarly, because HOG only looks at the local orientation statistics, it will have a high matching score as well. Our GN method looks for image edges which are long, connected and match the shape. The fragmented match in this example would be down-weighted.

Single	combined	orientation	color
GN	0.91	0.88	0.76
GN-hard	0.92	0.87	0.77
GN-gPb	0.93	0.85	0.86
Multiple	combined	orientation	color
GN	0.93	0.74	0.88
GN-hard	0.93	0.73	0.87
GN-gPb	0.91	0.65	0.66

Table 4.3: Average detection rate at 1.0 FPPI on CMU_KO8.

show the false positive per image (FPPI) versus the detection rate (DR) for these parts respectively. Table 4.2 summarizes the performance with the detection rate at 1.0 FPPI.

From the tables, GN significantly outperforms the other algorithms. The relative performance of the algorithms is similar to the shape matching evaluation. For objects with vibrant colors, such as the shaker (red) and colander (orange), GN-gPb performs slightly better than GN because these objects receive high gPb edge potentials. However for typical, un-colorful objects, gPb gives less confident edge potentials and this results in worse overall performance of GN-gPb for these objects. In addition, HOG performs worse than GN because it only captures the shape very coarsely and the cells covering the object boundary are easily corrupted by background clutter and occlusions. Figure 4.12 shows the detector responses for rL2D, OCM, HOG and GN. The response of the GN algorithm is more peaked at the correct location, while the other methods have many spurious responses leading to high scoring false positives.

Figure 4.13 shows the performance under different levels of occlusions. While many of the systems perform fairly well at low occlusion levels (0-15%), they perform significantly worse at high occlusion levels (>35%). L2D, rL2D and OCM often incorrectly have high point confidences in background clutter which result in false positives with higher score than true positives under heavy occlusion. HOG performs especially poorly because occlusions severely corrupt the descriptors of HOG cells. Our GN and GN-gPb algorithms are more robust to object occlusions, since they predict better shape similarities.

Table 4.3 analyzes the performance of soft versus hard shape model, different edge potentials and the effects of gradient orientation and color. Again, a soft shape model performs equivalently to the hard model, and both the orientation and color contribute to the detection accuracy.

Figure 4.14 shows typical false positives of GN. These detections have long contours which align well to the image. Additional information such as occlusion reasoning [49] or interior appearance of the object is needed to filter these false positives. Another failure

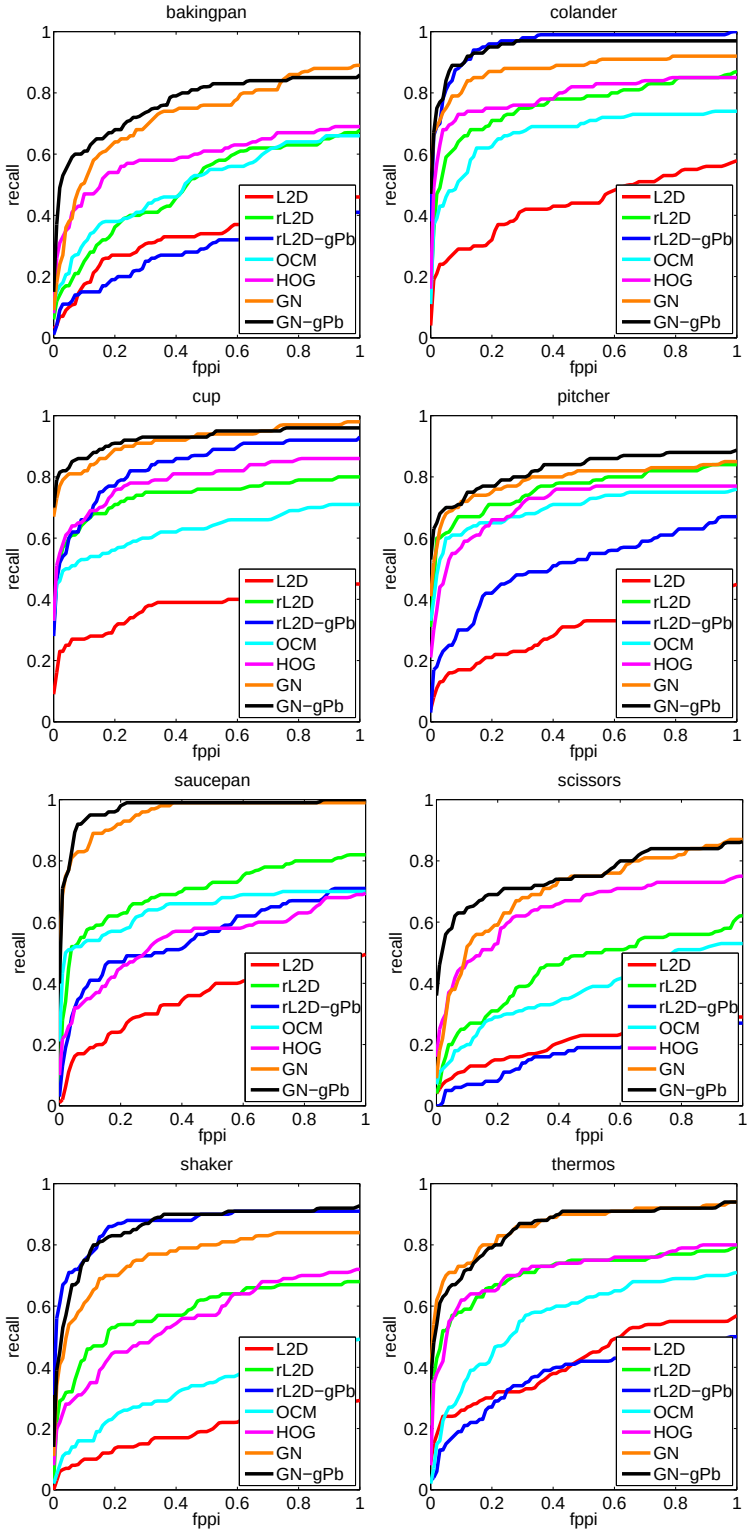


Figure 4.10: FPPI/DR results for single view on CMU_KO8.

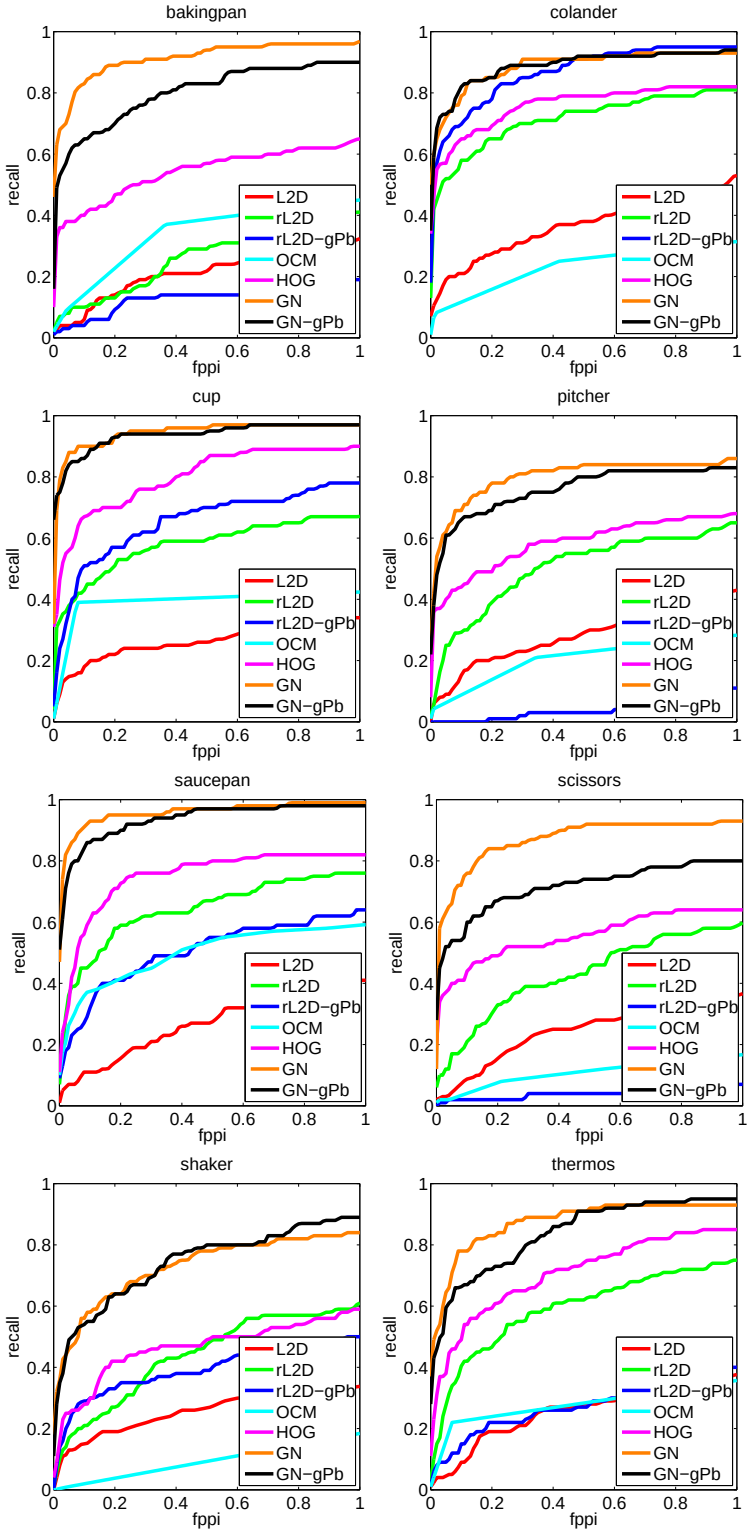
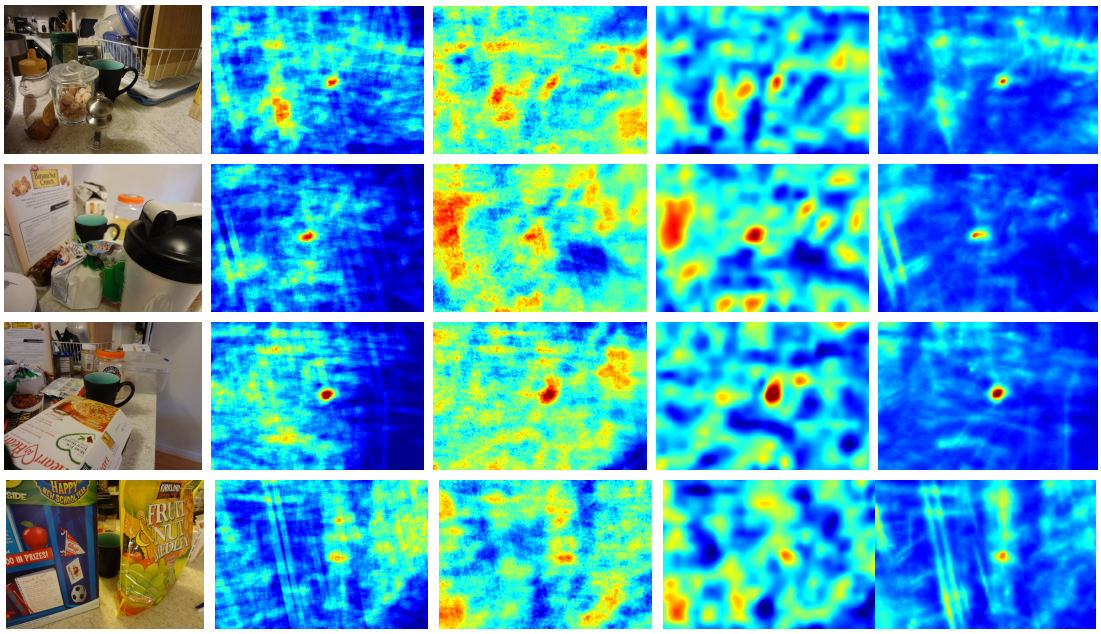


Figure 4.11: FPPI/DR results for multiple view on CMU_KO8.



(a) image (b) rL2D (c) OCM (d) HOG (e) GN

Figure 4.12: Response maps for detecting a cup. The response of GN is more peaked at the true detection than the other methods.

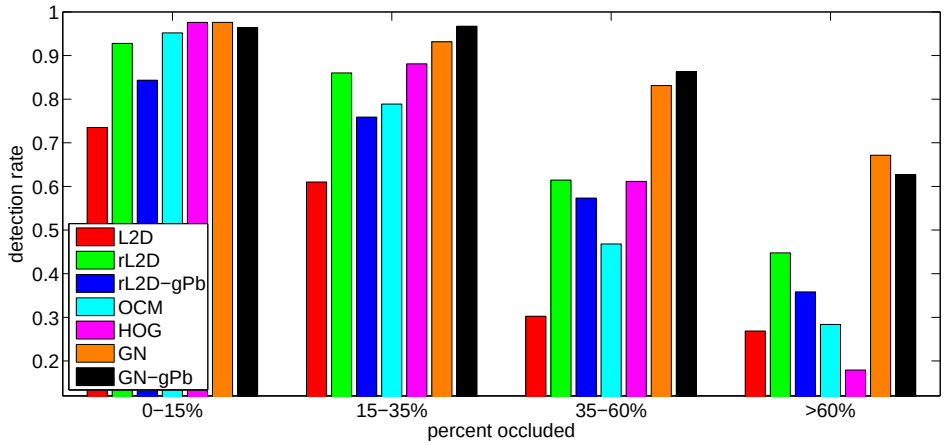
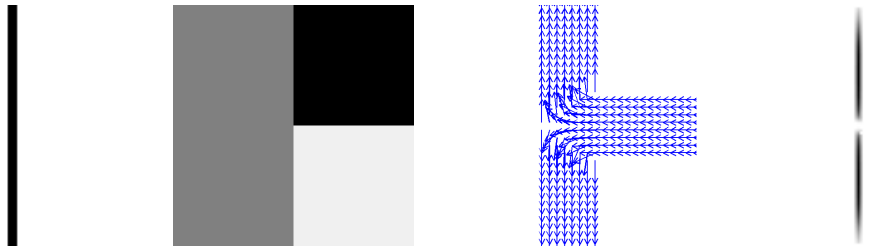


Figure 4.13: Detection rate under different occlusion levels. GN and GN-gPb are more robust to occlusions.

mode of GN is at junctions shown by the example in Figure 4.15. When the flow of the image gradients is broken, the shape matching information can not be propagated correctly by the message passing algorithm. This results in a fragmented match shown in Figure 4.15d.

Figure 4.14: False positives of GN. Each triplet shows (1) template, (2) false positive window and (3) predicted match in red overlaid on the Canny edgemap.



(a) template (b) input window (c) flow field (d) GN output

Figure 4.15: GN and junctions. The GN method is unable to propagate information past a junction if there is a significant change in the gradient orientation. As shown in (d), there is a gap in the shape match. Another level of reasoning is necessary to handle junctions.

4.7 Discussion

The main contribution of this chapter is to demonstrate that shape matching can incorporate edge connectivity directly on low-level gradients without extracting contours. We create a gradient network where each pixel is connected with its neighbors in the local tangent direction. Long paths which match the template shape are found using a message passing algorithm. Our results on a challenging dataset of feature-poor objects in realistic environments with severe occlusions demonstrate significant improvement over state-of-the-art methods for shape matching and object instance detection.

Chapter 5

Combining Boundary and Region Information

Even though feature-poor objects are primarily defined by their shape, many common objects have simple shape which can be easily confused with background clutter, resulting in false positives as shown in Figure 5.1. These objects often have large uniform regions which have typically been viewed as uninformative. However, from the figure, it can be seen that the appearance of the object, in particular the lack of texture, can be used to filter these false positives. In this chapter, we explore the benefits of combining both an explicit boundary representation with region appearance information using Boundary and Region Templates (BaRT). We show that the BaRT representation (Figure 5.2) achieves significant improvement over state-of-the-art methods for object instance detection.

5.1 Boundary Representation

While some current approaches, such as the popular HOG method, do implicitly capture both boundary and region information using grid of coarse gradient statistics, they are unable to capture the fine details of the shape. In addition, grid cells located on the object boundary are easily corrupted by background gradients (Figure 5.3). Thus, in our BaRT representation, we capture the shape explicitly. In the evaluation, we compare using an explicit representation of the boundary with a coarse representation.

We incorporate the boundary explicitly with BaRT using two methods. The first is the robust-LINE2D (rLINE2D) method [49] and the second is the Gradient Networks (GN) [50] approach from the previous chapter. Both methods are based on sparse edge templates which have been shown to be efficient at recognizing objects under arbitrary viewpoint. For a template with N_B sampled edge points P, the boundary descriptor is $B = [b_1, \ldots, b_{N_B}]$, where b_i the similarity measure return by rLINE2D and GN.

Figure 5.1: Example of false positives when using shape only. The edge information aligns well to the image but the interior does not match well. From left to right, we show (1) the model object, (2) a false positive detection, (3) the zoomed in view of the false positive, and (4) the edge points matched using Gradient Networks on top of the edgemap. The hotter the color, the better the match.

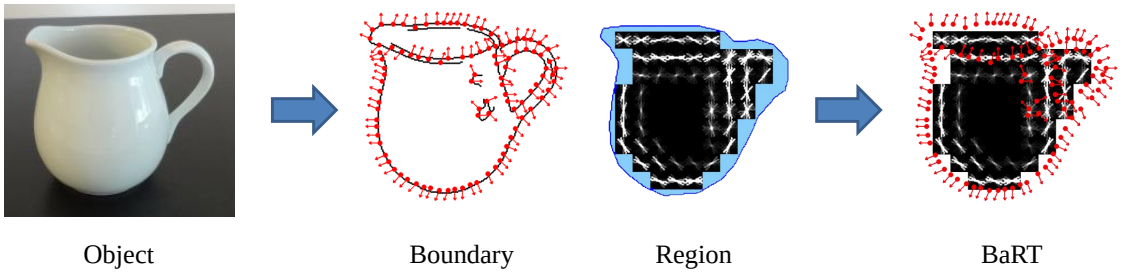


Figure 5.2: Boundary and Region Templates (BaRT). We propose to model an object by using an explicit boundary representation and a coarse representation of the interior region. The boundary is captured by a sparse set of edge points (*red*), each with an oriented gradient. The region is modeled using Histogram of Oriented Gradient (HOG) cells inside the object mask (*blue*). We modify HOG to handle large uniform regions.

5.2 Region Representation

As shown previously, there is information in the object interior even if the appearance is uniform. While the interior region of an object is unaffected by background clutter, its appearance can change due to lighting effects such as specularities and shadows. Computing statistics at a coarse resolution can mitigate these lighting effects, since they are either isolated or low frequency.

To demonstrate the importance of representing the object interior, we propose to explore the use of HOG and color for capturing region information, although other approaches and information may be used as well. We use rectangular cells and since cells on the boundary can be corrupted by the background, we consider only grid cells

Figure 5.3: Effect of strong gradients on the object boundary. The object behind the cup causes many of the cells near the boundary of the object, especially on the handle, to have low activation scores. The correct gradients are at those locations as shown by the edge point matches, but the overall statistics of those cells are incorrect.

that lie within the interior of an object which is specified by a mask. For a template of $N \times M$ cells, let $h_{i,j}$ be a 39-dimensional vector (36 dimensions for HOG and 3 dimensions for the L*a*b color) for the spatial cell (i, j), for $i = 1 \ldots N$ and $j = 1 \ldots M$. Let $\mathcal{I}_\alpha$ be the set of cells (i, j) that are completely contained within the object mask. Then the region descriptor, R, is a $39 \cdot |\mathcal{I}_\alpha|$ dimensional vector obtained by concatenating the histogram of gradients and color for all cells in $\mathcal{I}_\alpha$:

$$R = [h_{i,j} : (i, j) \in \mathcal{I}_\alpha]. \tag{5.1}$$

5.2.1 Grid Optimization

Since we are using a rectangular grid, the position of the grid on the object can have a major effect on performance. For some grid positions, there may be very few cells that lie completely within the object. To capture the most information about the interior, we want to maximize the number of cells inside the object.

We perform a search over all grid positions $\mathcal{X}$ to maximize the number of cells. Let $X_0 = (x_0, y_0)$ be the position of the grid such that the top left corner is aligned with object mask as shown in Figure 5.4, and let w be the width in pixels of each cell. Then the set of grid positions we enumerate is:

$$\mathcal{X} = \{(x_0 - dx, y_0 - dy) : dx \in [0, w - 1], dy \in [0, w - 1]\}. \tag{5.2}$$

We define $\mathcal{I}_\alpha(X)$ to be the set of cells (i, j) that are contained entirely inside the object at grid position X, and $\mathcal{I}_\beta(X)$ to be the set of cells (i, j) that overlap some portion of the object. Then the optimum grid position X^* is:

$$X^* = \underset{X \in \mathcal{X}}{\operatorname{argmax}} |\mathcal{I}_\alpha(X)|. \tag{5.3}$$

If there are multiple grid positions that maximize $|\mathcal{I}_\alpha(X)|$, we choose the one that minimizes $|\mathcal{I}_\beta(X)|$. This maximizes the number of cells that are completely inside the interior and minimizes the overall number of grid cells used.

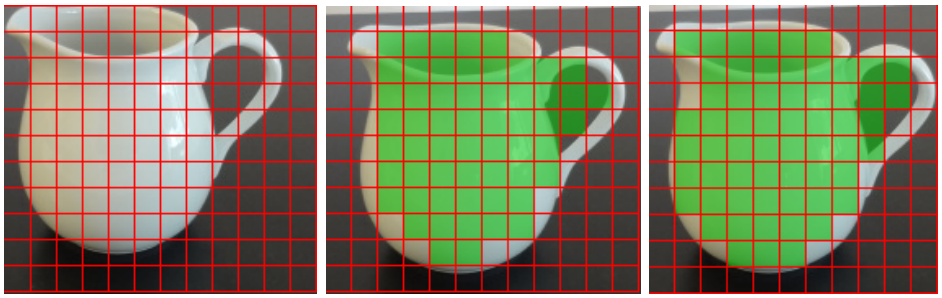


Figure 5.4: Grid optimization for region template. (left) Initial grid position X_0 with the top left corner of the grid aligned to the object. (center) Non-optimized grid position centered on the object contains 57 interior cells. (right) Optimized grid position contains 60 interior cells.

5.2.2 Uniform Regions

A HOG descriptor normalizes the gradient histogram of a cell with respect to the gradient magnitude of its neighbors. While normalization has been shown to be essential for good recognition performance [23], Figure 5.5 shows that it effectively amplifies noise in uniform regions when the neighboring cells are also uniform. If many training images of an object are provided, the HOG descriptor in these regions will essentially be random and the SVM will give zero weights to these regions. This essentially ignores these areas when performing recognition. For category recognition, where the interior appearance changes due to intraclass variations, such as from clothes on humans or materials differences of furniture, these regions are not very informative and should be ignored. However, for instance recognition these regions are from the exact same object and are informative. In the case of ESVMs with one training instance, the noise in these regions will just add random noise to the detection scores. Due to lighting effects, the descriptor in these regions will rarely match the model.

Ideally for uniform regions, the gradient magnitude for all orientations are 0, and negative weights are learned to penalize any gradient in these regions. Since HOG performs four different normalizations, we simply consider the magnitude of each normalization factor. Let $h_{i,j,k}$ be the histogram of gradients for the k^{th} normalization of cell (i, j), and $n_{i,j,k}$ be the normalization factor. For $n_{i,j,k} > \tau$, the relative gradient is useful, and we keep the normalization. However, for $n_{i,j,k} < \tau$, the normalization is effectively just amplifying the noise. In this case, we set the histogram of gradient with that normalization $h_{i,j,k}$ to be 0. Figure 5.6 shows the activation scores of individual cells with and without handling uniform regions. Accounting for uniform regions results in better cell-wise confidences of the object interior. For images with pixel values between 0 and 1, we choose τ using an average gradient magnitude of 0.2. This value was tuned on a pair of model images and kept constant for all the experiments.

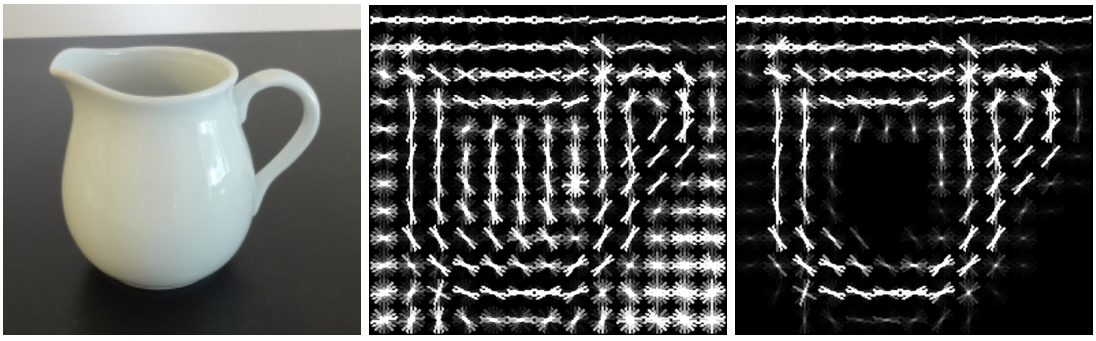


Figure 5.5: Modification of HOG to handle uniform regions. We show (left) the model image of a pitcher, (center) the original HOG descriptor, and (right) our HOG descriptor which accounts for uniform regions. The original HOG descriptor has random gradients from lighting effects in the uniform regions.

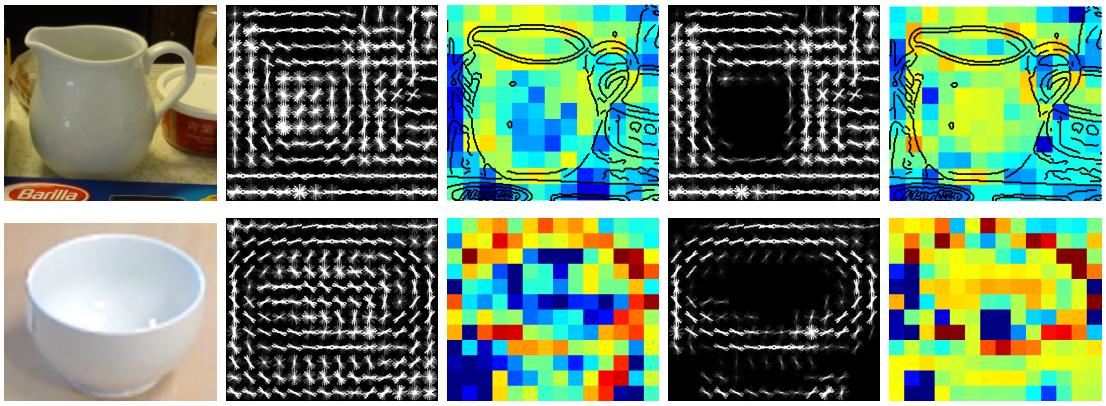


Figure 5.6: Comparison of HOG with and without uniform handling. We show the activation scores of individual cells computed using the method of Wang *et al.* [120]. From left to right, we show (1) a test image window, (2) the HOG descriptor and cell activation scores without uniform handling and (3) with uniform handling. The cell-wise confidences are much more representative of the visible portions of the object with uniform handling.

5.3 Implementation Details

For object instance detection, we assume that only one model image per viewpoint of an object is provided for training, similar to [22, 44, 58, 94, 98]. Unlike category recognition where training data for a given category can be mined from the Internet, obtaining multiple training images and ground truth for each object viewpoint is significantly more time consuming.

While positive data is very difficult to obtain, negative images are readily available from image search engines such as Flickr and Google Images. For training, it has been shown that the negative set can simply be a large set of random images [73]. For

evaluation, we use the background images provided by the datasets as the negative set.

To train BaRT for object detection, we concatenate the descriptors of boundary, B, and region, R, into a BaRT descriptor, $D = [B, R]$. We learn the weight of individual features for object detection using an ESVM. Object detection on a new image is performed by using a sliding window detector. We scan the boundary portion of the template at the pixel resolution, and we scan HOG at the standard cell resolution. Both sources of information are combined by resizing the output to the resolution of the image, and summing the boundary and region scores. Finally, detections are obtained by using mean-shift to perform non-maximal suppression.

5.4 Evaluation

In order to validate our BaRT representation for object instance detection, we compare our method with the baseline approaches of HOG+ESVM [73], rLINE2D [49], Gradient Networks [50] and using color only. We also systematically analyze the performance of each component of our representation, evaluating different combinations of boundary and region templates. In addition, we analyze the effect of using only the cells inside the object mask (+I) for HOG and color, and incorporating uniform region handling (+U) for HOG.

5.4.1 Object Detection

We evaluate the performance of BaRT for object instance detection. In the following experiments, an object is correctly detected if it satisfies the PASCAL overlap criterion [28, 58] with the ground truth bounding box. Figure 5.8 shows the Precision/Recall curves for each object and Table 5.1 summarizes the performance for each object using the Mean Average Precision (mAP). Example detection results are shown in Figure 5.7 using our full representation HOG+GN+color+I+U. For these experiments, we fix the size of each HOG and color cell to be the standard 8×8 for all objects. The object interior is specified by the object masks provided in the dataset.

From the table, using boundary alone with GN already performs significantly better than HOG. HOG cells on the boundary are easily corrupted by background clutter. By considering only the interior cells, HOG+I already perform significantly better than HOG. Combining explicit boundary information of rLINE2D and GN with HOG+I provides substantial improvement of 12% and 19% respectively over HOG. This validates our hypothesis that an explicit representation of boundary information is better than gradient statistics for specific objects. Using color by itself is not very good as many objects in cluttered scenes have similar colors. Combining color with the baseline templates provides 4-7% improvement, but only minimal gains when combined with HOG+rLINE2D

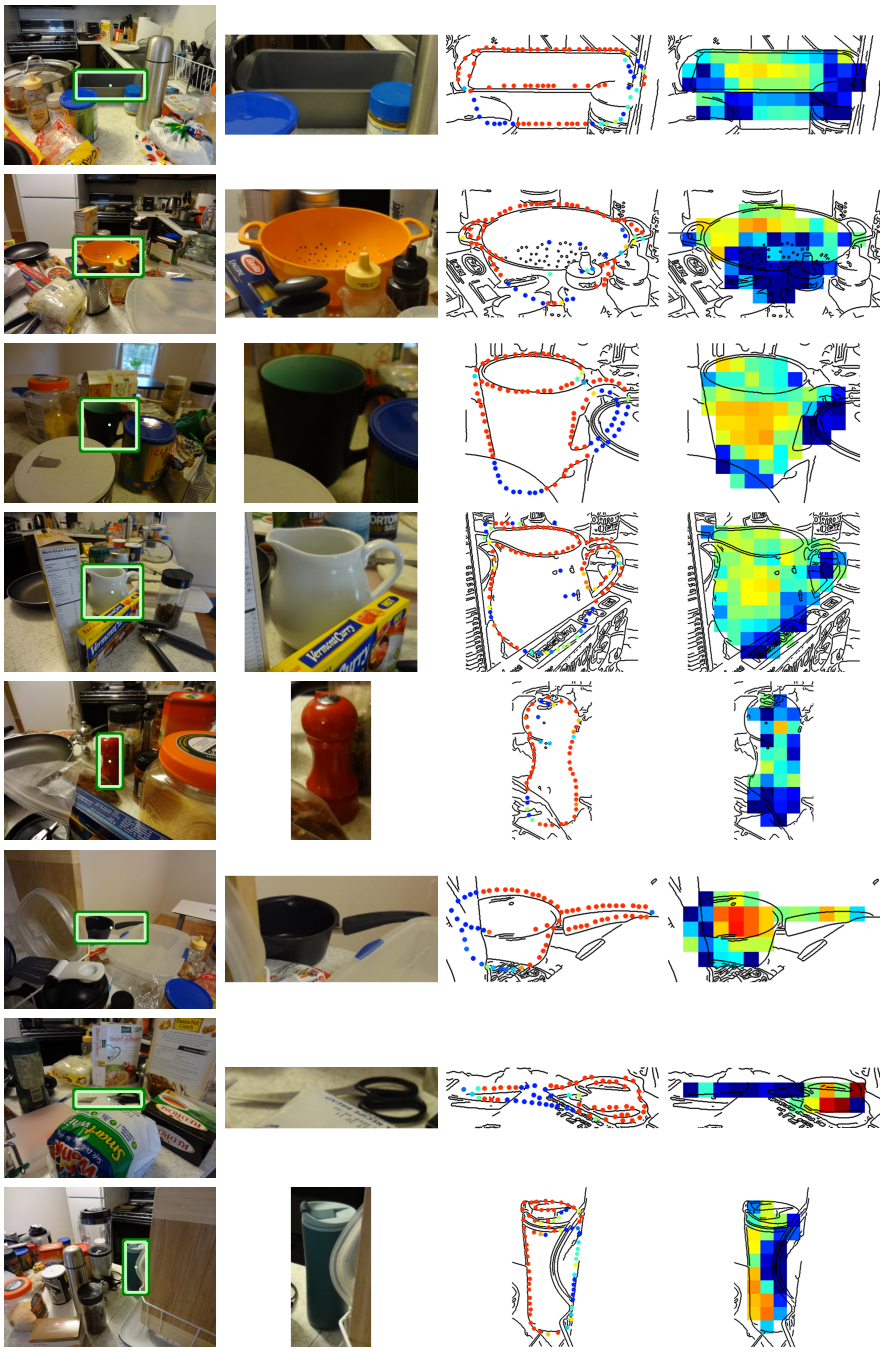


Figure 5.7: Example detections using HOG+GN+color+I+U in cluttered household environments. From left to right, we show (1) the original image with bounding box of detection, (2) the zoomed in view of the detection with HOG grid in white, (3) the HOG descriptor modified to handle uniform regions, (4) the boundary point matches, and (5) the region activation scores. Using our representation, the confidences of the boundary points and region cells correlate to whether they are visible or not.

Single	baseline	+I	+U	+I+U
HOG	0.49	0.59	0.46	0.55
LINE2D	0.16	-	-	-
rLINE2D	0.49	-	-	-
GN	0.65	-	-	-
color	0.06	0.05	-	-
HOG+rLINE2D	0.68	0.71	0.68	0.72
HOG+GN	0.78	0.78	0.75	0.77
HOG+color	0.53	0.65	0.50	0.63
rLINE2D+color	0.56	0.56	-	-
GN+color	0.72	0.71	-	-
HOG+rLINE2D+color	0.72	0.74	0.72	0.74
HOG+GN+color	0.78	0.78	0.77	0.77
Multiple	baseline	+I	+U	+I+U
HOG	0.46	0.53	0.40	0.50
LINE2D	0.09	-	-	-
rLINE2D	0.29	-	-	-
GN	0.70	-	-	-
color	0.07	0.12	-	-
HOG+rLINE2D	0.56	0.56	0.52	0.56
HOG+GN	0.80	0.80	0.78	0.80
HOG+color	0.51	0.59	0.44	0.55
rLINE2D+color	0.37	0.36	-	-
GN+color	0.76	0.77	-	-
HOG+rLINE2D+color	0.58	0.58	0.56	0.58
HOG+GN+color	0.80	0.80	0.78	0.80

Table 5.1: Object detection: Mean Average Precision. We show the performance of different combination of HOG, rLINE2D, GN and color templates. In addition, we evaluate the effect of using only interior cells (+I), handling uniformity (+U) as well as their combination (+I+U).

and HOG+GN as these templates are already able to reject most false positives.

Finally, we augment HOG to handle uniform regions. The performance was essentially the same when augmenting HOG+rLINE2D+color and HOG+GN+color. However, a benefit of correctly handling uniform regions is increased accuracy of cell-wise confidences for uniform regions as shown in Figure 5.6. Figure 5.7 shows that the scores of individual cells and the boundary correlate to whether the cell is visible or not. This is important for additional post-processing steps such as object segmentation and occlusion reasoning, where incorrect activation scores will negatively affect performance.

Figure 5.10 shows typical false positives of HOG+GN+color+I+U. In these cases, both the boundary and region align well to the image. Additional information such as depth or color would be needed to filter these false positives.

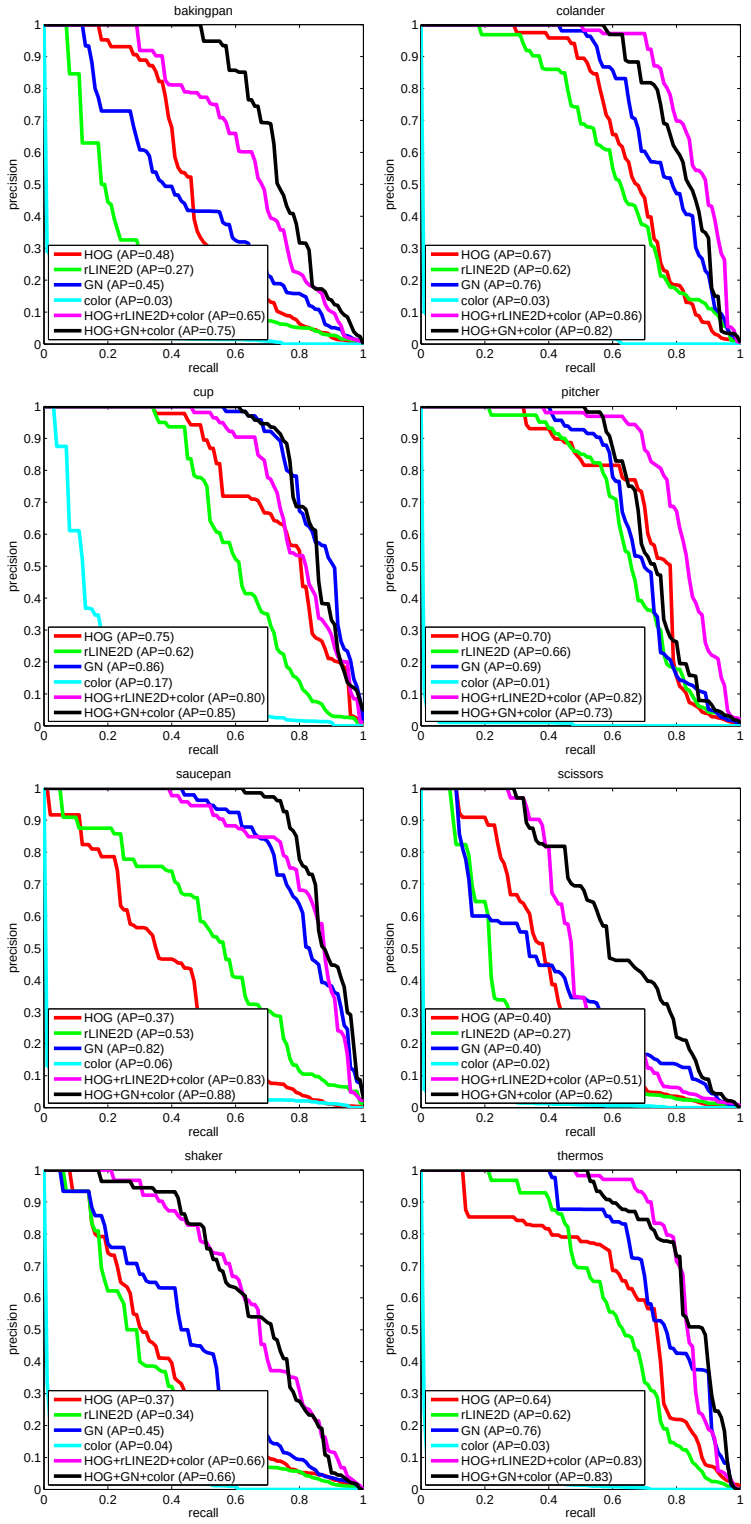


Figure 5.8: Precision/Recall curves for single view of CMU_KO8.

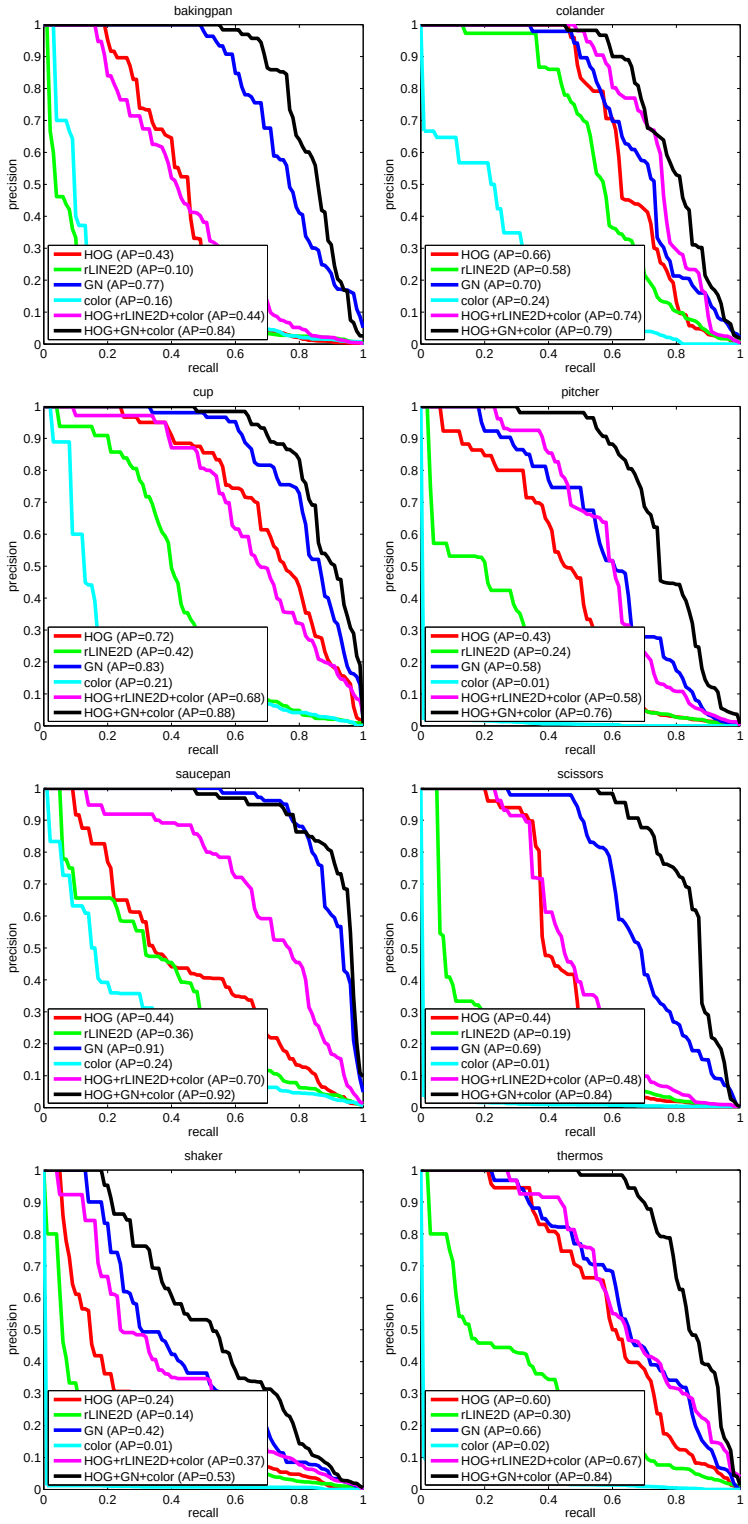


Figure 5.9: Precision/Recall curves for multiple view of CMU_KO8.

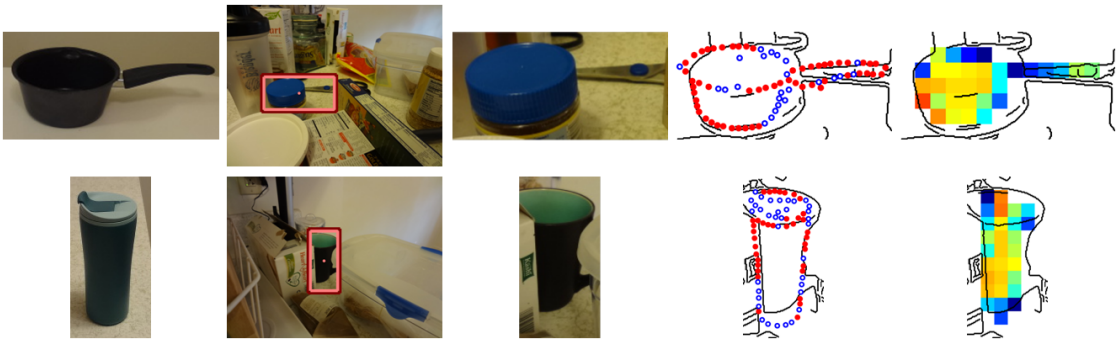


Figure 5.10: Example false positives of HOG+GN+color+I+U. From left to right, we show (1) the model image, (2) a false detection, (3) the zoomed in view of the false detection, (4) the boundary matches, and (5) the activation scores of region cells. These false detections have high scores for both boundary and region.

5.5 Discussion

In this chapter, we propose BaRT, a more complete representation for tackling the challenging problem of detecting feature-poor objects in cluttered environments. This representation improves over existing state-of-the-art approaches which represent feature-poor objects by using only boundary [44, 49] or coarse gradient statistics [24, 73]. The main contribution of this chapter is to demonstrate that combining the strengths of an explicit representation of the boundary and a coarse representation of the interior region results in a significantly better representation of feature-poor objects. We model the boundary of an object using a sparse set of edge points and the region using HOG cells contained within the silhouette of the object. Since HOG does not correctly capture uniform regions, we modify HOG to represent both textured and untextured information. Our results demonstrate that the BaRT representation can significantly improve performance over state-of-the-art systems for object instance detection.

Part IV

Occlusion Reasoning

Chapter 6

Representation under Arbitrary Viewpoint

Occlusions are common in real world scenes and are a major obstacle to robust object detection. In particular, instance detection requires recognizing objects under arbitrary viewpoint with severe occlusions as shown in Figure 6.1. For feature-poor objects, occlusions increase the shape ambiguity leading to false positives with higher scores than true positives that are severely occluded. In this chapter, we propose (i) a concise model of occlusions under arbitrary viewpoint without requiring additional training data and (ii) a method to capture global visibility relationships without combinatorial explosion.

In the past, occlusion reasoning for object detection has been extensively studied [41, 89, 108]. One common approach is to model occlusions as regions that are inconsistent with object statistics. Girshick *et al.* [38] use an occluder part in their grammar model when all parts cannot be placed. Wang *et al.* [120] use the scores of individual HOG filter cells, while Meger *et al.* [75] use depth inconsistency from 3D sensor data to classify occlusions. Local coherency of occlusions are often enforced with a Markov Random Field [33] to reduce noise in these classifications. Li *et al.* [66] use RANSAC to generate a large set of hypotheses and hallucinate points at positions where there is high error.

While assuming that any inconsistent region is an occlusion is valid if occlusions happen uniformly over an object, it ignores the fact there is structure to occlusions for many objects. For example, in real world environments, objects are usually occluded by other objects resting on the same surface. Thus it is often more likely for the bottom of an object to be occluded than the top of an object [27].

Recently, researchers have attempted to learn the structure of occlusions from data [36, 57]. With enough data, these methods can learn an accurate model of occlusions. However, obtaining a broad sampling of occluder objects is usually difficult, resulting in biases to the occlusions of a particular dataset. This becomes more problematic when considering object detection under arbitrary view [44, 109, 112]. Learning approaches need to

Figure 6.1: Example detections of (left) cup and (right) pitcher under severe occlusions after occlusion reasoning.

learn a new model for each view of an object. This is intractable, especially when recent studies [44] have claimed that approximately 2000 views are needed to sample the view space of an object. A key contribution of our approach is to represent occlusions under arbitrary viewpoint without requiring additional training data of occlusions. We demonstrate that our approach accurately models occlusions, and that learning occlusions from data does not give better performance.

Researchers have shown in the past that incorporating 3D geometric understanding of scenes [7, 47] improves the performance of object detection systems. Following these approaches, we propose to reason about occlusions by explicitly modeling 3D interactions of objects. For a given environment, we compute physical statistics of objects in the scene and represent an occluder as a probabilistic distribution of 3D blocks. The physical statistics need only be computed once for a particular environment and can be used to represent occlusions for many objects in the scene. By reasoning about occlusions in 3D, we effectively provide a unified occlusion model for different viewpoints of an object as well as different objects in the scene.

We incorporate occlusion reasoning with object detection by: (i) a bottom-up stage which hypothesizes the likelihood of occluded regions from the image data, followed by (ii) a top-down stage which uses prior knowledge represented by the occlusion model to score the plausibility of the occluded regions. We combine the output of the two stages into a single measure to score a candidate detection.

The focus of this chapter is to demonstrate that a relatively simple model of 3D interaction of objects can be used to represent occlusions effectively for instance detection of feature-poor objects under arbitrary view. Recently, there has been significant progress in simple and efficient template matching techniques [44, 45] for instance detection. These approaches work extremely well when objects are largely visible, but degrade

rapidly when faced with strong occlusions in heavy background clutter. We evaluate our approach by incorporating occlusion reasoning with the Boundary and Region Templates of Chapter 5, and demonstrate significant improvement in detection performance on a challenging occlusion dataset.

6.1 Occlusion Model

Occlusions in real world scenes are often caused by a solid object resting on the same surface as the object of interest. In our model, we approximate occluding objects by their 3D bounding box and demonstrate how to compute occlusion statistics of an object under different camera viewpoints, c, defined by an elevation angle ψ and azimuth θ.

Let the 2D view of an object under camera c be represented by K markers[1] $\mathcal{Z} = \{Z_1, ..., Z_K\}$. Each marker Z_i captures the local information centered around coordinate (x_i, y_i) on the object. These markers can capture any type of information from local shape to texture and color. A marker, for example, can be the center of a HOG cell [23], a SIFT keypoint [70], a LINE2D edge point [44], or a Hough voting patch [35]. By using this object representation, our occlusion model can augment any object detector which returns the probability, p_i, that each marker, Z_i, matches an image location. Here, we follow previous research and assume that the matching probability, p_i, is a good indicator of how likely a marker is visible [38, 120]. In addition, let each marker, Z_i, have a weight, w_i, which indicates its importance and influence. We define the set of tuples $\mathcal{M} = \{(Z_i, p_i, w_i) | 1 \leq i \leq K\}$ as the *matching pattern*.

In this chapter, we begin by assuming that the matching probabilities are binary (i.e., $p \in \{0, 1\}$) and that the weights are 1. We show how to relax these assumptions in the next chapter. For conciseness of notation, let the visibility states of the K markers be represented by a set of binary variables $\mathcal{V} = \{V_1, ..., V_K\}$ such that if $V_i = 1$, then Z_i is visible. For occlusions O_c under a particular camera viewpoint c, we want to compute occlusion statistics for each marker in $\mathcal{Z}$. Unlike other occlusion models which only compute an occlusion prior $P(V_i|O_c)$, we propose to also model the global relationship between visibility states, $P(V_i|\mathcal{V}_{\text{-}i}, O_c)$ where $\mathcal{V}_{\text{-}i} = \mathcal{V} \backslash V_i$. Through our derivation, we observe that $P(V_i|O_c)$ captures the classic intuition that the bottom of the object is more likely to be occluded than the top. More interesting is $P(V_i|\mathcal{V}_{\text{-}i}, O_c)$ which captures the structural layout of an occlusion. The computation of these occlusion properties reduce to integral geometry [100] (an entire field dedicated to geometric probability theory).

We make a couple of approximations to tractably derive the occlusion statistics. Specifically, since objects which occlude each other are usually physically close together,

[1] We introduce the term *marker* to deliberately avoid using the term *feature* which has been significantly overloaded in the literature.

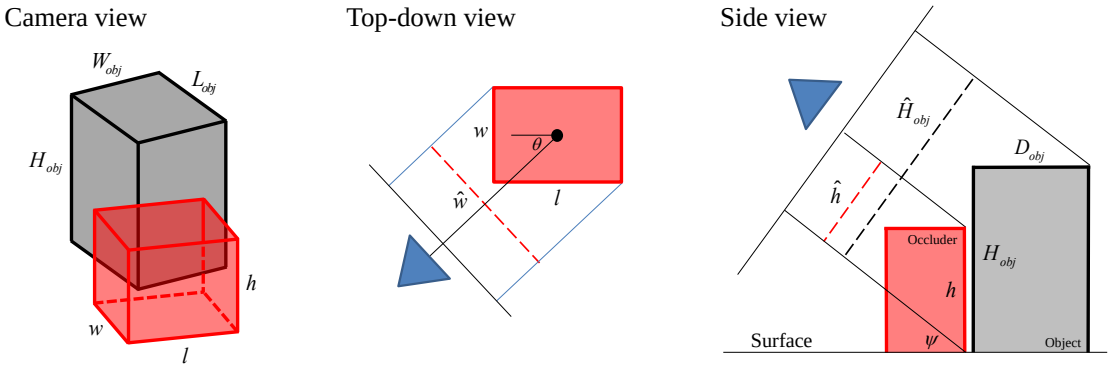


Figure 6.2: Occlusion model under arbitrary viewpoint. (left) Example camera view of an object (gray) and occluder (red). (center) Projected width of occluder, $\hat{w}$, for a rotation of θ. (right) Projected height of occluder, $\hat{h}$, and projected height of object, $\hat{H}_{obj}$, for an elevation angle of ψ. An occluder needs a projected height of $\hat{h} \geq \hat{H}_{obj}$ to fully occlude the object.

we approximate the objects to be on the same support surface. and we approximate the perspective effects over the range of object occlusions to be negligible.

6.1.1 Representation Under Different Viewpoints

The likelihood that a marker on an object is occluded depends on the angle the object is being viewed from. Most methods that learn the structure of occlusions from data [36] require a separate occlusion model for each view of every object. These methods do not scale well when considering detection of many objects under arbitrary view.

In the following, we propose a unified representation of occlusions under arbitrary viewpoint of an object. Our method requires only the statistics of object dimensions, which is obtained once for a given environment and can be shared across many objects for that environment.

The representation we propose is illustrated in Figure 6.2. For a specific viewpoint, we represent the portion of a block that can occlude the object as a bounding box with dimensions corresponding to the projected height $\hat{h}$ and the projected width $\hat{w}$ of the block. The projected height and width are the observed height and width of a block to the viewer.

The object of interest, on the other hand, is represented by its silhouette in the image. Initially, we derive our model using the bounding box of the silhouette with dimensions $\hat{H}_{obj}$ and $\hat{W}_{obj}$, and then relax our model to use the actual silhouette (Section 6.1.4).

First, we compute the projected width $\hat{w}$ of an occluder with width w and length l as shown by the top-down view in Figure 6.2. In our convention, $\hat{w} = w$ for an azimuth

of $\theta = 0$. Using simple geometry, the projected width is:

$$\hat{w}(\theta) = w \cdot |\cos\theta| + l \cdot |\sin\theta|. \tag{6.1}$$

Since θ is unknown for an occluding object, we obtain a distribution of $\hat{w}$ assuming all rotations about the vertical axis are equally likely. The distribution of $\hat{w}$ over $\theta \in [0, 2\pi]$ is equivalent to the distribution over any $\frac{\pi}{2}$ interval. Thus, the distribution of $\hat{w}$ is computed by transforming a uniformly distributed random variable on $[0, \frac{\pi}{2}]$ by Equation 6.1. The resulting probability density of $\hat{w}$ is given by:

$$p_{\hat{w}}(\hat{w}) = \begin{cases} \frac{2}{\pi}\left(1 - \frac{\hat{w}^2}{w^2+l^2}\right)^{-\frac{1}{2}}, & w \leq \hat{w} < l \\ \frac{4}{\pi}\left(1 - \frac{\hat{w}^2}{w^2+l^2}\right)^{-\frac{1}{2}}, & l \leq \hat{w} < \sqrt{w^2+l^2}. \end{cases} \tag{6.2}$$

The full derivation of this density is provided in Appendix B.1.

Next, we compute the projected height $\hat{h}$ of an occluder as illustrated by the side view of Figure 6.2. For an elevation angle ψ and occluding block with height h, the projected height $\hat{h}$ is:

$$\hat{h}(\psi) = h \cdot \cos\psi. \tag{6.3}$$

This corresponds to the maximum height that can occlude the object given our assumptions.

The projected height of the object, $\hat{H}_{obj}$, is slightly different in that it accounts for the apparent height of the object silhouette. An object is fully occluded vertically only if $\hat{h} \geq \hat{H}_{obj}$. To compute $\hat{H}_{obj}$, we need the distance, D_{obj}, from the closest edge to the farthest edge of the object. Following the computation of the projected width $\hat{w}$, we have $D_{obj}(\theta) = W_{obj} \cdot |\sin\theta| + L_{obj} \cdot |\cos\theta|$. The projected height of the object at an elevation angle ψ is then given by:

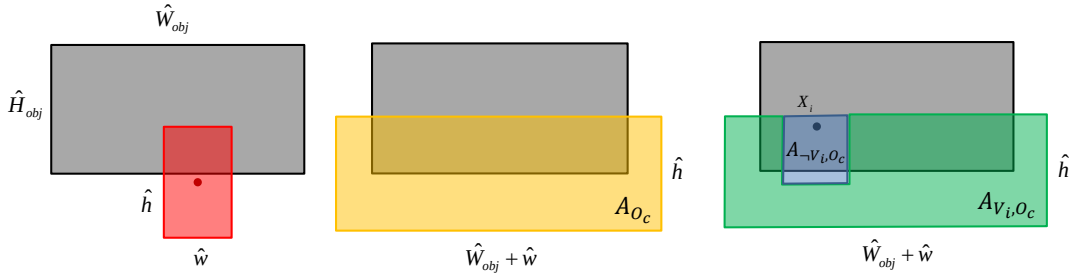
$$\hat{H}_{obj}(\theta, \psi) = H_{obj} \cdot |\cos\psi| + D_{obj}(\theta) \cdot |\sin\psi|. \tag{6.4}$$

Finally, the projected width of the object $\hat{W}_{obj}$ is computed using the aspect ratio of the silhouette bounding box.

6.1.2 Occlusion Prior

Given the representation derived in Section 6.1.1, we want to compute a probability for a marker on the object being occluded. Many systems which attempt to address occlusions assume that they occur randomly and uniformly across the object. However, recent studies [27] have shown that there is structure to occlusions for many objects.

We begin by deriving the occlusion prior using an occluding block with projected dimensions $(\hat{w}, \hat{h})$ and then extend the formulation to use a probabilistic distribution of occluding blocks. The occlusion prior specifies the probability $P(V_i|O_c)$ that a marker,



(a) (b) (c)

Figure 6.3: Computation of the occlusion prior. (a) We consider the center positions of a block (red) which occlude the object. The base of the block is always below the object, since we assume they are on the same surface. (b) The set of positions is defined by the yellow rectangle which has area A_{O_c}. (c) The set of positions which occlude the object while keeping Z_i visible is defined by the green region which has area A_{V_i,O_c}.

Z_i, on the object at position (x_i, y_i) is visible given an occlusion of the object. This involves estimating the area, A_{O_c}, covering the set of block positions that occlude the object (shown by the yellow region in Figure 6.3b), and estimating the area, A_{V_i,O_c}, covering the set of block positions that occlude the object while keeping Z_i visible (shown by the green region in Figure 6.3c). The occlusion prior is then just a ratio of these two areas:

$$P(V_i|O_c) = \frac{A_{V_i,O_c}}{A_{O_c}}. \tag{6.5}$$

From Figure 6.3b, a block (red) will occlude the object if its center is inside the yellow region. The area of this region, A_{O_c}, is:

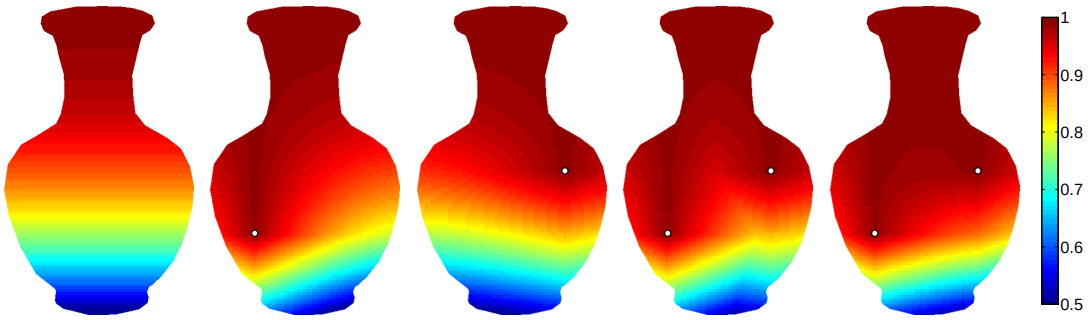
$$A_{O_c} = (\hat{W}_{obj} + \hat{w}) \cdot \hat{h}. \tag{6.6}$$

Next, from Figure 6.3c, this region can be partitioned into a region where the occluding block occludes Z_i (blue) and a region which does not (green). A_{V_i,O_c} corresponds to the area of the green region and can be computed as:

$$A_{V_i,O_c} = \hat{W}_{obj} \cdot \hat{h} + \hat{w} \cdot \min(\hat{h}, y_i). \tag{6.7}$$

The derivation is provided in Appendix B.2.

Now that we have derived the occlusion prior using a particular occluding block, we extend the formulation to a distribution of blocks. Let $p_{\hat{w}}(\hat{w})$ and $p_{\hat{h}}(\hat{h})$ be distributions of $\hat{w}$ and $\hat{h}$ respectively. To simplify notation, we define $\mu_{\hat{w}} = \mathbb{E}_{p_{\hat{w}}(\hat{w})}[\hat{w}]$ and $\mu_{\hat{h}} = \mathbb{E}_{p_{\hat{h}}(\hat{h})}[\hat{h}]$ to be the expected width and height of the occluders under these distributions, and define $\beta_y(y_i) = \int \min(\hat{h}, y_i) \cdot p_{\hat{h}}(\hat{h})\, d\hat{h}$. The average areas, A_{O_c} and A_{V_i,O_c}, are then



(a) (b) (c) (d) (e)

Figure 6.4: Example of (a) occlusion prior $P(V_i|O_c)$, (b,c) conditional likelihood $P(V_i|V_j, O_c)$ and $P(V_i|V_k, O_c)$ given two separate markers Z_j and Z_k individually, (d) approximate conditional likelihood $P(V_i|V_j, V_k, O_c)$ from Equation 6.12, and (e) explicit conditional likelihood $P(V_i|V_j, V_k, O_c)$ from Equation 6.10.

given by:

$$A_{O_c} = (\hat{W}_{obj} + \mu_{\hat{w}}) \cdot \mu_{\hat{h}}, \tag{6.8}$$

$$A_{V_i,O_c} = \hat{W}_{obj} \cdot \mu_{\hat{h}} + \mu_{\hat{w}} \cdot \beta_y(y_i). \tag{6.9}$$

This derivation assumes that the distribution $p_{\hat{w},\hat{h}}(\hat{w}, \hat{h})$ can be separated into $p_{\hat{w}}(\hat{w})$ and $p_{\hat{h}}(\hat{h})$. For household objects, we empirically verified that this approximation holds. In practice, the areas are computed by discretizing the distributions and Figure 6.4(a) shows an example occlusion prior. Figures 6.5 and 6.6 show how the distribution changes under different camera viewpoints. Our model is able to capture that the top of the object is much less likely to be occluded when viewed from a higher elevation angle than from a lower one. This is because the projected height of occluders is shorter the higher the elevation angle.

6.1.3 Occlusion Conditional Likelihood

Most occlusion models only account for local coherency and the prior probability that a marker on the object is occluded. Ideally, we want to compute a global relationship between all visibility states $\mathcal{V}$ on the object. While this is usually infeasible combinatorially, we show how a tractable approximation can be derived in the following section.

Let $\mathcal{Z}_{\mathcal{V}_{\text{-}i}}$ be the visible subset of $\mathcal{Z}$ according to $\mathcal{V}_{\text{-}i}$. We want to compute the probability $P(V_i|\mathcal{V}_{\text{-}i}, O_c)$ that a marker Z_i is visible given the visibility of $\mathcal{Z}_{\mathcal{V}_{\text{-}i}}$. Following Section 6.1.2, the conditional likelihood is given by:

$$P(V_i|\mathcal{V}_{\text{-}i}, O_c) = \frac{A_{V_i,\mathcal{V}_{\text{-}i},O_c}}{A_{\mathcal{V}_{\text{-}i},O_c}}. \tag{6.10}$$

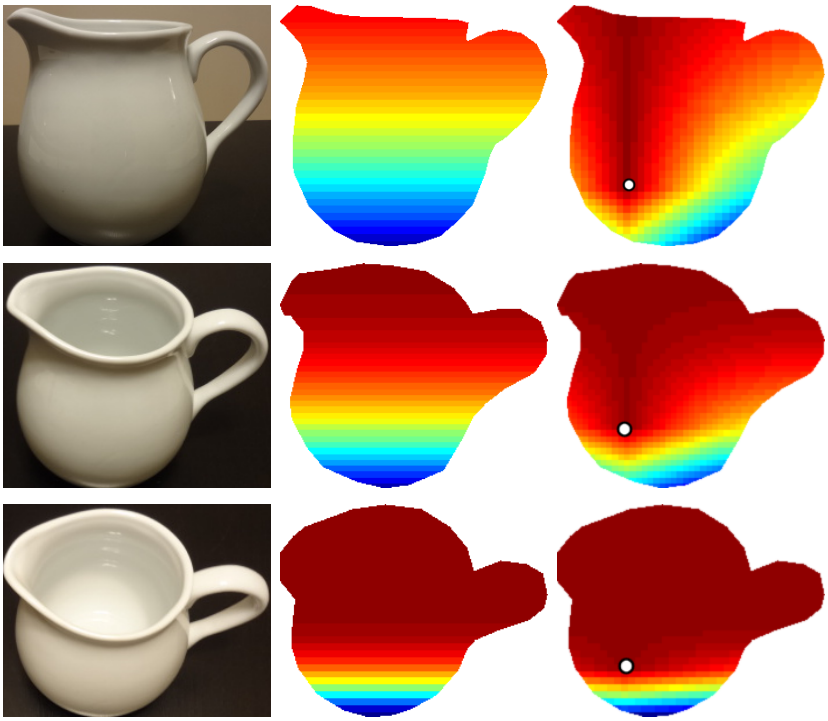


Figure 6.5: The occlusion prior and conditional distribution under different camera viewpoints for a pitcher. We show the (left) model viewpoint, (center) occlusion prior and (right) occlusion conditional likelihood.

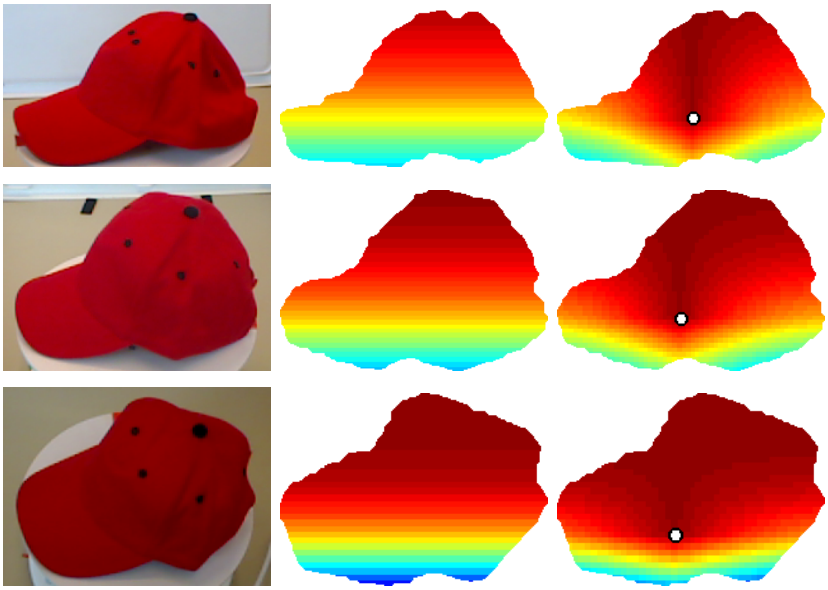


Figure 6.6: The occlusion prior and conditional distribution under different camera viewpoints for a cap. We show the (left) model viewpoint, (center) occlusion prior and (right) occlusion conditional likelihood.

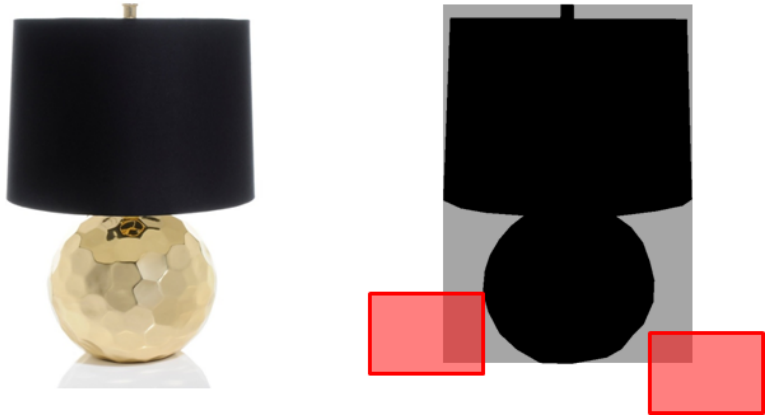


Figure 6.7: Using an arbitrary object silhouette. (left) Object. (right) Two blocks in red which occlude the object bounding box in gray, but not the silhouette in black.

We first consider the case where we condition on one visible marker, Z_j (i.e., $\mathcal{Z}_{\text{-}i} = \{Z_j\}$). To compute $P(V_i|V_j, O_c)$, we already have A_{V_j,O_c} from Equation 6.9, so we just need A_{V_i,V_j,O_c}. The computation follows from Section 6.1.2, so we omit the details and just provide the results below. The detailed derivation is provided in Appendix B.3. If we let $\beta_x(x_i, x_j) = \int \min(\hat{w}, |x_i - x_j|) \cdot p_{\hat{w}}(\hat{w})\ d\hat{w}$, then:

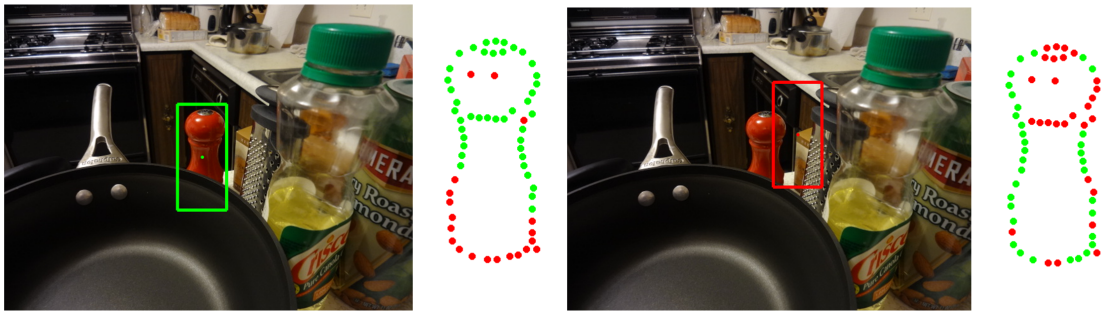
$$\begin{aligned} A_{V_i,V_j,O_c} = (\hat{W}_{obj} - |x_i - x_j|) \cdot \mu_{\hat{h}} \\ + \left(\int_0^{|x_i - x_j|} (|x_i - x_j| - \hat{w}) \cdot p_{\hat{w}}(\hat{w})\ d\hat{w} \right) \cdot \mu_{\hat{h}} \\ + \beta_x(x_i, x_j) \cdot \beta_y(y_i) + \mu_{\hat{w}} \cdot \beta_y(y_j). \end{aligned} \tag{6.11}$$

We can generalize the conditional likelihood to k visible markers (i.e., $|\mathcal{Z}_{\text{-}i}| = k$) by counting as above, however, the number of cases increases combinatorially. We make the approximation that the marker $Z_j \in \mathcal{Z}_{\text{-}i}$ with the highest conditional likelihood $P(V_i|V_j, O_c)$ provides all the information about the visibility of Z_i. This observation assumes that $V_i \perp \{\mathcal{V}_{\text{-}i} \backslash V_j\}|V_j$ and allows us to compute the global visibility relationship $P(V_i|\mathcal{V}_{\text{-}i}, O_c)$ without combinatorial explosion. The approximation of $P(V_i|\mathcal{V}_{\text{-}i}, O_{\text{-}i})$ is then:

$$P(V_i|\mathcal{V}_{\text{-}i}, O_c) \approx P(V_i|V_j^*, O_c), \tag{6.12}$$

$$V_j^* = \underset{V_j \in \mathcal{V}_{\text{-}i}}{\operatorname{argmax}} P(V_i|V_j, O_c). \tag{6.13}$$

For example, Figure 6.4(d,e) shows the approximate conditional likelihood and the exact one for $|\mathcal{Z}_{\text{-}i}| = 2$. Figures 6.5 and 6.6 show how the distribution changes under different camera viewpoints.



(a) True positive (b) False positive

Figure 6.8: Examples of occlusion hypotheses. (a) For a true detection, the occluded markers (red) are consistent with our model. (b) For a false positive, the top of the object is hypothesized to be occluded while the bottom is visible, which is highly unlikely according to our model.

6.1.4 Arbitrary Object Silhouette

The above derivation can easily be relaxed to use the actual object silhouette. The idea is to subtract the area, A_s, covering the set of block positions that occlude the object bounding box but not the silhouette from the areas described in Sections 6.1.2 and 6.1.3. Figure 6.7 shows two example block positions. An algorithm to compute A_s is provided in Appendix B.4. The occlusion prior and conditional likelihood are then given by:

$$P(V_i|O_c) = \frac{A_{V_i,O_c} - A_s}{A_{O_c} - A_s}, \tag{6.14}$$

$$P(V_i|\mathcal{V}_{\text{-}i}, O_c) = \frac{A_{V_i,\mathcal{V}_{\text{-}i},O_c} - A_s}{A_{\mathcal{V}_{\text{-}i},O_c} - A_s}. \tag{6.15}$$

6.2 Combining with Object Detection

Given our occlusion model from Section 6.1, we augment an object detection system by (i) a bottom-up stage which hypothesizes occluded regions using the object detector, followed by (ii) a top-down stage which measures the consistency of the hypothesized occlusion with our model. We explore using the occlusion prior and occlusion conditional likelihood for scoring and show in our evaluation that both are informative for object detection.

6.2.1 Occlusion Hypothesis

Our algorithm can be used to score the matching patterns of any object detector which returns a binary prediction that markers on the object are matched. Obtaining the matching pattern depends on the individual object detector. Some detectors, such as our

Gradient Network method, return marker scores that can be interpreted as probabilities. This interpretation is important for our model as we need to know when a marker is more likely to be matched than not matched. For these detectors, we can simply threshold the matching probabilities at 0.5 to obtain the occlusion hypothesis.

However, many detectors do not return marker scores which can be interpreted as matching probabilities (e.g., the decomposed cell-wise score [120] of HOG and the point-wise similarity metrics of LINE2D [44] and rLINE2D [49]). To calibrate the raw marker scores, we use the Extreme Value Theory [101] because obtaining many positive examples of occlusions is tedious and the method only requires the distribution of scores on negative examples. We calibrate each marker independently using its raw matching scores on randomly sampled detections in background clutter as negatives. Once the marker scores are calibrated, we obtain the occlusion hypothesis by thresholding at 0.5.

6.2.2 Occlusion Scoring

Given the hypothesized visibility labeling $\mathcal{V}^\zeta$ for a sliding window location ζ from Section 6.2.1, we want a metric of how well the occluded regions agree with our model. Intuitively, we should penalize markers that are hypothesized to be occluded by the object detector (Section 6.2.1) but are highly likely to be visible according to our occlusion model. From this intuition, we propose the following detection score:

$$\text{score}_f(\mathcal{V}^\zeta) = \frac{1}{K}\sum_{i=1}^{K} V_i^\zeta - f(\mathcal{V}^\zeta), \tag{6.16}$$

where $f(\mathcal{V})$ is a penalty function for occlusions. A higher score indicates a more confident detection, and for detections with no occlusion, the score is 1. For detections with occlusion, the penalty $f(\mathcal{V})$ is higher the more occluded markers which are inconsistent with the model. In the following, we propose two penalty functions, $f_{\text{OPP}}(\mathcal{V})$ and $f_{\text{OCLP}}(\mathcal{V})$, based on the occlusion prior and occlusion conditional likelihood of Section 6.1.

Occlusion Prior Penalty

The occlusion prior penalty (OPP) gives high penalty to locations that are hypothesized to be occluded but have a high prior probability $P(V_i^\zeta|O_c)$ of being visible. Intuitively, once the prior probability drops below some level λ, the marker should be considered part of a valid occlusion and should not be penalized. This corresponds to a hinge loss function $\Gamma(P,\lambda) = \max\left(\frac{P-\lambda}{1-\lambda}, 0\right)$. The linear penalty we use is then:

$$f_{\text{OPP}}(\mathcal{V}) = \frac{1}{K}\sum_{i=1}^{K}\left[(1-V_i)\cdot\Gamma\left(P(V_i|O_c),\lambda_p\right)\right]. \tag{6.17}$$

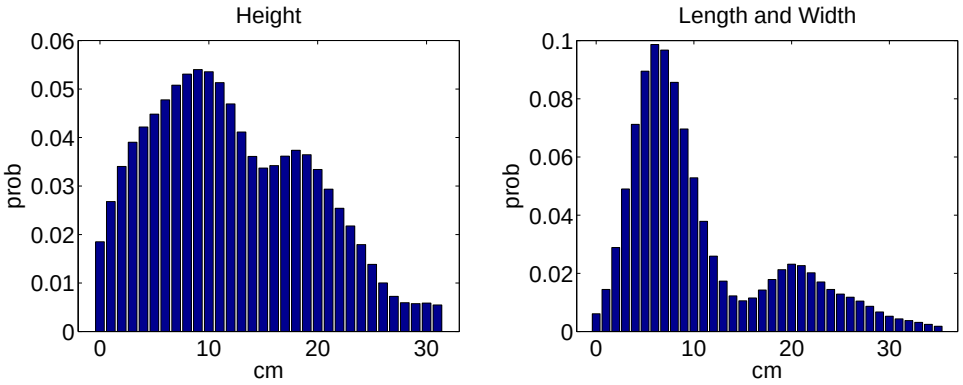


Figure 6.9: Distribution of (left) heights and (right) length and width of occluders in household environments.

Occlusion Conditional Likelihood Penalty

The occlusion conditional likelihood penalty (OCLP), on the other hand, gives high penalty to locations that are hypothesized to be occluded but have a high probability $P(V_i|\mathcal{V}_{\text{-}i}, O_c)$ of being visible given the visibility labeling of all other markers $\mathcal{V}_{\text{-}i}$. Using the same penalty function formulation as the occlusion prior penalty, we have that:

$$f_{\text{OCLP}}(\mathcal{V}) = \frac{1}{K}\sum_{i=1}^{K}\left[(1-V_i)\cdot\Gamma\left(P(V_i|\mathcal{V}_{\text{-}i}, O_c), \lambda_c\right)\right]. \tag{6.18}$$

6.3 Evaluation

We evaluate our occlusion model's performance for object instance detection on the CMU Kitchen Occlusion Dataset by incorporating occlusion reasoning with the Boundary and Region Template representation of Chapter 5. We explore the benefits of (i) using only the bottom-up stage and (ii) incorporating prior knowledge of occlusions with the top-down stage. When evaluating the bottom-up stage, we hypothesize the occluded region and consider the score of only the visible portions of the detection.

The parameters of our occlusion model were calibrated on images not in the dataset and were kept the same for all objects and all experiments. The occlusion parameters were set to $\lambda_p = 0.5$ and $\lambda_c = 0.95$. We show in Section 6.3.5 that our model is not sensitive to the exact choice of these parameters.

6.3.1 Distribution of Occluder Sizes

The distribution of object sizes varies in different environments. For a particular scenario, it is natural to only consider objects as occluders if they appear in that environment. The statistics of objects can be obtained from the Internet [59] or, in the household

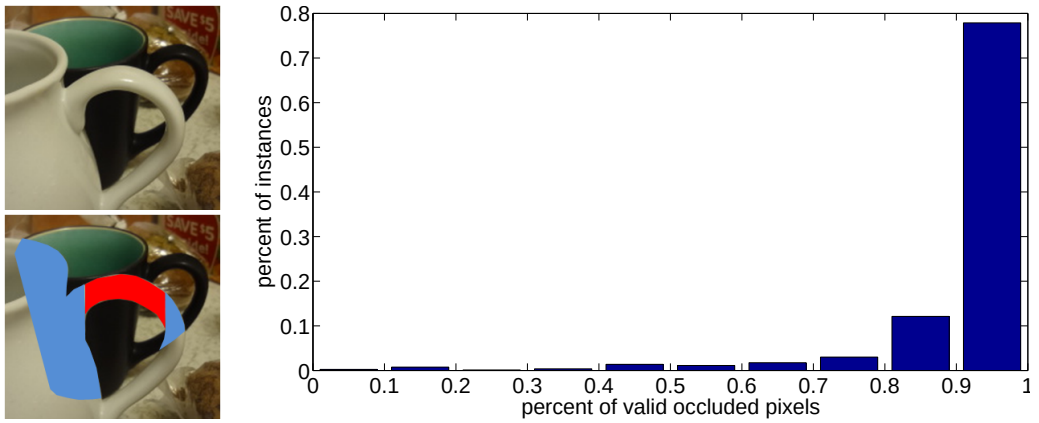


Figure 6.10: Validity of occlusion model. (left) We show in blue, the occluded pixels which satisfy our approximation, and in red, those that do not. (right) For each object instance in the CMU_KO8 dataset [49], we evaluate the percentage of occluded pixels which satisfy our approximation that they can be explained by a bounding box with a base lower than the object. For 80% of the images, over 90% of the occluded pixels can be explained with our model.

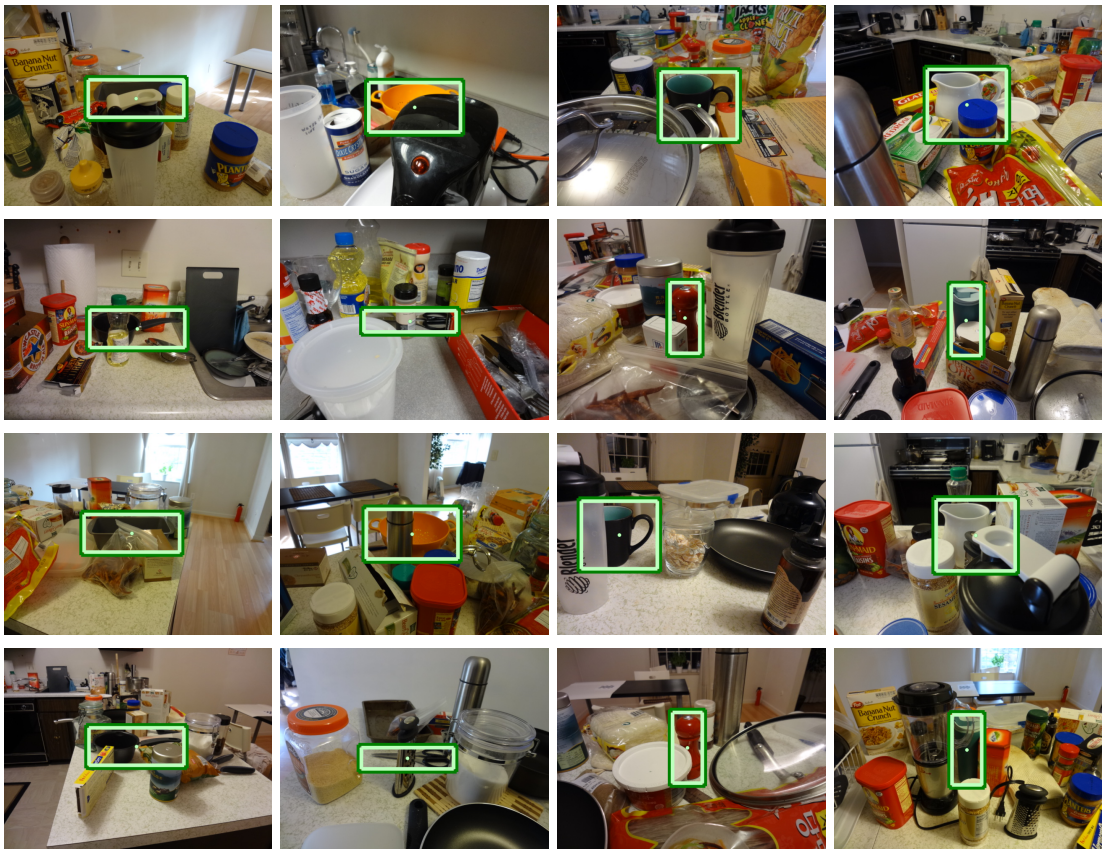


Figure 6.11: Example detection results under severe occlusions in cluttered household environments.

scenario, simply from 100 common household items. Figure 6.9 shows the distributions for household objects.

From real world dimensions, we can compute the projected width and height distributions, $p_{\hat{w}}(\hat{w})$ and $p_{\hat{h}}(\hat{h})$, for a given camera viewpoint. The projected width distribution is the same for all viewpoints and is obtained by computing the probability density from Equation 6.2 for each pair of width and length measurement. These densities are discretized and averaged to give the final distribution of $\hat{w}$.

The projected height distribution, on the other hand, depends on the elevation angle ψ. From Equation 6.3, $\hat{h}$ is a factor $\cos\psi$ of h. Thus, the projected height distribution, $p_{\hat{h}}(\hat{h})$, is computed by subsampling $p_h(h)$ by $\cos\psi$.

6.3.2 Validity of Occlusion Model

To derive the occlusion probabilities, we approximated occluder objects to be bounding boxes which are on the same surface as the object. While this approximation is consistent with the occlusion types observed by Dollar *et al.* in [27], we further validate the approximation on the CMU Kitchen Occlusion Dataset. Given the groundtruth occlusion labels in the dataset, we consider an occluded pixel to be consistent with the approximation if there are no un-occluded object pixels below it. From Figure 6.10, for 80% of the images, over 90% of the occluded pixels are consistent.

6.3.3 Object Detection

We evaluate the performance for object instance detection. An object is correctly detected if the intersection-over-union (IoU) of the predicted bounding box and the ground truth bounding box is greater than 0.5. Figure 6.12 and 6.13 shows the Precision/Recall plot for single and multiple view respectively and Table 6.1 summarizes the performance with the Mean Average Precision. A few example detections are shown in Figure 6.11.

From the table, OPP improves the performance for some templates, but hurts performance for others. OCLP, on the other hand, provides significant gains for all of the templates except for color and GN+color. Contrary to what one would naturally believe, color is actually not a very good cue for whether a marker is occluded or not because many objects have very similar colors. When the color of the occluder is similar to the object, the matching probability and thus the occlusion hypothesis will be incorrect.

The disparity between the gains of OPP and OCLP suggests that accounting for global occlusion layout by OCLP is more informative than considering the *a priori* occlusion probability of each point individually by OPP. In particular, OCLP improves over OPP for cases such as the example shown in Figure 6.15 where one side of the object is completely occluded. Although the top of the object is validly occluded, OPP assigns a high penalty and over-penalizes the true detections.

Single	baseline	+OPP	+OCLP
HOG	0.55	0.56	0.55
rLINE2D	0.49	0.53	0.61
GN	0.65	0.65	0.73
color	0.05	0.04	0.03
HOG+rLINE2D	0.72	0.73	0.76
HOG+GN	0.77	0.78	0.81
HOG+color	0.63	0.64	0.64
rLINE2D+color	0.56	0.59	0.62
GN+color	0.71	0.71	0.74
HOG+rLINE2D+color	0.74	0.75	0.78
HOG+GN+color	0.77	0.78	0.81
Multiple	baseline	+OPP	+OCLP
HOG	0.50	0.48	0.55
rLINE2D	0.29	0.33	0.37
GN	0.70	0.69	0.76
color	0.12	0.02	0.02
HOG+rLINE2D	0.56	0.58	0.60
HOG+GN	0.80	0.81	0.83
HOG+color	0.55	0.54	0.61
rLINE2D+color	0.36	0.42	0.43
GN+color	0.77	0.76	0.76
HOG+rLINE2D+color	0.58	0.61	0.64
HOG+GN+color	0.80	0.81	0.83

Table 6.1: Object detection performance. Mean Average Precision.

Figure 6.14 shows the performance under different levels of occlusion for all the templates. Here, the performance is the percentage of top detections which are correct. Our occlusion reasoning improves object detection under both low (0-35%) and high levels (>35%) of occlusions, but provides significantly larger gains for heavy occlusions.

Figure 6.8 shows a typical false positive that can only be filtered by our occlusion reasoning. Although a majority of the points match well and the missing parts are largely coherent, the detection is not consistent with our occlusion model and is thus heavily penalized and filtered.

Figure 6.16 shows a couple of failure cases where our assumptions are violated. In the first image, the pot occluding the pitcher is not accurately modeled by its bounding box. In the second image, the occluding object rests on top of the scissor. Even though we do not handle these types of occlusions, our model represents the majority of occlusions and is thus able to increase the overall detection performance.

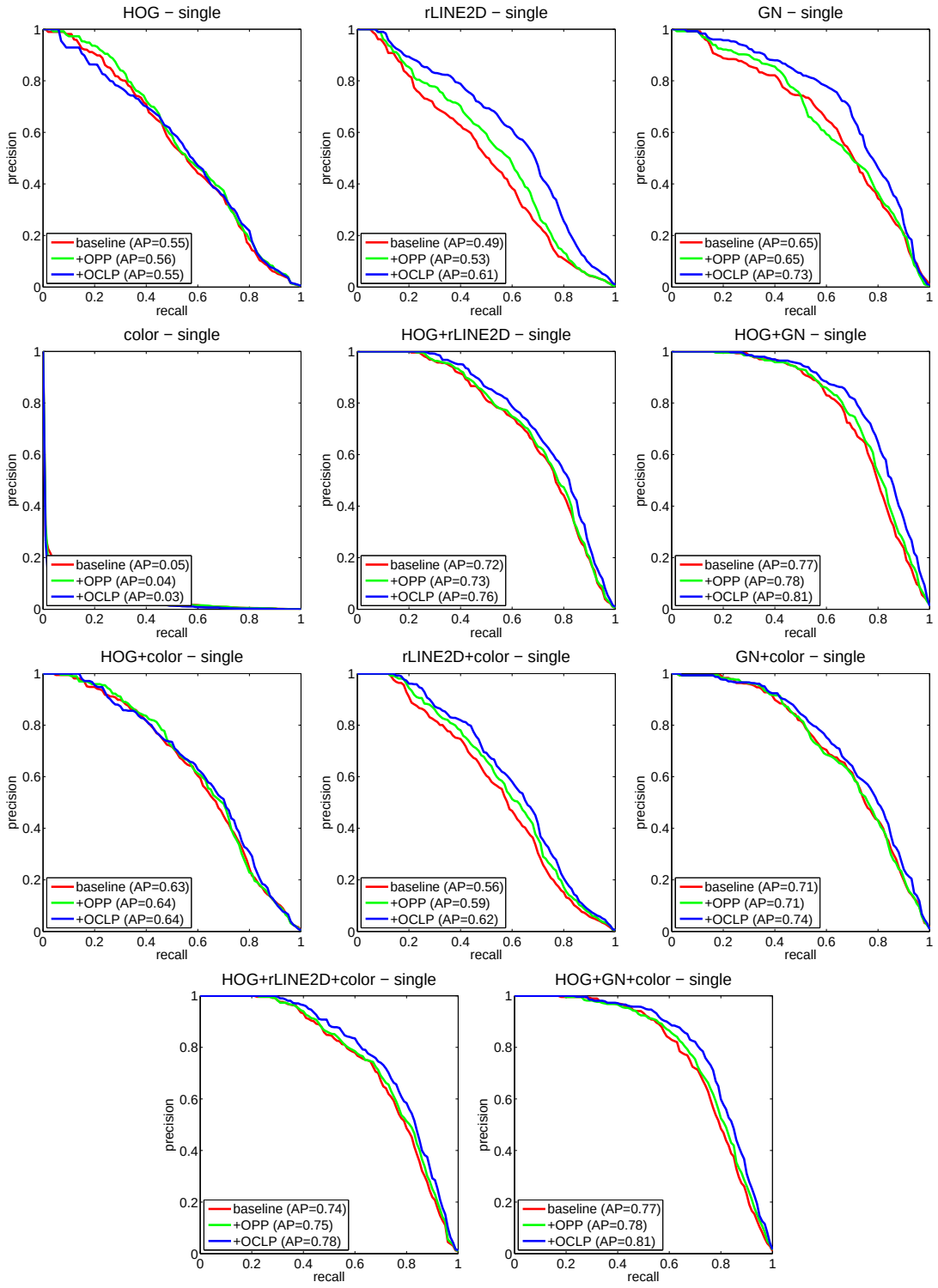


Figure 6.12: Precision/Recall plots for single view on CMU_KO8 for all templates.

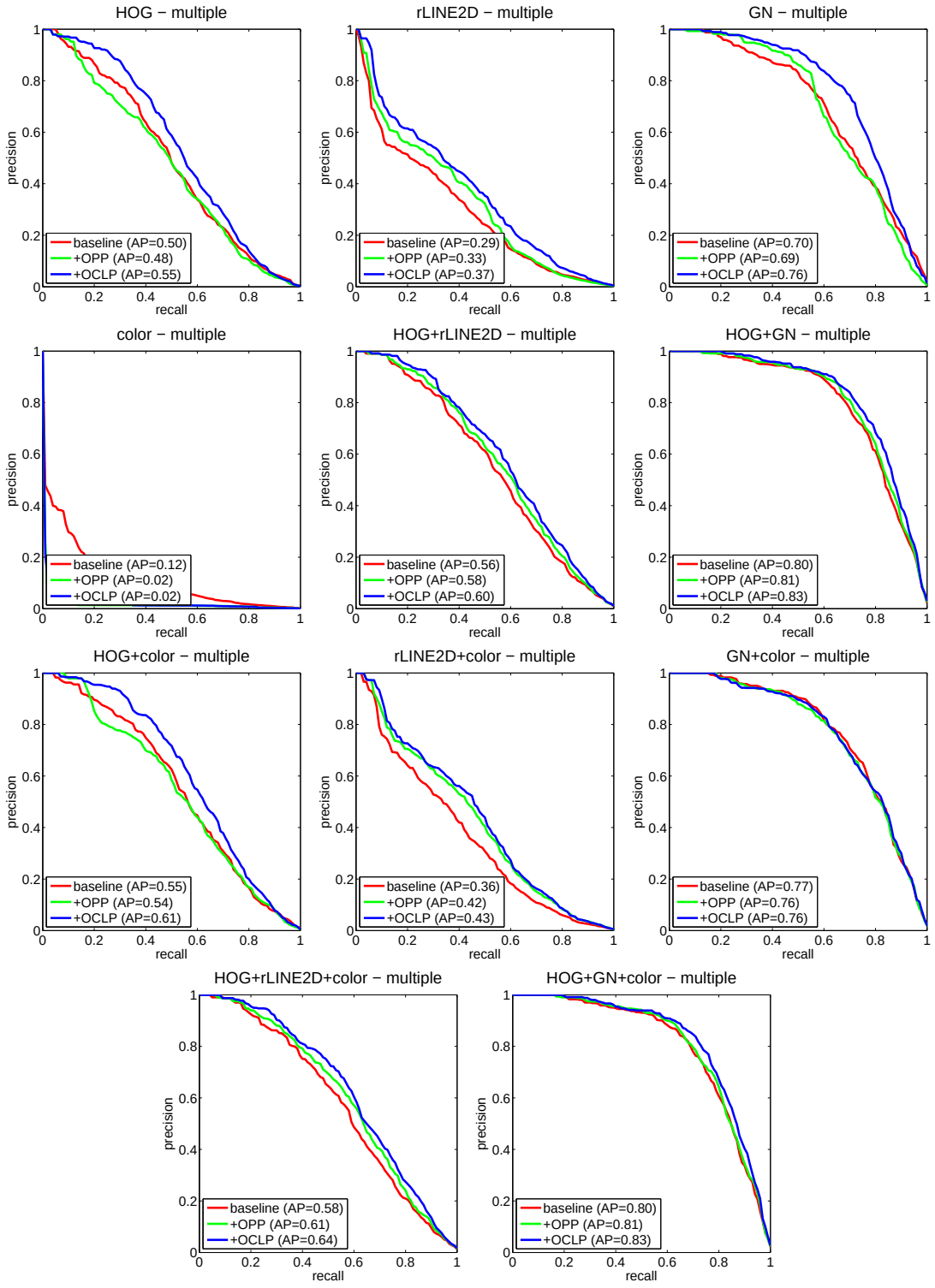


Figure 6.13: Precision/Recall plots for multiple views on CMU_KO8 for all templates.

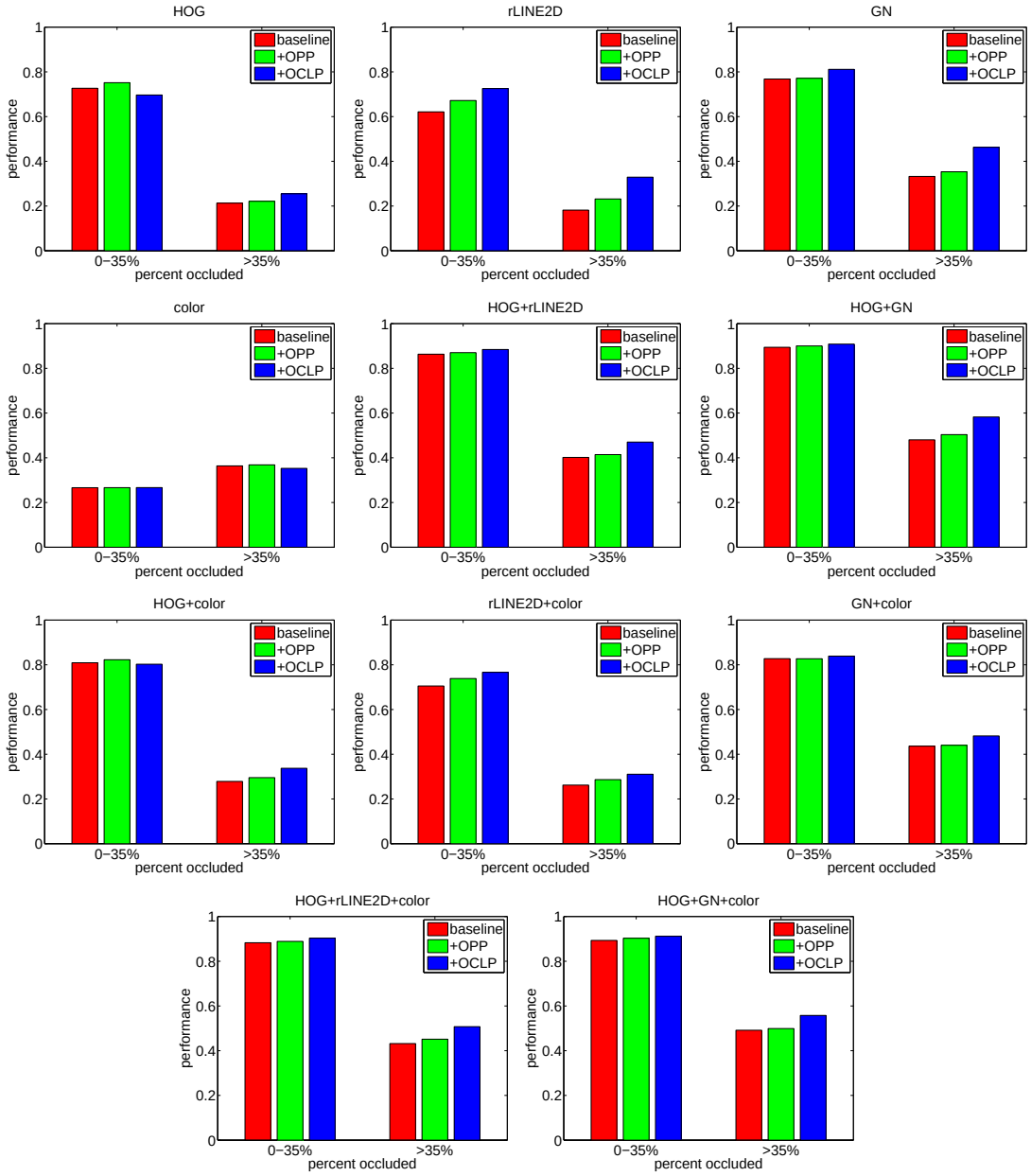


Figure 6.14: Performance under different occlusion levels. While our methods improve performance under all levels of occlusions, we see larger gains under heavy occlusions.

6.3.4 Learning From Data

To verify that our model accurately represents occlusions in real world scenes, we rerun the above experiments with occlusion priors and conditional likelihoods learned from data. We use the detailed groundtruth occlusion masks in the single view portion of the dataset to obtain the empirical distributions. Figure 6.17 compares learning the occlusion prior and occlusion likelihood using 10, 20, 40 and 80 images with our analytical model.

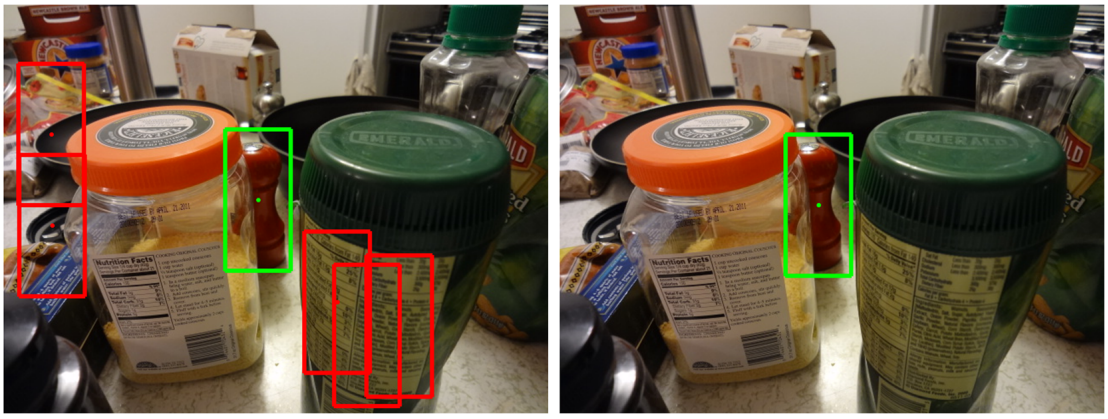
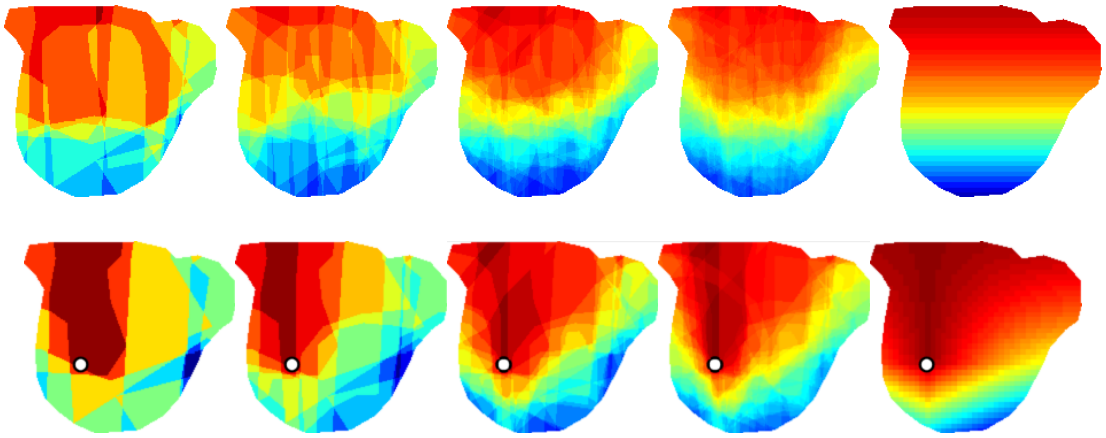


Figure 6.15: A typical case where OCLP performs better than OPP. (left) For OPP, the false positives in red have higher scores than the true detection in green. The occluded region at the top of the true detection is over-penalized. (right) For OCLP, the true detection is the top detection.

Figure 6.16: Typical failure cases of OCLP. (left) The pitcher is occluded by the handle of the pot which is not accurately modeled by a block. (right) The scissor is occluded by a plastic bag resting on top of it. In these cases, OCLP over penalizes the detections.

The distributions from the the empirical and analytical model are very similar.

We use 5-fold cross-validation for quantitative evaluation and Figure 6.18 shows the results using different number of images for learning. The learned occlusion properties, lOPP and lOCLP, correspond to their explicit counterparts, OPP and OCLP. The learned occlusion prior, lOPP, performs slightly better than OPP. This is a result of the slightly different distribution seen in Figure 6.17 where the sides of the object are more likely to be occluded in the dataset. The learned occlusion conditional likelihood, lOCLP, performs essentially the same as OCLP, but requires 80 images for every view of every object to achieve the same level of performance.



(a) 10 images (b) 20 images (c) 40 images (d) 80 images (e) analytical

Figure 6.17: Visualization of occlusion distributions learned using different amounts of data. From left to right, columns 1-4 show using 10, 20, 40, and 80 images for learning the occlusion prior (top) and the occlusion conditional likelihood (bottom). The last column shows the distribution of our analytical model.

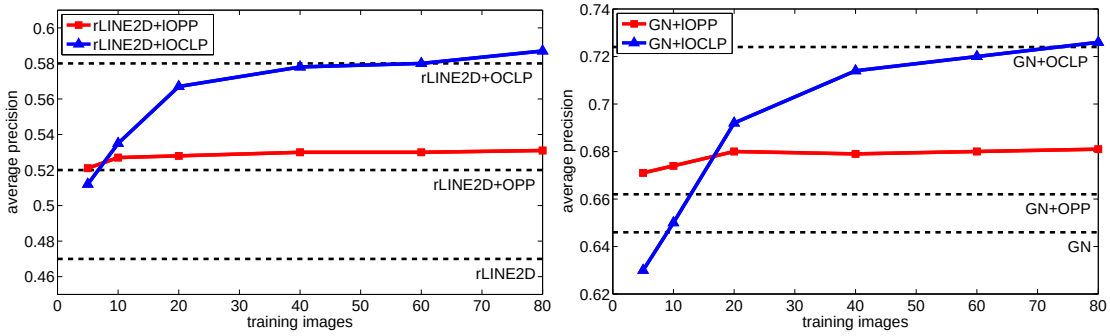


Figure 6.18: Learning the occlusion prior and conditional likelihood from data. The dotted lines show the performance of our analytic approach, which does not depend on the number of training images. We show the learned occlusion properties, lOPP (red) and lOCLP (blue), corresponding to OPP and OCLP. While lOPP performs slightly better than OPP, it needs about 20 images and is only slightly better. lOCLP needs about 60 to 80 images to achieve the same level of performance as OCLP.

6.3.5 Parameter Sensitivity

The two parameters of our occlusion reasoning approach are λ_p and λ_c, corresponding to the hinge loss parameters for OPP and OCLP. To verify that our approach is not sensitive to the exact choice of these parameters, we evaluated the performance of the occlusion reasoning for a range of parameter values. Figure 6.19 shows the sensitivity of λ_p and λ_c when augmenting both rLINE2D and GN. From the figure, the performance is relatively constant for a wide range of parameter values and is thus robust to the exact choice.

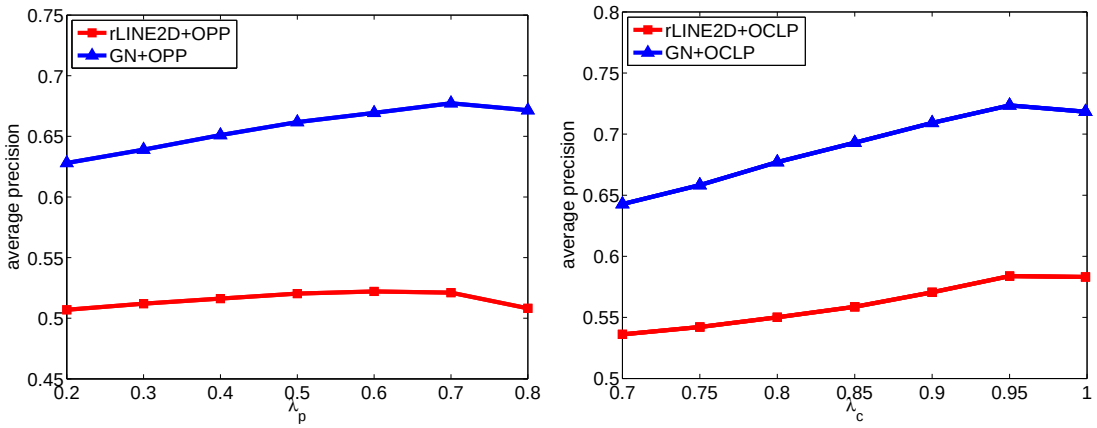


Figure 6.19: Parameter sensitivity for (left) OPP and (right) OCLP. The exact choice of the parameter does not affect the performance of the methods significantly.

6.4 Discussion

The main contribution of this chapter is to demonstrate that a simple model of 3D interaction of objects can be used to represent occlusions effectively for object detection under arbitrary viewpoint without requiring additional training data. We propose a tractable method to capture global visibility relationships and show that it is more informative than the typical *a priori* probability of a point being occluded. Our results on a challenging dataset of feature-poor objects under severe occlusions demonstrate that our approach can significantly improve object detection performance.

Chapter 7

Coherent Reasoning through Efficient Search

In the previous chapter, we proposed an occlusion model for object detection under arbitrary viewpoint to score binary matching patterns. One drawback of requiring the object detector to return a binary decision on whether a boundary point is matched or not matched is that a small set of misclassifications can significantly affect the occlusion distribution as shown in Figure 7.1. In this chapter, we take the core idea of representing occluders as bounding boxes which are on the same surface as the object of interest and reformulate the problem of occlusion reasoning as efficient search. Our approach directly operates on probabilities and thus does not require any hard commitment by the object detector. We also return the predicted occlusion mask which can be used to better understand the layout of the scene.

The main contributions of this chapter are three-fold: 1) formulating occlusion reasoning as efficient search, 2) providing a coherent method for probabilistic reasoning on multiple cues, and 3) scoring the matching pattern of an object detector. Our approach provides a more accurate estimate of the occlusion mask (Figure 7.2) and improves object detection performance.

7.1 Occlusion Model

The occlusion model from the previous chapter marginalized over all possible occluders to compute an occlusion likelihood which was then used to score an occlusion hypothesis. In this chapter, we propose to directly search for a set of valid occluders to explain the matching pattern, $\mathcal{M}$, from Section 6.1. The hypothesis is that if $\mathcal{M}$ can be explained well by a set of valid occluders, it is more likely to be a true detection than those matching patterns that cannot. Instead of requiring that the matching probabilities of markers be binary, we allow $p_i \in [0, 1]$ and the weights to be arbitrary.

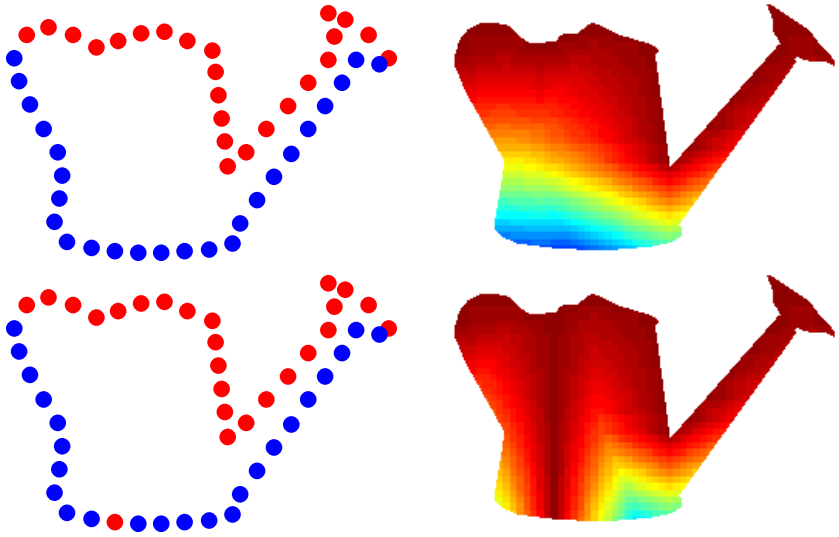


Figure 7.1: Example of OCLP's sensitivity to misclassifications. (top) We show the conditional likelihood distribution for an ideal case. (bottom) We show the conditional likelihood distribution when a single point is misclassified. The distributions are significantly different.

We follow the previous chapter and approximate objects occluding objects as bounding boxes and consider valid occluders as those boxes which touch the base of the object. Boxes with a base lower than this do not need to be considered as we only care about matching probabilities on the object.

Given this model of occluders, the goal is to search for the best set of occluder boxes $\mathbb{b}^*$ (oboxes) to explain $\mathcal{M}$. Each obox is parameterized by its top, left and right coordinates (t, l, r) with the bottom fixed to the base of the object.

7.1.1 Formulation

We define the value of each marker Z_i to be:

$$v_i = w_i \cdot (2p_i - 1). \tag{7.1}$$

For uniform marker weights (i.e., $w_i = 1$), definitely visible markers have a value of $v_i = 1$ and definitely occluded markers have a value of $v_i = -1$. When a marker Z_i falls inside any obox, b, its value is negated, essentially rewarding markers more likely to be occluded and penalizing markers more likely to be visible. An object occlusion is thus represented by a set of oboxes, $\mathbb{b}$. We define the occlusion quality function $q : \mathcal{B} \to \mathbb{R}$ to be:

$$q(\mathbb{b}) = \sum_{Z_i \notin \mathbb{b}_\cup} v_i - \sum_{z_j \in \mathbb{b}_\cup} v_j, \tag{7.2}$$

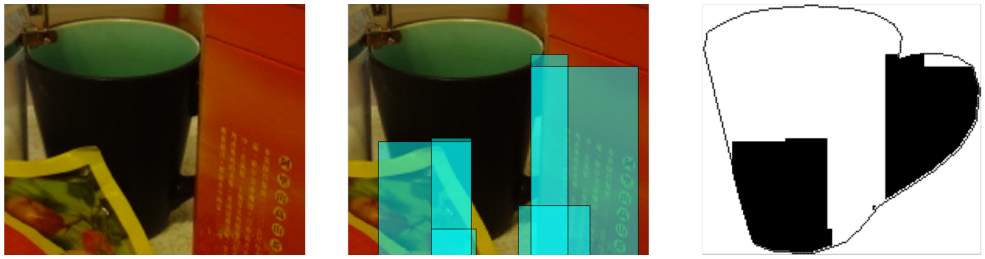


Figure 7.2: Example occlusion predictions. Given a hypothesis detection from a cup detector, the proposed Occlusion Efficient Subwindow Search (OESS) method predicts the occlusion mask and determines how likely it belongs to a true detection. From left to right we show (1) detection, (2) predicted occluder boxes and (3) predicted occlusion mask.

where $\mathcal{B}$ is the set of all possible oboxes, $\mathbb{b} \subset \mathcal{B}$, and $\mathbb{b}_\cup$ is the union of all oboxes $b \in \mathbb{b}$. The first term considers all markers outside the union of the obox set, giving positive score to visible markers and penalizing occluded markers. The second term considers all markers inside the union of the obox set, giving positive score to occluded markers that are explained and penalizing visible markers. The best obox set is given by:

$$\mathbb{b}^* = \underset{\mathbb{b} \subset \mathcal{B}}{\operatorname{argmax}}\, q(\mathbb{b}). \tag{7.3}$$

For an $m \times n$ pixel sized object, $\mathcal{B}$ has on the order of $\mathcal{O}(mn^2)$ elements which would be time consuming to exhaustively search, even for a single obox. In addition, it is difficult to determine the number of oboxes *a priori*. We extend the branch and bound scheme of Efficient Subwindow Search (ESS) [60] to find $\mathbb{b}^*$. Table 7.1 shows that our proposed approach, Occlusion Efficient Subwindow Search (OESS), is significantly faster than applying brute force search for a single obox iteratively.

7.1.2 Occlusion Efficient Subwindow Search (OESS)

Instead of an exhaustive search over all possible obox sets, we formulate the problem as a greedy global branch-and-bound search [60]. Each search iteration returns a single globally optimal obox for the current set of markers $\mathcal{Z}$. In the following, we describe how to find a single obox.

The branch-and-bound search is performed by hierarchically splitting the parameter space into disjoint sets while maintaining the upper bound of performance for each subset. The most promising subsets of the space are explored first. Sets of oboxes in the search are represented by intervals over each of the coordinates $[T, L, R]$, where for example $T = [t_{lo}, t_{hi}]$.

For each obox set $\mathbb{b}$, we compute an upper bound $\hat{q}(\mathbb{b})$ on the occlusion quality $q(b)$ in Equation 7.2 for any obox $b \in \mathbb{b}$. The search is performed by always looking at the

Algorithm 1 Occlusion Efficient Subwindow Search

Require: markers $\mathcal{Z}$, values v_i and bounding function $\hat{q}$

initialize $\mathbb{b}^* = \emptyset$
while $\mathcal{Z} \neq \emptyset$ **do**
    initialize priority queue P to empty
    set $\hat{\mathbf{x}}$ and $\hat{\mathbf{y}}$ to be the unique coordinates of $\mathcal{Z}$
    set $\mathbb{b} = [T, L, R] = \hat{\mathbf{y}} \times \hat{\mathbf{x}} \times \hat{\mathbf{x}}$
    repeat
        split $\mathbb{b}$ into $\mathbb{b}_1$ and $\mathbb{b}_2$
        push $[\mathbb{b}_1, \hat{q}(\mathbb{b}_1)]$ onto P
        push $[\mathbb{b}_2, \hat{q}(\mathbb{b}_2)]$ onto P
        retrieve $\mathbb{b}$ with highest upper bound from P
        if $\hat{q}(\mathbb{b}) \leq \sum_{Z_i \notin \mathbb{b}^*_\cup} v_i$ **then**
            return $\mathbb{b}^*$
        end if
    until $\mathbb{b}$ is a single rectangle
    add $\mathbb{b}$ to $\mathbb{b}^*$
    remove markers $Z_i \in \mathbb{b}_\cup$ from $\mathcal{Z}$
end while
return $\mathbb{b}^*$

set with the highest upper bound. If the most promising set contains a single obox, the search is terminated as its quality will be at least as good as all remaining sets. If the set contains more than one obox, it is split into two disjoint sets along the largest coordinate and upper bounds are computed for each. To avoid having many thin oboxes, we require that each obox have at least a minimum width of γ. After each split, we enforce that $l_{hi} = \min(l_{hi}, r_{hi} - \gamma)$ and $r_{lo} = \max(r_{lo}, l_{lo} + \gamma)$. A priority queue is used to maintain the ordering of the obox sets. Since the markers are often sparse, we speed up the search by considering only unique marker coordinates instead of all the values in the interval.

Once the branch-and-bound search terminates, a single obox with the maximum quality is returned. Since many objects have multiple occluders, we continue the search by removing all markers in $\mathcal{Z}$ covered by the current predicted obox set $\mathbb{b}^*$ and rerun the search on the remaining markers. We iterate until either the upper bound of the quality function for any $\mathbb{b}$ is less than or equal to not putting an obox at all (i.e., $\hat{q}(\mathbb{b}) \leq \sum_{Z_i \notin \mathbb{b}^*_\cup} v_i$), or $\mathcal{Z}$ is empty. Algorithm 1 shows the pseudocode for OESS.

7.1.3 Occlusion Quality Bound

The bounding function $\hat{q}$ is an upper bound on the occlusion quality for any obox $b \in \mathbb{b}$. It has to satisfy two properties: 1) $\hat{q}(\mathbb{b}) \geq \max_{b \in \mathbb{b}} q(b)$, and 2) $\hat{q}(\mathbb{b}) = q(b)$ if b is the only element in $\mathbb{b}$. In ESS, only markers contained within the box matter. However for occlusion reasoning, we care about whether markers outside the obox are correctly

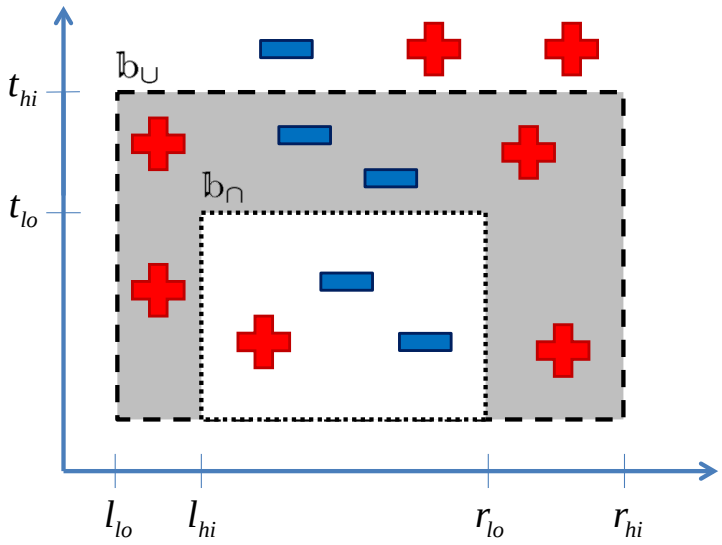


Figure 7.3: Illustration of the upper bound $\hat{q}$ of the occlusion quality for $\mathbb{b} = [T, L, R]$ on a binary visibility hypothesis with $w = 1$. In this example, $q^p_{out}(\mathbb{b}_\cap) = 6$, $q^n_{out}(\mathbb{b}_\cup) = -1$, $q^p_{in}(\mathbb{b}_\cap) = 1$ and $q^n_{in}(\mathbb{b}_\cup) = -4$. Thus, $\hat{q} = 6 + (-1) - 1 - (-4) = 8$.

# of markers	brute force (ms)	OESS (ms)	speedup
100	58.1	2.1	28x
250	1106.4	11.8	94x
500	10365	54.6	190x
1000	77548	221.2	351x

Table 7.1: OESS vs. brute force speed. Times were averaged over 10 trials using random points and matching probabilities.

classified as well. We can partition Equation 7.2 into $q = q_{out} - q_{in}$, where q_{out} contains the markers outside b and q_{in} contains the markers inside b. We can further partition $q_{in} = q^p_{in} + q^n_{in}$ and $q_{out} = q^p_{out} + q^n_{out}$, where q^p_{in} and q^p_{out} contain the positive terms and q^n_{in} and q^n_{out} contain the negative terms. Thus, we have that:

$$q = q^p_{out} + q^n_{out} - q^p_{in} - q^n_{in}. \tag{7.4}$$

An upper bound for $\mathbb{b}$ that satisfies the two properties is:

$$\hat{q}(\mathbb{b}) = q^p_{out}(\mathbb{b}_\cap) + q^n_{out}(\mathbb{b}_\cup) - q^p_{in}(\mathbb{b}_\cap) - q^n_{in}(\mathbb{b}_\cup), \tag{7.5}$$

where $\mathbb{b}_\cup$ is the union of all $b \in \mathbb{b}$ and $\mathbb{b}_\cap$ is their intersection. When $\mathbb{b}$ is a single obox, $\mathbb{b} = \mathbb{b}_\cup = \mathbb{b}_\cap$ and property 2 is satisfied. To show that property 1 is true, we first consider markers outside b. The markers outside any $b \in \mathbb{b}$ is a subset of markers outside $\mathbb{b}_\cap$ and a superset of markers outside $\mathbb{b}_\cup$. Since q^p_{out} contains only positive elements and

q^n_{out} contains only negative elements, $q^p_{out}(\mathbb{b}_\cap) \geq q^p_{out}(b)$ and $q^n_{out}(\mathbb{b}_\cup) \geq q^n_{out}(b)$ for all $b \in \mathbb{b}$. The converse is true for markers inside the obox, and it can be shown that $q^p_{in}(\mathbb{b}_\cap) \leq q^p_{in}(b)$ and $q^n_{in}(\mathbb{b}_\cup) \leq q^n_{in}(b)$. Combining these inequalities, we have that for any rectangle in $\mathbb{b}$:

$$\hat{q}(\mathbb{b}) = q^p_{out}(\mathbb{b}_\cap) + q^n_{out}(\mathbb{b}_\cup) - q^p_{in}(\mathbb{b}_\cap) - q^n_{in}(\mathbb{b}_\cup) \geq q(b). \tag{7.6}$$

The intersection $\mathbb{b}_\cap$ and union $\mathbb{b}_\cup$ can be computed efficiently as $\mathbb{b}_\cap = [t_{lo}, l_{hi}, r_{lo}]$ and $\mathbb{b}_\cup = [t_{hi}, l_{lo}, r_{hi}]$. If $l_{hi} > r_{lo}$, then $\mathbb{b}_\cap = \emptyset$. The computation of q^p_{in}, q^n_{in}, q^p_{out} and q^n_{out} can be done efficiently using integral images [118]. Figure 7.3 shows an illustration of the upper bound.

7.2 Combining with Object Detection

Given the matching pattern $\mathcal{M}$, our OESS method returns the best obox set, $\mathbb{b}^*$, and its occlusion quality $q(\mathbb{b}^*)$. While this occlusion quality can be used by itself as the score of the detection, it does not account for how well the object is matched. A detection with all the markers being occluded (i.e., $p_i = 0$) would receive a very high occlusion quality score since it can be fully explained with an obox that covers the whole object. However, no object markers are matched. To incorporate the OESS occlusion quality q with the raw score s returned by the object detector, we learn a linear weighting using an Exemplar SVM (ESVM) [73] between, s, q and their product sq:

$$score = \alpha_1 s + \alpha_2 q + \alpha_3 sq. \tag{7.7}$$

The single positive example (s^+, q^+) is the ideal detection where s^+ is the score of the detector on the training image, and $q^+ = \sum w_i$ is the maximum occlusion quality when all points are visible. The goal of the ESVM is to determine the weighting which best separates the false positives from this point. An ideal detection would thus have both a high matching score as well as a high occlusion quality. The parameters for training the ESVM are the same as [73] and we use three iterations of hard negative mining. Since the ESVM output is not calibrated between different detectors, it is difficult to choose the best scoring template for object recognition. We calibrate the scores using the Extreme Value Theory [101], as it does not require positive examples, which are often difficult to obtain. Negative data are easily obtained by sampling background images.

The OESS method can be used to rank the matching patterns of any object detector which returns the probability that markers on the object are matched. We use the method described in Section 6.2.1 to obtain the matching probabilities. The only difference is that we do not threshold the probabilities here.

In addition, our method can easily integrate multiple cues. Different marker types, however, capture different spatial extent of information around their positions. Boundary

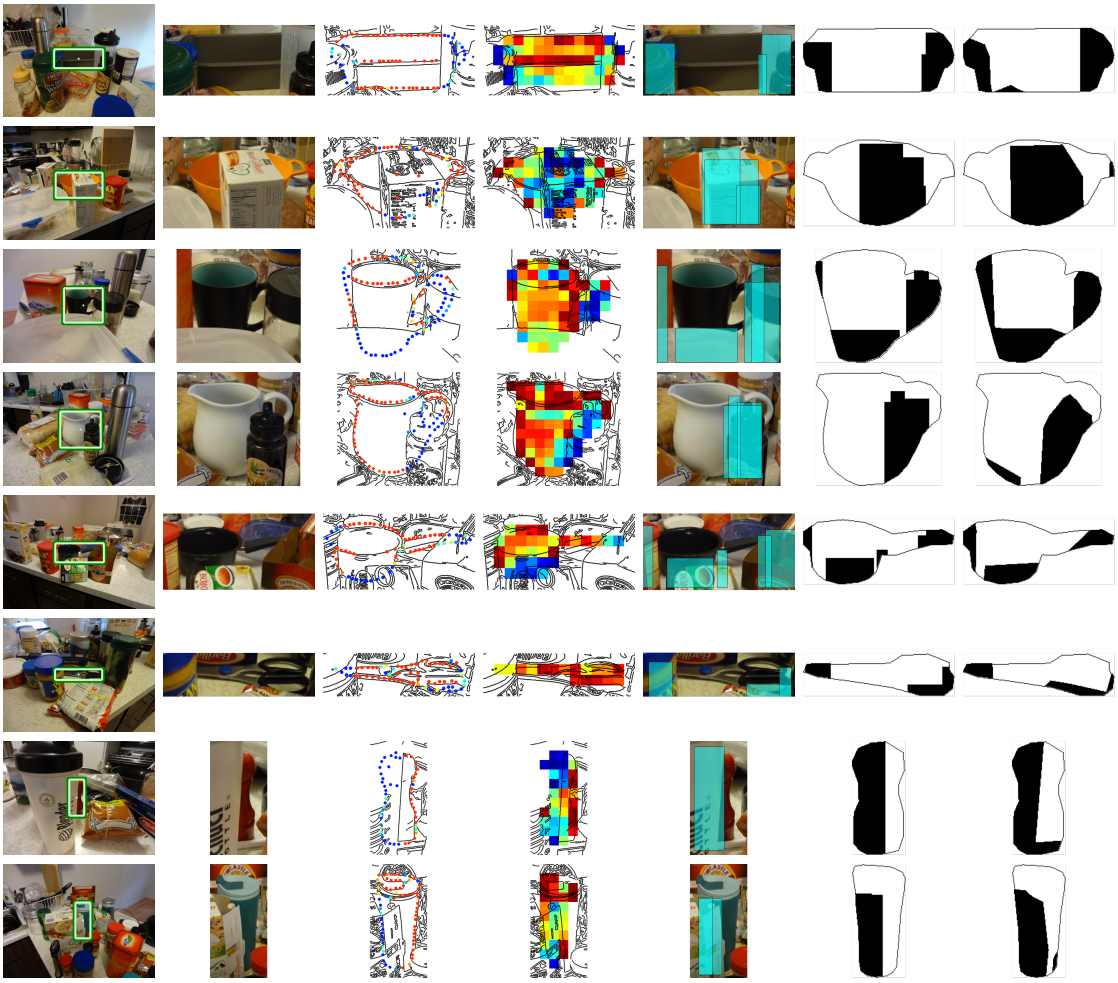


Figure 7.4: Example detections and occlusion reasoning using OESS on CMU_KO8. From left to right, we show (1) the original image with bounding box of detection, (2) the zoomed in view of the detection, (3) boundary matches, (4) activation scores of region cells using texture and color, (5) hypothesized oboxes, (6) predicted occlusion mask, and (7) groundtruth occlusion mask. For columns 3 and 4, the hotter the color, the better the match. To be consistent, red points indicate matched boundary points and blue points indicate points that are not matched.

cues, such as LINE2D, use sampled edge points which only consider information very locally. Grid-based approaches, such as HOG, cover a much larger area for each grid cell. Intuitively, we want to give more weight to points that have a larger region of influence. We weight each marker by the area in pixels of the region it represents. For grid based methods, this is the area of the cell. For point-based methods, this is the area of the sampling circle.

7.3 Evaluation

We evaluate our occlusion model's performance in object instance detection by conducting two sets of experiments. The first evaluates the algorithm's accuracy in predicting occlusions, and the second evaluates the algorithm's ability to detect objects. We systematically analyze the combination of boundary, texture, and color cues with our Boundary and Region Templates of Chapter 5 and their effect on occlusion reasoning. We set $\gamma = 16$ for all of the experiments. We evaluate our approach on the CMU Kitchen Occlusion Dataset.

7.3.1 Occlusion Prediction

First, we evaluate the performance in predicting the occluded region. We convert the best obox set, $\mathbb{b}^*$, of each detection window into a binary occlusion mask (Figure 7.4). The performance is evaluated using the standard intersection-over-union (IoU) metric between the predicted mask and the groundtruth mask from the dataset. We average the IoU for all the images of all the objects. We compare our method against thresholding the matching probabilities at 0.5, the mean-shift occlusion reasoning approach of [120] which enforces local coherency, and OCLP [49]. Since thresholding and mean-shift on B produce only point classifications, we dilate the classifications by the sampling radius of rLINE2D to produce an occlusion mask. This dilation captures the local region of influence of each point. In addition, OCLP is a scoring mechanism and does not predict an occlusion mask. We generate a mask by first thresholding the matching probabilities at 0.5 to get the matched points. Then, we evaluate the occlusion conditional likelihood [49] at all points on the object mask given these matched points and threshold it to predict the occlusion. Figure 7.5 shows a bar graph comparing the different occlusion prediction approaches.

From the figure, OESS significantly outperforms the other methods for all templates except for color. Again, poor confidences by using color hurt the occlusion reasoning performance. This suggests that most of the information is contained in texture and boundary.

In addition, thresholding and mean-shift perform significantly worse since they do not account for occlusion structure in the real world. OCLP captures some structure and works well for rLINE2D, but is sensitive to binary misclassifications, especially when applied to approaches that use a dense grid. OESS operates directly on probabilities and captures higher level occlusion structure which results in more accurate occlusion prediction.

Figure 7.6 shows example failure cases of the full system (HOG+GN+color with OESS). In the first case, the occluding object violates our bounding box assumption and we are unable to recover a good occlusion mask. In the second case, the matching prob-

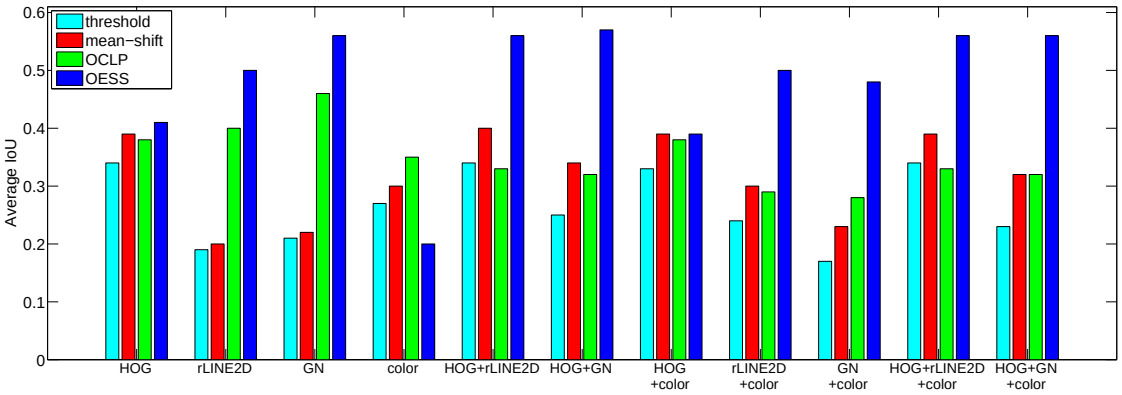


Figure 7.5: Occlusion prediction performance on CMU_KO8. We compare OESS with thresholding the matching probabilities, using the mean-shift approach of [120] to enforce local occlusion coherency, and OCLP. We report the average intersection-over-union (IoU) between the predicted mask and the groundtruth. OESS significantly outperforms all the approaches.

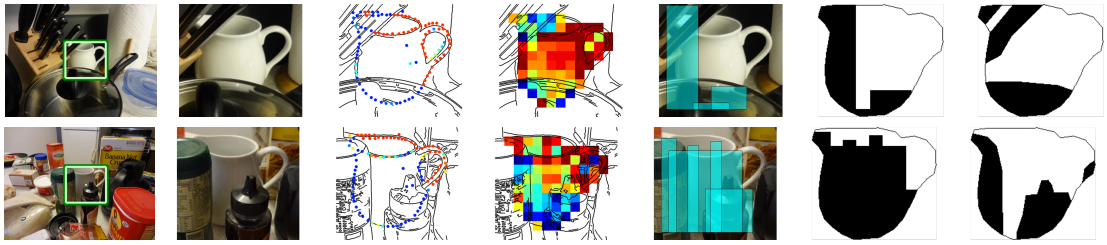


Figure 7.6: Example failure cases using OESS for occlusion segmentation of a pitcher. In the first row, the occlusion does not satisfy the bounding box approximation of occluders. In the second row, the region template produces inaccurate activation scores.

abilities are inaccurate. More robust templates which obtain more accurate matching probabilities will aid in improving the performance of occlusion reasoning and OESS.

7.3.2 Object Detection

We also need to verify that our method maintains or improves the detection performance while significantly improving the occlusion prediction. In the following experiments, an object is correctly detected if it satisfies the PASCAL overlap criterion [28] with the ground truth bounding box. We compare our occlusion reasoning approach with the OPP and OCLP approaches. Since these methods require binary matching classifications, we threshold the matching probabilities at 0.5.

We compare the occlusion reasoning on different combinations of boundary (rLINE2D and GN), texture (HOG), and color cues. Figure 7.7 shows the Precision/Recall curves. Table 7.2 summarize these curves using the Mean Average Precision (mAP).

From the tables, OESS improves the performance over the baseline for all the cues.

Single	baseline	+OPP	+OCLP	+OESS
HOG	0.55	0.56	0.55	0.56
rLINE2D	0.49	0.53	0.61	0.61
GN	0.65	0.65	0.73	0.73
color	0.05	0.04	0.03	0.04
HOG+rLINE2D	0.72	0.73	0.76	0.75
HOG+GN	0.77	0.78	0.81	0.81
HOG+color	0.63	0.64	0.64	0.64
rLINE2D+color	0.56	0.59	0.62	0.60
GN+color	0.71	0.71	0.74	0.75
HOG+rLINE2D+color	0.74	0.75	0.78	0.78
HOG+GN+color	0.77	0.78	0.81	0.80
Multiple	baseline	+OPP	+OCLP	+OESS
HOG	0.50	0.48	0.55	0.54
rLINE2D	0.29	0.33	0.37	0.47
GN	0.70	0.69	0.76	0.85
color	0.12	0.02	0.02	0.17
HOG+rLINE2D	0.56	0.58	0.60	0.61
HOG+GN	0.80	0.81	0.83	0.85
HOG+color	0.55	0.54	0.61	0.61
rLINE2D+color	0.36	0.42	0.43	0.43
GN+color	0.77	0.76	0.76	0.79
HOG+rLINE2D+color	0.58	0.61	0.64	0.64
HOG+GN+color	0.80	0.81	0.83	0.85

Table 7.2: Object detection on CMU_KO8: Mean Average Precision.

Importantly, it never performs worse, unlike OPP and OCLP. In addition, OESS outperforms OPP and OCLP in all cases for the typical recognition scenario with multiple object viewpoints. By reasoning on matching probabilities instead of hard classifications, our approach is more robust when fusing multiple cues.

Figure 7.9 shows the images containing the hardest true positives to detect for each object. The detections in these images have the lowest score among all the true positives. For these images, the objects are heavily occluded, making it difficult even for humans to find them.

Figure 7.10 shows the highest scoring false positives using HOG+GN+color and OESS. These detections have large coherent portions which match the shape and interior appearance of the object well, and which also largely obey the occlusion model. To reject these false positives, additional information such as image segmentations [115] and geometric context [47] are needed.

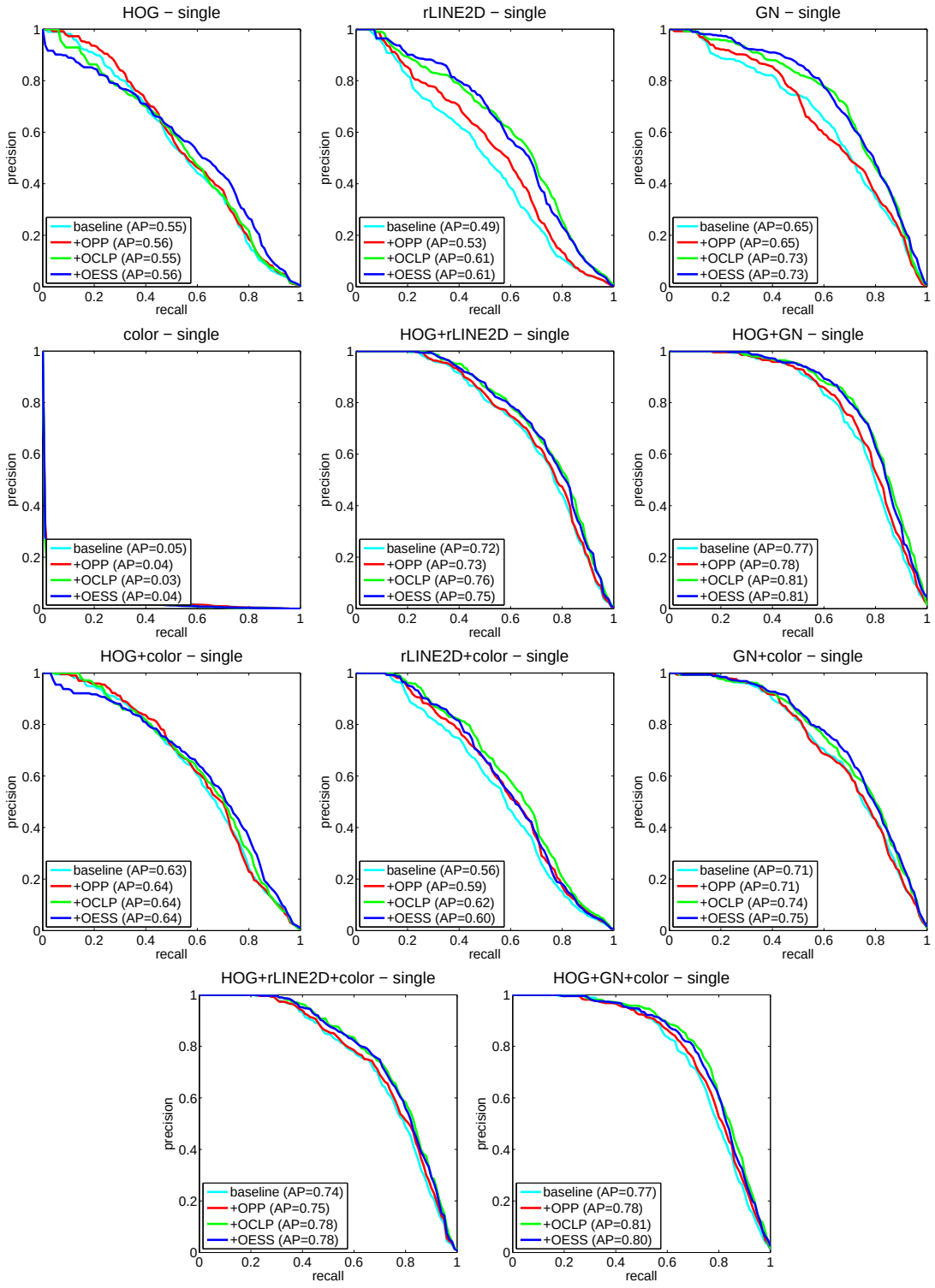


Figure 7.7: Precision/Recall plots on CMU_KO8 for single view for all templates.

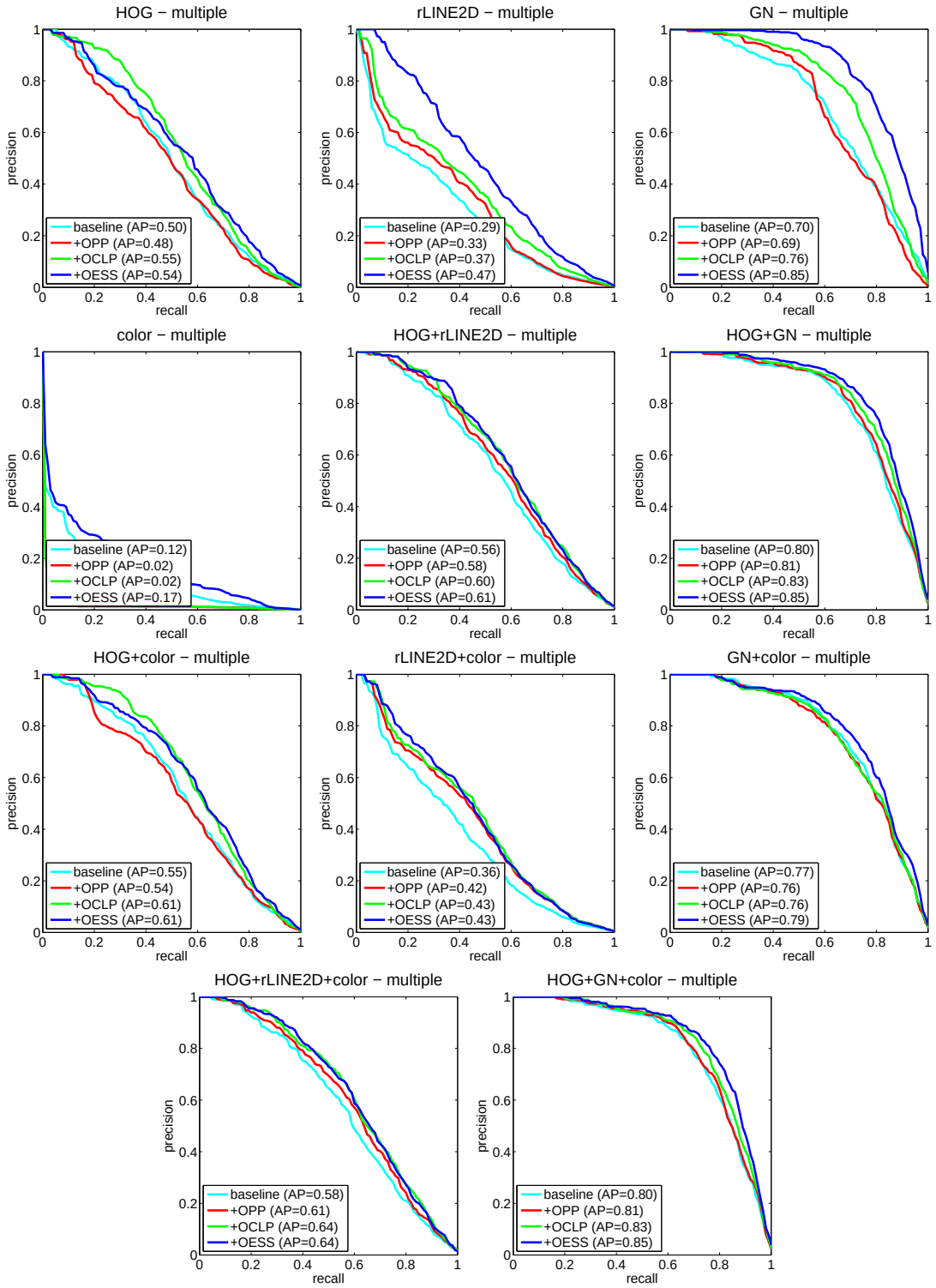


Figure 7.8: Precision/Recall plots on CMU_KO8 for multiple views for all templates.

(a) baking pan (b) colander (c) cup (d) pitcher

(e) saucepan (f) scissors (g) shaker (h) thermos

Figure 7.9: Images containing the hardest true positives to detect. The objects in these images are heavily occluded and have scores lower than many false positives.

7.4 Discussion

The main contribution of this chapter is to formulate occlusion reasoning as an efficient search over occluding blocks which best explain a probabilistic matching pattern. Our approach is able to coherently reason on matching patterns returned from multiple cues. Given a set of hypothesis object detections, we effectively score them based on how well the matching pattern can be explained by a set of valid occluding boxes. Our results on a challenging dataset of objects under severe occlusions and in heavy clutter demonstrate significant improvement over state-of-the-art methods for occlusion prediction and instance detection.

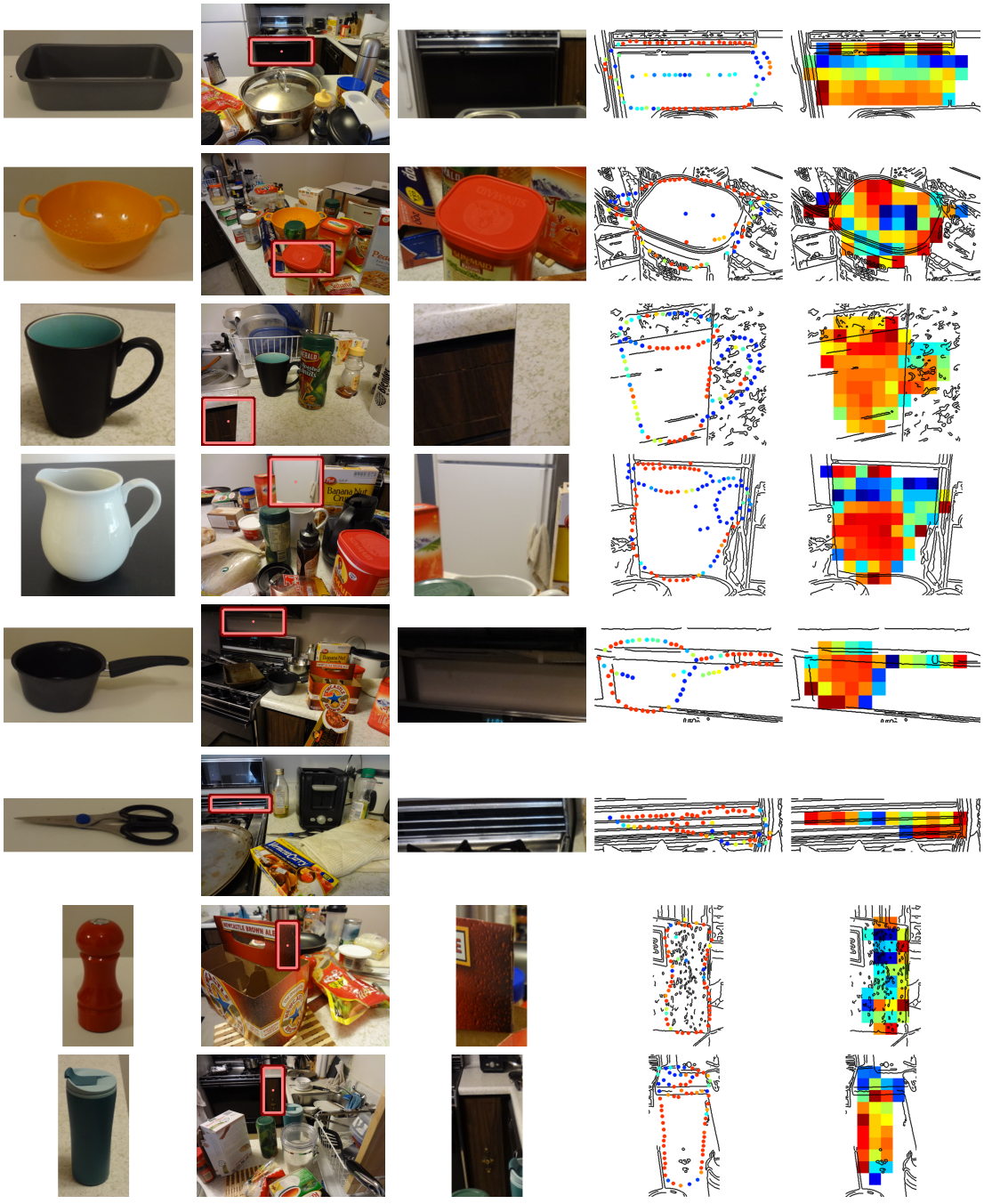


Figure 7.10: Highest scoring false positives. These false positives have large portions which match the object of interest. Additional information is needed to filter these windows.

Part V

Conclusion

Chapter 8

Contributions

In this dissertation, we have explored techniques for addressing ambiguity when detecting objects in scenes with severe clutter and occlusions. Our work has focused on the three key components of this problem: similar features, feature-poor objects, and occlusions. In this chapter, we summarize our findings in these areas.

Our main contributions are as follows:

- *Similar Features*: We address the problem of similar/repeated features when using discriminative keypoints for 3D object detection. Our Discriminative Hierarchical Matching (DHM) method preserves feature ambiguity at the matching stage, and disambiguates the correspondences at hypothesis testing using View-constrained RANSAC. We show that adding Simulated Affine features to the 3D model allows the system to handle viewpoints significantly different from the model images and that the quantization framework addresses matching these Simulated Affine features which can often be very similar.

- *Gradient Networks*: We address the problem of detecting feature-poor objects based on shape. Our approach captures contour connectivity directly on low-level image gradients without having to extract edges. Our shape matching framework seamlessly incorporates local appearance information around the shape such as gradient orientation, color and texture, and can utilize arbitrary edge potentials. We demonstrate an overall improvement of 19% in Average Precision over HOG and other shape matching methods, and we show that our approach is more robust when objects are under severe occlusions.

- *Boundary and Region Templates*: We introduce a framework for detecting feature-poor objects which incorporates both an explicit representation of the boundary as well as the appearance of the interior (e.g., texture and color). We demonstrate that the lack of texture is actually informative and that while approaches such as HOG

do capture a coarse representation of the boundary, using an explicit representation can significantly improve the detection performance. Our full representation yields an improvement of 34% in Average Precision over HOG and 10% over Gradient Networks which uses shape alone.

- *Occlusion Reasoning*: We introduce two methods for increasing the robustness of object detection under arbitrary viewpoint when objects are under severe occlusions. Our first approach models the structure of occlusions with an Occlusion Conditional Likelihood by computing the probability a marker on the object is visible given the visibility labelings of all the other markers. We derive the likelihood under arbitrary viewpoint analytically by approximating objects as 3D bounding boxes and using a distribution of occluder dimensions. We score hypothesis detections based on how likely they conform to our occlusion model by penalizing markers which are inconsistent with the Occlusion Conditional Likelihood.

  Our second approach takes the core idea of representing occluders as bounding boxes and reformulates the problem of occlusion reasoning as efficient search. Our Occlusion Efficient Subwindow Search (OESS) method searches for a set of valid occluders which best explain a matching pattern from the detector and scores a detection based on how well the occlusions can be explained. The approach operates directly on matching probabilities and seamlessly incorporates many different types of cues.

Chapter 9

Future Directions

Despite our contributions, this work is only a small step towards detecting object instances robustly in cluttered scenes. The system we propose is not perfect and its performance is still far from that of human perception. In this chapter, we discuss several areas which may be useful for further investigation as follow-up works to this thesis.

9.1 Fine-grained Verification

The ultimate goal of object instance detection is to find an exact object in an image. However current state-of-the-art systems, as well as our own, can at best only produce hypotheses which look similar to the object of interest. In general, many man-made objects belonging to the same category have similar appearance. For example, almost all mugs are composed of a handle attached to a cylinder, and all laptops have a screen with a keyboard. The difference which separate two objects of the same category lie in the fine-grained details (Figure 9.1). Mugs, for example, may differ in properties such as the shape of the handle, the size of the cylinder, or the materials they are made from. Current approaches are unable to separate these details well. While matching templates pixel for pixel could capture fine-grained details, objects appearing under slightly different viewpoints and lighting conditions would make these approaches intractable. Allowing for variation on the other hand would implicitly allow for objects which may not be exactly the same.

The problem becomes significantly more challenging when objects are occluded. In these cases, the score of similar objects are likely to be higher than the correct object under high levels of occlusion. While humans can usually still discriminate the fine-grained details and correctly identify an object, there reaches a point where there is just not enough information since the occluded area is inherently ambiguous. If the entire set of possible objects in the world are known, it may be possible to determine if the occluded region is informative. But in general, this is not the case and the occluded

Figure 9.1: Fine-grained discrimination of mugs. The differences between these mugs are in the fine-grained details.

region can contain anything.

Recently, there has been a surge in interest in fine-grained discrimination within object categories such as birds [121], plant leafs [56], and dogs [54]. The key component of these algorithms is to obtain good alignment with a generic category model to perform the separation. However, even with good alignment, the problem of separating noise from distinctiveness remains, and there is still much work left to be done.

9.2 Scalable Representation

Currently, our object representation is primarily based on templates. While view-invariant techniques exist, we choose non-invariant methods as they are able to directly observe the projection of the object for each view. This is important for capturing the fine-grained details. However brute force template matching requires a large number of templates to cover all the object viewpoints. While approaches such as LINE2D have shown that thousands of templates can be matched in near real-time, the method is only able to scale linearly with the number of templates. In the real world, there exist billions of distinct objects, making methods which scale linearly intractable.

However objects in the real world often share very similar properties. For example many objects within the same category usually look very similar. One possibility is to design a hierarchy of templates where initial templates separate a broad class of similar objects from the background, and each level of the hierarchy performs further discrimination. For a well-balanced hierarchy, the computation complexity would on the order of the logarithm of the number of templates, making it more tractable.

Another possibility is to share features between the objects, such as the boosting method of Torralba *et al.* [114] and the steerable basis of Pirsiavash and Ramanan [88]. Shape grammar [129] methods have also been used to build a hierarchy of feature parts where each successive level uses the features from the previous level. However one drawback of feature sharing is controlling the level of variation within each shared feature. If too much variation is allowed, the object instance will not be the same. If not enough

variation is allowed, there may be no sharing at all. One possible solution may be to do an initial coarse matching by sharing features and then perform fine-grained verification as proposed in the previous section.

9.3 Incorporating Depth Information

In this thesis, we focused on recognizing object instances from a single 2D image. However for many applications, such as robotic manipulation, depth information from sensors such as laser scanners and stereo cameras are available as well. Particularly with the introduction of the Microsoft Kinect [105], the access to RGB-D and 3D data is becoming more pervasive. A possible extension of our work is to try and incorporate 3D information with the object representation and occlusion reasoning.

There has already been work on representing objects using RGB-D information such as by Hinterstoisser *et al.* [44] and Lai *et al.* [58]. These are primarily extensions of popular approaches such as sparse edge point matching methods and HOG from 2D to 3D. One possibility in our representation is to not only incorporate surface normal information such as in LINEMOD [44], but to incorporate smoothness using an approach similar to Gradient Networks. Instead of only matching based on local information, we can find smooth connected surfaces which match the object as well.

Having depth information can also significantly improve occlusion reasoning. Not only can we use it to classify occlusions based on whether the surface normals and depth match, we can determine which objects are in front of others. This information can then be used to obtain better segmentations of the object for ranking the hypothesis and fine-grained verification.

In addition, we can also use the depth to constrain the scale of the templates. By doing so, we only need to scan a template at one scale per image location instead of at all scales on a scale pyramid. This can potentially save a significant amount of computation.

Chapter 10

Closing Thoughts

Objects which lack discriminative features are *the* most difficult objects for Computer Vision systems to detect automatically. In cluttered environments, the lack of distinctive features makes it impossible to avoid some accidental alignment between the object and the background. Occlusions in these scenes further exacerbate the problem by increasing the overall ambiguity, resulting in false positives that have higher scores than occluded true positives. The work presented in this thesis is a step towards detecting objects robustly in these challenging scenarios. While object instance detection still remains an open area of research, the primary challenges we foresee moving forward are fine-grained discrimination and scalability.

We conclude this thesis with the three guiding principles which have been the driving forces behind this research. First, be non-committal; procrastinate hard decisions until as late as possible. Second, there is structure to the world; exploit it. Lastly, *one man's noise is another man's signal*[1]; use every piece of information available.

[1] Edward W. Ng, New York Times, 1990.

Appendices

Appendix A

Datasets

While there exist many vision datasets for object categories (e.g., PASCAL [28], ImageNet [25], LabelMe [99], SUN [126]), there are very few for evaluating object instance detection in natural scenes. Early work in object instance detection [41, 68] primarily evaluated their algorithms in controlled setups. While large datasets were collected, such as the Columbia University Image Library (COIL-100) [83] and the Amsterdam Library of Object Images (ALOI) [37], they primarily contained objects on simple, monotone backgrounds with very little clutter. Many recent datasets, such as the Table Top Object Dataset [7], often still contain objects only on simple backgrounds.

To evaluate feature-based methods in more cluttered environments, Rothganger *et al.* [98] proposed a database of eight objects. However, their evaluation set contains only 51 images and many of the scenes are staged and not natural. In addition, all of the algorithms they evaluated on the dataset achieve well over 90% performance. To perform detailed analysis of the failure modes of object detection systems, synthetic datasets such as NORB [61] have been used. However, results on images with synthetic backgrounds and lighting conditions can not be directly transferred to natural images.

Recently, Lai *et al.* [58] proposed an RGB-D object dataset collected in household environments. While their dataset contains objects in scenes that are more natural than previous datasets, the majority of objects are well spread out on a table with very little clutter and occlusions. The state-of-the-art LINE2D and LINEMOD algorithms by Hinterstoisser *et al.* [44] were evaluated on their self-collected dataset of six objects, where the test set for each object was a single video sequence. While the images are very cluttered, the clutter was the same for all the images and the objects have little to no occlusion. In addition, their template matching methods already achieve over 90% on the dataset. This high level of performance is consistent with our observation in Chapter 4 that many algorithms work well when objects are unoccluded, but degrade rapidly in natural scenes where occlusions are common. While we evaluated on a subset of these datasets, we collected two datasets: 1) CMU Grocery Dataset (CMU10_3D) for

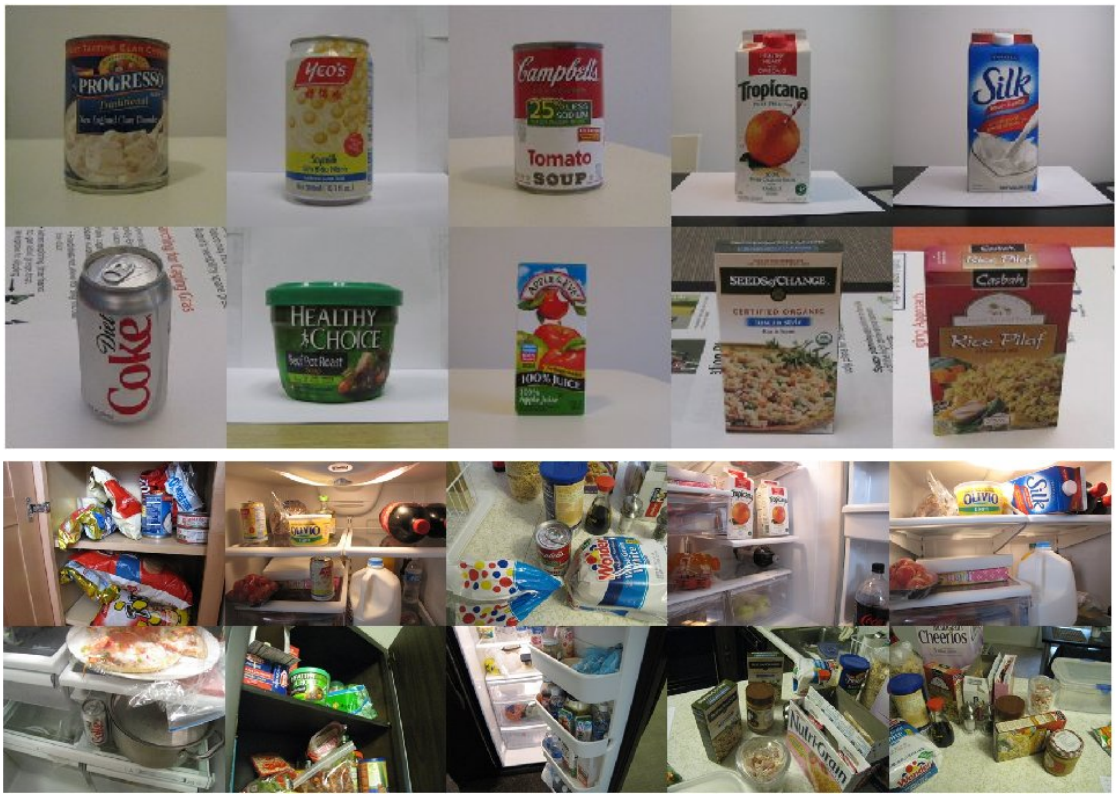


Figure A.1: CMU Grocery Dataset (CMU10_3D). (top) 10 feature-rich kitchen objects in the dataset. (bottom) Example images showing objects in severe clutter with lighting variation and occlusions.

feature-rich objects and 2) CMU Kitchen Occlusion Dataset (CMU_KO8) for feature-poor objects, to perform detailed analysis in more natural scenes with severe clutter and occlusions. Our datasets contain groundtruth object pose and occlusion labels.

A.1 CMU Grocery Dataset (CMU10_3D)

This dataset contains 10 feature-rich household objects (clam chowder can, soymilk can, tomato soup can, orange juice carton, soy milk carton, diet coke can, pot roast soup, juice box, rice tuscan box, rice pilaf box) with 62 images per object, for a grand total of 620 images. Figure A.1 shows the objects in the dataset as well as a few example images. For each object, three types of images were taken. 25 images contain one instance of the object, and 25 images contain two instances, both with their ground truth marked as regions and ID within the image. Finally, 12 more images were collected in a calibrated setup and their full 6D poses were groundtruthed.

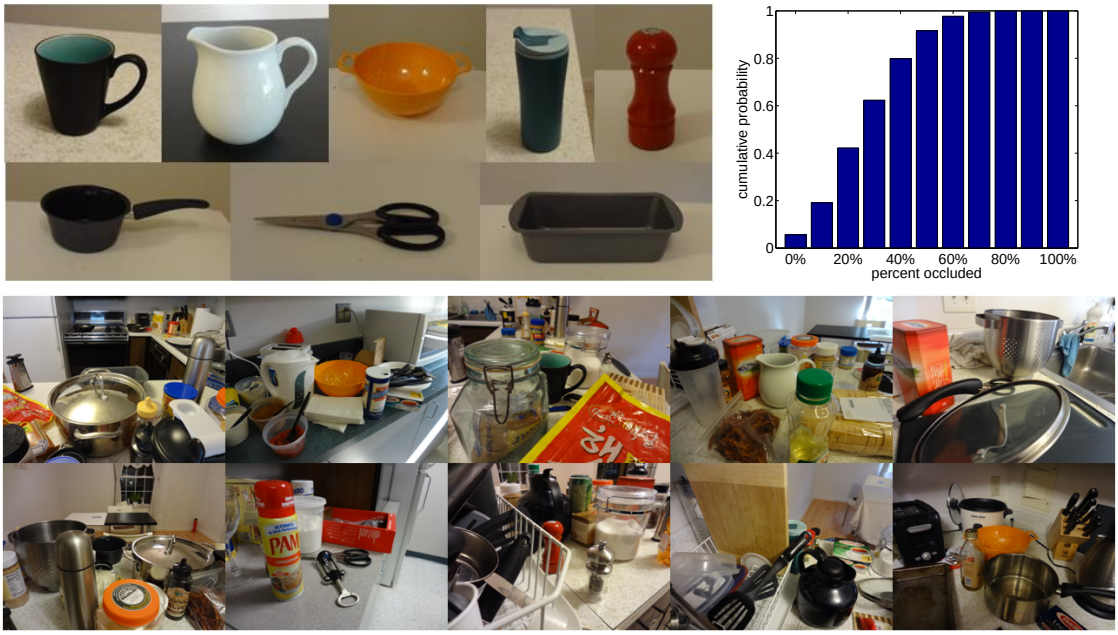


Figure A.2: CMU Kitchen Occlusion Dataset (CMU_KO8). (top-left) 8 feature-poor kitchen objects in the dataset. (top-right) The dataset contains roughly equal amount of partial occlusions (1-35%) and heavy occlusions (35-80%). (bottom) Example images showing objects in cluttered environment under severe occlusions.

A.2 CMU Kitchen Occlusion Dataset (CMU_KO8)

This dataset contains 1600 images of 8 feature-poor household objects and is split evenly into two parts; 800 for a single view of an object and 800 for multiple views of an object. Figure A.2 shows the 8 objects with a few example images.

The single-view part contains groundtruth labels of the occlusions and Figure A.2 shows that our dataset contains roughly equal amounts of partial occlusion (1-35%) and heavy occlusions (35-80%) as defined by [27], making this dataset very challenging.

For multiple-view evaluation, we focus our viewpoint variation to primarily the elevation angle as relative performance under different azimuth angles is similar. We use 25 model images for each object which is the same sampling density as [44]. Each model image was collected with a calibration pattern to groundtruth the camera viewpoint and to rectify the object silhouette to be upright. The test data was collected by changing the camera viewpoint and the scene around a stationary object. A calibration pattern was used to ground truth the position of the object.

Appendix B

Computing Occlusion Distributions

Many of the results derived for the Occlusion Prior and Occlusion Conditional Likelihood in Chapter 6 are from the classic field of integral geometry [100]. In the following, we show the detailed derivations.

B.1 Probability Density of $\hat{w}$

We show how to transform a uniform variable over a $\frac{\pi}{2}$ interval by Equation 6.1 using the *distribution function technique* [46]. First, let's simplify the equation for the projected width:

$$\hat{w}(\theta) = w \cdot \cos\theta + l \cdot \sin\theta \tag{B.1}$$

$$= \sqrt{w^2 + l^2} \cdot \left\{ \frac{w}{\sqrt{w^2 + l^2}} \cdot \cos\theta + \frac{l}{\sqrt{w^2 + l^2}} \cdot \sin\theta \right\} \tag{B.2}$$

$$= \sqrt{w^2 + l^2} \cdot \left\{ \cos\left[\tan^{-1}\left(\frac{l}{w}\right)\right] \cdot \cos\theta + \sin\left[\tan^{-1}\left(\frac{l}{w}\right)\right] \cdot \sin\theta \right\} \tag{B.3}$$

$$= \sqrt{w^2 + l^2} \cdot \cos\left[\theta - \tan^{-1}\left(\frac{l}{w}\right)\right]. \tag{B.4}$$

Since the transformation over any $\frac{\pi}{2}$ interval is equivalent, the shift by $\tan^{-1}\left(\frac{l}{w}\right)$ is irrelevant. For simplicity of derivation, consider the interval $[\theta_1, \theta_2]$, where $\theta_1 = \cos^{-1}\left(\frac{l}{\sqrt{w^2+l^2}}\right)$ and $\theta_2 = \cos^{-1}\left(\frac{w}{\sqrt{w^2+l^2}}\right)$. We define the random variable Θ to have a uniform density

over this interval:

$$p_\Theta(\theta) = \begin{cases} 2/\pi, & \theta_1 \leq \theta \leq \theta_2 \\ 0, & else. \end{cases} \tag{B.5}$$

To compute the probability density of $\hat{w}$, we apply the transformation:

$$g(\theta) = \hat{w}(\theta)/\sqrt{w^2 + l^2} = \cos\theta, \tag{B.6}$$

to Θ to produce the random variable Y (i.e., $Y = g(\Theta)$). The distribution function technique calculates the density $p_Y(y)$ of Y by first finding the cumulative distribution function $P_Y(y)$ and then taking the derivative. There are two cases.

Case 1: $\cos\theta_2 < y < \cos\theta_1$

$$P_Y(y) = \int_{\cos^{-1} y}^{\theta_2} \frac{2}{\pi} \cdot d\theta \tag{B.7}$$

$$= -\frac{2}{\pi}\cos^{-1} y + \frac{2}{\pi}\theta_2 \tag{B.8}$$

$$p_Y(y) = \frac{2}{\pi\sqrt{1-y^2}} \tag{B.9}$$

Case 2: $\cos\theta_1 < y < 1$

$$P_Y(y) = \int_{-\theta_1}^{-\cos^{-1} y} \frac{2}{\pi} \cdot d\theta + \int_{\cos^{-1} y}^{\theta_2} \frac{2}{\pi} \cdot d\theta \tag{B.10}$$

$$= -\frac{4}{\pi}\cos^{-1} y + \frac{2}{\pi}(\theta_1 + \theta_2) \tag{B.11}$$

$$p_Y(y) = \frac{4}{\pi\sqrt{1-y^2}} \tag{B.12}$$

Thus we have that:

$$p_Y(y) = \begin{cases} \frac{2}{\pi\sqrt{1-y^2}}, & \cos\theta_2 < y < \cos\theta_1 \\ \frac{4}{\pi\sqrt{1-y^2}}, & \cos\theta_1 < y < 1 \end{cases} \tag{B.13}$$

Substituting in θ_1, θ_2 and y, we obtain the probability density function $p_{\hat{w}}(\hat{w})$ in Equation 6.2.

B.2 Occlusion Prior

We show how to compute the area A_{V_i,O_c} covering all the possible positions of the red block in Figure 6.3a which occlude the object but keep the point X_i visible. This region is specified in green in Figure B.1. To compute the area A_{V_i,O_c}, we break it up into the area of three parts:

$$A_{V_i,O_c} = \Delta_{1,1} + \Delta_{1,2} + \Delta_2. \tag{B.14}$$

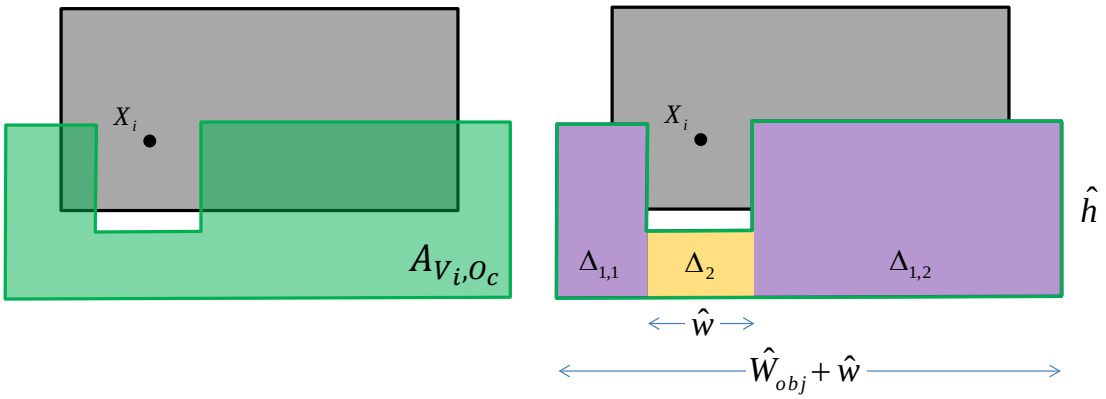


Figure B.1: Detailed illustration of how to compute the area A_{V_i,O_c} covering all the possible positions where the red block in Figure 6.3a occludes the object while keeping X_i visible.

From the figure, we can see that the purple region has area:

$$\Delta_{1,1} + \Delta_{1,2} = \hat{W}_{obj} \cdot \hat{h}. \tag{B.15}$$

Note that this area does not depend on the position of X_i. On the other hand, the area of yellow region does depend on the y coordinate of X_i. If the projected height of the block $\hat{h}$ is shorter than y_i, the height of the yellow region is $\hat{h}$. If it is taller, there are less possible positions of the red block and the height of the yellow region is y_i. Thus its area is:

$$\Delta_2 = \begin{cases} \hat{w} \cdot \hat{h}, & \hat{h} \leq y_i \\ \hat{w} \cdot y_i, & \hat{h} > y_i \end{cases} \tag{B.16}$$

Combining Equations B.15 and B.16, we get the area A_{V_i,O_c} in Equation 6.7.

B.3 Occlusion Conditional Likelihood

We show how to compute the area A_{V_i,V_j,O_c} covering all the possible positions of the red block in Figure 6.3a which occlude the object but keep both points X_i and X_j visible. Without loss of generality, assume that X_i is lower than X_j (i.e., $y_i \leq y_j$). If this is not the case, we can simply switch the points. The region is specified in green in Figure B.2. To compute the area A_{V_i,V_j,O_c}, we break it up into the area of five parts:

$$A_{V_i,V_j,O_c} = \Lambda_{1,1} + \Lambda_{1,2} + \Lambda_2 + \Lambda_3 + \Lambda_4 \tag{B.17}$$

From the figure, we can see that the purple region has area:

$$\Lambda_{1,1} + \Lambda_{1,2} = (\hat{W}_{obj} - |x_i - x_j|) \cdot \hat{h} \tag{B.18}$$

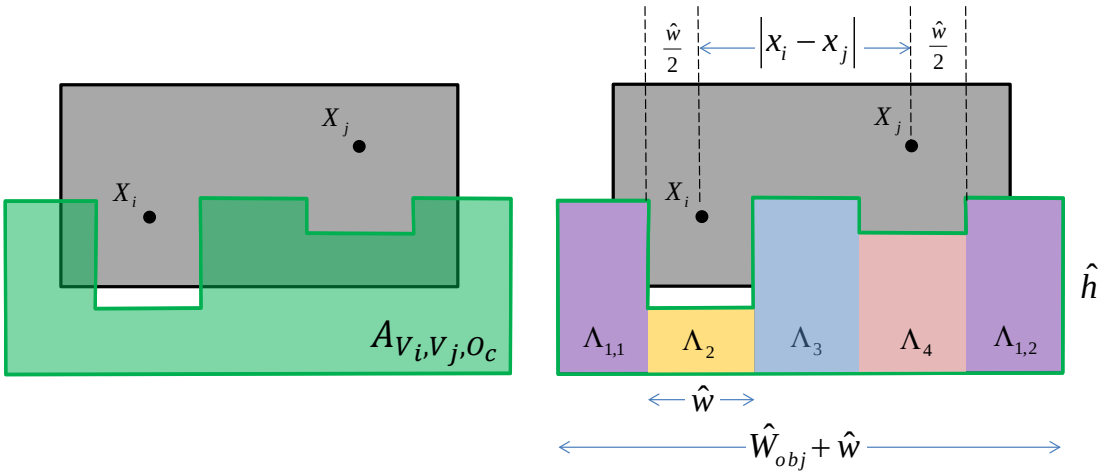


Figure B.2: Detailed illustration of how to compute the area A_{V_i,V_j,O_c} covering all the possible positions where the red block in Figure 6.3a occludes the object while keeping both X_i and X_j visible.

The area of the yellow region is the same as for the occlusion prior, and it does not depend on the location of X_j:

$$\Lambda_2 = \begin{cases} \hat{w} \cdot \hat{h}, & \hat{h} \leq y_i \\ \hat{w} \cdot y_i, & \hat{h} > y_i \end{cases} \tag{B.19}$$

The blue region covers the positions of the block which can fit in between X_i and X_j. If the projected width $\hat{w}$ is greater than $|x_i - x_j|$, it can not fit in this region and the area is 0. However, if it is less than $|x_i - x_j|$, the width of the blue region is $|x_i - x_j| - \hat{w}$. Thus the area is:

$$\Lambda_3 = \begin{cases} (|x_i - x_j| - \hat{w}) \cdot \hat{h}, & \hat{w} \leq |x_i - x_j| \\ 0, & \hat{w} > |x_i - x_j| \end{cases} \tag{B.20}$$

The orange region covers the positions of the block that are below X_j. When the projected width $\hat{w}$ is less than $|x_i - x_j|$, the computation of the area is similar to Λ_2. However, when it is greater, the possible horizontal positions of the block is restricted by point X_i. In this case, instead of $\hat{w}$ positions, there are only $|x_i - x_j|$ positions. Thus the area is:

$$\Lambda_4 = \begin{cases} \hat{w} \cdot \hat{h}, & \hat{w} \leq |x_i - x_j|, \hat{h} \leq y_j \\ \hat{w} \cdot y_j, & \hat{w} \leq |x_i - x_j|, \hat{h} > y_j \\ |x_i - x_j| \cdot \hat{h}, & \hat{w} > |x_i - x_j|, \hat{h} \leq y_j \\ |x_i - x_j| \cdot y_j, & \hat{w} > |x_i - x_j|, \hat{h} > y_j. \end{cases} \tag{B.21}$$

Combining Equations B.18, B.19, B.20, B.21 and simplifying it, we get:

$$\begin{aligned} A_{V_i,V_j,O_c} = (\hat{W}_{obj} - |x_i - x_j|) \cdot \hat{h} \\ + \hat{w} \cdot \min(\hat{h}, y_i) \\ + \delta(\hat{w} \leq |x_i - x_j|) \cdot (|x_i - x_j| - \hat{w}) \cdot \hat{h} \\ + \min(\hat{w}, |x_i - x_j|) \cdot \min(\hat{h}, y_i) \end{aligned} \tag{B.22}$$

Integrating over the projected width and height distributions $p_{\hat{w}}$ and $p_{\hat{h}}$, we get the average area in Equation 6.11.

B.4 Computing Area for Silhouette

We show how to compute the area of the silhouette A_s used in Equation 6.14 and 6.15. Given a mask M, we extract the height of the lowest point $\mathcal{Y}^M(x)$ relative to bottom of the mask for each unique position $x \in \mathcal{X}^M$. Then for an occluder with projected width and height $(\hat{w}, \hat{h})$, the area covering all the positions that intersect the bounding box but not the silhouette is given by,

$$A_s = \int \Omega(\mathcal{X}^M, \mathcal{Y}^M, \hat{w}, \hat{h}) \cdot dx, \tag{B.23}$$

where $\Omega(\mathcal{X}^M, \mathcal{Y}^M, \hat{w}, \hat{h})$ is the *Mask Sliding Min* shown in Algorithm 2. This function considers the highest position to place an occluder at position x while not intersecting the mask. The position is lower than than height of the occluder and lower than the height of all mask points within an interval $[-\hat{w}/2, \hat{w}/2]$ of x.

Algorithm 2 Mask Sliding Min, $\Omega(\mathcal{X}^M, \mathcal{Y}^M, \hat{w}, \hat{h})$

Require: bottom of mask $(\mathcal{X}^M, \mathcal{Y}^M)$, projected width $\hat{w}$, projected height $\hat{h}$

1: $\mathcal{Y}^M = \min(\mathcal{Y}^M, \hat{h})$
2: **for** $x = \min(\mathcal{X}^M) - \frac{\hat{w}}{2} \rightarrow \max(\mathcal{X}^M) + \frac{\hat{w}}{2}$ **do**
3: $\quad \mathcal{Z}(x) = \min_{\hat{x} \in [x - \hat{w}/2, x + \hat{w}/2]} \mathcal{Y}^M(\hat{x})$
4: **end for**
5: **return** $\mathcal{Z}$

For a distribution of occluding blocks $p_{\hat{w}}(\hat{w})$ and $p_{\hat{h}}(\hat{h})$ for $\hat{w}$ and $\hat{h}$ respectively, the average areas are then given by:

$$\iiint \Omega(\mathcal{X}^M, \mathcal{Y}^M, \hat{w}, \hat{h}) \cdot p_{\hat{w}}(\hat{w}) \cdot p_{\hat{h}}(\hat{h}) \cdot dx \cdot d\hat{w} \cdot d\hat{h}. \tag{B.24}$$

References

[1] Google Goggles. `http://www.google.com/mobile/goggles`.

[2] A. Ansar and K. Daniilidis. Linear pose estimation from points or lines. *IEEE Transactions on Pattern Analysis and Machine Intelligence (PAMI)*, 25(5):578–589, 2003.

[3] R. Arandjelović and A. Zisserman. Smooth object retrieval using a bag of boundaries. In *Proceedings of IEEE International Conference on Computer Vision (ICCV)*, 2011.

[4] P. Arbeláez, M. Maire, C. Fowlkes, and J. Malik. From contours to regions: An empirical evaluation. In *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2009.

[5] P. Arbeláez, M. Maire, C. Fowlkes, and J. Malik. Contour detection and hierarchical image segmentation. *IEEE Transactions on Pattern Analysis and Machine Intelligence (PAMI)*, 33(5):898–916, 2011.

[6] X. Bai, Q. Li, L. Latecki, W. Liu, and Z. Tu. Shape band: A deformable object detection approach. In *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2009.

[7] S. Bao, M. Sun, and S. Savarese. Toward coherent object detection and scene layout understanding. In *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2010.

[8] H. Barrow, J. Tenenbaum, R. Bolles, and H. Wolf. Parametric correspondence and chamfer matching: Two new techniques for image matching. In *Proceedings of International Joint Conference on Artificial Intelligence (IJCAI)*, 1977.

[9] H. Bay, A. Ess, T. Tuytelaars, and L. Van Gool. Speeded-up robust features (SURF). *Computer Vision and Image Understanding (CVIU)*, 110(3):346–359,

2008.

[10] S. Belongie, J. Malik, and J. Puzicha. Shape matching and object recognition using shape contexts. *IEEE Transactions on Pattern Analysis and Machine Intelligence (PAMI)*, 2002.

[11] A. Berg, T. Berg, and J. Malik. Shape matching and object recognition using low distortion correspondences. In *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2005.

[12] P. Bhat, C. Zitnick, M. Cohen, and B. Curless. GradientShop: A gradient-domain optimization framework for image and video filtering. *ACM Transactions on Graphics (TOG)*, 29(2):10, 2010.

[13] I. Biederman. Recognition-by-components: A theory of human image understanding. *Psychological Review*, 94(2):115, 1987.

[14] I. Biederman. Recognizing depth-rotated objects: A review of recent research and theory. *Spatial Vision, 13*, 2(3):241–253, 2000.

[15] G. Bilodeau and R. Bergevin. Generic modeling of 3D objects from single 2D images. In *Proceedings of International Conference on Pattern Recognition (ICPR)*, 2000.

[16] H. Breu, J. Gil, D. Kirkpatrick, and M. Werman. Linear time Euclidean distance transform algorithms. *IEEE Transactions on Pattern Analysis and Machine Intelligence (PAMI)*, 17(5):529–533, 1995.

[17] M. Calonder, V. Lepetit, C. Strecha, and P. Fua. BRIEF: Binary robust independent elementary features. In *Proceedings of European Conference on Computer Vision (ECCV)*, 2010.

[18] T. Cass. Robust affine structure matching for 3D object recognition. *IEEE Transactions on Pattern Analysis and Machine Intelligence (PAMI)*, 20(11):1265–1274, 1998.

[19] O. Chum, J. Philbin, J. Sivic, M. Isard, and A. Zisserman. Total recall: Automatic query expansion with a generative feature model for object retrieval. In *Proceedings of IEEE International Conference on Computer Vision (ICCV)*, 2007.

[20] M. Ciocarlie, K. Hsiao, E. G. Jones, S. Chitta, R. B. Rusu, and I. A. Sucan. Towards reliable grasping and manipulation in household environments. In *Pro-*

ceedings of International Symposium on Experimental Robotics (ISER), 2010.

[21] A. Collet, D. Berenson, S. Srinivasa, and D. Ferguson. Object recognition and full pose registration from a single image for robotic manipulation. In *Proceedings of IEEE International Conference on Robotics and Automation (ICRA)*, 2009.

[22] A. Collet, M. Martinez, and S. Srinivasa. The MOPED framework: Object recognition and pose estimation for manipulation. *International Journal of Robotics Research (IJRR)*, 2011.

[23] N. Dalal. *Finding people in images and videos.* PhD thesis, Institut National Polytechnique de Grenoble / INRIA Grenoble, 2006.

[24] N. Dalal and B. Triggs. Histograms of oriented gradients for human detection. In *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2005.

[25] J. Deng, W. Dong, R. Socher, L.-J. Li, K. Li, and L. Fei-Fei. ImageNet: A large-scale hierarchical image database. In *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2009.

[26] P. Dollár, Z. Tu, and S. Belongie. Supervised learning of edges and object boundaries. In *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2006.

[27] P. Dollár, C. Wojek, B. Schiele, and P. Perona. Pedestrian detection: An evaluation of the state of the art. *IEEE Transactions on Pattern Analysis and Machine Intelligence (PAMI)*, 2011.

[28] M. Everingham, L. Van Gool, C. Williams, J. Winn, and A. Zisserman. The PASCAL Visual Object Classes (VOC) challenge. *International Journal of Computer Vision (IJCV)*, 2010.

[29] P. Felzenszwalb and D. Huttenlocher. Distance transforms of sampled functions. *Theory of Computing*, 8, 2012.

[30] V. Ferrari, L. Fevrier, F. Jurie, and C. Schmid. Groups of adjacent contour segments for object detection. *IEEE Transactions on Pattern Analysis and Machine Intelligence (PAMI)*, 2008.

[31] V. Ferrari, T. Tuytelaars, and L. Van Gool. Object detection by contour segment networks. In *Proceedings of European Conference on Computer Vision (ECCV)*,

2006.

[32] P. Flynn and A. Jain. CAD-based computer vision: From CAD models to relational graphs. *IEEE Transactions on Pattern Analysis and Machine Intelligence (PAMI)*, 13(2):114–132, 1991.

[33] R. Fransens, C. Strecha, and L. Van Gool. A mean field EM-algorithm for coherent occlusion handling in MAP-estimation prob. In *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2006.

[34] W. Freeman and E. Adelson. The design and use of steerable filters. *IEEE Transactions on Pattern Analysis and Machine Intelligence (PAMI)*, 13(9):891–906, 1991.

[35] J. Gall and V. Lempitsky. Class-specific Hough forests for object detection. In *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2009.

[36] T. Gao, B. Packer, and D. Koller. A segmentation-aware object detection model with occlusion handling. In *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2011.

[37] J. Geusebroek, G. Burghouts, and A. Smeulders. The Amsterdam library of object images. *International Journal of Computer Vision (IJCV)*, 61(1), 2005.

[38] R. B. Girshick, P. F. Felzenszwalb, and D. McAllester. Object detection with grammar models. In *Proceedings of Neural Information Processing Systems (NIPS)*, 2011.

[39] I. Gordon and D. G. Lowe. What and where: 3D object recognition with accurate pose. In *Toward Category-Level Object Recognition*, 2006.

[40] W. Grimson and T. Lozano-Perez. Localizing overlapping parts by searching the interpretation tree. *IEEE Transactions on Pattern Analysis and Machine Intelligence (PAMI)*, 1987.

[41] W. Grimson, T. Lozano-Pérez, and D. Huttenlocher. *Object recognition by computer*. MIT Press, 1990.

[42] C. Harris and M. Stephens. A combined corner and edge detector. In *Proceedings of Alvey Vision Conference*, 1988.

[43] S. Hinterstoisser, S. Benhimane, and N. Navab. N3M: Natural 3D markers for real-

time object detection and pose estimation. In *Proceedings of IEEE International Conference on Computer Vision (ICCV)*, 2007.

[44] S. Hinterstoisser, C. Cagniart, S. Ilic, P. Sturm, N. Navab, P. Fua, and V. Lepetit. Gradient response maps for real-time detection of texture-less objects. *IEEE Transactions on Pattern Analysis and Machine Intelligence (PAMI)*, 2011.

[45] S. Hinterstoisser, V. Lepetit, S. Ilic, P. Fua, and N. Navab. Dominant orientation templates for real-time detection of texture-less objects. In *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2010.

[46] R. V. Hogg and E. Tanis. *Probability and Statistical Inference*. Pearson, 8th edition, 2009.

[47] D. Hoiem, A. Efros, and M. Hebert. Putting objects in perspective. *International Journal of Computer Vision (IJCV)*, 2008.

[48] E. Hsiao, A. Collet, and M. Hebert. Making specific features less discriminative to improve point-based 3D object recognition. In *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2010.

[49] E. Hsiao and M. Hebert. Occlusion reasoning for object detection under arbitrary viewpoint. In *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2012.

[50] E. Hsiao and M. Hebert. Gradient networks: Explicit shape matching without extracting edges. In *Proceedings of AAAI Conference on Artificial Intelligence (AAAI)*, 2013.

[51] E. Hsiao and M. Hebert. Shape-based instance detection under arbitrary viewpoint. In Z. Pizlo and S. Dickinson, editors, *Shape Perception in Human and Computer Vision: An Interdisciplinary Perspective*. Springer, 2013.

[52] D. Huttenlocher and S. Ullman. Recognizing solid objects by alignment with an image. *International Journal of Computer Vision (IJCV)*, 5(2), 1990.

[53] K. Ikeuchi. Generating an interpretation tree from a CAD model for 3D-object recognition in bin-picking tasks. *International Journal of Computer Vision (IJCV)*, 1(2):145–165, 1987.

[54] A. Khosla, N. Jayadevaprakash, B. Yao, and L. Fei-Fei. Novel dataset for fine-grained image categorization. In *First Workshop on Fine-Grained Visual Cat-*

egorization, Proceedings of IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2011.

[55] J. Koenderink. *Solid shape*. MIT Press, 1990.

[56] N. Kumar, P. N. Belhumeur, A. Biswas, D. W. Jacobs, W. J. Kress, I. C. Lopez, and J. V. Soares. Leafsnap: A computer vision system for automatic plant species identification. In *Proceedings of European Conference on Computer Vision (ECCV)*, 2012.

[57] S. Kwak, W. Nam, B. Han, and J. H. Han. Learn occlusion with likelihoods for visual tracking. In *Proceedings of IEEE International Conference on Computer Vision (ICCV)*, 2011.

[58] K. Lai, L. Bo, X. Ren, and D. Fox. A large-scale hierarchical multi-view RGB-D object dataset. In *Proceedings of IEEE International Conference on Robotics and Automation (ICRA)*, 2011.

[59] J. F. Lalonde, D. Hoiem, A. A. Efros, C. Rother, J. Winn, and A. Criminisi. Photo clip art. In *ACM SIGGRAPH*, 2007.

[60] C. Lampert, M. Blaschko, and T. Hofmann. Beyond sliding windows: Object localization by efficient subwindow search. In *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2008.

[61] Y. LeCun, F. Huang, and L. Bottou. Learning methods for generic object recognition with invariance to pose and lighting. In *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2004.

[62] B. Leibe, A. Leonardis, and B. Schiele. Combined object categorization and segmentation with an implicit shape model. In *Workshop on Statistical Learning in Computer Vision, Proceedings of European Conference on Computer Vision (ECCV)*, 2004.

[63] M. Leordeanu, M. Hebert, and R. Sukthankar. Beyond local appearance: Category recognition from pairwise interactions of simple features. In *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2007.

[64] V. Lepetit and P. Fua. Keypoint recognition using randomized trees. *IEEE Transactions on Pattern Analysis and Machine Intelligence (PAMI)*, 28(9):1465–1479, 2006.

[65] V. Lepetit, F. Moreno-Noguer, and P. Fua. EPnP: An accurate O(n) solution to the PnP problem. *International Journal of Computer Vision (IJCV)*, 81(2), 2009.

[66] Y. Li, L. Gu, and T. Kanade. Robustly aligning a shape model and its application to car alignment of unknown pose. *IEEE Transactions on Pattern Analysis and Machine Intelligence (PAMI)*, 33(9):1860–1876, 2011.

[67] T. Lindeberg. Feature detection with automatic scale selection. *International Journal of Computer Vision (IJCV)*, 30(2):79–116, 1998.

[68] D. G. Lowe. *Perceptual organization and visual recognition.* PhD thesis, Stanford University, 1984.

[69] D. G. Lowe. Local feature view clustering for 3D object recognition. In *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2001.

[70] D. G. Lowe. Distinctive image features from scale-invariant keypoints. *International Journal of Computer Vision (IJCV)*, 2004.

[71] J. Luo and C. Guo. Perceptual grouping of segmented regions in color images. *Pattern Recognition*, 36(12):2781–2792, 2003.

[72] J. Mairal, M. Leordeanu, F. Bach, M. Hebert, and J. Ponce. Discriminative sparse image models for class-specific edge detection and image interpretation. In *Proceedings of European Conference on Computer Vision (ECCV)*, 2008.

[73] T. Malisiewicz and A. A. Efros. Ensemble of exemplar-SVMs for object detection and beyond. In *Proceedings of IEEE International Conference on Computer Vision (ICCV)*, 2011.

[74] J. Matas, O. Chum, M. Urban, and T. Pajdla. Robust wide baseline stereo from maximally stable extremal regions. In *Proceedings of British Machine Vision Conference (BMVC)*, 2002.

[75] D. Meger, C. Wojek, B. Schiele, and J. J. Little. Explicit occlusion reasoning for 3D object detection. In *Proceedings of British Machine Vision Conference (BMVC)*, 2011.

[76] K. Mikolajczyk and C. Schmid. Indexing based on scale invariant interest points. In *Proceedings of IEEE International Conference on Computer Vision (ICCV)*, 2001.

[77] K. Mikolajczyk and C. Schmid. An affine invariant interest point detector. In *Proceedings of European Conference on Computer Vision (ECCV)*, 2002.

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

[79] K. Mikolajczyk and C. Schmid. A performance evaluation of local descriptors. *IEEE Transactions on Pattern Analysis and Machine Intelligence (PAMI)*, 27(10):1615–1630, 2005.

[80] K. Mikolajczyk, T. Tuytelaars, C. Schmid, A. Zisserman, J. Matas, F. Schaffalitzky, T. Kadir, and L. Van Gool. A comparison of affine region detectors. *International Journal of Computer Vision (IJCV)*, 65(1-2):43–72, 2005.

[81] B. Moghaddam and A. Pentland. Probabilistic visual learning for object representation. *IEEE Transactions on Pattern Analysis and Machine Intelligence (PAMI)*, 19(7):696–710, 1997.

[82] J.-M. Morel and G. Yu. ASIFT: A new framework for fully affine invariant image comparison. *SIAM Journal on Imaging Sciences*, 2(2):438–469, 2009.

[83] S. Nene, S. Nayar, and H. Murase. Columbia object image library. *TR CUCS-006-96, Columbia University*, 1996.

[84] D. Nister and H. Stewenius. Scalable recognition with a vocabulary tree. In *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2006.

[85] A. Opelt, A. Pinz, and A. Zisserman. A boundary-fragment-model for object detection. In *Proceedings of European Conference on Computer Vision (ECCV)*, 2006.

[86] M. Ozuysal, M. Calonder, V. Lepetit, and P. Fua. Fast keypoint recognition using random ferns. *IEEE Transactions on Pattern Analysis and Machine Intelligence (PAMI)*, 32(3):448–461, 2010.

[87] N. Payet and S. Todorovic. From contours to 3D object detection and pose estimation. In *Proceedings of IEEE International Conference on Computer Vision (ICCV)*, 2011.

[88] H. Pirsiavash and D. Ramanan. Steerable part models. In *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2012.

[89] H. Plantinga and C. Dyer. Visibility, occlusion, and the aspect graph. *International Journal of Computer Vision (IJCV)*, 1990.

[90] T. Poggio and S. Edelman. A network that learns to recognize 3D objects. *Nature*, 343(6255):263–266, 1990.

[91] M. Prasad, A. Zisserman, A. Fitzgibbon, M. Kumar, and P. Torr. Learning class-specific edges for object detection and segmentation. *Computer Vision Graphics and Image Processing (CVGIP)*, 2006.

[92] V. Prisacariu and I. Reid. fastHOG-a real-time GPU implementation of HOG. *Department of Engineering Science, Oxford University, Technical Report*, 2009.

[93] M. Radovanović, A. Nanopoulos, and M. Ivanović. Nearest neighbors in high-dimensional data: The emergence and influence of hubs. In *Proceedings of International Conference on Machine Learning (ICML)*, 2009.

[94] X. Ren and M. Philipose. Egocentric recognition of handled objects: Benchmark and analysis. In *Egovision Workshop, Proceedings of IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2009.

[95] L. Renninger and J. Malik. When is scene identification just texture recognition? *Vision Research*, 44(19), 2004.

[96] E. Rosten, R. Porter, and T. Drummond. Faster and better: A machine learning approach to corner detection. *IEEE Transactions on Pattern Analysis and Machine Intelligence (PAMI)*, 32(1):105–119, 2010.

[97] F. Rothganger, S. Lazebnik, C. Schmid, and J. Ponce. 3D object modeling and recognition using affine-invariant patches and multi-view spatial constraints. In *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2003.

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

[99] B. C. Russell, A. Torralba, K. P. Murphy, and W. T. Freeman. LabelMe: A database and web-based tool for image annotation. *International Journal of Computer Vision (IJCV)*, 77(1-3):157–173, 2008.

[100] L. Santalo. *Integral geometry and geometric probability*. Addison-Wesley Publishing

Co., Reading, MA, 1976.

[101] W. J. Scheirer, N. Kumar, P. N. Belhumeur, and T. E. Boult. Multi-attribute spaces: Calibration for attribute fusion and similarity search. In *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2012.

[102] C. Schmid and R. Mohr. Local grayvalue invariants for image retrieval. *IEEE Transactions on Pattern Analysis and Machine Intelligence (PAMI)*, 19(5):530–535, 1997.

[103] G. Schweighofer and A. Pinz. Robust pose estimation from a planar target. *IEEE Transactions on Pattern Analysis and Machine Intelligence (PAMI)*, 28(12), 2006.

[104] J. Shotton, A. Blake, and R. Cipolla. Multiscale categorical object recognition using contour fragments. *IEEE Transactions on Pattern Analysis and Machine Intelligence (PAMI)*, 30(7):1270–1281, 2008.

[105] J. Shotton, A. Fitzgibbon, M. Cook, T. Sharp, M. Finocchio, R. Moore, A. Kipman, and A. Blake. Real-time human pose recognition in parts from single depth images. In *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2011.

[106] S. Srinivasa, D. Ferguson, C. Helfrich, D. Berenson, A. Collet, R. Diankov, G. Gallagher, G. Hollinger, J. Kuffner, and M. V. Weghe. HERB: A home exploring robotic butler. *Autonomous Robots*, 28(1):5–20, 2010.

[107] P. Srinivasan, Q. Zhu, and J. Shi. Many-to-one contour matching for describing and discriminating object shape. In *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2010.

[108] M. Stevens and J. Beveridge. *Integrating Graphics and Vision for Object Recognition*. Kluwer Academic Publishers, 2000.

[109] H. Su, M. Sun, L. Fei-Fei, and S. Savarese. Learning a dense multi-view representation for detection, viewpoint classification and synthesis of object categories. In *Proceedings of IEEE International Conference on Computer Vision (ICCV)*, 2009.

[110] R. Szeliski and S. Kang. Recovering 3D shape and motion from image streams using non-linear least squares. In *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 1993.

[111] A. Thayananthan, B. Stenger, P. Torr, and R. Cipolla. Shape context and chamfer

matching in cluttered scenes. In *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2003.

[112] A. Thomas, V. Ferrari, B. Leibe, T. Tuytelaars, B. Schiele, and L. Van Gool. Towards multi-view object class detection. In *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2006.

[113] E. Tola, V. Lepetit, and P. Fua. DAISY: An efficient dense descriptor applied to wide-baseline stereo. *IEEE Transactions on Pattern Analysis and Machine Intelligence (PAMI)*, 32(5):815–830, 2010.

[114] A. Torralba, K. Murphy, and W. Freeman. Sharing visual features for multiclass and multiview object detection. *IEEE Transactions on Pattern Analysis and Machine Intelligence (PAMI)*, 29(5):854–869, 2007.

[115] A. Toshev, B. Taskar, and K. Daniilidis. Object detection via boundary structure segmentation. In *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2010.

[116] Z. Tu. Probabilistic boosting-tree: Learning discriminative models for classification, recognition, and clustering. In *Proceedings of IEEE International Conference on Computer Vision (ICCV)*, 2005.

[117] T. Tuytelaars and L. Van Gool. Wide baseline stereo matching based on local, affinely invariant regions. In *Proceedings of British Machine Vision Conference (BMVC)*, 2000.

[118] P. Viola and M. Jones. Robust real-time object detection. *International Journal of Computer Vision (IJCV)*, 2001.

[119] N. Wang and H. Ai. Who blocks who: Simultaneous clothing segmentation for grouping images. In *Proceedings of IEEE International Conference on Computer Vision (ICCV)*, 2011.

[120] X. Wang, T. Han, and S. Yan. An HOG-LBP human detector with partial occlusion handling. In *Proceedings of IEEE International Conference on Computer Vision (ICCV)*, 2009.

[121] P. Welinder, S. Branson, T. Mita, C. Wah, F. Schroff, S. Belongie, and P. Perona. Caltech-UCSD Birds 200. Technical Report CNS-TR-2010-001, California Institute of Technology, 2010.

[122] G. Willems, T. Tuytelaars, and L. Gool. An efficient dense and scale-invariant spatio-temporal interest point detector. In *Proceedings of European Conference on Computer Vision (ECCV)*, 2008.

[123] J. Winn, A. Criminisi, and T. Minka. Object categorization by learned universal visual dictionary. In *Proceedings of IEEE International Conference on Computer Vision (ICCV)*, 2005.

[124] A. Witkin and J. Tenenbaum. On the role of structure in vision. *Human and Machine Vision*, 1:481–543, 1983.

[125] B. Wu and R. Nevatia. Detection of multiple, partially occluded humans in a single image by bayesian combination of edgelet part detectors. In *Proceedings of IEEE International Conference on Computer Vision (ICCV)*, 2005.

[126] J. Xiao, J. Hays, K. A. Ehinger, A. Oliva, and A. Torralba. SUN database: Large-scale scene recognition from abbey to zoo. In *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2010.

[127] M. Zerroug and R. Nevatia. Part-based 3D descriptions of complex objects from a single image. *IEEE Transactions on Pattern Analysis and Machine Intelligence (PAMI)*, 21(9):835–848, 1999.

[128] D. Zhao and J. Chen. Affine curve moment invariants for shape recognition. *Pattern Recognition*, 30(6):895–901, 1997.

[129] S. Zhu and D. Mumford. A stochastic grammar of images. *Foundations and Trends in Computer Graphics and Vision*, 2(4):259–362, 2006.