Context-driven Object Detection and Segmentation with Auxiliary Information

Tao Wang

A thesis submitted for the degree of
Doctor of Philosophy
of The Australian National University

November 2016

Except where otherwise indicated, this thesis is my own original work.

Tao Wang

Tao Wang
8·11·2016

Acknowledgments

Although supervisors always play an instrumental role in one's journey towards thesis completion, I believe my own supervisors, Xuming He and Nick Barnes, deserve a very special mention for their continual encouragement, exceptional guidance and unwavering support over the years. I am deeply indebted to Xuming, for his dedication and understanding in vision and learning opened up a whole new world for me. Xuming has always been a role model, a caring mentor and a close friend of mine. I want to thank him for encouraging me develop a well-rounded understanding of the problems I wanted to tackle, for providing very practical guidance while it was difficult to make progress, and for repeatedly helping me with presenting my findings in a logical and lucid manner. It is fair to say that I would not be who I am now had I not worked with him. I am deeply grateful to Nick, for the countless times of inspiring and encouraging discussions, for the many insightful ideas he contributed, and also for tirelessly fixing imperfections in my presentations and writings. It has been a privilege and an infinite source of gratitude to be guided by Nick who has always been understanding and supportive throughout my study.

I would like to extend my sincerest thanks to my supervisory panel, Chunhua Shen and Richard Hartley for their sound advice and thought-provoking questions. In particular, I have learned a great deal from discussions with Chunhua in the first year of my study. I also appreciate his encouragement and support while I write up my thesis. Furthermore, I am fortunate to have been a member of the BVA vision processing team, the NICTA computer vision research group and the broader ANU vision and robotics group. I have immensely benefited from seminars, summer schools and other group activities which allow me to meet and learn from students and scholars from around Australia. I want to especially thank Hanxi Li, Miaomiao Liu and Mathieu Salzmann for the excellent discussions and enlightenment in our weekly reading groups. In addition, I want to thank NICTA's Intelligent Transport Systems project for providing me an opportunity to work as a part-time research programmer.

I also want to thank my lab mates for their many insightful ideas on my thesis project, their company, and friendship: Buyu Liu, Lin Gu, Fang Wang, Weipeng Xu, Wei Zhuo, Zongyuan Ge, Zeeshan Hayder, Lachlan Horne, Samunda Perera, David Feng, Kyoungup Park, Zhihui Hao, Cong Phuoc Huynh... The days we worked in the same lab and the happy times we spent together in the beautiful landscapes of Canberra have been seared in my memory.

Last but not least, thank all my friends outside my academic life. Thanks to my parents, for their life-long love and encouragement.

Abstract

One fundamental problem in computer vision and robotics is to localize objects of interest in an image. The task can either be formulated as an object detection problem if the objects are described by a set of pose parameters, or an object segmentation one if we recover object boundary precisely. A key issue in object detection and segmentation concerns exploiting the spatial context, as local evidence is often insufficient to determine object pose in the presence of heavy occlusions or large object appearance variations. This thesis addresses the object detection and segmentation problem in such adverse conditions with auxiliary depth data provided by RGBD cameras. We focus on four main issues in context-aware object detection and segmentation: 1) what are the effective context representations? 2) how can we work with limited and imperfect depth data? 3) how to design depth-aware features and integrate depth cues into conventional visual inference tasks? 4) how to make use of unlabeled data to relax the labeling requirements for training data?

We discuss three object detection and segmentation scenarios based on varying amounts of available auxiliary information. In the first case, depth data are available for model training but not available for testing. We propose a structured Hough voting method for detecting objects with heavy occlusion in indoor environments, in which we extend the Hough hypothesis space to include both the object's location, and its visibility pattern. We design a new score function that accumulates votes for object detection and occlusion prediction. In addition, we explore the correlation between objects and their environment, building a depth-encoded object-context model based on RGBD data. In the second case, we address the problem of localizing glass objects with noisy and incomplete depth data. Our method integrates the intensity and depth information from a single view point, and builds a Markov Random Field that predicts glass boundary and region jointly. In addition, we propose a nonparametric, data-driven label transfer scheme for local glass boundary estimation. A weighted voting scheme based on a joint feature manifold is adopted to integrate depth and appearance cues, and we learn a distance metric on the depth-encoded feature manifold. In the third case, we make use of unlabeled data to relax the annotation requirements for object detection and segmentation, and propose a novel data-dependent margin distribution learning criterion for boosting, which utilizes the intrinsic geometric structure of datasets. One key aspect of this method is that it can seamlessly incorporate unlabeled data by including a graph Laplacian regularizer. We demonstrate the performance of our models and compare with baseline methods on several real-world object detection and segmentation tasks, including indoor object detection, glass object segmentation and foreground segmentation in video.

Publications

Several contributions presented in this thesis have been published elsewhere by the author. We list these below:

- **Chapter 3 - Structured Hough Voting for Joint Object Detection and Occlusion Prediction:**

  Tao Wang, Xuming He and Nick Barnes, 'Learning Structured Hough Voting for Joint Object Detection and Occlusion Reasoning', In *Proceedings of IEEE International Conference on Computer Vision and Pattern Recognition (CVPR 2013)*, Portland, USA, 2013.

- **Chapter 4 - Glass Object Segmentation by Joint Inference of Boundary and Depth:**

  Tao Wang, Xuming He and Nick Barnes, 'Glass Object Localization by Joint Inference of Boundary and Depth', In *Proceedings of IEEE International Conference on Pattern Recognition (ICPR 2012)*, Tsukuba, Japan, 2012.

- **Chapter 5 - Glass Object Segmentation by Label Transfer on Joint Depth and Appearance Manifolds:**

  Tao Wang, Xuming He and Nick Barnes, 'Glass Object Segmentation by Label Transfer on Joint Depth and Appearance Manifolds', In *Proceedings of IEEE International Conference on Image Processing (ICIP 2013)*, Melbourne, Australia, 2013.

- **Chapter 6 - Laplacian Margin Distribution Boosting for Learning from Sparsely Labeled Data:**

  Tao Wang, Xuming He, Chunhua Shen, Nick Barnes, 'Laplacian Margin Distribution Boosting for Learning from Sparsely Labeled Data', In *Proceedings of IEEE International Conference on Digital Image Computing: Techniques and Applications (DICTA 2011)*, Noosa, Australia, 2011.

Contents

Acknowledgments **v**

Abstract **vii**

Publications **ix**

1 Introduction **1**
- 1.1 Our research problems . . . 4
- 1.2 Object detection with depth-encoded context . . . 6
- 1.3 Glass segmentation by joint inference of boundary and region . . . 8
- 1.4 Depth-aware features and label transfer . . . 9
- 1.5 Learning from sparsely labeled data . . . 11
- 1.6 Thesis outline . . . 11
- 1.7 Major contributions . . . 13

2 Literature Review **15**
- 2.1 Object detection in computer vision . . . 15
  - 2.1.1 Sliding window detectors . . . 16
  - 2.1.2 Hough transform detectors . . . 20
  - 2.1.3 Object detection with RGBD data . . . 27
  - 2.1.4 Occlusion reasoning for object detection . . . 28
  - 2.1.5 Context modeling for object detection . . . 31
- 2.2 Object segmentation in computer vision . . . 33
  - 2.2.1 Foreground object segmentation . . . 34
  - 2.2.2 Context modeling with Markov Random Fields . . . 36
  - 2.2.3 Inference in Markov Random Fields . . . 41
  - 2.2.4 Glass object segmentation . . . 43
- 2.3 Boosting for learning from sparsely labeled data . . . 47
- 2.4 Summary . . . 51

3 Structured Hough Voting for Joint Object Detection and Occlusion Prediction **53**
- 3.1 Introduction . . . 53
- 3.2 Our approach . . . 55

3.2.1 Structured Hough voting . . . 55
3.2.2 Depth-encoded context . . . 58
3.2.2.1 Second-order features . . . 59
3.3 Model learning and inference . . . 60
3.3.1 Joint inference for object detection and occlusion prediction . . . 60
3.3.2 Learning with depth-augmented data . . . 62
3.4 Experimental evaluation . . . 63
3.4.1 Dataset and setup . . . 63
3.4.2 Model details . . . 64
3.4.3 Quantitative results . . . 64
3.4.4 Segmentation performance analysis . . . 67
3.4.5 More detailed examples . . . 68
3.5 Conclusion . . . 69

4 Glass Object Segmentation by Joint Inference of Boundary and Depth 77
4.1 Introduction . . . 77
4.2 Our approach . . . 79
4.2.1 Boundary and region graph . . . 79
4.2.2 A Markov Random Field on boundaries and superpixels . . . 83
4.2.3 Joint prediction . . . 87
4.2.4 Depth reconstruction . . . 89
4.3 Experimental evaluation . . . 89
4.3.1 Dataset and setup . . . 89
4.3.2 Recall statistics for glass proposal . . . 89
4.3.3 Segmentation results and comparisons . . . 91
4.3.4 Qualitative analysis for joint inference . . . 93
4.4 Conclusion . . . 95

5 Glass Object Segmentation by Label Transfer on Joint Depth and Appearance Manifolds 99
5.1 Introduction . . . 99
5.2 Our approach . . . 101
5.2.1 Superpixels and features . . . 101
5.2.2 Boundary label transfer . . . 102
5.2.3 Object model and inference . . . 103
5.3 Experimental evaluation . . . 104
5.3.1 Data specifications and setup . . . 104
5.3.2 Ablation studies . . . 104

5.3.3 Results and discussion . . . 107
5.3.4 Building subset-specific manifolds . . . 109
5.4 Conclusion . . . 111

6 Laplacian Margin Distribution Boosting for Learning from Sparsely Labeled Data 115
6.1 Introduction . . . 115
6.2 Our approach . . . 117
6.2.1 Margin distribution and Laplacian MDBoost . . . 117
6.2.2 Semi-supervised Laplacian MDBoost . . . 119
6.3 Experimental evaluation . . . 121
6.3.1 Datasets and setup . . . 122
6.3.2 Laplacian MDBoost for supervised learning . . . 123
6.3.3 Semi-supervised Laplacian MDBoost . . . 124
6.3.4 Video segmentation with Semi-supervised Laplacian MDBoost . . . 125
6.3.5 RGBD glass object segmentation with Semi-supervised Laplacian MDBoost . . . 126
6.4 Conclusion . . . 128

7 Conclusion 135
7.1 Primary contributions . . . 135
7.2 Future work . . . 137
7.2.1 3D scene structure reasoning . . . 137
7.2.2 Holistic scene understanding . . . 138
7.2.3 Other types of auxiliary information . . . 138

References 141

List of Figures

1.1 Example of object detection and segmentation. **(a)** input image. **(b)** object detection with bounding boxes. **(c)** semantic segmentation. **(d)** object instance segmentation. 2

1.2 Illustration of the proposed object detector. **(a)** RGB frame with object bounding box (red) and visible part bounding box (green). **(b)** Object centroid voting from multiple layers. **(c)** Combined object centroid voting results. **(d)** Detector output (red) with visibility pattern prediction (green). **(e)** Object visibility pattern prediction results. **(f)** Final segmentation results. 7

1.3 Illustration of the proposed glass object segmentation system. **(a)** Intensity image with ground truth foreground mask overlaid. **(b)** Edge detector output. **(c)** Triangulation result. **(d)** Boundary classifier output (magnified). **(e)** Superpixel classifier output (magnified). **(f)** Reconstructed depth with joint inference result overlaid. 8

1.4 **Top:** Illustration of feature manifold based glass boundary classification. We use a learned feature manifold to match every boundary fragment in a test scene (shown as image patches) to training set in order to predict its label. **Bottom:** Large variation on glass boundaries: patches examples. 10

2.1 Visualization of HOG feature space. **(a)** input image. **(b)** HOG cells and local gradient orientations. **(c)** A visualization of HOG features using method in [208]. 17

2.2 Example of RGBD imagery. The point cloud was reconstructed from a video sequence including the color and depth frames. Depth images are color coded so that pixels close to the camera are shown in blue, and far-away pixels are in red. Missing depth values are shown in white. 27

2.3 The frontal views of two visually similar chairs (cropped). For each chair the original image is shown on the left (**(a)** and **(c)**), with the visualized HOG feature map [208] on the right (**(b)** and **(d)**). For the partially occluded chair, the seat and the base are occluded by a table in the front. See text for details. . . 29

2.4 Example of indoor scenes. Note how objects are occluded or truncated by image boundaries. Groups of objects are also arranged together to facilitate human interactions. 31

2.5 Two examples of neighborhood graphs for Markov Random Fields. **Left Panel:** A 4-connected grid of image pixels. **Right Panel:** An 8-connected grid of image pixels. 36

2.6 The factor graph of the pairwise MRF in Equation 2.20. For simplicity, only three nodes are shown. 37

2.7 Example of image labeling results with TextonBoost [185] using unary terms only, and with pairwise terms added. **(a)** input image. **(b)** ground-truth labeling. **(c)** image labeling result with unary terms only. **(d)** image labeling result with pairwise terms added. 40

2.8 Example RGBD image pairs containing glass objects. Note the distinctive but irregular missing patterns in and around glass regions. See text for details. . . . 44

3.1 Illustration of structured Hough voting. **(a)** RGB frame with object bounding box (red) and visible part bounding box (green). **(b)** Object centroid voting from multiple layers. **(c)** Combined object centroid voting results. **(d)** Detector output (red) with visibility pattern prediction (green). **(e)** Object visibility pattern prediction results. **(f)** Final segmentation results. 54

3.2 Top-ranked clusters (presented with the patches closest to the cluster centers) for 3 contextual layers on the Berkeley 3D object dataset. 55

3.3 Illustration of multiple layered object centroid and mask voting. L1 corresponds to the object layer, and L2, L3, L4 correspond to far-away context, close-up context and occluder layers, respectively. For mask voting, brighter regions indicate a higher response, while darker regions indicate a lower response. 56

3.4 Illustration of the impact of patch pair terms on hypothesis scoring. **Upper panel**: A specific example, with **(a)** RGB frame with an example of a patch pair (in blue rectangles). **(b)** Object centroid voting results without patch pair terms. **(c)** Object centroid voting results with patch pair terms added. **(d)** Shape voting results without patch pair terms. **(e)** Shape voting results with patch pair terms added. **Lower panel**: The highest ranked patch pairs on the Berkeley 3D object dataset. The first row shows on-object patches, and the second row shows off-object patches. Each column corresponds to a patch pair. 59

3.5 An illustration of how iterative inference updates the object centroid and supporting mask hypotheses. The first row on the right shows object centroid voting, with the corresponding supporting mask estimations shown in the second row. 62

3.6 Detection examples of our approach. See text for details. 65

3.7 Detection precision-recall curves on the Berkeley 3D Object dataset (left) and the NYU Depth dataset (right). The solid curve corresponds to our approach (**Ours**). The dashed curves correspond to baseline methods: Deformable Parts Model (**DPM**) [46], Max-margin Hough transform (**M^{2}HT**) [129], and Max-margin Hough transform with 2D geometric context (**2D**). See details in text. 66

3.8 Detection precision-recall curves on the Berkeley 3D Object dataset (left) and the NYU Depth dataset (right). The solid curve corresponds to our full model (**Full**). The dashed curves correspond to diagnostic results with various components in our full model turned off, i.e., single layer context (**Single**), patch pair term off (**P Off**), and segmentation off (**S Off**). See details in text. 66

3.9 Precision-recall curves on the Berkeley 3D Object dataset (left) and the NYU Depth dataset (right) for segmentation at 50% recall rate in Figure 5.5. Simultaneously voting for local feature position and whole object hypothesis yields the best segmentation results. 68

3.10 More experimental results of the proposed approach on Berkeley 3D Object Dataset [77] and NYU Depth Dataset [145]. Each row corresponds to a specific instance on a test image. See text for detailed discussion. 69

3.11 Per-class detection precision-recall curves on the Berkeley 3D Object dataset (B3DO). The solid curve corresponds to our approach (**Ours**). The dashed curves correspond to baseline methods: Deformable Parts Model (**DPM**) [46], Max-margin Hough transform (**M^{2}HT**) [129], and Max-margin Hough transform with 2D geometric context (**2D**). See details in text. 71

3.12 Per-class detection precision-recall curves on the NYU Depth dataset (NYU). The solid curve corresponds to our approach (**Ours**). The dashed curves correspond to baseline methods: Deformable Parts Model (**DPM**) [46], Max-margin Hough transform (**M^{2}HT**) [129], and Max-margin Hough transform with 2D geometric context (**2D**). See details in text. 72

3.13 Per-class detection precision-recall curves on the Berkeley 3D Object dataset (B3DO). The solid curve corresponds to our full model (**Full**). The dashed curves correspond to diagnostic results with various components in our full model turned off, i.e., single layer context (**Single**), patch pair term off (**P Off**), and segmentation off (**S Off**). See details in text. 73

3.14 Per-class detection precision-recall curves on the NYU Depth dataset (NYU). The solid curve corresponds to our full model (**Full**). The dashed curves correspond to diagnostic results with various components in our full model turned off, i.e., single layer context (**Single**), patch pair term off (**P Off**), and segmentation off (**S Off**). See details in text. 74

4.1 Illustration of the proposed approach. **(a)** Intensity image with ground truth foreground mask overlaid. **(b)** Edge detector output. **(c)** Triangulation result. **(d)** Boundary classifier output (magnified). **(e)** Superpixel classifier output (magnified). **(f)** Reconstructed depth with joint inference result overlaid. 78

4.2 Examples of the boundary and region graph construction. **(a)** Input intensity image. **(b)** Input depth image (missing readings are shown in white). **(c)** Glass region proposal with proposed glass regions in black. **(d)** Triangulation result. . 80

4.3 An example of boundary proposal including glass, depth and RGB boundary. **(a)** BGTG boundary detector output. **(b)** Glass region proposal results. **(c)** depth boundary detector output before alignment. **(d)** low-threshold RGB edge detector output. See text for details. 81

4.4 The factor graph of the MRF model for our glass detector. Each black square represents a term in Equation 5.2. Each circular node represents a random variable. Shaded nodes are observations. 83

4.5 Illustration of our angle preference for boundary pairwise term. In the top and middle examples, the angles between connected boundary fragments are obtuse and straight respectively. These are commonly found in ground-truth glass boundaries. In the bottom example, however, the angle is acute and is more likely a result from incorrectly identified glass boundary. 86

4.6 Illustration of our superpixel pairwise term. Assume the arrow points towards glass regions in red, and the non-glass regions are in blue. See text for details. 87

4.7 The precision-recall curves based on boundary matching (left panel) and pixelwise region matching (right panel). 91

4.8 Examples of glass detection results on our new RGBD Glass dataset. Note that missing areas are shown in white, and depth readings are recovered by a piece-wise planar model. 93

4.9 Failure examples of glass detection on our RGBD Glass dataset. See text for details. 94

4.10 Examples of boundary and region unary terms (magnified, the viewing window is marked as a red bounding box in the RGB images). The boundary orientation is shown as a red arrow pointing towards glass regions. Local boundary and region classifiers provide complementary information for glass object segmentation. See text for details. 96

4.11 Examples of iterative joint inference. While the initial boundary inference smoothes the unary classifier output, we obtain much cleaner boundary inference results with the joint inference. See text for details. 97

5.1 **Top:** Illustration of feature manifold based glass boundary classification. We use a learned feature manifold to match boundary fragments in a test scene (shown as image patches) to a training set in order to predict their labels. **Bottom:** Large variation on glass boundaries: patches examples. 100

5.2 Example of SLIC [2] superpixels with initial region sizes of 10 px (left) and 30 px (right) respectively. 102

5.3 Example images from the three subsets of our RGBD Glass dataset. See text for details. 105

5.4 Qualitative comparisons between triangulation-based image partitioning method (left two columns, partitions shown in orange) used in Chapter 4 and SLIC [2] superpixels (right two columns, partitions shown in red). Note how SLIC superpixels more closely follow glass boundaries, especially in regions highlighted with blue circles. The SLIC initial region size shown here is 10 px. 106

5.5 The overall precision and recall on RGBD Glass dataset for various methods. **Left**: Performance based on boundary pixel accuracy. **Middle**: Performance based on region pixel accuracy on the whole dataset. **Right**: Performance based on region pixel accuracy in the glass boundary neighborhoods (i.e., regions within 10 px of ground-truth glass boundaries). 112

5.6 Hard examples of glass detection results on the RGBD glass dataset. Column **(a)**: RGB image frame. **(b)**: Unary responses from local glass boundary classifiers in Chapter 4. **(c)**: Joint inference and depth recovery results in Chapter 4. **(d)**: Glass boundary label transfer results. **(e)**: Inference and depth recovery results with the proposed method. Note that missing depth readings are recovered by a piece-wise planar model for glass region and smoothed out using a median filter elsewhere. 113

5.7 Normalized accumulated weight for different features on the RGBD Glass dataset (All) and its subsets (Floor, Laboratory and Office). We add up the absolute values of weights for feature dimensions belong to specific types of features. The resulting bar graph illustrates the relative accumulated "importance" of various types of features used in our method. The feature types are: Hue and Saturation (HS), Blurring (Blur), Blending and Emission (BE), Texture Distortion (Texture), Missing Depth (Missing), Color histogram (Color), HOG on depth data (HOD), Range histogram (Range). See text for details. . . 114

6.1 Examples of different choices of the edge weights w_{ij} for the graph Laplacian. See text for details. 119

6.2 Average test errors (with standard deviations) of AdaBoost, AdaBoost-CG, LPBoost, MDBoost and Laplacian MDBoost on 13 UCI benchmark datasets. . 130
6.3 Performance of Laplacian MDBoost (dash-dot line) and Semi-supervised Laplacian MDBoost (solid line) on UCI datasets **banana** (green), **ringnorm** (blue) and **splice** (red). 131
6.4 Examples of video segmentation with three different semi-supervised algorithms: Semi-supervised Laplacian MDBoost (SemiLap-MDBoost), Learning with local and global consistency combined with MDBoost (LLGC+MDBoost) and SemiBoost. The video data are sequences 0370 and 0950 from the Youtube Celebrity Face Tracking and Recognition Dataset [83]. 132
6.5 Examples of coarse ground-truth superpixel labelings used for our glass region classification experiment. Each red bounding box covers a ground-truth glass object. The center-aligned green bounding boxes cover one fourth the area of the red bounding boxes. A superpixel is labeled as glass if it has 50% or more overlap with the green bounding box, and non-glass if it has 50% or more overlap with the region outside the red bounding box. All superpixels inside the red bounding box but outside the green bounding box are treated as unlabeled data. See text for details. 133

List of Tables

3.1 Per-class average precision on the Berkeley 3D Object dataset and the NYU Depth dataset. Mean average precision values are calculated separately for each dataset. 75

4.1 The overall glass region recall rate, near-boundary glass region recall rate, and the proposed glass area under different dilation disk radii r. Setting $r = 15$ px gives a good tradeoff between recall rates and the proposed area. See text for details. 90

4.2 The glass boundary recall rates from various boundary cues. The first three columns give the recall rates for the three boundary cues in Equation 4.1. The last two columns give the recall rates using the combination of the three cues, before and after triangulation. See text for details. 91

4.3 F-measures at 50% recall for boundary and region accuracy metrics, respectively. 91

4.4 Comparison of average runtime per image (in seconds) between detached and joint inference. The numbers report here are a comparison of MRF inference times (not including feature extraction and local classification). 92

5.1 Glass boundary recall rates for triangulation-based method used in Chapter 4 versus SLIC [2] used in this chapter. e_{max} denotes the pixel error tolerance. See text for details. 106

5.2 Precision (in percentage $\%$) and F-measures at 25%, 50% and 75% recall for glass boundary label transfer. Column Base refers to baseline performance without the feature pool. Columns (1) through (3) refer to (1) image partitioning at multiple scales, (2) sampling features on multiple scales, and (3) sampling features at multiple locations. Column Full refers to our full model with all of the three components. See text for details. 107

5.3 Precision (in percentage $\%$) and F-measures at 25%, 50% and 75% recall for glass boundary label transfer. The first three columns refer to scenarios in which we remove certain depth-aware features. Specifically, they refer to No Color histogram (NC), No HOG on depth data (NH) and No Range histogram (NR), respectively. The fourth column, kNN, refers to the case where we disable the distance metric learning. The final column, Full, refers to our full model. See text for details. 107
5.4 F-measures at 50% recall for boundary and region accuracy metrics. The final row (Bound Region) is based on region pixel accuracy in the glass boundary neighborhoods (i.e., regions within 10 px of ground-truth glass boundaries). . . 108
5.5 Per-image runtime statistics for the method in Chapter 4 and the proposed method. On average the proposed method is about 8 times faster. See text for details. 108
5.6 Precision (in percentage $\%$) at 25%, 50% and 75% recall for glass boundary label transfer on the three subsets of our RGBD Glass dataset. Columns under "Single manifold" refer to results from our proposed approach with a single manifold built on the entire dataset. Columns under "Subset-specific manifold" refer to results obtained with subset-specific manifolds. 110

6.1 Test error and standard deviation (in percentage $\%$) of Laplacian MDBoost (using only labeled data), Semi-supervised Laplacian MDBoost (SemiLap-MDBoost), Learning with Local and Global Consistency combined with MDBoost (LLGC+MDBoost), and SemiBoost on UCI datasets. 124
6.2 Average test and training error (in percentage $\%$) of Semi-supervised Laplacian MDBoost (SemiLap-MDBoost), Learning with Local and Global Consistency combined with MDBoost (LLGC+MDBoost), and SemiBoost on the YouTube Celebrities Face Tracking and Recognition Datasets over 10 tests. 126
6.3 Precision (in percentage $\%$) at 25%, 50% and 75% recall for glass region and boundary classification using fully labeled dataset and partially labeled dataset, respectively. Methods using fully labeled dataset include SVM and Random Forest (RF) from Chapter 4 and MDBoost. Methods using partially labeled dataset include Laplacian MDBoost (without using unlabeled data), Semi-supervised Laplacian MDBoost (SemiLap-MDBoost), Learning with Local and Global Consistency combined with MDBoost (LLGC+MDBoost), and SemiBoost. 128

Chapter 1

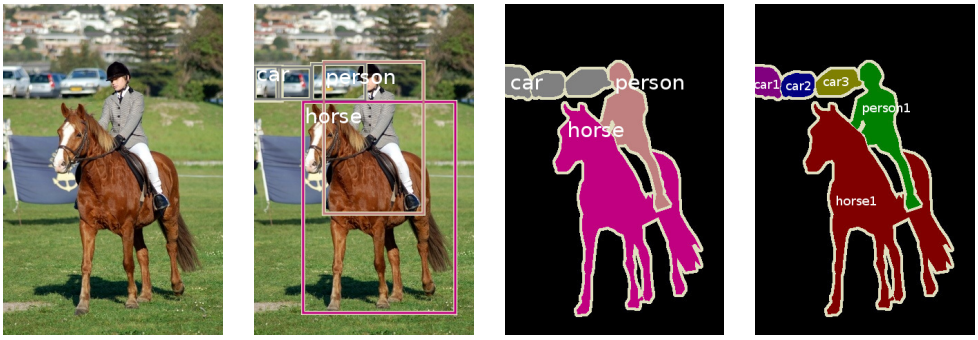
Introduction

One fundamental problem in computer vision and robotics is to make computers capable of understanding three-dimensional scenes from visual information. Such capacity is one of the most impressive features of the human visual system: we all have the ability to quickly, accurately and comprehensively interpret the visual world. The various tasks involved here are referred to as *scene understanding* in computer vision. Broadly speaking, scene understanding aims at resolving the gap between low level image features and high level semantic concepts. One of the core problems here is to localize objects of interest. Take the picture in Figure 1.1 (a) for example, a human can effortlessly (1) recognize the person, the horse, and the cars in the picture, and (2) delineate where these objects are.

These abilities give rise to two popular paradigms for localizing objects in computer vision, i.e., *object detection* and *object segmentation*. Both tasks involve inferring the location of objects belonging to a specific category from an image with different levels of details. Object detection, as depicted in Figure 1.1 (b), parametrizes object location with a rectangular bounding box. The bounding box has an associated category label (e.g., person, horse, or car) and optional pose parameters (e.g., frontal-view, rear-view or side-view for cars). Object segmentation, as depicted in Figure 1.1 (c), is more accurate in the sense that it computes a pixelwise segmentation for the objects. The segmentation may additionally identify individual object instances as shown in Figure 1.1 (d), because multiple object instances belonging to the same category may spatially overlap.

Being able to localize objects is an essential functionality for many real-world applications including autonomous vehicles, content-based image search, event detection in video surveillance, inspection and quality control, etc. In general, localizing objects is arguably one of the most essential steps towards understanding a scene, and it opens the possibility for interacting with identified objects in the environment. In particular, object detection and segmentation link together the semantics and the geometry of a scene, which means it has close ties with other scene understanding problems in computer vision such as image classification and geometry estimation.

Although localizing objects seems to be an effortless task for humans in most cases, it



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

Figure 1.1: Example of object detection and segmentation. **(a)** input image. **(b)** object detection with bounding boxes. **(c)** semantic segmentation. **(d)** object instance segmentation.

remains perhaps one of the most challenging problems in computer vision. The challenging nature of this problem lies in the fact that objects in realistic settings exhibit considerable intraclass appearance variability due to multiple factors such as occlusion, viewpoint variations, and background clutter. In addition, more image data obtained in unstructured environment are constantly being created as low-cost consumer cameras become ubiquitous. There has been an increasing number of pictures taken in cluttered environments from more arbitrary viewpoints, with many potentially interesting objects being only partially visible. These objects would adversely affect both the training and the evaluation of object detection and segmentation algorithms due to their large appearance variations.

In face of the problem, the visual information from a single color image may be insufficient for computers to localize challenging object classes. In fact, information in the real world comes through multiple input modalities, and we may utilize auxiliary input to explain away some of the ambiguities in color images. The benefit of learning to localize objects with multimodal input is at least threefold. Firstly, different modalities may exhibit distinct statistical properties due to the underlying fact that they typically carry different kinds of information. This allows us to discover useful information about the objects and the scene. For example, we may consider the auxiliary information provided by a textual input modality, which may provide concepts such as the psychological perception of an object (e.g., beautiful, interesting, valuable) that is usually not obvious from visual information. As an another example, one particular problem in 2D imaging is that it is challenging to infer 3D configurations of the objects and the scene from a single color image. Without a depth estimation, it requires a lot of effort to manually label the 3D configurations of objects and their parts. Therefore we may consider the multi-modal input provided by depth-capable cameras. For example, the RGBD cameras such as Kinect can collect high quality depth maps and registered color images for indoor environments. The depth maps provide a 2.5D point cloud representation of the scene from which we may infer credible cues for the underlying 3D geometry. Secondly, we may learn joint

representations by fusing data from different modalities to capture real-world concepts and resolve ambiguities. Take depth maps again as an example, they encode useful information about the interactions between an object and its environments, such as the distance between two objects and the occlusion relationships. This allows us, for example, to learn a depth-encoded object appearance model. In addition, we may build a joint feature representation with better class separability by including depth-aware features. Lastly, an important finding from our work is that learning an object model with multimodal input helps even when some modalities are absent during model evaluation. This opens up a new perspective to localizing objects in which we use auxiliary information to help us train a better object model, and apply the model to a test setting where the auxiliary information may be unavailable. Similar ideas have also been suggested in both the psychological [181] and computer vision [191, 32, 22, 186, 234] communities.

In this thesis, we make use of auxiliary information to build object detection and segmentation algorithms, with a focus on modeling the object and its environments with multi-modal input generated from RGBD cameras. We discuss both generic object detection and semi-transparent object (e.g., glass) segmentation. The latter is a specific type of objects which lacks homogeneity of surface appearance and therefore requires purposely designed features and inference algorithms. It should also be noted that the amount of available depth data may vary in practice. For instance, the majority of cameras equipped on handheld devices today do not come with a depth sensor. Therefore, assuming depth maps as a part of an algorithm's input may limit its applicability. This reality prompts us to train an object model with auxiliary depth information and to test without depth maps as discussed. In addition, we address a general problem in object detection and segmentation that it is expensive to obtain precise and complete ground-truth for large datasets. More specifically, we address the four primary challenges as follows:

1) **Partial object observation.** Localizing objects remains challenging for cluttered/crowded scenes, such as indoor environments, where objects are frequently occluded by neighboring objects or the viewing window. The partial objects being observed usually provide limited information on the object position and pose, so many previous object detection approaches are prone to failure as they solely rely on image cues from objects themselves. In this regard, it is important to seek additional information from the environment. Specifically, the availability of depth imagery enables modeling the environment in 3D. Depth maps can provide direct evidence to resolve the ambiguities resulting from projecting the underlying 3D world to a 2D image. In particular, occlusion can be viewed as a special type of contextual relationship in 3D, which would become an intrinsic component of object and scene models.

2) **Partial sensory data.** Localizing semi-transparent objects from a single color image is challenging due to lack of locally discriminative visual features and homogeneity of surface appearance. Therefore, the auxiliary information such as depth maps can be an important cue

for localizing these objects. However, depth maps provided by RGBD cameras are imperfect in the sense that they may contain incorrect or missing readings due to various local refractive properties of the structured light being projected. Therefore, it is important for our algorithms to adapt to the imperfections and counter their negative impact. In addition, we can improve depth reconstruction of the scene if we are able to partially correct the artefacts in depth maps.

3) **Fusing data from different modalities.** It is a non-trivial issue to fuse data from different modalities, e.g., to integrate depth cues into conventional visual inference tasks. The integration can happen at different granularities such as either at the local image patch level or the object and scene level. It could also happen at various stages of the algorithm such as either during feature extraction or model inference. Therefore, the integration of data from multiple modalities is an important aspect of our methods.

4) **Partial ground-truth annotation.** Auxiliary information can also be provided by unlabeled data. By designing a semi-supervised learning algorithm, we are able to work with large datasets with only a fraction of images labeled. In addition, we may also relax the labeling requirements for object segmentation. For example, we can train algorithms with coarse labels (e.g., an object bounding box) without the need to specify exact object boundaries. In fact, a generic semi-supervised learning technique can be applied to a range of real-world applications that involve a classification problem.

In the following section, we discuss four specific research problems raised in this thesis, addressing the primary challenges above. In Sections 1.2 to 1.5, we outline the main ideas of our work in response to the research problems. Section 1.6 summarizes the content of each chapter, and Section 1.7 lists the major contributions from this thesis.

1.1 Our research problems

To reliably detect or segment objects in the presence of background clutter and heavy occlusion, and in order to address different levels of auxiliary information availability, we provide solutions to the following four central research problems in this thesis:

1) **Depth-aware context modeling.** In each image, the structural prior information of its scene essentially defines a *context*[1]. For a single intensity image, an important class of context is the two-dimensional projection of 3D scenes. Co-registered RGBD imagery allows for modeling contextual elements in the underlying 3D world. Context reasoning can be carried out at multiple levels. We can discard relative geometric relationships between objects and context and describe context at a geometry-free level, e.g., the presence of a table in an image raises our expectation of seeing chairs. By encoding the relative geometric distributions between objects and context, we are able to provide more specific cues, e.g., the vicinity of a table is more

[1] There are many sources of contextual information (e.g., spatial context, semantic context and temporal context). In this thesis we focus on the spatial context. See [40] for details on various sources of contextual information.

likely to contain chairs. Another option is to discard the notion of objects and look at local image patches and the interactions among them. For instance, we can reason about interactions between local image regions and boundary. Therefore, it is a non-trivial issue to model the spatial context in RGBD images. The discussion above presents our first research problem: *at what level, and how can we model the spatial context, in order to integrate the most relevant information into an object detection or segmentation framework?*

2) **Inference with imperfect depth data.** Low-cost RGBD cameras (with Kinect as a prominent example) can provide depth maps as the auxiliary information. There is great potential benefit from having a high quality dense depth map registered to the color image, as geometric information plays a vital role in scene understanding. In fact, depth inference from a single color image is a well-studied problem in computer vision (e.g., [177, 114]). However, depth sensors are far from being a standard addition to RGB cameras. Furthermore, the Kinect depth maps mainly work in indoor environments within a certain distance range, and tend to contain artefacts such as missing or incorrect readings due to sensor limitations. Therefore, our second research problem is: *how can we work with limited availability of depth maps? Further, when depth maps are available, how can we deal with the artefacts to counter their negative impact, or even use them as a useful image cue?*

3) **Depth-aware features and label transfer.** Object detection and segmentation with static color images have been extensively studied. Yet, the popularity of consumer depth-capable sensors put forward the question of how to sensibly make use of this additional depth information. As discussed in our first research problem, depth cues can help resolve the ambiguities in the underlying 3D world in terms of the scene geometry. Apart from that, depth maps can also facilitate the design of novel features and feature manifolds for figure/ground classification and label transfer. Therefore, our third research problem is: *can we design effective depth-aware features for object categories that are difficult to localize with color cues, such as semi-transparent objects? How do we integrate depth cues into conventional visual inference tasks?*

4) **Learning with unlabeled data.** In computer vision, many different types of sensory data are available, with different levels of ground-truth annotation. Another type of the auxiliary information we focus in this thesis is the unlabeled data. For large datasets, detailed object ground-truth annotation (e.g., pixelwise segmentation masks) can be expensive to obtain. Therefore it would be appealing for object detection and segmentation algorithms to either assume only a fraction of images as labeled, or require only coarse object labels. This brings out our last research question: *how can we make use of unlabeled data and relax the labeling requirements for training data?*

Our investigations reported in this thesis are centered around the four research problems above. We will discuss these questions and provide our solutions by building an object detection system and an object segmentation system. The detection system jointly detects objects

and estimates occlusion, while the segmentation system focuses on localizing semi-transparent objects. Sections 1.2 to 1.5 present the main ideas of our work corresponding to the four research questions, followed by thesis outline and our primary contributions.

1.2 Object detection with depth-encoded context

Although many context-aware object detection methods have been proposed [219, 201, 127, 16], most existing contextual models focus on 2D spatial relationships between objects on the image plane and fewer works have extended the modeling to 3D scenarios [8, 193]. Modeling context from a 3D perspective has several advantages over its 2D counterpart conceptually. First, spatial relationships have smaller variations and are easier to interpret semantically; in addition, more spatial relationships in physical world can be captured, instead of being limited to relative positions on image plane. In particular, joint modeling of an object class and its 3D context may provide effective constraints on the object's scope on image plane and lead to a coarse-level object segmentation. See Figure 1.2 for an example.

However, the appearance variability of the context around an object could be large. It is therefore challenging to use context as a cue, because we would need to model the variability in the appearance of all of the objects around an object of interest. One key challenge is to generate proper training data to capture all the appearance variations. In addition, moving from 2D to 3D (i.e., depth-encoded) context adds a dimension to be sampled, thus seems to make the problem more difficult.

In response to the difficulty outlined above, the practicality of our method is based on both the problem setting and the model design. Firstly, we consider indoor scenes where object-context spatial regularities such as supporting and attachment are more restrictive (e.g., many objects are either supported by floor or by tables), and scene regularities such as orthogonality and vanishing points are more common due to features of man-made structures. In addition, our model uses depth maps to guide us in building a cleaner context representation, such as separating nearby co-occuring objects (e.g., tables and chairs, keyboards and mice) against wall and floor structures further away. During inference, our depth-encoded codebook design enables an image region to contribute to each object hypothesis in a different manner based on its depth layer. Intuitively, context region produces less concentrated vote for object locations as the increased distance from objects leads to higher uncertainty.

More specifically, we propose a structured Hough voting method that incorporates depth-dependent context into a codebook based object detection model. We design a multi-layer representation of context by sorting image regions into different layers depending on their distance to the object. Each layer provides support for the object hypotheses with information from different aspects of the scene. Intuitively, image cues from the object provide the most informative estimation of object location. Further, the surrounding environment can provide

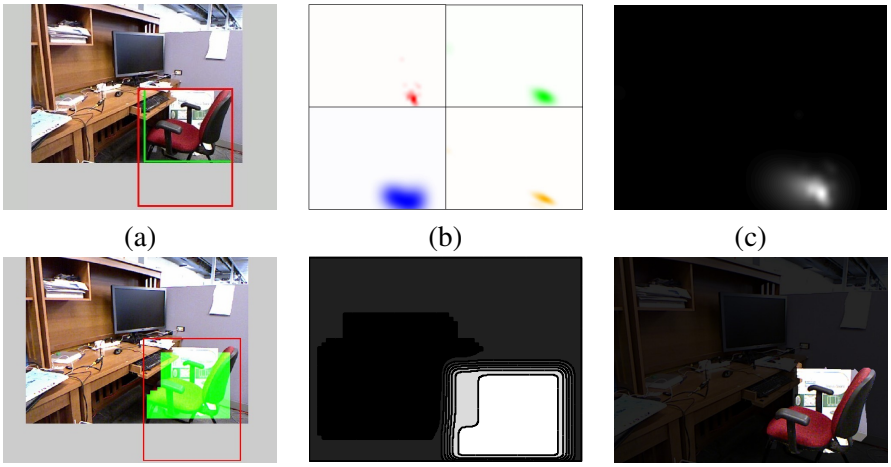


Figure 1.2: Illustration of the proposed object detector. **(a)** RGB frame with object bounding box (red) and visible part bounding box (green). **(b)** Object centroid voting from multiple layers. **(c)** Combined object centroid voting results. **(d)** Detector output (red) with visibility pattern prediction (green). **(e)** Object visibility pattern prediction results. **(f)** Final segmentation results.

less concentrated but useful information on object location, particularly when the contribution from the object itself is weaker due to occlusion.

In addition to the depth-encoded context codebook, our model generalizes the traditional Hough voting detection methods in two other ways. Firstly, we define a new object hypothesis space in which both the object's center and its visibility mask will be predicted. Each image patch will generate a weighted vote to a joint score of the object center and its support mask in the image. Secondly, we view occlusion as special contextual information, which could provide cues for object detection and help with reasoning about visibility of object parts. The overall output of our approach is a simultaneous object detection and coarse segmentation.

Finally, the varying availability of auxiliary information is a specific issue we wanted to address in this work. Although RGBD cameras are gaining popularity rapidly, the majority of image data are color images. Therefore, we would like our object detector to train with RGBD data but to test without depth maps. The training process aims to learn a context-aware object detection model which encodes depth cues and a coarse level of 3D relationships. The learned depth-encoded object and context model is then applied to 2D images. More specifically, we use depth to sort image features into different layers, and learn codebook entries so that they minimize appearance and 3D geometric distribution variations.

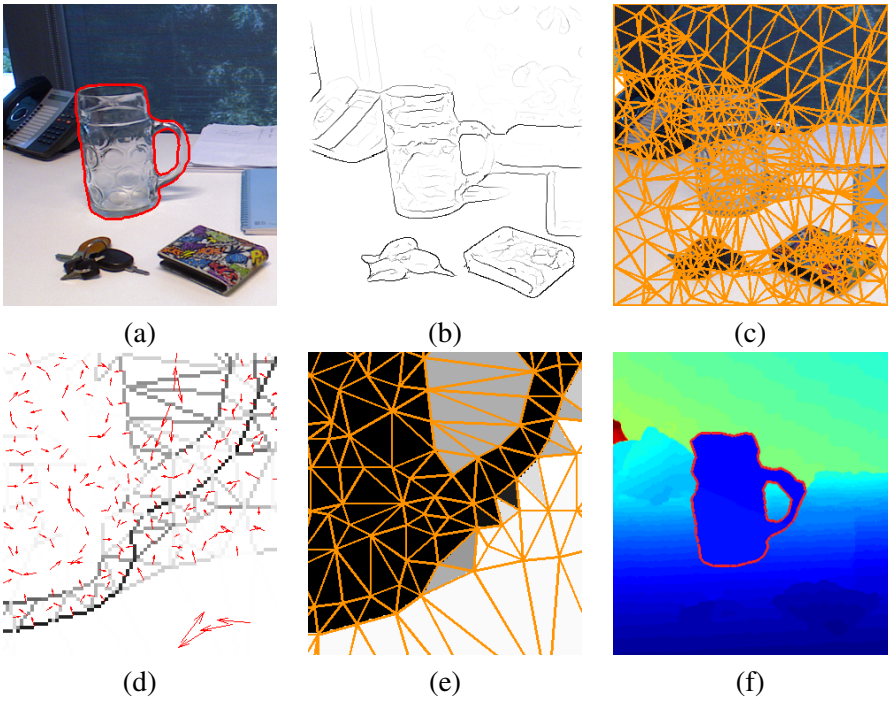


Figure 1.3: Illustration of the proposed glass object segmentation system. **(a)** Intensity image with ground truth foreground mask overlaid. **(b)** Edge detector output. **(c)** Triangulation result. **(d)** Boundary classifier output (magnified). **(e)** Superpixel classifier output (magnified). **(f)** Reconstructed depth with joint inference result overlaid.

1.3 Glass segmentation by joint inference of boundary and region

We aim to localize semi-transparent surfaces by exploring multimodal sensors and incorporating depth information. In particular, we seek to exploit RGBD cameras to fuse the intensity and depth information from a single view point for indoor environments. While recent work with RGBD cameras is mainly for generic object detection [98, 99, 49], here our goal is joint detection, segmentation and depth inference, which can facilitate many interactive tasks such as robotic manipulation. There has been some work exploiting range devices to detect or reconstruct semi-transparent objects [209, 84]. Unlike those methods, we rely on a single view RGBD image and combine both intensity and depth cues.

Unlike in Section 1.2 where we take an object-centric view and build an object model that jointly considers the possible object shapes and poses, in this and the following section we focus on the local appearance and depth properties of glass boundary and region. One of the key reasons of taking this local perspective is that glass objects do not have just a few canonical shapes in comparison to some object categories such as cups, bottles, and bowls. See Figure 2.8 for some examples. Arguably, glass objects include subsets of the above object categories: glass cups, glass bottles and glass bowls, etc. While the problem of exploring

the shape and pose constraints for glass objects is interesting, here we focus on capturing the properties of glass objects based on their being made of glass, and the interaction between glass and non-glass regions. Additionally, modeling the specificity of glass material has been proven effective for localizing glass objects in prior literature. For example, some early work focused on detecting special properties of the glass surfaces and their interaction with the opaque environment in images [151, 3, 144] while later ones model the relative features on two sides of a local glass boundary fragment [135, 134] based on a combination of appearance cues. See Section 2.2.4 for a more detailed discussion on the literature.

Taking the local perspective mentioned above, the key idea of our work is to incorporate the spatial context by constructing a Markov Random Field (MRF) [15] on triangularized contour fragments and the corresponding superpixels. Based on spatial neighborhood, we incorporate constraints between local boundary pairs, superpixel pairs, and boundary-superpixel cliques. More specifically, for each image contour fragment, we estimate if it is likely part of the glass/non-glass boundary, and an orientation for the glass region. For superpixels, we estimate their likelihood being part of the glass region. We add different potentials into our energy function to encourage valid configurations, and penalize incompatible ones. For instance, the orientation for glass regions of two connected glass contour fragments must be the same. For a local clique consisting of a glass contour fragment and two neighboring superpixels, the glass/non-glass labels of both regions must be consistent with the boundary orientation. In addition, a joint inference scheme is designed to predict the glass boundary and region simultaneously. Our work is the first that jointly optimizes boundary and region properties and constraints for glass object segmentation.

Furthermore, we exploit the refraction and attenuation that will be experienced by an active structured light signal passing through glass objects. This physical process is difficult to model, but it provides a distinctive missing-vs-nonmissing pattern in the depth map. We integrate boundary cues from color with region cues from depth to build a glass boundary and region detector. After we obtained a glass region segmentation with MRF inference described above, we fill in the missing depth values and reconstruct the scene in 3D.

1.4 Depth-aware features and label transfer

The third research problem we discussed in Section 1.1 is the design of depth-aware features and the integration of depth cues into visual inference tasks. In response, we design a number of novel depth-aware features for glass boundary estimation. Most importantly is the distinctive missing-vs-nonmissing pattern which we found to be highly effective for coarsely localizing glass objects, so we compute the ratio of pixels with missing depth readings in a local image region as a depth feature. Other features include range (depth) histograms and histogram of oriented gradient (HOG) features computed on depth maps. We also explore building a flex-

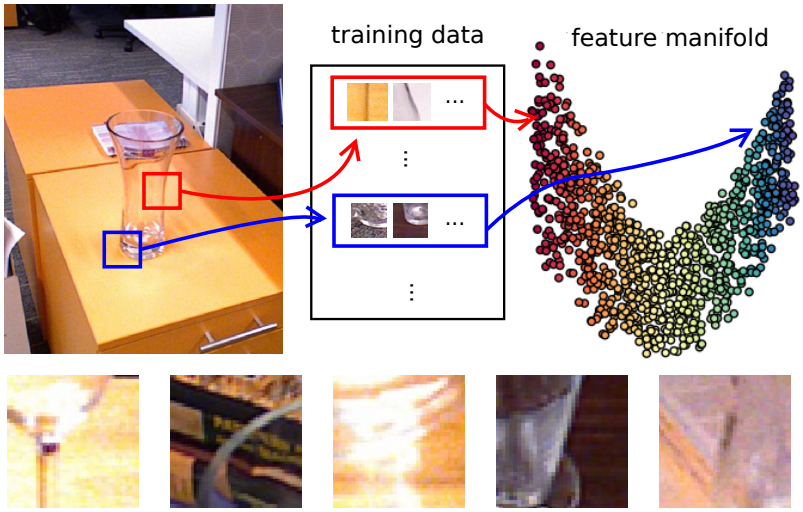


Figure 1.4: **Top:** Illustration of feature manifold based glass boundary classification. We use a learned feature manifold to match every boundary fragment in a test scene (shown as image patches) to training set in order to predict its label. **Bottom:** Large variation on glass boundaries: patches examples.

ible feature pool which contains both depth and color features. We augment the image cues by sampling features on multiple scales and at multiple locations.

One key reason for the challenging nature of glass object segmentation is the large appearance variations at glass boundaries, as shown in a few examples in Figure 1.4. Training a generic classifier for glass boundaries tends to produce unreliable predictions. Even with RGBD cameras, the missing patterns on depth maps can be noisy, or distorted due to local refractive properties. To address this feature variation issue, we propose an image adaptive approach to predicting glass boundaries. The main idea is to generate boundary proposals based on a nonparametric feature model. Our model is represented by a joint depth and appearance feature manifold, on which each point is the glass boundary feature of an image patch pair. The boundary label of any pair of neighboring patches is predicted by a weighted voting of its nearest neighbors on the feature manifold. The distance metric on the manifold is learned in a supervised manner.

We then integrate the locally adapted glass boundary predictor into a superpixel-based pairwise MRF for glass object segmentation. The MRF labels every superpixel as glass versus non-glass, in which our boundary prediction is used to modulate the smoothing terms in random fields. Our work is the first to explore nonparametric label transfer within the context of glass object segmentation, and exploit a joint depth-appearance manifold for transductive learning.

1.5 Learning from sparsely labeled data

The ground-truth annotation availability issue led us to the development of a boosting-based semi-supervised learning algorithm. Our method adopts a novel data-dependent margin distribution learning criterion, which utilizes the intrinsic geometric structure of datasets. One key aspect of our method is that it can seamlessly incorporate unlabeled data by including a graph Laplacian regularizer.

Boosting algorithms have achieved great popularity in a spectrum of computer vision problems due to their good generalization, robust performance, and intrinsic feature selection mechanism. One key observation related to our work is that the appealing properties of boosting are closely related to the *margin distribution* (MD) instead of solely the minimum margin [168]. Notably, Shen and Li [182] proposed a totally corrective boosting algorithm, termed MDBoost, to maximize the average margin while minimizing margin variance. The new boosting method achieves competitive performance and faster convergence (i.e., fewer weak learners) on several classification tasks.

Inspired by manifold learning, we propose to improve MDBoost by incorporating a local representation of margin variance, in which only neighboring points on the data manifold contribute to the variance computation. Intuitively, the data-dependent margin variance may give a better description of the margin distribution. Due to its resemblance to the Laplacian Eigenmap [10], we refer to this new boosting approach as *Laplacian MDBoost*. Importantly, our learning criterion can be naturally generalized to a semi-supervised learning scenario. Given both labeled and unlabeled data, we augment the supervised learning criterion with a graph Laplacian-based regularization term, which encourages the classifier outputs on unlabeled data to satisfy the data manifold constraint. This combined learning criterion provides a coherent framework and admits a simple convex quadratic dual formulation such as MDBoost. We employ a column-generation (CG) based optimization procedure to incrementally add informative weak learners, yielding a boosting-like algorithm. The efficacy of the proposed algorithm has been demonstrated in our glass object segmentation experiment, in addition to another video object segmentation task.

1.6 Thesis outline

The next chapter discusses some prior literature that is relevant to the problems addressed in this thesis. It first reviews object detection algorithms, and categorizes them according to two most popular paradigms: the sliding window detector and the Hough transform detector, and their variants and extensions. Next, we discuss work on object detection with RGBD data and context reasoning. The chapter then moves on to object segmentation algorithms, focusing on foreground object segmentation and context modeling with MRFs. After that, we discuss work

on glass object segmentation. Finally, we review work related to the proposed semi-supervised boosting algorithm.

In Chapter 3, we describe a structured Hough voting method for detecting objects with heavy occlusion in indoor environments. First, we extend the Hough hypothesis space to include both object localization, and the object's visibility pattern. We design a new score function that accumulates votes for object detection and occlusion prediction. In addition, we explore the correlation between objects and their environment, building a depth-encoded object-context model based on RGBD data. Particularly, we design a layered context representation and allow image patches from both objects and backgrounds to vote for the object hypotheses. We demonstrate that using a data-driven 2.1D representation we can learn visual codebooks with better quality, and obtain more interpretable detection results in terms of the spatial relationship between objects and viewer. We test our algorithm on two challenging RGBD datasets with significant occlusion and intraclass variation, and demonstrate the superior performance of our method.

Chapter 4 addresses the problem of localizing glass objects with a multimodal RGBD camera. Our method integrates the intensity and depth information from a single view point, and builds an MRF that predicts glass boundary and region jointly. Based on the segmentation, we also reconstruct the depth of the scene and fill in the missing depth values. The efficacy of our algorithm is validated on a new RGBD glass dataset of 43 distinct glass objects.

Chapter 5 also addresses the glass object segmentation problem with an RGBD camera. Our approach uses a nonparametric, data-driven label transfer scheme for local glass boundary estimation. A weighted voting scheme based on a joint feature manifold is adopted to integrate depth and appearance cues, and we learn a distance metric on the depth-encoded feature manifold. Local boundary evidence is then integrated into an MRF framework for spatially coherent glass object detection and segmentation. The efficacy of our approach is verified on our RGBD dataset where we obtained a clear improvement over the state-of-the-art both in terms of accuracy and speed.

In Chapter 6, we propose a novel data-dependent margin distribution learning criterion for boosting, termed Laplacian MDBoost, which utilizes the intrinsic geometric structure of datasets. One key aspect of our method is that it can seamlessly incorporate unlabeled data by including a graph Laplacian regularizer. We derive a dual formulation of the learning problem that can be efficiently solved by column generation. Experiments on various datasets validate the effectiveness of the new graph Laplacian based learning criterion in both supervised and unsupervised learning settings. We also show that our algorithm outperforms the state-of-the-art semi-supervised learning algorithms on a variety of inductive inference tasks, including glass region classification and real world video segmentation.

Chapter 7 summarizes the main results from this thesis and discusses future research directions.

1.7 Major contributions

In this section, we summarize the main differences between our methods and other object detection and segmentation methods, and list the most important results reported in this thesis.

- We propose a structured Hough voting model for indoor object detection and occlusion prediction. We extend the original Hough voting based detection model by introducing a joint Hough space of object location and visibility pattern. The structured Hough model can naturally incorporate both the object and its spatial context, which is especially important for cluttered indoor scenes.

- We utilize depth information at the training stage of the structured Hough voting model to build a multilayer object-context model so that a better visual codebook is learned and more detailed object-context relationships can be captured. We use depth information only in the model training stage to learn an appearance model for the surrounding environment of an object with higher quality, which transfers the depth knowledge for a test scenario which uses color images only.

- We propose a novel joint inference approach to glass object segmentation with RGBD cameras. By setting up an MRF which jointly encodes boundary fragment and superpixel properties and constraints, we propose a global optimization procedure for glass detection, segmentation and scene reconstruction.

- We propose a glass boundary detection approach by label transfer on joint depth and appearance manifolds. We design novel features for glass object segmentation and a flexible feature pool for improving performance. In addition, our work is the first to explore nonparametric label transfer within the context of glass object segmentation, and exploit a joint depth-appearance manifold for transductive learning.

- We propose a semi-supervised boosting algorithm based on the margin distribution boosting. We use the graph Laplacian as an effective means of manifold regularization on both labeled and unlabeled data. The algorithm is totally-corrective and a column generation based optimization technique is used to facilitate minimizing the objective function. The efficacy of this algorithm has been demonstrated on two object segmentation tasks.

Chapter 2

Literature Review

Object detection and object segmentation are two popular paradigms for object recognition, which is a key aspect of resolving the gap between low level image features and high level semantic concepts in a scene. There is an abundance of prior literature on both problems. In addition, both problems are based on a classification model for the object/non-object membership. In this chapter, we review object detection and segmentation approaches in the literature, with a focus on those that overlap with our research problems discussed in Section 1.1: 1) occlusion and context reasoning, 2) object detection with RGBD data and 3) semi-transparent object detection and segmentation. We also review work on semi-supervised learning that aims at utilizing unlabeled data for classification.

The rest of this chapter is organized as follows. We first discuss popular object detection algorithms in Section 2.1. In particular, we look at methods with occlusion and context reasoning. Section 2.2 reviews foreground object segmentation algorithms, with a focus on those based on Markov Random Fields (MRFs), a unifying framework for object segmentation and image labeling. In addition, we discuss methods designed to localize semi-transparent objects, a class of objects that are particularly challenging to detect due to their special refractive properties. We then discuss learning a classification model for these systems with partially labeled data in Section 2.3, followed by a summary in Section 2.4.

2.1 Object detection in computer vision

The object detection task is to infer the location of objects belonging to a specific category in an image. In most cases, we are interested in identifying objects from a *basic and entry level category* [80, 150], which is at a level of abstraction in a taxonomy that carries the most information, possesses the highest category cue validity, and are, thus, the most differentiated from one another [172]. For the horse in Figure 1.1, for example, we will use the entry level category *horse* instead of *animal* or *Equus ferus caballus*. Recognizing objects requires discriminating them from other objects, while also generalizing over appearance variations within that category. The challenge of this task lies in the delicate contention between specificity and

generality. For example, detecting *horse* requires us to differentiate them from *cow*, *sheep* and *person*, while being able to detect different subspecies and from various viewpoints.

More specifically, suppose we have an image I and an object category of interest o. An object is parametrized by a hypothesis $\mathbf{x} \in \mathcal{X}$ where $\mathcal{X}$ is the object pose space in I. A basic and common parametrization of $\mathbf{x}$ is a bounding box $\mathbf{x} = (a_x, a_y, a_s, a_r)$, where a_x and a_y are the image coordinates of the object center, a_s is a scale, and a_r is an aspect ratio. Most object detection systems define a scoring function $S(o, \mathbf{x})$ for each valid location $\mathbf{x}$ on the image plane, and all hypotheses with a score $S(o, \mathbf{x})$ above a certain threshold are claimed as detected objects.

Evaluation of bounding box predictions can be performed by the Jaccard index defined as

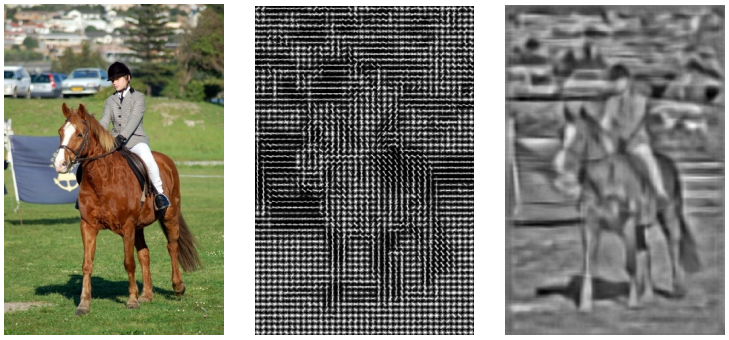
$$J = \frac{area(\mathbf{x}_1 \cap \mathbf{x}_2)}{area(\mathbf{x}_1 \cup \mathbf{x}_2)} \tag{2.1}$$

where usually a predicted bounding box that has more than 50% Intersection-over-Union (IoU) overlap with the ground-truth is considered correct [43].

In the next two sections, we discuss two popular object detection strategies, i.e., sliding window detectors and Hough transform-based detectors. The former takes a top-down, object centric view by examining all possible object locations, while the latter takes a bottom-up, feature centric view by accumulating votes for object locations. However, we note the two strategies are not fundamentally different from each other. The actual difference is more of an algorithmic nature, i.e., how the score is evaluated for all possible object hypotheses [107]. In Section 2.1.3, we continue our discussion by looking at the impact of RGBD camera on object detection with the new challenges it presents. This is followed by discussions on difficult cases for object detection, specifically occlusion reasoning in Section 2.1.4 and context handling in Section 2.1.5.

2.1.1 Sliding window detectors

One of the most popular object detection paradigms is the sliding-window classifier, e.g., [207, 38]. The underlying assumption is the label (e.g., object/non-object) for each bounding box can be obtained independently from labels of other bounding boxes, so the algorithm exhaustively scans through the image with candidate object windows at various locations and scales. This strategy is straightforward, as it evaluates one object candidate at a time and ignores the spatial context that can be more intricate to consider. More importantly, the scan can be naturally viewed as a matching process, so we can define a score that quantifies the match between an object candidate and the object template (e.g., the parameters of the classifier). In its basic form, the scoring function for object detection in this scenario can be written as a linear model:



(a) (b) (c)

Figure 2.1: Visualization of HOG feature space. **(a)** input image. **(b)** HOG cells and local gradient orientations. **(c)** A visualization of HOG features using method in [208].

$$S(o, \mathbf{x}) = \beta^T \cdot \Phi(\mathbf{x}, I) \tag{2.2}$$

where $\Phi(\mathbf{x}, I)$ is a feature function and β is the associated weight vector. The feature function takes the image I and a bounding box $\mathbf{x}$ as input and returns a feature vector that encodes the appearance of the bounding box. To counter the intraclass appearance variation within a specific category, feature functions usually provide some level of invariance to color, shape, deformation, etc. Although in principle it is possible to use raw pixel values from the bounding box $\mathbf{x}$ as the feature vector, more effective features that can achieve a higher level of invariance are commonly used, including Haar-like features [207], SIFT descriptors [121], Local Binary Patterns (LBP) [146], and Histograms of Oriented Gradients (HOG) features [38]. See Figure 2.1 for an example of visualizations of the HOG feature space. The weight vector β is usually obtained by discriminative training algorithms such as Support Vector Machines (SVM) [15] or Boosting [207].

Since the advent of deep Convolutional Neural Networks (CNNs) such as AlexNet [94], ZFNet [231] and VGGNet [187], they have been successfully applied to object detection and is now a key component of many state-of-the-art object detection algorithms. Girshick et al. [58] proposed to use a computationally expensive CNN to compute features for a relatively small number of image region proposals. The region proposal step has later been able to share computation with feature extraction [57] and further fully integrated into the CNN as a region proposal network [167]. The most important contribution of a CNN is its ability to extract low dimensional (e.g., 4096-D) but high quality image features, due to the deep structure of the network. Very recently, He et al. [68] proposed residual networks that are significantly deeper than previously used networks. OverFeat [180] is another CNN-based object detection method that uses an efficient sliding window scheme to share computations and apply a CNN densely

over an image. YOLO [165], on the other hand, casts object detection as a regression problem to provide a very efficient algorithm. The method regresses from an image to a fixed-sized tensor and lacks the ability to detect densely populated objects.

It should be noted that the image region outside the bounding box, or the *spatial context*, can also provide useful information for localizing objects. For example, the image from Figure 2.1 (a) shows a horseback riding activity so the person and the horse in that image can be predictive of the presence of each other. Also, these activities usually take place in grassland areas, therefore a grassland landscape can also support the detection of a horse. However, such information is discarded in many sliding-window classifiers, if our aim is to capture the appearance commonalities of an object category from within the bounding box $\mathbf{x}$. We discuss related work in the literature that specifically addresses this issue in Sections 2.1.4 and 2.1.5.

The sliding window detector in Equation 2.2 is also referred to as based on *template matching*, as the feature function defines a rigid transformation of the image data. The template is sometimes referred to as an *appearance model*, which is the characteristic and discriminative appearance (e.g., shape) we learned for the specific object category. The model makes strong assumptions about the rigidity of an object by only allowing small local deformations and appearance changes. However, many object categories present deformable shape variations, and even with rigid objects their appearance can greatly change locally. To address the appearance variation issue, there have been various extensions to the basic detector. For example, non-linear template matching [171, 128] and a classifier cascade [207] can be used to encode high order interactions among object features. Particularly, part-based models naturally allow for appearance variations caused by shape deformations, as we will show next.

Part-based models. The appearance of most object categories exhibit some amount of deformation, and the strong rigidity assumption in the basic model does not allow certain parts of an object moving too far from its anchored location in the template. Therefore, modeling deformable objects may require a large number of templates and consequently more training images. In fact, deformable objects can be represented in terms of other objects (e.g., object parts) through compositional rules [47]. In part-based models, an object is represented by a fixed number of rigid templates (primitive parts), and the deformations are modeled by the spatial relationships among them.

In particular, the Deformable Parts Model (DPM) [46] defines a star-shaped object model that combines a root node and a number of vertically symmetric parts. The root node is similar to the rigid template in Equation 2.2, and each part captures detailed local appearance by a rigid template at higher feature resolution. Deformation is modeled for each part with an anchor position and a deformation term that penalizes parts moving away from their anchor positions. Let (dx_i, dy_i) be a deformation vector for the i-th part from its anchor location, and define

$$\Phi_d(d_x, d_y) = (dx, dy, dx^2, dy^2) \tag{2.3}$$

as the deformation feature function, the scoring function for their model can be written as

$$S(o, p_0, \cdots p_n) = \sum_{i=0}^{n} \beta_i^T \cdot \Phi(p_i, I) - \sum_{i=1}^{n} d_i \cdot \Phi_d(dx_i, dy_i) + b \tag{2.4}$$

where the object hypothesis $\mathbf{x}$ in Equation 2.2 is replaced with the positions of the root template p_0 and n part templates $p_1, \cdots, p_n$. The first term on the right hand side of Equation 2.4 is the score for each rigid template, and the second term is the penalty for part deformations. d_i is the deformation cost for the i-th part and b is a real-valued bias term. For example, if we set $d_i = (0, 0, 1, 1)$ the deformation cost for the i-th part will be the squared distance between its actual position and the anchor position relative to the root template.

The parts in DPM are placed initially using heuristics and updated by discriminatively training an appearance model. Bourdev and Malik [22] propose the notion of "Poselets" where appearance and 3D configurations are jointly considered for selecting informative object parts. On the other hand, although we can use a mixture of templates to handle extreme viewpoint variations, it is usually difficult to use just a few templates to capture the large structural variations for highly deformable object categories (e.g., cats and dogs). Recent work by Endres et al. [42] proposes to learn a collection of part detectors and use them to classify bottom-up image regions. These part activations are then evaluated by a boosting classifier for bounding box predictions. Wang et al. introduce a regionlet-based object representation which also accommodates deformations by the regionlet group selection [214].

Many part-based models for object detection (e.g., [46]) are an instantiation of a more general compositional model. For example, Girshick et al. [59] propose a grammar model that allows for multiple part subtypes, optional parts, and explicit reasoning of occlusions. Other work towards a more general compositional model includes part sharing [152] and building hierarchical tree structures [237].

In summary, it is generally acknowledged that a move to compositional models is needed for the detection of object categories that naturally present pose and shape variations. However, it is a broad and challenging problem to move to richer models while maintaining a high level of performance (e.g., [46, 59]). Richer models typically involve more computationally expensive inference problems, and it is arguably impossible to find an "optimal" part-based structure for a certain object category due to limited data and annotation availability at hand. In this sense, this problem also relates to the automatic part discovery problem (see [154] for a recent work and review). Once again, all the abovementioned object detection methods focus

on exploiting the image cues from within the object bounding box and consider the spatial context as a less relevant issue. We continue our discussion in the next section by looking at another typical object detection strategy that takes a part-centric view. This alternative view aligns well with the needs to reuse and share object parts and, more importantly, it allows us to naturally incorporate contextual support into object detection under a unified and coherent framework.

2.1.2 Hough transform detectors

It is a non-trivial issue to search over the pose space in an image with a sliding window detector. To avoid the time-consuming exhaustive search, methods based on selective search [101] and object proposals [5, 204, 41, 242] have been proposed to reduce the number of object hypotheses to be examined. The Generalized Hough transform [7] provides a different way of dealing with the complexity in searching over the object pose space. A visual codebook, instead of a feature template, is learned to capture the appearance of object parts. This is usually done by clustering of object (and background) features of image patches. During testing, each image patch casts probabilistic votes for the object center. This is done by matching the patch against the codebook to obtain similarity scores between the patch and each of the codebook entries. These similarity scores will then be used to re-weigh the probabilistic votes stored in the codebook.

The Hough transform defines object in terms of parts. It thus naturally allows for part location variations as in part-based models discussed in the previous section. More importantly, the votes for object center can be stored nonparametrically which makes the detector capable of encoding sophisticated part location distributions. This is in contrast to the previously discussed part-based models such as DPM that define part deformation costs in terms of anchor locations only. Consider a detector for side view of cars for example, the front and rear wheels can be encoded as a single part in Hough transform-based detectors. This leads to a cleaner representation of parts and, in particular, facilitates the reuse and sharing of parts for scalable object detection [164]. In addition, the voting process in Hough transform can be easily extended to encode pose parameters beyond the object center. Indeed, the presence and appearance of certain object parts can be predictive of object pose such as orientation and scale. In Chapter 3, we will show how to include object masks into voting for joint object detection and occlusion estimation.

Mathematically, denote each image patch $I_{\mathbf{y}}$ by its location $\mathbf{y} = (b_x, b_y)$ and feature descriptor $\mathbf{f}_{\mathbf{y}}$, the basic Hough transform detector assumes that the overall detection score $S(o, \mathbf{x})$ is obtained by factorizing $p(o, \mathbf{x}|I)$ into individual probabilities $p(o, \mathbf{x}, \mathbf{y}, \mathbf{f}_{\mathbf{y}})$ over all observations:

$$
\begin{aligned}
S(o,\mathbf{x}) &= \prod_{\mathbf{y}} p(o,\mathbf{x},\mathbf{y},\mathbf{f}_{\mathbf{y}}) \\
&\approx \sum_{\mathbf{y}} p(o,\mathbf{x},\mathbf{y},\mathbf{f}_{\mathbf{y}}) \\
&= \sum_{\mathbf{y}} p(o,\mathbf{x}|\mathbf{y},\mathbf{f}_{\mathbf{y}})p(\mathbf{y},\mathbf{f}_{\mathbf{y}})
\end{aligned} \tag{2.5}
$$

where $p(\mathbf{y},\mathbf{f}_{\mathbf{y}})$ is the prior on features and locations. We note the "summation hack" used here (i.e., replacing the product by a summation in Equation 2.5) has a natural probabilistic interpretation as an outlier model [141]. Assuming an appearance-based codebook is learned from the image patches in object class o, denoted by $\mathcal{C} = \{C_i\}_{i=1}^{K}$, and a uniform prior $p(\mathbf{y},\mathbf{f}_{\mathbf{y}})$, we can marginalize Equation 2.5 over the *codebook entries* or *codewords* C_i:

$$
\begin{aligned}
S(o,\mathbf{x}) &\propto \sum_{\mathbf{y}} p(o,\mathbf{x}|\mathbf{y},\mathbf{f}_{\mathbf{y}}) \\
&= \sum_{i=1}^{K}\sum_{\mathbf{y}} p(o,\mathbf{x}|C_i,\mathbf{y},\mathbf{f}_{\mathbf{y}})p(C_i|\mathbf{y},\mathbf{f}_{\mathbf{y}})
\end{aligned} \tag{2.6}
$$

We can further simplify Equation 2.6 by the fact that codebook entries are matched by appearance only, i.e., $p(C_i|\mathbf{y},\mathbf{f}_{\mathbf{y}}) = p(C_i|\mathbf{f}_{\mathbf{y}})$. Also, the distribution $p(o,\mathbf{x}|C_i,\mathbf{y},\mathbf{f}_{\mathbf{y}})$ only depends on the matched codebook entry C_i and the location of the image patch $\mathbf{y}$:

$$
\begin{aligned}
S(o,\mathbf{x}) &= \sum_{i=1}^{K}\sum_{\mathbf{y}} p(o,\mathbf{x}|C_i,\mathbf{y})p(C_i|\mathbf{f}_{\mathbf{y}}) \\
&= \sum_{i=1}^{K}\sum_{\mathbf{y}} \underbrace{p(o|C_i)}_{\text{weight}}\underbrace{p(\mathbf{x}|C_i,\mathbf{y})}_{\text{location}}\underbrace{p(C_i|\mathbf{f}_{\mathbf{y}})}_{\text{matching}}
\end{aligned} \tag{2.7}
$$

where the codebook likelihood $p(o,\mathbf{x}|C_i,\mathbf{y})$ is decomposed into a weight term and a location term. We now discuss the common choices for the three terms in the basic Hough transform detector.

The weight term. The weight term $p(o|C_i)$ quantifies how confident we are that the codebook entry C_i matches the object as opposed to the background. The simplest choice would be a uniform weight, i.e., assuming each codeword is equally likely to be an object part. In fact, when we have negative samples there is a better way for estimating the weight [129]:

$$p(o|C_i) \propto \frac{p(C_i|o)}{p(C_i)} \tag{2.8}$$

where $p(C_i|o)$ is the relative frequency of the codeword C_i on the object features, while $p(C_i)$ is the relative frequency of both positive and negative training images. The weight in this case is referred to as a naive-Bayes weight, as the weight is set independently for each codeword.

The location term. The location term $p(\mathbf{x}|C_i, \mathbf{y})$ is the probabilistic Hough vote for the location of the object. It can be estimated by a Mixture of Gaussian [106] or encoded nonparametrically during codebook learning by observing the geometric distribution of the codebook activations relative to the object center [109]. For example, in the implicit shape model [109] each codebook entry C_i consists of a typical patch descriptor $\mathbf{f}_{ci}$ and a set D_i that contains geometric features of training patches associated with the i-th entry. A typical geometric feature is the relative positions $\boldsymbol{d}$ of image patches w.r.t. the corresponding object centers. The location term in this case can be written as

$$p(\mathbf{x}|C_i, \mathbf{y}) = \frac{1}{Z} \sum_{\boldsymbol{d} \in D_i} e^{\left(-\frac{\|(\mathbf{y}-\mathbf{x}_c)-\boldsymbol{d}\|^2}{2\sigma_d^2}\right)} \tag{2.9}$$

where $\mathbf{x}_c = (a_x, a_y)$ is the center of bounding box and $(\mathbf{y} - \mathbf{x}_c)$ is the offset from the object center to the image patch, and σ_d is the standard deviation of a Gaussian filter for the object center. We can also use other radially symmetric kernels for the density estimation.

The matching term. The matching term in Equation 2.7, which is the likelihood that the codebook entry C_i generated the feature $\mathbf{f}_\mathbf{y}$, can be estimated by the distance between the codebook entry and the feature as follows:

$$p(C_i|\mathbf{f}_\mathbf{y}) = \begin{cases} \frac{1}{Z}\exp(-\gamma d(\mathbf{f}_{ci}, \mathbf{f}_\mathbf{y})) & \text{if } d(\mathbf{f}_{ci}, \mathbf{f}_\mathbf{y}) \leq t \\ 0 & \text{otherwise} \end{cases} \tag{2.10}$$

where Z is a normalizing factor, $d(\cdot,\cdot)$ is a distance function, and γ, t are positive parameters. Here γ controls the sensitivity to distance variations, and t defines a cut-off threshold for matching.

The basic Hough transform detector described above shares some similarities with the bag-of-words model widely used for image classification tasks. Also, the simple and flexible nature of the Hough voting process leads to various extensions to the original model. Progress has been made in discriminative codebook learning [52, 226], efficient inference methods [106], joint recognition and segmentation [109, 166], scalable multiclass detection [164], maxima

search in high-dimensional Hough spaces [179, 139, 148, 163], and among others. We now discuss several variants and extensions to the basic Hough transform-based object detector.

Bag-of-words model. The bag-of-words model can be seen as a special case of Hough transform-based detectors. The spatial relationships of the features within an object hypothesis are ignored; the model only captures appearance of object parts, not their geometric distributions. The location term in Equation 2.9 in this model only takes the presence of a feature in the object hypothesis bounding box as

$$p(\mathbf{x}|C_i,\mathbf{y}) = \begin{cases} 1 & \text{if } area(\mathbf{x}\cap\mathbf{y}) = area(\mathbf{y}) \\ 0 & \text{otherwise} \end{cases} \tag{2.11}$$

Although the model is typically used for image classification [37, 45], it has also been used for unsupervised object discovery and detection [189].

Implicit shape model. The Hough voting process implicitly reasons about the location of object parts. Therefore it would be possible to link a detected object with the contributing parts, in order to obtain a coarse segmentation. Furthermore, the part-object relations can be used to verify detection results. This idea leads to the implicit shape model [109] that obtains a segmentation of a detection, without any additional labeling. The main idea is to exploit the influence of a given patch $I_\mathbf{y}$ on the object hypothesis:

$$p(I_\mathbf{y}|o,\mathbf{x}) = \frac{\sum_{i=1}^{K} p(o,\mathbf{x}|C_i,I_\mathbf{y})p(I_\mathbf{y})}{p(o,\mathbf{x})} \tag{2.12}$$

where $p(o,\mathbf{x}|C_i,I_\mathbf{y})$ can be computed as shown in Equation 2.7. $p(I_\mathbf{y})$ and $p(o,\mathbf{x})$ are usually assumed as uniform priors. For a specific pixel, the figure-ground probability is estimated by summing up all patches that contain this pixel. The segmentation can be further used to verify multiple detections with a Minimal Description Length (MDL, also known as the Occam's Razor) criterion.

Learning discriminative codebooks. The basic Hough transform-based detector uses clustering (e.g., K-means) to learn visual codebooks. While a certain level of discriminative power can be achieved (e.g., by setting naive-Bayes weights for codewords in Equation 2.8), it is preferable to adopt a discriminative codebook learning approach so that each codeword can be optimized to be as discriminative as is possible. To this end, the class-specific Hough forest [52] has been proposed to learn a discriminative codebook where leaf nodes of each tree directly optimize voting performance by minimizing class impurity and offset variance. At each tree node, the algorithm picks one of those uncertainty measures at random, and splits the image features into two subsets for its children nodes by minimizing the chosen uncertainty.

More formally, each tree T is constructed through a series binary tests $t(I_{\mathbf{y}}) \rightarrow \{0,1\}$ defined on a set of training patches $\{I_{\mathbf{y}} = (\mathbf{f}_{\mathbf{y}}, c_{\mathbf{y}}, \boldsymbol{d}_{\mathbf{y}})\}$, where $\mathbf{f}_{\mathbf{y}}$ refers to the appearance of patch $I_{\mathbf{y}}$, $c_{\mathbf{y}} \in \{0,1\}$ the class label, and $\boldsymbol{d}_{\mathbf{y}}$ the offset w.r.t. object center ($\boldsymbol{d}_{\mathbf{y}}$ is undefined for background patches, i.e., those with $c_{\mathbf{y}} = 0$). The *class-label uncertainty* measure is defined by:

$$U_1(\{I_{\mathbf{y}}\}) = -|\{I_{\mathbf{y}}\}| \cdot Entropy(\{c_{\mathbf{y}}\}) \tag{2.13}$$

where $Entropy(\{c_{\mathbf{y}}\}) = -c \cdot \log c - (1-c) \cdot \log(1-c)$ in which c is the proportion of patches with label $c_{\mathbf{y}} = 1$ in $\{I_{\mathbf{y}}\}$. $|\{I_{\mathbf{y}}\}|$ is the size of $\{I_{\mathbf{y}}\}$. The *offset uncertainty* is specified by the variance:

$$U_2(\{I_{\mathbf{y}}\}) = \sum_{I_{\mathbf{y}}: c_{\mathbf{y}}=1} (\boldsymbol{d}_{\mathbf{y}} - \boldsymbol{d}_A)^2 \tag{2.14}$$

where $\boldsymbol{d}_A$ is the mean offset vector for all patches reaching the node.

At each node during training, a pool of binary tests $\{t^k(\cdot)\}$ is randomly generated. A tree T recursively grows each node by finding a binary test that minimizes the following criterion:

$$\min_k \Big(U_\star(\{I_{\mathbf{y}} | t^k(I_{\mathbf{y}}) = 0\}) + U_\star(\{I_{\mathbf{y}} | t^k(I_{\mathbf{y}}) = 1\}) \Big) \tag{2.15}$$

where $\star = 1$ or 2 which corresponds to the random choice of uncertainty measure. If the depth of the node has reached the maximum depth of a tree or the number of patches associated with the node is smaller than a threshold, the node is declared as a leaf. For the i-th leaf, the confidence score $c_{ci} \in [0,1]$ is the ratio of foreground patches in all patches reaching the leaf. The offset vectors of the foreground patches, denoted as D_i, are stored for voting at test time.

During testing, each image patch is evaluated against the binary tests until they reach a leaf node. Given an image patch at location $\mathbf{y}$, its vote for the object center $\mathbf{x}$ is computed by:

$$p\big(o, \mathbf{x} | \mathbf{y}; T\big) \propto \frac{c_{ci}}{|D_i| \sigma^2} \sum_{d \in D_i} e^{\left(-\frac{||(\mathbf{y} - \mathbf{x}_c) - \boldsymbol{d}||^2}{2\sigma^2} \right)} \tag{2.16}$$

where c_{ci} and D_i are the confidence score and offset vectors associated with the reached leaf node, respectively. For the entire forest, the average of the probabilities coming from all trees is used for the forest-based estimate.

Learning discriminative codebook weights. As discussed, the weight term in Equation 2.7 can either be set to uniform or according to Bayes' theorem (resulting in the naive-Bayes weight in Equation 2.8). Ideally, we would prefer to learn the weight term so that parts that are both repeatable and occur at a consistent location obtain higher weights. In the max-

margin Hough transform framework, once a codebook is generated we can discriminatively learn weights for each entry to directly optimize classification performance [129].

Note that the scoring function in Equation 2.7 is linear w.r.t. the weight term $p(o|C_i)$, we can replace it with a weight w_i for the i-th codebook entry and get

$$S(o,\mathbf{x}) = \sum_{i=1}^{K} w_i \cdot \sum_{\mathbf{y}} p(\mathbf{x}|C_i,\mathbf{y}) p(C_i|\mathbf{f}_\mathbf{y}) \tag{2.17}$$

For notation simplicity, we further define an activation vector $A^T = [a_1, \cdots, a_K]$, where $a_i(\mathbf{x}) = \sum_{\mathbf{y}} p(\mathbf{x}|C_i,\mathbf{y}) p(C_i|\mathbf{f}_\mathbf{y})$. The max-margin Hough transform [129] learns w_i as follows

$$\begin{aligned} \min_{\mathbf{w},b,\xi} \ & \frac{1}{2}\mathbf{w}^T\mathbf{w} + C\sum_{i=1}^{T}\xi_i \\ \text{s.t. } & z_i(\mathbf{w}^T A_i + b) \geq 1 - \xi_i, \\ & \mathbf{w} \succcurlyeq 0, \xi_i \geq 0, \forall i = 1,2,\ldots,T \end{aligned} \tag{2.18}$$

where $\mathbf{w}^T = [w_1, \cdots, w_K]$ is the weight vector, A_i is the activation vector for the i-th training sample, $z_i \in (-1, +1)$ is the binary label for each training sample. The formulation is similar to the objective function of a linear SVM [15] with an additional positivity constraint on the weights.

Learning the weight vector $\mathbf{w}$ requires negative training samples, and the number of negative samples in a typical object detection setting can be much larger than positives. To retrieve hard negative instances, one can bootstrap the hard mining process by finding peaks in the voting space using uniform weights.

The naive-Bayes weight in Equation 2.8 takes into account only the appearance of a codeword, while the max-margin Hough transform weight jointly considers the codeword appearance and the spatial distributions of feature positions w.r.t. to the object center to derive its importance.

Beyond voting with patches. The Hough voting procedure does not have any restrictions on the voting elements, i.e., it refers to any detection process based on an additive aggregation of evidence coming from local image elements. Image patches are typically selected as voting elements for their simplicity and ease of implementation. However, it would be useful to consider voting elements beyond image patches that, for instance, carry more resemblance to human perception. In particular, fragments of outline contour have been shown to be useful for object detection. For example, Shotton, Blake and Cipolla [184] propose a codebook learning scheme purely based on local contour features. Opelt, Pinz and Zisserman [149] use both

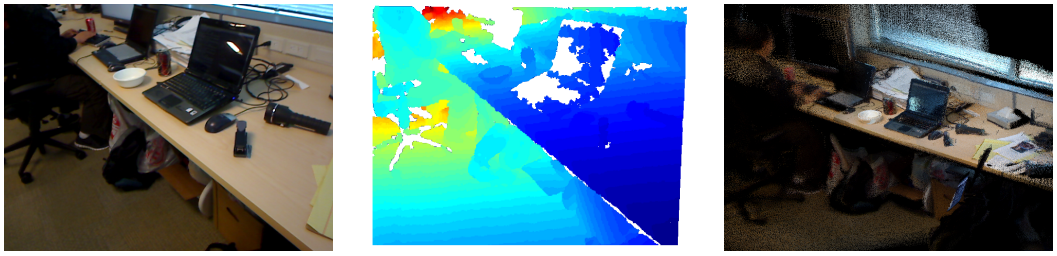
image patches and boundary fragments, and then AdaBoost [207] to select pairs of these voting elements as weak learners. Common to both methods is that the spatial distributions of the image features are used as a cue for codebook learning, whereas in the basic Hough voting model only appearance is considered for clustering.

In addition, regions can be an appealing choice for voting elements as they encode shape and scale of objects naturally, and are only mildly affected by background clutter. Gu et al. [63] propose to learn codewords from a bag of overlaid regions for Hough voting based detection. Another work from Yu et al. [230] explores Hough voting under a joint detection-and-tracking setting in video, in which they aggregate votes with both spatial and temporal structural information.

High dimensional Hough spaces. One important advantage of Hough transform-based detectors is their flexibility to encode different pose parameters. The basic Hough voting space is the 2D image coordinate space, where each point (a_x, a_y) corresponds to an object hypothesis centered at $\mathbf{x}_c = (a_x, a_y)$, with a given scale and aspect ratio. It is straightforward that Hough voting can vote for scales and aspect ratios. For example, Seemann, Leibe and Schiele [179] propose a multi-aspect detection approach based on Hough voting. Ommer and Malik [148] propose pairwise clustering of voting lines to obtain object hypothesis in the joint location and scale space. We can also allow codebooks to encode other aspects of an object model. For example, Mikilajczyk, Leibe and Schiele [139] propose a Hough voting method with rotation recovery. In the latent Hough transform [163], Razavi et al. cast the grouping of object properties such as pose, color, shape, or subcategory as a latent assignment problem, and learn the grouping from training data. In Chapter 3, we extend the Hough space to include occlusion estimation and propose a mask voting scheme to efficiently search over the extended high dimensional Hough space.

So far, we have shown that the implementation of probabilities $p(o|C_i)$, $p(\mathbf{x}|C_i, \mathbf{y})$ and $p(C_i|\mathbf{f_y})$ in Hough transform detectors are highly flexible, and this has been a major reason for researchers to adopt this framework. In addition, the codebook-based representation allows for designing a more structured representation of voting elements. In particular, under the Hough transform framework it is not restrictive to assume that the voting elements must be groups of object parts collected from within the bounding box. This naturally permits us to incorporate occlusion and context reasoning within a unified framework. In Chapter 3, we present a novel Hough transform detector that features a structured codebook representation to explicitly reason about object parts, occlusions, and the spatial context. Closely related to this issue are the recent advances in image sensors that allow people to collect high quality depth data co-registered with color images. This provides a convincing understanding of the underlying 3D configuration of objects in a 2D image. In the next section, we discuss object detection with RGBD data, and then continue our discussion by looking at occlusion reasoning and context modeling for object detection in Sections 2.1.4 and 2.1.5.

2.1.3 Object detection with RGBD data



Color image Depth image Point cloud

Figure 2.2: Example of RGBD imagery. The point cloud was reconstructed from a video sequence including the color and depth frames. Depth images are color coded so that pixels close to the camera are shown in blue, and far-away pixels are in red. Missing depth values are shown in white.

Despite the rapid progress in object detection we reviewed so far, generic object detection is far from a solved problem. In one of the most widely recognized visual recognition challenge in the computer vision community, the VOC Challenge [43], average precision for state-of-the-art object detectors hovers around 30% to 60% depending on the difficulty of individual object categories. This prompts researchers to look into alternative sensory data, among which depth sensors are the most prevalent. In particular, depth sensors make it easier to identify major scene structures, allowing the extraction of accurate geocentric information about a scene. Depth can also be beneficial for context and occlusion reasoning. Indeed, contextual interactions happen in 3D and depth would provide credible cues for the spatial relations among objects.

Recently, the advent and popularity of affordable RGBD sensors have seen an increased interest in building depth-aware object models. Foremost to the research is the availability of high quality, dense RGBD data. There have been a few public RGBD datasets made available for a range of scene understanding tasks, such as object detection, object segmentation and image labeling [98, 145, 77, 55]. See Figure 2.2 for a sample RGBD frame and a point cloud from [98].

The most easily perceivable opportunity for improvement is perhaps to design novel depth-aware features. For example, Spinello and Arras [190] show that directly applying the highly successful HOG features [38] on depth data can help improve pedestrian detection performance. This feature has also been successfully applied to hand pose estimation [169, 170] and generic object detection [98, 66, 30]. Lai et al. [98] compare the effectiveness of shape and visual features on a large-scale RGBD dataset and demonstrate that the combination of two gives best object recognition performance. The shape features used in their work include spin images [79] and SIFT descriptors [121]. In their subsequent work [18, 19] they further propose to use learned depth features leading to improved results. The detection results can

also facilitate other scene understanding tasks, such as image labeling [100]. Yebes et al. [228] propose 3D-aware features computed from stereo images for objects in road scenes.

In addition to depth-aware features, Choi et al. [34] develops a conditional random field model that jointly reasons about object appearance, geometry, and scene-object relations with RGBD data. Similarly, Lin et al. [113] recognize 3D cuboids by jointly exploiting 2D segmentation, 3D geometry, as well as contextual relations between the scene and objects. Gupta et al. [66] propose geocentric embedding for learning depth-aware feature representations with convolutional neural networks. Liu et al. [119] detect objects with 3D sliding boxes using deep Boltzmann machines to piece together appearance and depth features learned with [58].

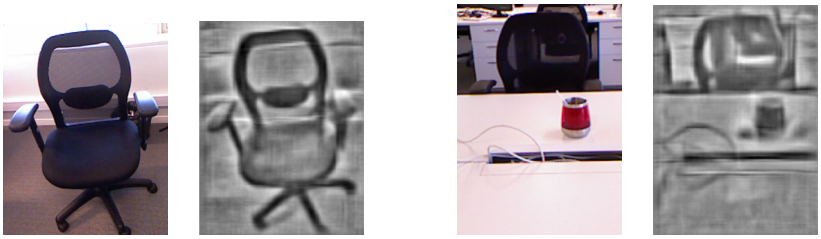
One particular problem of the abovementioned methods is that they require depth information in both model training and evaluation. As depth-capable sensors are far from ubiquitous compared to color cameras, it would be advantageous if we are able to apply models learned with depth information to 2D cases where only color images are available. This particular angle that uses auxiliary depth information is not fully explored in the literature. In Chapter 3 we aim at learning a depth-encoded object detection algorithm that can be applied to 2D images. In fact, we can learn an appearance model with better quality when depth information is available during training, and transfer the depth knowledge for a test scenario with color images only. For example, Zhang et al. [234] demonstrate that depth information in the training phase can benefit scene classification and instance level object recognition. Shrivastava and Gupta [186] propose to learn a geometry-driven DPM from RGBD images. In particular, Sun et al. [195] use a depth-encoded patch selection process for Hough transform-based detection. They use depth to prune out patches of incorrect scales, and to create a generative depth model of an object.

Occlusion reasoning and context modeling, as will be discussed in the next two sections, can also benefit from additional depth information. In particular, when modeling the context in 3D, occlusion can be naturally viewed as a special type of contextual relationship, which would become an intrinsic component of object and scene models. Also, if we reason about geometric relations among objects in 2D we have to deal with uncertainties introduced by the projection from the 3D world to a 2D image. By reasoning the context directly in 3D, we can potentially eliminate some of these geometric uncertainties introduced by the 3D-to-2D projection.

2.1.4 Occlusion reasoning for object detection

Most object detection methods introduced in the previous sections rely on one important assumption: the majority of images used for both training and testing should only include fully visible views of an object. There is no special handling for partially visible objects. Therefore these objects could negatively impact the training and testing process. This is because the al-

gorithms can confuse between the very different appearances of a fully visible object and that of a partially visible one. See Figure 2.3 for an example. In Figure 2.3 (c), the appearance of the table in front of the chair is very different from the chair seat and base being occluded, yet it could still be predictive of the presence of a chair behind as these table-chair configurations are commonly found in an office scene.



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

Figure 2.3: The frontal views of two visually similar chairs (cropped). For each chair the original image is shown on the left (**(a)** and **(c)**), with the visualized HOG feature map [208] on the right (**(b)** and **(d)**). For the partially occluded chair, the seat and the base are occluded by a table in the front. See text for details.

One possible way to deal with this partial visibility problem is to require all bounding box annotations to include only visible object parts, and treat those partially visible objects as separate subcategories using methods such as mixture models. For example, in the chair category we may have a dedicated subcategory that detects backrests. In fact, this simple strategy has been proven effective in state-of-the-art object detection systems such as the DPM [46]. The downside, however, is that it requires more training data to cover all typical viewpoint variations. Conceptually, it is preferable to treat the backrests of the two chairs in Figure 2.3 as a single object part, and build an object model that allows certain parts of an object to be occluded.

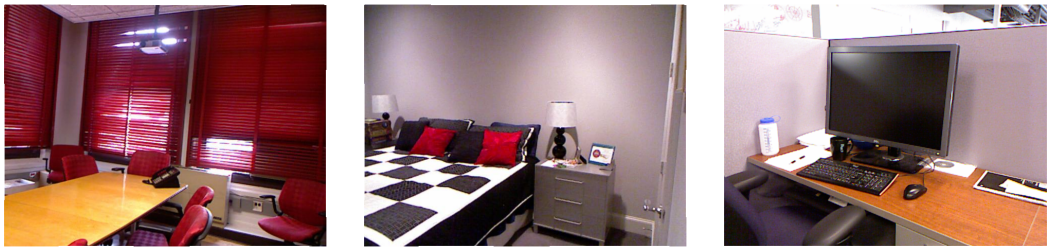
It should also be noted that the partial observation issue is more prevalent in indoor object detection problems. This is primarily due to two underlying facts that produce two typical partial observation scenarios. Firstly, due to the compact nature of indoor spaces, many objects have to be arranged closely to each other. In particular, some objects are arranged in functional groups to facilitate human interactions. Examples include the typical configurations of table and chairs, and the various components of a desktop computer (e.g., a monitor, a keyboard and a mouse). We refer to this scenario where one object blocks the view of another object as *occlusion*. Another typical scenario is when the viewer (or camera) is too close to the object so that the object is unable to fit in the viewing window. This results in a partially visible object truncated by image boundaries. We refer to this case as *truncation*.

The presence of occlusion and truncation makes object detection more challenging. For detectors not explicitly reasoning about occlusion and truncation, it is likely that inconsistent

part appearances or geometric distributions will be mixed up with regular ones, resulting in much larger intraclass appearance variations. The models introduced in the previous sections could easily fail in the presence of occlusion, as features from the occluded parts will adversely contribute to the score of object hypotheses. In this regard, explicit occlusion reasoning is necessary for objects that are frequently being occluded.

Because of its prevalence in many real-world applications, occlusion has been well studied in the computer vision literature. One basic strategy is to allow object detectors identify partial occlusion so that the occluder would not adversely affect the score of an object hypothesis. For the simple template matching based sliding window detector in Section 2.1.1, we can use the scores of individual HOG cells to infer occlusion [213]. For part-based models, Girshick et al. [59] use an occluder part in their grammar model when all parts cannot be placed. Tang et al. [197] leverage the fact the occlusions often form characteristic patterns and extend the DPM for joint person detection and tracking. Wojek et al. [218] combine object and part detectors based on their expected visibility using a 3D scene model. Wu and Nevatia [220] maximize a joint likelihood that involves responses of multiple part detectors for multiple, partially occluded humans. Li et al. [111] present a method for detecting partially occluded cars based on And-Or models. Brox et al. [26] use a part-based poselet detector and align the corresponding part masks to image boundary cues. Another work that also reasoned about occlusion within bounding boxes for object detectors is [53]. The bounding box representation was augmented with a set of latent variables to generate a binary occlusion pattern. In addition, they enforce consistency between visibility patterns of multiple objects and their relative depth ordering. This is inspired by an earlier paper that uses structured output regression for detection with partial truncation [206]. To reduce noise in occlusion classifications, local coherency of regions is often enforced [50]. One common feature for the papers mentioned above is that they mainly focus on modeling occlusion without complex reasoning about the underlying 3D scene, partially due to the fact that depth data is not easily accessible, making it difficult to study the real 3D configuration of objects in a scene.

Recently with accessible 3D data collected from affordable RGBD sensors, there has been an increasing amount of work on occlusion reasoning in 3D. For example, Meger et al. [136] use depth inconsistency from 3D sensor data to classify occlusions. Pepik et al. [157] leverage fine-grained 3D annotated urban street scenes to mine distinctive, reoccurring occlusion patterns. Detectors based on DPM with explicit occluder parts are then trained for each of these patterns. Zia et al. [241] model occlusions on a 3D geometric object class model by enumerating a finite number of occlusion patterns. Hsiao and Hebert [74] explicitly model occlusions by reasoning about 3D interactions of objects. These works reason about 3D geometric configurations of parts, objects and cameras in 3D that help to explain occlusions more naturally. In addition, Bonde et al. [20] address the problem of object instance recognition in clutter that allows them to learn discriminative 3D shape features for individual object instances. Simi-



(a) (b) (c)

Figure 2.4: Example of indoor scenes. Note how objects are occluded or truncated by image boundaries. Groups of objects are also arranged together to facilitate human interactions.

larly, Tejani et al. [198] propose a latent-class Hough forest in which the class distributions at leaf nodes are treated as latent variables. Unlike our work, their method focuses on 3D pose estimation where a dense 3D model of each object instance is needed.

Despite the progress, 3D occlusion reasoning in general is less studied due to poorer data availability. As discussed in Section 2.1.3, although there have been a few large publicly available RGBD datasets, most imagery data available and being created nowadays are color images only. Therefore, one key issue here is to train a better occlusion-aware object detector with auxiliary depth information and apply it to a test scenario without depth. Another issue is to integrate the depth-aware occlusion reasoning into a coherent object detection framework. In Chapter 3, we present an object detection system that aims at resolving these issues.

2.1.5 Context modeling for object detection

Most algorithms we discussed in the previous sections disregard information from outside the object bounding box, which we referred to as the *spatial context* of an object. For human observers, however, we understand a scene holistically and would utilize any part of the context that is predictive of object locations to quickly identify possible image regions that may contain an object of interest. In particular, occlusion can be viewed as an integral part of context in 3D. As discussed in the previous section, occlusion can help improve object detection performance if properly modeled.

Despite the obviously larger appearance variations of the context, it would be helpful if we can roughly predict potential locations of certain objects with the help of their spatial context. This is particularly possible for man-made environments where objects are typically arranged in specific ways to facilitate human interactions. Consider the indoor scenes depicted in Figure 2.4 for an example. Most people will have little difficulty seeing the partially occluded or truncated chairs in (a), the bed in (b), and the table in (c). Also, for small objects with limited visual cues such as the mouse in (c), the surrounding context (e.g., the keyboard and monitor near the mouse) makes it easier to recognize them. The same is true for the lamps in (b). Con-

textual relationships are an integral component of a coherent visual story told by an image, and they can be particularly useful in indoor scenes as viewers are typically closer to the objects, making them partially visible in many cases.

The basic frameworks for both detection strategies discussed in Sections 2.1.1 and 2.1.2 do not explicitly reason about the spatial context. Object detection in those cases is solved locally using cues from within the bounding boxes only and contextual cues are discarded. This results in difficulty in recognizing certain objects that have limited visual information from the objects themselves, but the spatial context explains away the uncertainties about their presence. Below we review related work that specifically addresses these issues in object detection.

In both the psychophysics and the computer vision communities, it is widely acknowledged that contextual information plays an important role in detecting and localizing objects (e.g., [14, 73]). In general, two sources of contextual information are most widely used in object detection methods. One is to rely on semantic contextual information at an object level (e.g., in terms of previously detected objects). The drawback of this conceptualization is that it renders the complexity of context analysis to be at par with the problem of semantic understanding of the scene (e.g., object detection). Another way is to use the entire scene information holistically, e.g., using contextual features without explicitly reasoning about the semantic context.

Context-aware object detection has been well studied, and many context-aware object detection methods have been proposed. See [219] for a recent review and [40] for an empirical study. For example, Wolf and Bileschi [219] use relative positions of other detected objects in a scene as well as low-level cues such as global positions, color and texture to build a map of the contextual support for the target object. Torralba and Sinha [201] show that context can 'prime' an object detection system by providing strong cues for location and scale selection, from a holistic representation of context based on the spatial layout of spectral components. Torralba, Murphy and Freeman [202] propose boosted random fields, which learn contextual relationships by assembling graph fragments in an additive model. Maire, Yu and Perona [127] propose to jointly solve image segmentation, figure-ground organization and object detection as a grouping problem based on a graph that captures interactions among pixels, object parts and its surroundings. Blaschko and Lampert [16] use local and global context kernels with SVMs to learn the importance of different context contributions during training. Pan and Kanade [153] generate 3D geometry hypotheses with a generalized RANSAC algorithm and integrate them into an MRF that jointly considers object-context and object-object compatibilities. In addition, many works rely on semantic contextual information at an object level (e.g., [185, 87, 161]). In particular, Mottaghi et al. [143] exploit both the local and global context by reasoning about the presence of contextual classes, and propose a context-aware improvement to the DPM. Zhu et al. [239] use convolutional neural networks to obtain contextual scores for object hypotheses, in addition to scores obtained with object appearance. Yang

et al. [223] have shown that reasoning about a 2.1D layered object representation in a scene can positively impact object detection.

It should also be noted that context and occlusion reasoning is closely relevant to holistic scene understanding approaches, e.g., those jointly solve object detection and segmentation, among other scene understanding tasks. For example, Yao et al. [225] propose a holistic scene understanding model that jointly solves object detection, segmentation and scene classification. However, they did not incorporate explicit context and occlusion modeling.

Despite the progress, most existing contextual models focus on 2D spatial relationships among objects on the image plane and fewer works have extended the modeling to 3D scenarios. One main difficulty in modeling the 3D context was the lack of accessible 3D data. As discussed in Section 2.1.3, it has recently become feasible to collect a large amount of high quality depth and co-registered color images for indoor environments with the recent progress in consumer-level depth sensors. Sudderth et al. [193] propose a system that models object categories over the 3D locations and appearances of visual features. The 3D geometry required for training are obtained from binocular stereo images. More recently, Bao, Sun and Savarese [8] proposed a coherent object detection and supporting surface reasoning algorithm that maximizes the joint probability of having a number of detected objects on a few supporting planes given the observations. They also propose a geometric context feedback loop [194] that iteratively solves object detection, support region segmentation and layout estimation. Unlike their work, we aim to utilize RGBD datasets to learn a context-aware object detection model that encodes depth cues and a coarse level of 3D relationships in Chapter 3. More specifically, we train a depth-dependent appearance model for each object class and its context. The learned depth-encoded object and context model is then applied to 2D images during test. Our model is a structured Hough transform detector that jointly solves for object detection and occlusion estimation. This is made possible by modeling occlusion as an integral part of the depth-encoded context.

2.2 Object segmentation in computer vision

While object detection algorithms provide a good estimation of object locations within an image, the bounding box representation may not be sufficiently descriptive for scenarios where detailed shape or pose is desired. In such scenarios, we can instead infer a pixelwise mask for objects belonging to a specific category, labeling every pixel in the image with either a foreground or background membership. This problem is often referred to as *(foreground) object segmentation* in the literature. More generally, if we assign a finite (possibly large) discrete set of labels to every pixel in the image, the problem is also referred to as *image labeling*.

For object segmentation, one typical solution is to use a statistical classifier for each and

every pixel (or superpixel) based on local appearance, then use Markov Random Fields (MRFs) to incorporate contextual information. The MRF allows for joint reasoning of local and contextual cues, and efficient inference methods exist in many scenarios. More importantly, one can incorporate class-specific object appearance (e.g., shape) information to bias local segmentation results. Despite being one of the most widely adopted methods, the MRF is not a panacea for any object segmentation problem. Some of the most common issues researchers need to consider in their model design include difficulty in dealing with long-range contextual interactions or a large number of semantic categories, and the combinatorial nature of the inference problem.

In this section, we firstly review work on foreground object segmentation in Section 2.2.1, and then discuss two major aspects involved in MRF-based object segmentation: context modeling and inference in Sections 2.2.2 and 2.2.3. Particularly, an important yet challenging problem in object segmentation is the localization of semi-transparent objects. These objects are commonly found in indoor environments and play a key role in daily human activities. The challenging nature of this problem lies in the fact that the appearance of semi-transparent objects varies greatly and largely depends on the background. In Section 2.2.4, we look at existing work on glass object segmentation, which is the problem we wanted to address in Chapters 4 and 5.

2.2.1 Foreground object segmentation

Depending on the number of object categories we are interested in, foreground object segmentation can either be a binary *figure-ground segmentation* problem, or a more general image labeling one. In general, foreground object segmentation relies on two broad types of cues. One is *bottom-up* cues based on local appearance. For example, one can first segment an image into homogeneous regions and then classify them using local color and texture. An important assumption for this approach is the (local) uniformity and continuity of object appearance. However, the appearance variation within an object instance can be potentially large, and background clutter renders the problem of identifying accurate object boundaries even more difficult. The *top-down* approach, on the other hand, reconciles object detection and segmentation by applying learned object detection models to guide the segmentation process. Properties that can be used to guide segmentation include possible shape, color and texture of an object category. The main difficulty for the top-down approach is similar to training object detectors: the large structural variability for certain object categories can be difficult to capture using a concise object appearance model.

Most researchers build their models by designing a method to jointly consider top-down and bottom-up cues. For example, Liu and Sclaroff [117] propose a deformable shaped-based segmentation algorithm where bottom-up segmentation and top-down deformable templates

are combined by split and merge. The implicit shape model [109] discussed in Section 2.1.2 learns segmentation masks corresponding to visual codebook entries. Mori et al. [142] tackle the problem of joint detection and segmentation of baseball players by assembling detected salient parts such as limbs and torsos. Similarly, the ObjCut framework [95] uses a part-based model to bias a bottom-up grouping process. Their model is among the first to combine MRFs with pictorial structure models for foreground object segmentation. See [223] for another similar work on multi-layer object segmentation. In particular, Ladicky et al. [97] propose a hierarchical MRF model that jointly reasons about pixels, segments and objects with a single global energy function. More recently, Guo et al. [64] harnessed the arbitrariness of foreground appearance, the spatial-temporal smoothness of foreground, and the correlation of background for foreground segmentation.

It should be noted that image contours are a natural link between low level image features and high level semantics. For example, the classic active contour model [82] can be used to segment foreground objects. Prasad et al. [160] propose to learn class-specific edges for object detection and segmentation. More recently, Brox et al. [26] used image contours and texture patches as two complementary bottom-up features for foreground object segmentation. The link between bottom-up features and top-down semantics is established by non-rigidly aligning poselet activations to the corresponding edge structures in an image. Parkhi et al. [155] use a template-based model to detect distinctive parts of an object, followed by segmentation with image specific information to complete the detection spatially. This method has been proven to be particularly useful for highly deformable object categories such as cats and dogs. In addition, the problem of salient closed contour detection is closely related to foreground object segmentation. For instance, Mahamud et al. [126] develop a foreground segmentation method using saliency relations based on the global property of contour closure. Arbelaez and colleagues [6] propose a method to transform detected image contours into a hierarchy of regions based on Oriented Watershed Transform and agglomerative clustering. They also develop an approach to detect occlusion boundary for video data based on motion cues [196], in which they can use the above strategy for figure/ground assignment. Inspired by the prior literature, we will show in Chapter 4 that contour-based cues are essential to glass object segmentation performance.

For more flexible shape templates, Borenstein and Malik [21] use a hierarchy of image segments at multiple scales for shape template matching. To deal with the weakly structured object classes, Larlus and Jurie [102] use a bag-of-words based object model to allow for strong viewpoint variations and ensure long range consistency of labelings.

Finally, MRF-based foreground segmentation such as GrabCut [173] can achieve impressive figure-ground segmentation results with the help of user interactions. The key idea is to estimate the color distributions of both foreground and background regions iteratively using graph cut with the aid of sparse user input. More recently, Jain and Grauman [76] proposed

a method to predict the easiest input modality that will be sufficiently strong to successfully segment foreground.

2.2.2 Context modeling with Markov Random Fields

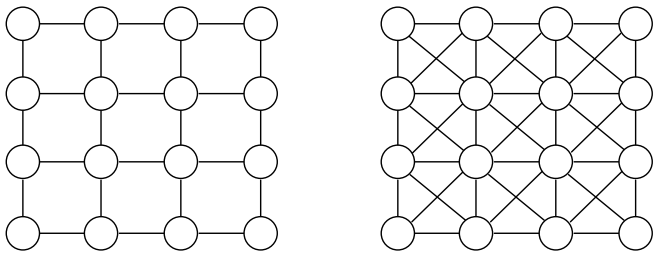


Figure 2.5: Two examples of neighborhood graphs for Markov Random Fields. **Left Panel:** A 4-connected grid of image pixels. **Right Panel:** An 8-connected grid of image pixels.

In this section, we discuss MRF-based object segmentation in detail. The MRF is a probabilistic graphical model that provides a flexible and consistent framework for a variety of probabilistic inference problems [89]. One of the most impressive features of the MRF is the ability to jointly consider local and contextual information in a consistent optimization framework. As opposed to object detection, object segmentation works on a finer granularity to provide detailed object locations. This comes at the cost of more difficult contextual modeling, as the local perspective usually means less awareness of long-range interactions and, by default, a lack of a model for individual object instances. Therefore, as we will show in this section, there has been an abundance of methods focusing on context modeling both working on the local feature level and the second-to-high order constraint level. In Chapters 4 and 5, we will also show how context modeling can facilitate glass object segmentation.

Generally, an MRF models a joint probability distribution over a set of random variables. For object segmentation, these variables are usually associated with image *sites* (i.e., pixels or superpixels). Each image site has a corresponding random variable in the MRF and a node in a neighborhood graph. See Figure 2.5 for two examples of the neighborhood graphs. Markov models explicitly reason about only the connections between relatively few pairs of image sites, typically between neighboring image sites. The explicit short-range interactions then give rise to implicit long-range correlations with a knock-on effect. Researchers also devise graphical models with more effective long-range interactions by adding context-aware features or auxiliary nodes.

We begin our discussion by revisiting a standard pairwise MRF for foreground object segmentation. Mathematically, denote the set of all image sites as $\mathcal{S}$. Let $\mathcal{G} = (\mathcal{V}, \mathcal{E})$ be the neighborhood graph on $\mathcal{S}$ based on the spatial relationship in the image. Here $\mathcal{V}$ and $\mathcal{E}$ are the vertices and edges of the neighborhood graph, respectively. Both $\mathcal{V}$ and $\mathcal{S}$ can be indexed

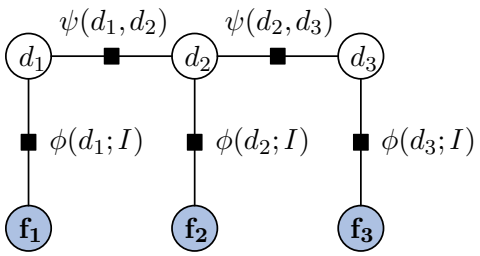


Figure 2.6: The factor graph of the pairwise MRF in Equation 2.20. For simplicity, only three nodes are shown.

by image sites, i.e., $\mathcal{V} = \mathcal{S} = (1, 2, \cdots, i, \cdots, N)$. Note that in practice we may have auxiliary nodes (e.g., nodes at regional or global levels) so $\mathcal{S}$ and $\mathcal{V}$ do not always have this one-to-one correspondence. A typical edge $(i, j) \in \mathcal{E}$ is defined by $(i, j), i, j \in \mathcal{V}$. Also in practice we may have higher order *cliques* that involve explicit interactions among more than two nodes.

Denote $\mathbf{D} = \{d_i\}, i \in \mathcal{V}$ as a set of binary variables associated with $\mathcal{V}$, and we assume a binary state space $\{0, 1\}$ for d_i, with 1 indicating foreground and 0 for background. Denote $\mathbf{F} = \{\mathbf{f}_i\}, i \in \mathcal{V}$ as the observed data (i.e., feature vectors) from an input image. In the MRF framework, the posterior over the labels given the observed data is obtained with the Bayes' rule:

$$P(\mathbf{D}|\mathbf{F}) \propto P(\mathbf{D}, \mathbf{F}) = P(\mathbf{D})P(\mathbf{F}|\mathbf{D}) \tag{2.19}$$

where $P(\mathbf{F}|\mathbf{D})$ is usually assumed to have a factorized form for computational feasibility, i.e., $P(\mathbf{D}|\mathbf{F}) = \prod_{i \in \mathcal{V}} P(\mathbf{f_i}|d_i)$. In this case, the joint probability in Equation 2.19 can be modeled by an MRF that minimizes an energy function of binary labels $\mathbf{D}$:

$$E(\mathbf{D}; \mathbf{F}) = \sum_{i \in \mathcal{V}} \phi(d_i; \mathbf{f}_i) + \beta \sum_{(i,j) \in \mathcal{E}} \psi(d_i, d_j) \tag{2.20}$$

where $\mathbf{f}_i$ is the feature vector associated with the i-th image site. The two terms in the energy function are referred to as *unary potential* and *pairwise potential* respectively. Note that the energy can be seen as a negative log-probability so it essentially factorizes the joint probability distribution into unary and pairwise terms. For ease of interpretation, factor graph representations are commonly used in the literature to explicitly illustrate how the joint distribution over all random variables are factorized. Figure 2.6 shows the factor graph of the MRF in Equation 2.20. Each circular node represents a random variable. Each rectangular node represents a factor. Shaded nodes are observations (i.e., image features).

Local classifier. The first and an essential step towards image labeling is to obtain a local

estimate of labels. Due to the noisy nature of local appearances for a semantic category, the resulting labeling with purely local information is usually not spatially coherent. Yet it provides a good starting point for the more complex context modeling, see Figure 2.7(c) for an example. One typical way of labeling an image is to use a statistical classifier based on local information only. For example, Konishi and Yuille [92] propose to use Bayesian classification based on local color and texture cues for image labeling. He et al. [69] use a multilayer perceptron taking in color, edge magnitude and texture information. In general, when we use a classifier for local estimations, the unary potential can be formulated as the negative log-probability from the classifier output:

$$\phi(d_i; \mathbf{f}_i) = -\log(P(d_i|\mathbf{f}_i)) \tag{2.21}$$

It should be noted that the factorization of $P(\mathbf{D}|\mathbf{F})$, i.e., $P(\mathbf{D}|\mathbf{F}) = \prod_{i \in \mathcal{V}} P(\mathbf{f}_i|d_i)$, can be restrictive for the analysis of natural images where it is important to make use of the spatial dependencies. In particular, different objects can share similar local appearances. For example, in the Discriminative Random Fields (DRF) [96] work Kumar and Hebert consider a structured-vs-nonstructured object segmentation problem. At a local image patch level, buildings (structured) and sky (nonstructured) can have similar color and texture. It is the regional and global context that makes the semantic class clear to the viewer. Generalized Linear Models [133] are used in their work to model the label posteriors given the whole set of observations, instead of observation from a single image site. He et al. [69] propose to learn region and global label features from labeled images in order to incorporate contextual cues at multiple scales. Shotton et al. [185] propose a texture-layout filter to record patterns of textons, and exploit the textural appearance of objects and its contextual layout. The inclusion of contextual cues helps resolve the label ambiguities at the local level. In particular, local appearance exhibits large variations for semi-transparent objects (e.g., the appearance of glass and non-glass surface could be similar locally), and we will show how to incorporate spatial context for local glass segmentation and build a flexible feature pool for glass boundary estimation in Chapters 4 and 5.

Label transfers of local estimates. Similar to object detection, the emergence of large databases of images allows researchers to build nonparametric models for label prediction in image labeling. The basic idea is to explain an image by matching its parts to other images from the database.

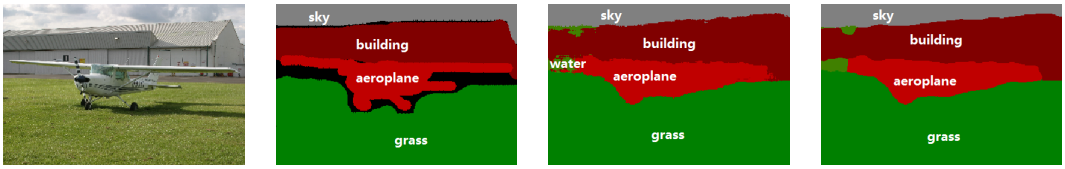
Many of these methods follow a two-step approach. They firstly generate a reasonably sized retrieval set (or a few, see [232]) from a large database by coarse scene matching so that the retrieval set contains scenes with similar object categories and geometric setup to the query. The label transfer then happens at a local level (e.g., a few pixels wide) within the retrieval set. For example, Liu, Yuen and Torralba [115] first retrieve nearest neighbors of

a query image with distance derived from global scene descriptors such as GIST [147] and spatial pyramid intersection of HOG visual words [103]. This is followed by a coarse-to-fine SIFT flow algorithm to establish dense pairwise correspondences between the query scene and each of its nearest neighbors. Finally, they use an MRF to combine the likelihood obtained from SIFT flow, the semantic class location priors, and smoothness constraints. Similarly, Tighe and Lazebnik propose SuperParsing [199] that performs label transfer at the superpixel level to avoid the expensive inference via SIFT flow. This also lowers the need for finding similar scenes in terms of the spatial layout of semantic classes. For both global and superpixel matching they use an extensive set of image features, which is essential for the performance of their method. Other related work includes integrating image parsing with per-exemplar object detectors [200], and building a superpixel graph to allow metric learning for superpixel matching [60]. In particular, Fathi et al. [44] take a semi-supervised learning approach to learn a metric for label propagation in videos.

The benefit of nonparametric methods to scene parsing is at least three-fold. Firstly, we can use simple matching schemes such as nearest neighbor search to obtain a local label estimate, thus the methods are usually computationally fast. This also eliminates the need for training a universal unary classifier, which could be time-consuming. Secondly, nonparametric models can easily adapt to large datasets and a large number of semantic categories, and we do not need to retrain the classifier when more data are added. Finally and perhaps more importantly, appearance of local image regions (e.g., a few pixels wide) usually exhibit large variations. Therefore, it would be difficult to train generic classifiers to capture the variations of small local features. On the contrary, nonparametric models are well-suited for this scenario. Although the work on context features discussed earlier in this section also aims to address this problem, it still requires a universal classifier which makes it difficult to deal with extreme appearance variations or a large number of semantic classes. Related work based on nonparametric label transfer has achieved state-of-the-art results on large benchmark datasets such as the SIFT flow database [115] and the SUN database [221].

In Chapter 5, we introduce a glass object segmentation method based on label transfer on joint depth and appearance manifolds. Our work is the first to explore nonparametric label transfer within the context of glass detection, and exploit a joint depth-appearance manifold for transductive learning. Label transfer is particularly effective for glass objects as the appearance variations at glass boundaries are large. We will discuss the glass object segmentation problem in more details in Section 2.2.4.

Smoothing constraints. One important feature of an MRF is the ability to eliminate noise in local estimates by modeling second or high order constraints among variables. In the pairwise MRF as shown in Equation 2.20, the pairwise term is the summation of pairwise potentials between each pair of nodes in the neighborhood graph. In particular, the Potts model, first developed in statistical physics, is one of the simplest pairwise potentials commonly used in



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

Figure 2.7: Example of image labeling results with TextonBoost [185] using unary terms only, and with pairwise terms added. **(a)** input image. **(b)** ground-truth labeling. **(c)** image labeling result with unary terms only. **(d)** image labeling result with pairwise terms added.

computer vision:

$$\psi(d_i, d_j) = \begin{cases} 0, & \text{if } d_i = d_j, \\ 1, & \text{otherwise} \end{cases} \tag{2.22}$$

This pairwise term gives a constant penalty for inconsistent neighboring labels. We also refer to the binary case of the model above as an Ising model. An image-adaptive version of this pairwise term, the contrast-sensitive Potts model, is also widely used in the image labeling literature. It replaces the constant penalty in Equation 2.22 with an edge feature $\mathbf{g_{ij}}$ based on the difference in colors of neighboring pixels [25, 173, 185]:

$$\mathbf{g}_{ij} = \theta_p + \theta_v \exp(-\theta_\beta ||I_i - I_j||^2) \tag{2.23}$$

where I_i and I_j are the color vectors of pixels i and j respectively. θ_p, θ_v and θ_β are model parameters learned from training data. This contrast-sensitive Potts model penalizes neighboring nodes in the graph having different labels except where there is a corresponding edge in the image. See Figure 2.7 for an example of the smoothing effect of the pairwise potential. Note how the rough edges and the small isolated regions misclassified as water are removed.

One main drawback of the pairwise terms above is that the interactions among observations are restricted to site pairs. The DRF framework [96] proposes to address this issue by learning a data-dependent pairwise discriminative model in the pairwise terms, in addition to the smoothing term of the Ising model. Another problem with the pairwise terms above is that it has an over-smoothing effect in many cases, making the MRF incapable of following fine contours of certain semantic classes such as trees and bushes. To address this issue, He et al. use a superpixel representation of images with the assumption that all pixels from a particular image segment belong to the same semantic class [70]. Instead of using this hard constraint, Kohli et al. [88] propose a quality sensitive and robust high-order $\mathcal{P}^n$ Potts model that favors all pixels belong to an image segment taking the same label, while setting the penalty as a linear truncated function to allow for variables in a clique taking different labels. Krahenbuhl and

Koltun [90] propose a fully connected conditional random field model in which the pairwise potentials are defined by a linear combination of Gaussian kernels. Both the last two methods allow for efficient inference while being able to obtain high quality labeling results in terms of preserving finer details at object boundaries.

2.2.3 Inference in Markov Random Fields

Inference in MRF-based object segmentation is the process of predicting the label values by combining cues from different energy terms, or equivalently, minimizing the energy defined by the energy function. In a probabilistic framework, the possible label configurations are fully described by the posterior distribution of the label variables given the input. In practice, we usually want to obtain a certain point estimator, such as the mean or mode for the distribution, as our labeling output. Each estimator has an associated loss function that quantifies the discrepancy between the estimated configuration and the "ideal" configuration. The estimator minimizes the corresponding loss function. In practice, the *MAP estimate* and the *MPM estimate* are widely used:

- *MAP estimate:* Maximum A Posterior (MAP) of labeling $\mathbf{D}$ given image I is the mode of the posterior distribution,

$$\mathbf{D}^* = \arg\max_{\mathbf{D}} P(\mathbf{D}|I), \tag{2.24}$$

  where the loss function is the 0-1 loss: $L(\mathbf{D}, \hat{\mathbf{D}}) = \delta(\mathbf{D}, \hat{\mathbf{D}})$.

- *MPM estimate:* Marginal Posterior Mode (MPM) is the mode of the marginal posterior distribution,

$$d_i^* = \arg\max_{d_i} P(d_i|I), \forall i, \tag{2.25}$$

  where the loss function is the Hamming loss: $L(\mathbf{D}, \hat{\mathbf{D}}) = |\{i : d_i \neq \hat{d}_i\}|$.

Exact computation of the estimators is feasible for certain probabilistic models with special structures. For all other model structures we have to use approximate algorithms since the exact inference in NP-hard. We will discuss four types of inference algorithms as follows. Note that the three latter types are all approximate inference algorithms.

Exact inference. In certain restricted situations, it is possible to efficiently compute the MAP labeling in MRFs by constructing a specialized graph. In particular, [62] presents the *graph*

cut algorithm, or the *minimum cut/maximum flow algorithm* for binary image segmentation. In the case of a tree-structured graph, the Belief Propagation (BP) [156] algorithm is able to compute the marginals or modes of the model distribution. The BP algorithm propagates a set of messages carrying the interaction information through a tree model until they achieve consistency.

Approximate deterministic inference. In the context of image labeling, the computation of MAP is essentially a combinatorial optimization problem. Therefore, the MAP estimation is an energy minimization in which the domain is discrete. In general, two approximate approaches are commonly used for this minimization-based labeling, i.e., heuristic local search and relaxation-based methods.

Heuristic local search-based methods search for the local minima in a state space neighborhood of an energy function from an initial estimate. Therefore, the quality of solution usually relies on the initial estimate and the size of neighborhood. The neighborhood in the state space is defined with respect to certain transformations of the state configuration. For example, the Iterative Conditional Mode (ICM) [112] approach defines the transformation as changing the label for a single node. Boykov et al. propose an effective local search method with a large neighborhood [24]. The algorithm defines two transformations (or moves), the α-expansion and α-β-swap, generating a much larger neighborhood in the state space. It greedily searches for the local minima based on the current estimate, and in each step finds the locally optimal transformation that gives the largest decrease of energy. In particular, the local search avoids bad local minima, and can be shown to come within a factor of 2 of the energy minimum. Each local move can be formulated as a graph cut problem that can be efficiently solved.

General discrete energy minimization can be viewed as an integer programming problem. In relaxation-based methods, linear programming relaxations have been adopted for approximately solving for the MAP solution in MRFs [217, 224]. Firstly, the MAP problem is formulated as an Integer Linear Problem (ILP). By relaxing the integer constraints, the problem can be converted to a Linear Program (LP) that can be more efficiently solved. The integer solution can be recovered from the fractional solution of the LP [85].

Variational inference. In variational approximation, we use an approximating family of label probability distributions that are simpler than the original distribution and in which the inference is tractable. During inference, we choose a specific distribution from the approximating family to match the original distribution. The marginals or modes of the approximating distribution are used as substitutes for the original ones.

The simplest approximate inference, called mean field approximation, is originally a method of approximation for the computation of the mean of an MRF. Originating in statistical mechanics, mean field approximation uses an approximating family with a fully factorized form [229]. In general, mean field approximation can only obtain a result with good quality when the nodes do not fluctuate a lot around their mean values. The algorithm can be thought of as a parallel

message-passing algorithm where each node sends an identical message to each of its neighbors at a particular time step. The message is, in turn, based on the message it received from its neighbors. It should be noted that we can improve the approximation of mean field by taking factorial distributions where each component is a larger but tractable subgraph of the original factor graph, leading to the structured mean field approach [176]. The fully factorized mean field algorithm is sometimes referred to as naive mean field in comparison.

A more sophisticated approximation based on BP, called the Loopy Belief Propagation (Loopy BP), uses a more complicated approximating family that includes pairwise marginals. In particular, the messages sent from a node to its neighbors at a given time step are different. See [215] for a comparison between the mean field and Loopy BP algorithms.

Sampling-based inference. Sampling methods are a general optimization approach commonly used to handle intractable posterior distributions in MRFs. The Markov Chain Monte Carlo (MCMC) sampling methods, including Gibbs sampling [56] and Metropolis-Hastings sampling [216], are widely used in practice. The basic idea behind MCMC is to define a Markov chain in such a way that its stationary distribution is the target distribution. After drawing samples from the Makov chain, we can derive the distribution or statistics from those samples. In contrast to deterministic methods, MCMC is guaranteed to be unbiased and converge in the limit.

In Gibbs sampling, the algorithm repeatedly sweeps through the MRF updating one node at a time. At each step, a node is updated to be a random draw from its conditional distribution, holding all neighboring nodes fixed. Metropolis-Hastings algorithm provides a more general approach that uses a proposal distribution to sample a candidate labeling given current configuration iteratively, and only changes the current labeling with a certain acceptance probability at each iteration.

Theoretically, the estimates provided by sampling become exact in the limit as the sample size grows to infinity. In practice, however, sampling-based methods are computationally expensive as many samples are needed to obtain a good estimate. Methods have been proposed to improve sampling efficiency, particularly in graphical models with special structures [71].

Simulated Annealing (SA) [205] is another sampling-based algorithm that can be used for MAP inference. It draws samples from the annealed posterior distribution as the temperature decreases. When the temperature gets close to zero, only MAP states have significant probability mass. SA also provides the global MAP estimate, but the annealing must take place in infinitesimal steps, and it uses Gibbs sampling each time the temperature is reduced.

2.2.4 Glass object segmentation

So far we have discussed generic foreground object segmentation with a focus on related work based on MRFs. In this thesis, we are particularly interested in the glass object segmentation

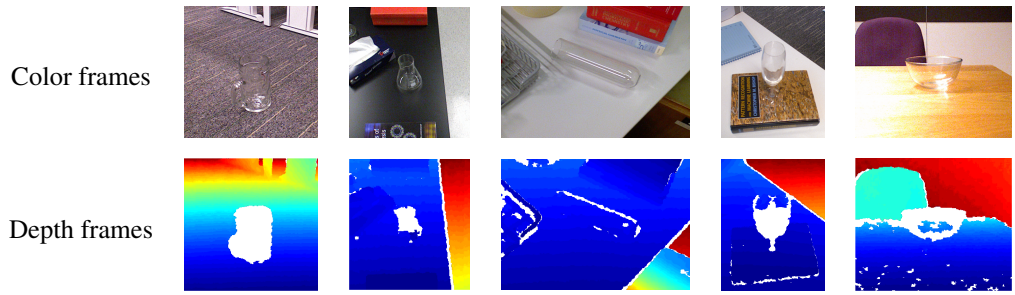


Figure 2.8: Example RGBD image pairs containing glass objects. Note the distinctive but irregular missing patterns in and around glass regions. See text for details.

problem. We make this choice because glass objects play an important role in daily human activities and are commonly found in indoor environments such as home, office and laboratory. Therefore, it is essential for a visual recognition system to be able to localize them.

Despite the progress in generic object segmentation, the segmentation of glass objects remains a particularly challenging problem in scene understanding [75, 137]. The main difficulty in detecting glass objects lies in the semi-transparent nature of glass surface that results in very large appearance variations depending on the background. Therefore, there is a lack of locally discriminative visual features to capture the appearance variations at glass regions and boundaries [135, 51]. For example, visual cues commonly used for image labeling such as color and texture are less effective due to the changing background. In fact, a glass surface can be seen as an overlay on the background so relative features that identify the difference between two image regions may better help localize glass boundaries. In addition, glass objects are usually made for a specific use, and could come in very different and irregular shapes. It is therefore difficult to assume shape templates for glass objects.

In this thesis, we are interested in pixelwise segmentation for semi-transparent objects (including not only glass but also some plastic objects, for example), and we use the term *glass objects* and *semi-transparent objects* interchangeably. In particular, we are interested in making use of RGBD data to localize glass objects. See Figure 2.8 for example RGBD image pairs containing glass objects. Note how the appearances of glass objects in color images are affected by background clutter, and the various overlay effects in glass regions such as blurring, texture distortion, and saturation changes. In addition, notice the distinctive but irregular missing patterns (shown in white) in depth images resulting from attenuation of structured light signals passing through glass. Although these patterns may roughly tell us about the presence of glass, the missing pattern could either be dilated or eroded based on local refractive properties. Moreover, these patterns could spatially overlap with missing patterns caused by other reasons such as occlusion boundaries. These missing patterns can be a nuisance for RGBD imaging but, as we will show in our work, can also be used as an effective feature for glass object segmentation. In this section, we review related work on glass object detection,

segmentation and pose estimation.

Localizing glass objects with color images. We begin our discussion with related work on localizing glass objects with color images only. In general, there are two major problems. Firstly, we have to obtain effective visual features to identify glass regions and boundaries locally. Secondly, we need to build an object model in order to piece together the local estimates and suppress any local noise if possible. For the first problem, as it is difficult to design features to identify a glass region by itself, most previous work has focused on detecting special properties of the glass surfaces and their interactions with the opaque environment in images [151, 144]. Metelli [138] is among the first to study the perception of transparency in terms of spatial and intensity relations of light reflected from a relatively wide field. See [188] for a review and study on the theory of perceptual transparency from the psychology community. One of the early works by Adelson and Anandan [3] in the computer vision community introduces a linear model for the intensity of a transparent surface:

$$I = \alpha I_B + e \tag{2.26}$$

where I_B is the intensity of the background, α is a blending factor, and e is the emission of the semi-transparent surface. They relate the characteristics of visual transparency to the characteristics of the X junctions resulting from patterns on overlapping distinct layers. In addition to this overlay model, highlights are another useful cue as glass is known to be highly specular, and highlights can be found in color images by assuming a dichromatic reflection model [86]. In particular, McHenry, Ponce and Forsyth [135] design a classifier that attempts to find a glass/non-glass boundary based on a combination of visual cues. They compute relative features at both sides of a boundary fragment to partially address the appearance variation issue. Similar cues are also used in [91]. The cues used in their papers include:

- **Color similarity:** the color tends to be similar of both sides of a glass boundary;
- **Blurring:** the texture on the glass side is blurrier;
- **Overlay consistency:** the intensity distribution on the glass side is constrained by the intensity distribution on the non-glass side. In particular, pixels on the glass side usually have a lower saturation value;
- **Texture distortion:** the texture on the glass side is slightly different;
- **Highlights and caustics:** the presence of highlights and caustics increases the probability of a possible transparent material around;
- **Cross-correlation:** distortion produced by a semi-transparent object can also be captured by region analysis, e.g., a cross-correlation measure.

Usually these cues are considered as noise and discarded in object detection and segmentation. However, they are characteristic of glass/non-glass boundaries. In particular, Osadchy et al. [151] recognize objects from specular reflections using knowledge of their 3D shapes.

In terms of object models, McHenry and Ponce [134] propose two complementary measures of affinity and another of discrepancy between regions to group image regions into glass/non-glass surfaces. The local predictions are combined using the geodesic active contour framework [29]. Their work focuses on the binary criteria that answer if two regions are made of the same material, and do not consider the unary region estimates.

Fritz et al. [51] model local patch appearances with an additive model of latent factors in order to detect transparent visual words, and then use latent topic activations to generate object hypotheses. The basic idea behind the additive latent model is that the appearance of a glass region is a combination of factors including background and one or more patterns that have been affected by refraction effects. Their method uses a sliding-window based approach to infer latent topic activations based on linear SVMs. Therefore, it only generates bounding boxes for likely glass object locations instead of a pixelwise segmentation.

Localizing glass objects with multimodal data. The challenging nature of glass object detection and segmentation encouraged researchers to utilize additional sensory information beyond single-view visual cues. In most cases, range (depth) cameras are employed to detect semi-transparent objects, in which the attenuation of signal intensities is exploited.

Klank, Carton and Beetz [84] use two images from a time-of-flight camera to detect and reconstruct transparent objects. Their active infrared camera is robust to illumination changes, however has a shadow-like behavior for glass objects. To deal with this, they adopt a two-step reconstruction scheme and assume glass objects as piecewise planar to get an initial reconstruction. Lee and Shim [105] use a stereo time-of-flight camera setup and derive a generalized depth imaging formulation for translucent objects. They find that the depth readings of a time-of-flight camera with semi-transparent objects present a systematic distortion and that the distorted depth values can be refined using an iterative optimization. Phillips and colleagues [159] use a stereo camera and exploit the fact that glass objects generate anomalies in the stereo inverse perspective map. Glass objects are assumed to be standing on a flat supporting plane. The plane needs to be somewhat textured to facilitate 2D homography estimation. Their method identifies extruding points from textured surfaces that violate the inverse perspective mapping, and use a dataset of 3D models to generate shape templates for detailed localization. In particular, they use a similarity score that maximizes the homography inconsistency inside the shape template while minimizing the inconsistency in the neighborhood around the template. Wallace and Csakany [209] develop a time-of-flight laser sensor based on photon counts to measure 3D data from transparent surfaces. Liu et al. [116] propose a frequency-based 3D reconstruction method, which incorporates a frequency-based matting method that is similar to structured light methods. Ma et al. [125] derive a formulation of light

transport in refractive media using light fields and the transport of intensity equation. Ye et al. [227] augment a Kinect camera with an ultra-sonic sensor that is able to measure distance to any object, including transparent surfaces. Xu et al. [222] use linearity in light-field images to estimate the likelihood of a pixel belonging to a transparent object or a Lambertian background. Lei et al. [108] use a LIDAR device along with a registered RGB camera for glass object segmentation. Object candidates are proposed by highlight spots in RGB images and refined by running GrabCut [173] on depth and laser reflectance intensity images. In addition, when viewpoint is fixed, Han et al. [67] develop an approach for dense transparent surface reconstruction based on refraction of light.

The closest to our work is from Lysenkov, et al. [123] in the sense that they also use an RGBD camera for glass object detection and pose estimation. They propose a model taking into account both silhouette and surface edges, and perform CAD-based pose estimation. An extension to this work from the same group [124] focuses on pose estimation in transparent clutter. Another extension proposed by Luo et al. [122] improves the method by integrating visual cues so that non-transparent objects that produce unknown depth values would not be considered as transparent objects. However, these methods require 3D models of objects obtained by covering transparent objects with paint, in order to make their surface Lambertian. In our work, we wanted to make our method more flexible with unseen objects and avoid using strong shape priors. Albrecht and Marsland [4] also propose a detection and reconstruction method for glass objects from point cloud data. Their method utilizes the shadows in RGBD images that are left in two or more distinct viewpoints to facilitate reconstruction. In our work, however, we are interested in glass object segmentation from a single viewpoint.

2.3 Boosting for learning from sparsely labeled data

So far we focused on two paradigms for localizing objects in computer vision, i.e., object detection and segmentation. Common to both problems is the need for a classification model that distinguishes image features between object and non-object. The training process of these classification models requires annotation that can be expensive to obtain for large datasets. For example, detailed object ground-truth annotation, usually being a segmentation mask, can be laborious to create manually. Therefore, it would be advantageous if we can relax the labeling requirements by assuming only partial or coarse annotation is available.

More generally, classification is the problem of assigning a class (or label) to a new observation, on the basis of a set of training data. The resulting model is commonly referred to as a classifier. A classification problem is *supervised* if the class membership of observations in the training set is known, or *unsupervised* otherwise. We call the training data *labeled* or *unlabeled*, respectively. Due to the annotation availability issue discussed before, in this thesis we focus on the *semi-supervised* classification problem where only partial class membership

information in the training set is available. Semi-supervised classification studies the problem of using both labeled and unlabeled data to learn a classifier. As we will show in Chapter 6, in some practical applications including object segmentation, semi-supervised classifiers achieve a level of performance comparable to fully supervised classifiers, therefore either reduce the amount of required annotation or eliminate the need for detailed annotation.

There has been a large amount of literature in semi-supervised learning and we refer the readers to the recent book [31] for a comprehensive review. Generally, semi-supervised learning methods can be categorized into either *transductive* or *inductive* based on the nature of inference. Transductive algorithms can only predict labels of data seen during training. Typical approaches include label propagation [238] and LLGC [236]. The goal of transductive learning is to predict labels for an observed and unlabeled transduction set, and the algorithm commonly makes use of the geometric properties of the data distribution. More specifically, many transductive learning algorithms are based on the *manifold assumption* which assumes that data lie in a low-dimensional manifold in a (high-dimensional) input feature space. The geometry of the data distribution can be captured by representing the dataset as a graph, with data points as vertices and pairwise similarities between data points as edge weights. Inductive methods, on the other hand, build a general decision rule over the input feature space and therefore can be used to predict the labels of data that are unseen during training. Examples of inductive methods include co-training [17] and semi-supervised SVM [12]. One of the most widely used underlying ideas in these methods is the *cluster assumption* which assumes that decision boundaries are more likely to pass through regions in the feature space with lower data density. It should be noted, however, although the manifold assumption is inherently transductive, we can also use it to regularize decision boundaries in inductive methods. For example, manifold regularization [11] adds a data-dependent geometric regularization term to the objective function of a max-margin classifier (e.g., an SVM). Our work in this thesis belongs to the inductive category and is inspired by this manifold regularization idea. Specifically, our method is based on the manifold assumption in Laplacian Eigenmaps [10].

Many classification algorithms are commonly used in the computer vision literature. This includes decision trees, ensemble learning (e.g., boosting and random forest), k-nearest neighbors, SVMs, to name a few [15]. In our work, we choose to make use of the boosting classification framework and, more specifically, extend the margin distribution boosting (MDBoost) algorithm [182] to support semi-supervised learning based on manifold regularization. We choose the boosting framework because the max-margin nature of boosting algorithms makes it straightforward to introduce manifold regularization for semi-supervised learning and induce an inductive learning algorithm. More importantly, the geometry of the (labeled and unlabeled) data distributions can be assimilated into the margin-cost based objective function. As a result, the algorithm can be efficiently and incrementally trained using column generation, thus retains the stage-wise gradient descent training procedure. This is in contrast to methods such as the

semi-supervised SVM [12] that involves solving a computationally expensive mixed integer program for the semi-supervised case.

Several works have extended supervised boosting algorithms to a semi-supervised setting. Semi-supervised MarginBoost [28] generalizes the margin concept to unlabeled data, and minimizes a margin-based loss by functional gradient descent. Chen and Wang also minimize the margin-based loss and introduce additional local smoothness into regularization in the Regularized Boost [33]. SERBoost [175] aims to scale up to large datasets by using expectation regularization. In ASSEMBLE [13] and SemiBoost [131], authors introduce the notion of pseudo-labels for unlabeled data and boost any supervised classifier by iteratively relabeling the unlabeled data. Unlike those existing approaches, the algorithm proposed in this thesis optimizes the margin distribution directly within a totally corrective framework, while incorporating manifold regularization on both labeled and unlabeled data coherently.

For completeness, we briefly review the AdaBoost and MDBoost algorithms below.

AdaBoost. AdaBoost is the first and most commonly used variant of boosting alogrithms [207]. Mathematically, let $\mathcal{D}_l = \{(\mathbf{x}_i, y_i)\}_{i=1,\cdots,M}$ be the training data set, where $\mathbf{x}_i \in \mathcal{X}$ is the input feature vector and $y_i \in \{-1, +1\}$ is the output label. Given the training data, our goal is to train a classifier to assign a binary label to any input vector $\mathbf{x}$. In the setting of boosting methods, the classifier consists of a weighted combination of weak learners (classifiers).

More specifically, denote $h(\cdot) \in \mathcal{H}$ as a weak learner that maps an input vector $\mathbf{x}$ into a binary output. We assume that we choose K weak learners from the set $\mathcal{H}$ in our boosted classifier, and define a matrix $H \in \mathbb{Z}^{M\times K}$ to be all the possible predictions of the training data using weak learners. That is, $H_{ij} = h_j(\mathbf{x}_i)$ is the label ($\{+1, -1\}$) given by the weak learner $h_j(\cdot)$ on the training example $\mathbf{x}_i$. We also use $H_{i:} = [H_{i1}\ H_{i2} \cdots H_{iK}]$ to denote the i-th row of H, which constitutes the output of all the weak learners on the training example $\mathbf{x}_i$. Let α be the weight vector for the weak learners. We can write the output of the final classifier on any training data $\mathbf{x}_i$ as $H_{i:}\alpha$, and the so-called (unnormalized) *margin* at data $\mathbf{x}_i$ is defined as $y_i H_{i:}\alpha$.

AdaBoost can be viewed as a gradient descent procedure that minimizes the exponential classification error (or loss) function. The training procedure of AdaBoost is a greedy algorithm that constructs an additive combination of weak classifiers such that the following exponential loss is minimized [36]:

$$L(y, f(\mathbf{x})) = \exp\big(-yH(\mathbf{x})\big). \tag{2.27}$$

where

$$H(\mathbf{x}) = \mathbf{sign}\big(\sum\nolimits_{i=1}^{N} \alpha_i h_i(\mathbf{x})\big), \tag{2.28}$$

Here α_i is the weight coefficient for the i-th weak learner, and N is the number of weak learners.

Margin theory and MDBoost. One way of deciphering the success of boosting lies in margin theory [178]. Several papers, such as LPBoost [39], adopt the minimum margin as an alternative learning criterion for boosting. Ryyzin and Schapire [168] point out that the generalization performance of boosting algorithms may depend more on the margin distribution instead of the minimum margin. Based on this observation, Shen and Li propose MDBoost and achieved promising classification performance by directly maximizing the average margin and minimizing the margin variance [182].

Specifically, let ρ_i denote the unnormalized margin for the i-th example datum, *i.e.*, $\rho_i = y_i H_{i:}\alpha, \forall i = 1, \cdots, M$. The cost function and the learning problem in MDBoost can be written as follows:

$$\begin{aligned} \min_{\alpha} \ & \frac{1}{2(M-1)} \sum_{i>j} (\rho_i - \rho_j)^2 - \sum_{i=1}^{M} \rho_i \\ \text{s.t.} \ & \alpha \succcurlyeq 0, 1^\top \alpha = D, \end{aligned} \tag{2.29}$$

where D is a regularization parameter. By defining a matrix $A \in \mathbb{R}^{M \times M}$, where

$$A = \begin{bmatrix} 1 & -\frac{1}{M-1} & \cdots & -\frac{1}{M-1} \\ -\frac{1}{M-1} & 1 & \cdots & -\frac{1}{M-1} \\ \vdots & \vdots & \ddots & \vdots \\ -\frac{1}{M-1} & -\frac{1}{M-1} & \cdots & 1 \end{bmatrix},$$

the optimization problem can be rewritten into the following form:

$$\begin{aligned} \min_{\alpha} \ & \tfrac{1}{2}\rho^\top A \rho - 1^\top \rho, \\ \text{s.t.} \ & \alpha \succcurlyeq 0, 1^\top \alpha = D, \\ & \rho_i = y_i H_{i:}\alpha, \forall i = 1, \cdots, M. \end{aligned} \tag{2.30}$$

It has been shown [183] the problem in (6.2) can be efficiently solved by considering its dual form, i.e.,

$$\begin{aligned} \min_{r,u} \ & r + \tfrac{1}{2D}(u-1)^\top A^{-1}(u-1), \\ & \text{s.t.} \sum_{i=1}^{M} u_i y_i H_{i:} \preccurlyeq r 1^\top. \end{aligned} \tag{2.31}$$

The form of the dual problem allows us to incrementally search the solution space by the column generation technique. At each iteration, we obtain a new weak classifier through searching

the most violated constraint:

$$h'(\cdot) = \underset{h(\cdot)}{\operatorname{argmax}} \sum_{i=1}^{M} u_i y_i h(\mathbf{x}_i). \tag{2.32}$$

While the MDBoost learning cost incorporates the margin variance information, the global variance can be restrictive and cannot describe the finer structure of the distribution beyond the second order statistics. In our work, we propose to use the "local" version of variance that considers the geometric properties of the data manifold. More importantly, the idea that we can make use of the geometric properties of the data distribution can be naturally extended to a semi-supervised learning setting. In Chapter 6, we propose the Semi-supervised Laplacian MDBoost algorithm that addresses the above shortcomings of MDBoost. In addition, we apply the new semi-supervised learning algorithm on a number of object segmentation tasks to verify its efficacy.

2.4 Summary

Object detection and segmentation have wide application in computer vision and robotics. For object detection, our task is to infer a bounding box-based parametrization of an object hypothesis. We reviewed two broad groups of methods based on sliding windows and the Hough transform respectively. Most importantly, the availability of RGBD data allows depth information to be incorporated both in terms of feature engineering and model design. Our focus in this thesis is to build an object detection system with better context and occlusion reasoning made possible by the addition of depth data. In particular, due to the limited availability of RGBD data compared to RGB imagery, we are interested in the scenario where depth data are only available during model training.

For object segmentation, our task is to infer a pixelwise foreground object mask. We reviewed relevant methods with a focus on those based on MRFs. In addition, we discussed two main issues in MRF-based object segmentation: context modeling and inference. The focus of our work in this thesis is the glass object segmentation problem, therefore we then discussed related work in the literature. Our work is among the first to leverage the additional depth data and the partial depth readings caused by irregular refractive properties of the glass surface. Also, to the best of our knowledge, we are the first to explore nonparametric label transfer for glass object segmentation.

Finally, we reviewed work on semi-supervised learning and boosting algorithms. We showed that boosting algorithms are an essential component of many object detection and segmentation systems. In addition, we revisited the MDBoost algorithm that directly optimizes the margin distribution. Its formulation provides us the flexibility to incorporate manifold regularization and to extend the algorithm to a semi-supervised learning scenario.

Despite the progress discussed in this chapter, many object detection and segmentation models have certain limitations when only partial information is available during either the model training or testing stage. Three main issues remain, although the auxiliary depth information provides promising outlook for resolving these limitations. The issues are partial object observation, incomplete and imperfect data modalities, and partial ground-truth annotation. A key problem here is depth-aware context modeling in the presence of occlusion and under varying levels of depth information availability. In this thesis, we are interested in utilizing auxiliary depth information to model the spatial context for localizing both generic and glass objects. Particularly, glass objects exhibit large appearance variations and depth information obtained with RGBD cameras can be noisy and incomplete near glass boundaries. In addition, it is important to incorporate unlabeled data for object detection and segmentation when precise and complete ground-truth annotations are expensive to obtain. This thesis proposes a series of context-driven object detection and segmentation approaches to address these issues.

Chapter 3

Structured Hough Voting for Joint Object Detection and Occlusion Prediction

3.1 Introduction

Object detection remains a challenging task for cluttered/crowded scenes, such as indoor environments, where objects are frequently occluded by neighboring objects or the viewing window [53, 206]. The partial objects being observed usually provide limited information on the object position and pose, so many previous object detection approaches are prone to failure as they solely rely on image cues from objects themselves.

It is widely acknowledged that contextual information plays an important role in detecting and localizing objects in such adverse conditions. Many context-aware object detection methods have been proposed recently [219, 201, 127, 16]. However, most existing contextual models focus on 2D spatial relationships between objects on the image plane and fewer works have extended the modeling to 3D scenarios [8, 193]. One main difficulty in modeling 3D context was the lack of accessible 3D data. With recent progress in consumer-level depth sensors (e.g., Kinect), however, it becomes feasible to collect a large amount of high quality depth and registered color images for indoor environments [77, 145].

Modeling context from a 3D perspective has several advantages over its 2D counterpart conceptually. Firstly, spatial relationships have smaller variations and are easier to interpret semantically; in addition, more spatial relationships in physical world can be captured, instead of being limited to relative positions on the image plane. In particular, occlusion can be viewed as a special type of contextual relationship in 3D, which would become an intrinsic component of object and scene models. Finally, joint modeling of an object class and its 3D context may provide effective constraints on the object's scope on the image plane and lead to a coarse-level object segmentation. See Figure 3.1 for an example.

Our work aims to utilize RGBD datasets to learn a context-aware object detection model

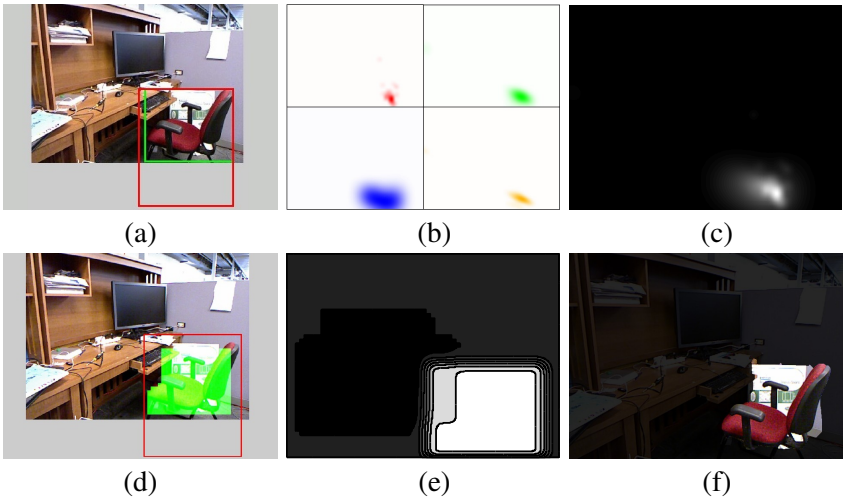


Figure 3.1: Illustration of structured Hough voting. **(a)** RGB frame with object bounding box (red) and visible part bounding box (green). **(b)** Object centroid voting from multiple layers. **(c)** Combined object centroid voting results. **(d)** Detector output (red) with visibility pattern prediction (green). **(e)** Object visibility pattern prediction results. **(f)** Final segmentation results.

which encodes depth cues and a coarse level of 3D relationships. We focus on training a depth-dependent appearance model for each object class and its context. The learned depth-encoded object and context model is then applied to 2D images during test so it can be used to facilitate generic object detection [195].

Specifically, we propose a structured Hough voting method that incorporates depth-dependent contexts into a codebook-based object detection model. Our model generalizes the traditional Hough voting detection methods in three ways. First, we design a multi-layer representation of *image context* for indoor scenes that captures the layout structure of scenes. An image region contributes to each object hypothesis in a different manner based on its depth layer. Secondly, we define a new object hypothesis space in which both the object's center and its visibility mask will be predicted. Each image patch will generate a weighted vote to a joint score of the object center and its support mask in the image. Finally, we view occlusion as special contextual information, which could provide cues for localizing objects and help with reasoning about visibility of object parts. The overall output of our approach is a simultaneous object detection and coarse segmentation.

Our detection and segmentation are achieved by maximizing the joint score of object center and visibility mask. We derive an efficient alternating ascent method to search modes of the Hough voting score maps. To learn the model from partially labeled RGBD data, we adopt an approximate learning procedure based on the max-margin Hough transform [129]. We evaluate our approach on two public RGBD datasets and demonstrate its efficiency.

The remainder of this chapter is organized as follows. The details of our model structure

are introduced in Section 3.2. Section 3.3 describes the inference procedure in our structured Hough voting, followed by the max-margin learning for model estimation. Details on experimental evaluation are reported in Section 3.4 and Section 3.5 summarizes this chapter.

3.2 Our approach

Figure 3.2: Top-ranked clusters (presented with the patches closest to the cluster centers) for 3 contextual layers on the Berkeley 3D object dataset.

3.2.1 Structured Hough voting

We first briefly review the original Hough voting based object detection method and introduce notation. Hough voting methods (e.g., [109, 52]) generally use object poses as their hypothesis, accumulate scores from each image patch into a confidence map for the hypothesis space, and search for the highest voting scores from the map [7].

Mathematically, suppose we have an image I and an object class of interest o. Let the object hypothesis be $\mathbf{x}_s \in \mathcal{X}$, where $\mathcal{X}$ is the object pose space. To simplify the notation, we assume each hypothesis is $\mathbf{x}_s = (\mathbf{x}, a_s)$ where $\mathbf{x} = (a_x, a_y)$ is the image coordinate location of the object center and a_s is a scale. At a specific object scale a_s, Hough voting methods define a scoring function $S(\mathbf{x})$ for each valid location $\mathbf{x}$ on the image plane, which is a summation of

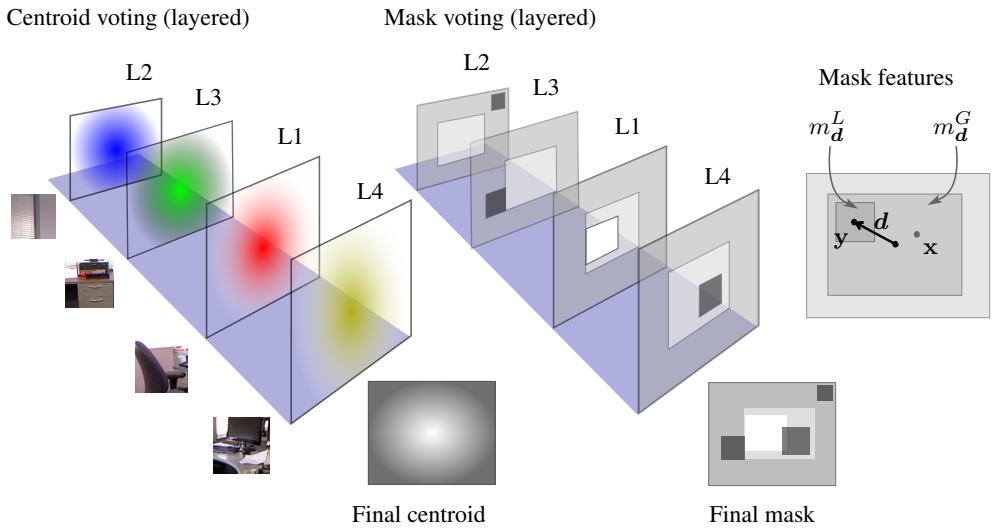


Figure 3.3: Illustration of multiple layered object centroid and mask voting. L1 corresponds to the object layer, and L2, L3, L4 correspond to far-away context, close-up context and occluder layers, respectively. For mask voting, brighter regions indicate a higher response, while darker regions indicate a lower response.

weighted votes from every local image patch. To compute the voting weights, an appearance-based codebook is usually learned from the image patches in object class o, denoted by $\mathcal{C} = \{C_i\}_{i=1}^K$. Each codebook entry C_i consists of a typical patch descriptor $\mathbf{f}_{ci}$ and geometric features D_i of training patches associated with the i-th entry. A typical geometric feature is the relative positions $\boldsymbol{d} = (d_x, d_y)$ of image patches w.r.t. the corresponding object centers.

Given the codebook $\mathcal{C}$, we can write the Hough score function as follows. Denote each image patch $I_{\mathbf{y}}$ by its location $\mathbf{y} = (b_x, b_y)$ and feature descriptor $\mathbf{f}_{\mathbf{y}}$,

$$S(\mathbf{x}) \propto \sum_{i=1}^{K} \sum_{\mathbf{y}} \omega_i p(C_i|\mathbf{y}) \sum_{\boldsymbol{d} \in D_i} e^{\left(-\frac{\|(\mathbf{y}-\mathbf{x})-\boldsymbol{d}\|^2}{2\sigma_d^2}\right)} \tag{3.1}$$

where $\omega_i = p(o|C_i)$ is the entry-to-class probability, $p(C_i|\mathbf{y})$ is the patch-to-entry matching probability, and σ_d is the standard deviation of a Gaussian filter for the object center. Notice that the object center $\mathbf{x}$ essentially specifies a bounding box. However, the bounding box hypothesis space is limited in its representation power as it is incapable of describing partial objects or visibility patterns.

We propose to extend the object hypothesis space from a single centroid $\mathbf{x}$ to a joint space $(\mathbf{x}, v)$ and define a new score function $S(\mathbf{x}, v)$. Here $\mathbf{x}$ specifies the object center (or equivalently its bounding box), and v is a visibility mask indicating which part of the object is visible, as shown in Figure 3.3. The mask v has the same size as the image I, and $v(\mathbf{y}) = 1$ if the image patch at $\mathbf{y}$ belongs to the object o, and 0 otherwise. For notation simplicity, we reshape v as an

1-D vector and denote its element at image location $\mathbf{y}$ as $\boldsymbol{v}_{\mathbf{y}}$.

Our key step is, instead of using Gaussian kernels in Equation 3.1, we introduce a class of voting masks that are capable of representing the relative positions as well as the object visibility pattern. As illustrated in the rightmost figure in Figure 3.3, we include a local mask and a global mask for each codebook entry. The local mask predicts if a local patch itself is part of the object, and the global mask casts a vote for the spatial extent of the whole object on the image plane based on the relative geometric feature $\boldsymbol{d}$.

Formally, each codebook entry C_i includes a new set of geometric features $\tilde{D}_i = \{\tilde{\boldsymbol{d}} = (\boldsymbol{d}, m^L_{\boldsymbol{d}}, m^G_{\boldsymbol{d}})\}$, where $m^L_{\boldsymbol{d}}$ is the local mask feature and $m^G_{\boldsymbol{d}}$ is the global mask feature. The local mask features describe local visibility of object regions, which is similar to the ISM [109]. The global mask features limit the scope of each object on the image plane. A natural choice is an object bounding box-shaped mask as illustrated in Figure 3.3. Note that by choosing a different family of mask features, our model allows for finer description of the object shape and/or visibility patterns.

For an image patch at $I_{\mathbf{y}}$ and object center hypothesis $\mathbf{x}$, we can compute two average voting masks from the i-th codebook entry as follows:

$$m^G_i(\mathbf{x},\mathbf{y}) \propto \sum_{\tilde{\boldsymbol{d}} \in \tilde{D}_i} m^G_{\boldsymbol{d}}(\mathbf{x}-\mathbf{y}+\boldsymbol{d}) * G(0,\sigma_d^2) \tag{3.2}$$

$$m^L_i(\mathbf{x},\mathbf{y}) \propto \sum_{\tilde{\boldsymbol{d}} \in \tilde{D}_i} m^L_{\boldsymbol{d}}(\mathbf{x}-\mathbf{y}) * G(0,\sigma_d^2) \tag{3.3}$$

where m^G and m^L are the average global and local voting mask, respectively; $m(\mathbf{x})$ represents the mask with its center shifted to $\mathbf{x}$, $G(\cdot)$ is the Gaussian kernel, and $*$ is the convolution operator. See Figure 3.3 for an illustration.

We define the new score function as a matching score between the visibility mask hypothesis v and a weighted sum of the voting mask values,

$$S(\mathbf{x}, v) = \sum_{i=1}^{K} \omega_i v^T \Big[\sum_{\mathbf{y}} \gamma(v(\mathbf{y})) \Big(m^G_i(\mathbf{x},\mathbf{y}) + \mu m^L_i(\mathbf{x},\mathbf{y}) \Big) p(C_i|\mathbf{y}) - w_b \Big] \tag{3.4}$$

where w_b is a global bias to the mask voting score, and μ is the relative weight of the local mask. $\gamma(u)$ is a weighting function with $\gamma(1) = 1$ and $\gamma(0) = \delta, \delta < 1$. Intuitively, we give a smaller weight to the votes that arise from features not on the object. ω_i gives a relative weight for each codebook entry. It can be shown that when $v = 1$, $\mu = 0$ and the global voting mask has the shape of an object bounding box, the new score function is equivalent to the Hough voting score in Equation 3.1.

3.2.2 Depth-encoded context

The structured Hough voting model can easily incorporate image contextual information by extending the codebook and including votes from both object and context patches. In this work, we design a multi-layer scene representation that captures different types of image cues for detection and integrates them into the model. The overall object model does not have a 3D or 2.5D point cloud like representation; it is a 2.1D (i.e., multiple layers) object-centric model. However, our model does encode 3D depth information as we discuss below.

Concretely, we group image patches into four layers according to their relationship with the target object: 1) An *object layer* which includes all the image patches from the object itself; 2) An *occluder layer* which has the patches occluding the object; 3) A *nearby context layer* which consists of the context patches within 1 meter of the average object depth; 4) A *far-away context layer* that has the rest of the context image patches.

We associate each layer with its own specific parameters as they contribute to object detection and occlusion prediction in different ways. We first learn a separate codebook-based appearance model for each layer using object labels and depth cues. Denote the i-th codebook entry of layer l as C_i^l, we define a context-aware structured Hough voting model by including the votes from all the layers:

$$S_c(\mathbf{x}, v) = \sum_{l=1}^{4} \sum_{i=1}^{K_l} \omega_i^l v^T \Big[\sum_{\mathbf{y}} \gamma(v(\mathbf{y})) \Big(m_{l,i}^G(\mathbf{x}, \mathbf{y}) + \mu^l m_{l,i}^L(\mathbf{x}, \mathbf{y}) \Big) p(C_i^l | \mathbf{y}) - w_b^l \Big] \tag{3.5}$$

where K_l is the size of the codebook in layer l. Note that each layer has its own Gaussian kernel width σ_d^l in the voting masks. The details of each layer are as follows.

A. Depth-encoded codebooks. We use HOG features [46] for image patches on the target object and Texton like [185] features for patches from context layers. In particular, we use the filter bank, color and HOG textons obtained with the implementation from [90]. The initial codebooks are generated by K-means clustering of randomly sampled patches. To capture discriminative patches, we also use an interest point detector to sub-sample the patch pool. The Texton feature, which is a coarser level descriptor, is better for capturing context in a scene. Some examples of image patches in our codebooks are shown in Figure 3.3. We can see that different types of scene structure are captured. We further refine the initial codebooks by utilizing depth information available during training. Specifically, we rank each cluster in each layer by its 3D offset variance, and prune out those ranked in the last 25%.

B. Layer-dependent voting masks. We design the global mask feature $m_{\boldsymbol{d}}^G$ and local mask feature $m_{\boldsymbol{d}}^L$ according to the properties of each layer. In this work, all the global masks have the same shape as the object bounding box. Thus all active patches contribute to limiting the scope

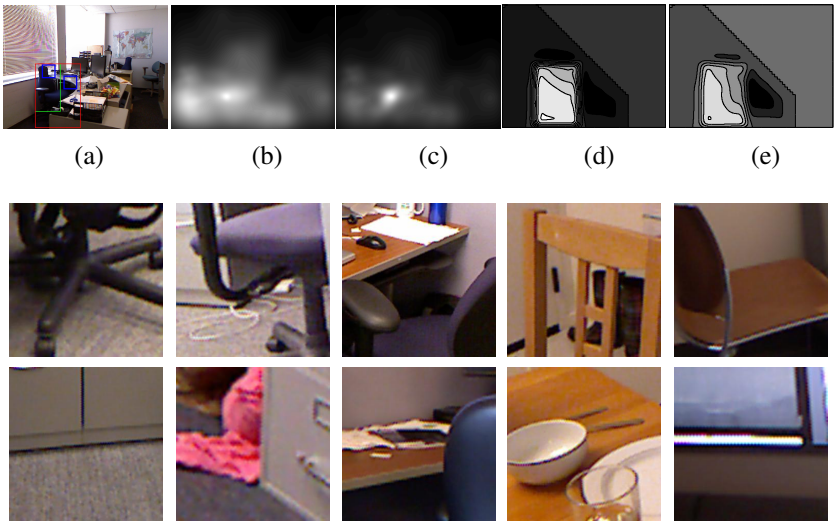


Figure 3.4: Illustration of the impact of patch pair terms on hypothesis scoring. **Upper panel**: A specific example, with **(a)** RGB frame with an example of a patch pair (in blue rectangles). **(b)** Object centroid voting results without patch pair terms. **(c)** Object centroid voting results with patch pair terms added. **(d)** Shape voting results without patch pair terms. **(e)** Shape voting results with patch pair terms added. **Lower panel**: The highest ranked patch pairs on the Berkeley 3D object dataset. The first row shows on-object patches, and the second row shows off-object patches. Each column corresponds to a patch pair.

of the object. For the local masks, the object layer has a positive 2D stump with $1/10$th of the object size, while other layers have a negative 2D stump with the same size. Intuitively, the active image patches from context layers help localize the object center but also indicate the local patches that do not belong to the object. In addition, we set the Gaussian blur parameter σ_d^l such that the far away context layer has larger variances in terms of center prediction (3 times).

3.2.2.1 Second-order features

In addition to layered codebooks, which are built on single patches, we utilize patch feature pairs to improve the discriminative power of the model [235]. In particular, we focus on co-occurring object and contextual feature pairs. These feature pairs can refine the context relationship and better predict the object boundary.

We incorporate the object-context feature pairs into our structured Hough voting model by adding a second-order term to the score function: $S(\mathbf{x}, v) = S_c(\mathbf{x}, v) + \alpha S_p(\mathbf{x}, v)$, where α is the relative weight, and S_p is the object-context feature pair term. Assume the first layer $l = 1$

Algorithm 1: Alternating Inference for $S(\mathbf{x}, v)$.

Input: Input Image I; Layered Codebooks $\mathcal{C} = \{C_i\}, i = 1 \cdots N_L$; Offsets D_i; Mask templates $m_d(y), m'_d(y), \forall d \in D_i$; Entry weights $\{\omega_i^l, \mu_j^l, \omega_{ij}^l, \mu_{ij}^l\}$; Model parameters $\tau, \alpha, \delta, \kappa$; Local maxima seeds N_{seed}; termination threshold $\varepsilon > 0$; Maximum iterations $T_{\max}$.

Initialization: Let $v = 1$, search for N_{seed} local maxima for $S(\mathbf{x}, 1)$: $\mathbf{x}_i, i = 1 \cdots N_{\text{seed}}$.

for each local maxima x_i **do**

  for iteration $= 1 : T_{\max}$ **do**

    1. Obtain a new v_i^* by solving Equation 3.8;
    2. Optimal solution check:
       if $S(\mathbf{x}_i, v_i) - S(\mathbf{x}_i, v_i^*) < \varepsilon$,
       then break and the problem is solved;
    3. $v \leftarrow v_i^*$, vote again for x_i^* with v_i, $\mathbf{x}_i \leftarrow \mathbf{x}_i^*$.

  end

  Mask Recalculation: Obtain a new v_i^* by solving Equation 3.8, $v \leftarrow v^*$.

end

Output: $argmax_{(\mathbf{x}_i, v_i)} S(\mathbf{x}_i, v_i)$

is the object layer, S_p can be written as

$$S_p(\mathbf{x}, v) = \sum_{i=1}^{K_1} \sum_{l=2}^{4} \sum_{j=1}^{K_l} \omega_{ij}^l v^T \Big[\sum_{\mathbf{y}, \mathbf{y}'} \gamma(v(\mathbf{y})) \Big(m_{1,i}^G \odot m_{l,j}^G + \mu^l m_{1,i}^L \oplus m_{l,j}^L \Big) \cdot \varphi - w_b^{1,l} \Big] \tag{3.6}$$

where $\odot$ and $\oplus$ are the element-wise product and addition operators, respectively. We omit the variables $(\mathbf{x}, \mathbf{y})$ in m for clarity of the notation. ω_{ij}^l is the weight for the object-context codebook entry pairs. The patch pair to entry matching probability $\varphi = p(C_j^l | C_i^1) p(C_i^1 | \mathbf{y}) p(C_j^l | \mathbf{y}')$ and $p(C_j^l | C_i^1)$ is estimated by the feature co-occurrence frequency matrix during training. We also use depth information to prune out geometrically unstable or inconsistent codebook pairs as in the previous subsection.

3.3 Model learning and inference

3.3.1 Joint inference for object detection and occlusion prediction

Once the structured Hough voting model is trained with depth-augmented image data, we can apply it to 2D images for object detection and occlusion prediction. Our method infers the object center hypothesis and its visibility mask by maximizing the Hough score function $S(\mathbf{x}, v)$.

However, due to the large hypothesis space of $(\mathbf{x}, v)$, it is difficult to use the original Hough voting approach, or conduct a brute-force search. In this section, we propose a coordinate-ascent method which finds the local maxima of the score function.

Specifically, we alternatively maximize the score function with respect to one variable, while keeping the other fixed. When v is fixed, the optimization is the same as the original Hough voting. We only need to carry out a weighted Hough voting step and the local maxima $\mathbf{x}_i^*$ can be retrieved from the Hough map. When the object center is fixed, our Hough score is a quadratic function of the binary vector v. To convert $S(\mathbf{x}, v)$ into its quadratic form, we notice that $\gamma(v(\mathbf{y})) = (1-\delta)v(\mathbf{y}) + \delta$. So we can write the first term (i.e., the global mask term) in Equation 3.5 as

$$\begin{aligned} S_{c1}(x,v) = &\sum_{l=1}^{4}\sum_{i=1}^{K_l}\omega_i^l v^T \\ &\left[\sum_{\mathbf{y}}\Big((1-\delta)v(\mathbf{y})m_{l,i}^G(\mathbf{x},\mathbf{y})p(C_i^l|\mathbf{y}) \right. \\ &\left. + \delta m_{l,i}^G(\mathbf{x},\mathbf{y})p(C_i^l|\mathbf{y}) - w_b^l\Big)\right] \\ = &\sum_{l=1}^{4}\sum_{i=1}^{K_l}\left[v^T\Big(\omega_i^l\sum_{\mathbf{y}}(1-\delta)m_{l,i}^G(\mathbf{x},\mathbf{y})p(C_i^l|\mathbf{y})\Big)v \right. \\ &\left. + v^T\Big(\omega_i^l\sum_{\mathbf{y}}\delta m_{l,i}^G(\mathbf{x},\mathbf{y})p(C_i^l|\mathbf{y}) - w_b^l\Big)\right] \end{aligned} \tag{3.7}$$

The other terms in Equations 3.5 and 3.6 can be written in this form similarly. Summing those terms together, we have the following overall scoring function:

$$S(x,v) = v^T A(\mathbf{x})v + v^T B(\mathbf{x}) \tag{3.8}$$

where

$$A(\mathbf{x}) = \begin{bmatrix} \omega_i^l\sum_{\mathbf{y}}(1-\delta)m_{l,i}^G(\mathbf{x},\mathbf{y})p(C_i^l|\mathbf{y}) \\ \vdots \\ \mu_i^l\sum_{\mathbf{y}}(1-\delta)m_{l,i}^L(\mathbf{x},\mathbf{y})p(C_i^l|\mathbf{y}) \\ \vdots \\ \omega_{ij}^l\sum_{\mathbf{y}}(1-\delta)(m_{1,i}^G\odot m_{l,j}^G)\cdot\varphi \\ \vdots \\ \mu_{ij}^l\sum_{\mathbf{y}}(1-\delta)(m_{1,i}^L\oplus m_{l,j}^L)\cdot\varphi \\ \vdots \end{bmatrix}, \tag{3.9}$$

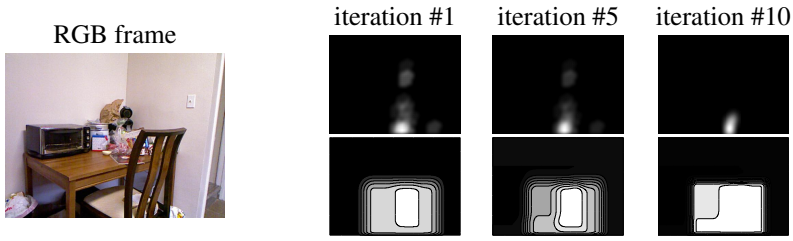


Figure 3.5: An illustration of how iterative inference updates the object centroid and supporting mask hypotheses. The first row on the right shows object centroid voting, with the corresponding supporting mask estimations shown in the second row.

$$B(\mathbf{x}) = \begin{bmatrix} \omega_i^l \sum_{\mathbf{y}} \delta m_{l,i}^G(\mathbf{x},\mathbf{y}) p(C_i^l|\mathbf{y}) - w_b^l \\ \vdots \\ \mu_i^l \sum_{\mathbf{y}} \delta m_{l,i}^L(\mathbf{x},\mathbf{y}) p(C_i^l|\mathbf{y}) - w_b^l \\ \vdots \\ \omega_{ij}^l \sum_{\mathbf{y}} \delta (m_{1,i}^G \odot m_{l,j}^G) \cdot \varphi - w_b^{1,l} \\ \vdots \\ \mu_{ij}^l \sum_{\mathbf{y}} \delta (m_{1,i}^L \oplus m_{l,j}^L) \cdot \varphi - w_b^{1,l} \\ \vdots \end{bmatrix}, \tag{3.10}$$

where $\odot$ and $\oplus$ are the element-wise product and addition operators, respectively. Please refer to Equation 3.5 for the definition of the variables. We choose to solve a relaxed version of this problem by allowing $v(\mathbf{y}) \in [0, +1]$, which is a constrained quadratic programming problem. We find an approximate binary solution by searching for an optimal threshold to binarize the solution vector. Note that the constraint for the relaxed quadratic programming problem will enforce invisibility for any image location $\mathbf{y}$ outside the bounding box $\mathbf{x}$, i.e., $v(\mathbf{y}) = 0, \forall \mathbf{y} \notin \mathbf{x}$. This greatly reduces the search space.

The inference algorithm is summarized in Algorithm 1. It initializes the object center hypothesis with the original Hough voting method, and search for object hypotheses at multiple scales. Figure 3.5 shows the iterative inference process.

3.3.2 Learning with depth-augmented data

Our model in Equations 3.5 and 3.6 is linear in terms of its weight vector $\mathbf{w} = \{\omega_i^l, \mu_j^l, \omega_{ij}^l, l = 1, \cdots, 4, i, j = 1, \cdots, K^l\}$. We utilize the max-margin Hough transform [129] framework to train our codebook entry and entry pair weight parameters $\mathbf{w} = \{\omega_i^l, \mu_j^l, \omega_{ij}^l, \mu_{ij}^l\}$. During training, our scoring function $S(\mathbf{x}, v)$ can be interpreted as a weighted sum of $\mathbf{w}$ so it can be trained using the objective function of the max-margin formulation as follows

$$\begin{aligned} \min_{\mathbf{w},b,\xi} \ & \frac{1}{2}\mathbf{w}^T\mathbf{w} + C\sum_{i=1}^{T}\xi_i \\ \text{s.t. } & z_i(\mathbf{w}^T D_i + b) \geq 1 - \xi_i, \\ & \mathbf{w} \succcurlyeq 0, \xi_i \geq 0, \forall i = 1, 2, \ldots, T \end{aligned} \tag{3.11}$$

where z_i is the label of the i-th training sample, ξ_i is the corresponding slack variable, and D_i^T is the activation matrix for the i-th sample defined as

$$D_i^T = \begin{bmatrix} v^T\Big(\sum_{\mathbf{y}}\gamma(v(\mathbf{y}))m_{l,i}^G(\mathbf{x},\mathbf{y})p(C_i^l|\mathbf{y}) - w_b^l\Big) \\ \vdots \\ v^T\sum_{\mathbf{y}}\gamma(v(\mathbf{y}))m_{l,i}^L(\mathbf{x},\mathbf{y})p(C_i^l|\mathbf{y}) \\ \vdots \\ v^T\Big(\sum_{\mathbf{y}}\gamma(v(\mathbf{y}))(m_{1,i}^G \odot m_{l,j}^G)\cdot\varphi - w_b^{1,l}\Big) \\ \vdots \\ v^T\sum_{\mathbf{y}}\gamma(v(\mathbf{y}))(m_{1,i}^L \oplus m_{l,j}^L)\cdot\varphi \\ \vdots \end{bmatrix} \tag{3.12}$$

We assume only a coarse labeling of the visibility is available for positive training data. To speed up training, we generate a negative example set that consists of incorrect labelings obtained from applying a simple version of our model with uniform weights, i.e., $\mathbf{w} = 1$. For all the other model parameters, we use cross-validation to find their values using a held-out validation set.

3.4 Experimental evaluation

3.4.1 Dataset and setup

We evaluate the proposed structured Hough voting method on two challenging RGBD object datasets: the Berkeley 3D Object (B3DO) Dataset (Version 1) [77] and a subset of object classes on the NYU Depth Dataset (Version 2) [145]. B3DO contains 849 images taken in 75 different scenes, and 8 object categories.The NYU Depth dataset has a total of 1449 labeled images. As the dataset was originally designed for pixelwise scene segmentation, it contains many background classes (e.g., wall, ceiling) which are not suitable for our object representation. Therefore, we run experiments with only the following 5 categories: table, chair, door, bed and sofa. For both datasets, we follow the training, validation and testing split supplied with their respective versions. See Figure 3.6 for some qualitative detection results using our

approach.

As the labelings of visibility masks are expensive to obtain, we assume only coarse-level labels for our masks. Two bounding boxes are used: one for the whole object and the other for visible parts. Some examples of the ground truth labelings are shown in Figure 3.6(a) (more in Section 3.4.5). For evaluation of segmentation accuracies we also manually label the visibility ground-truth using polygons on the B3DO dataset.

3.4.2 Model details

For codebook generation, we randomly sample 200 patches per image from the visible part bounding box and generate 400 clusters for non-object patches using K-means, then rank them according to the patches' offset variance. We then prune these clusters by discarding clusters with 20 or less members, and discard again remaining clusters with ranking in the last 25%. For other layers (i.e., context and occluder), we sample 400 patches per image and generate 800 clusters as the appearance variability is larger with context and occluders. For these layers we follow a similar pruning process after a second round of clustering is performed as discussed in Section 3.2.

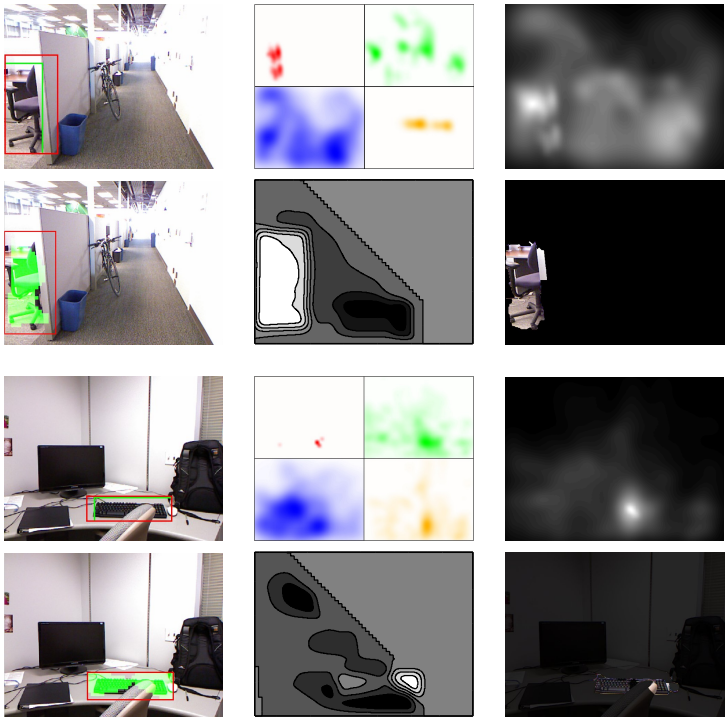
During test, we first prune down our search space for object hypotheses with edge boxes [242] as a pre-processing step. Afterwards, our detector searches for up to 100 local peaks in the Hough image with $v = 1$, and then runs a full version of inference and computes scoring functions for each of these peaks. Our alternate inference algorithm is likely to converge in a few iterations in most cases so we limit the maximum number of iterations to 20. The inference is efficient and complete detection takes around 5 seconds per image with a quad-core i7 desktop computer, using our paralleled MATLAB implementation.

After object location and the corresponding visibility mask are inferred, we run GrabCut [173] in the bounding box specified by $\mathbf{x}$ to generate a final segmentation mask to utilize bottom-up image cues and examine segmentation performance. Based on the shape voting results, we set regions with highest responses as foreground seeds and regions with lowest responses as background seeds, then run GrabCut for 10 iterations to get the final segmentation mask.

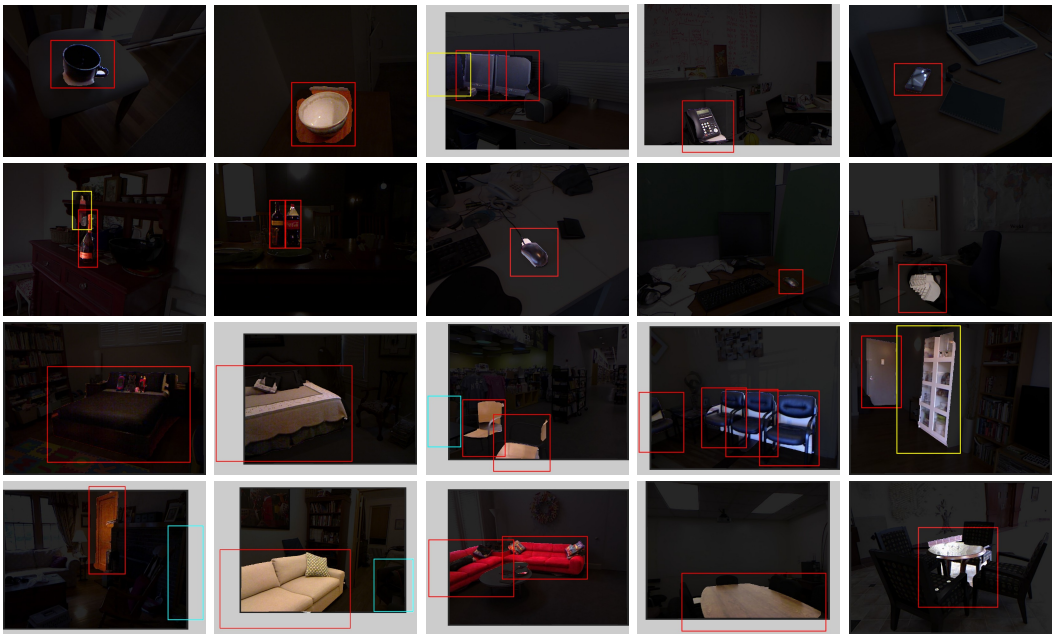
3.4.3 Quantitative results

In this section, we present quantitative evaluation results on the B3DO and NYU Depth datasets. Figure 3.7 reports the performance of our approach on the two datasets in comparison with three baseline methods.[1] Specifically, we compare our method with Deformable Parts Model

[1] Please note the results reported in our CVPR'13 paper [212] are not valid. There were errors in the experimental setup. Results obtained using the correct experimental setup are reported here.



(a) Detection examples with illustrations of intermediate steps. See the caption of Figure 3.1 for meanings of each step.



(b) More detection results in some challenging scenes. The red, yellow and cyan boxes indicate correct detections, false alarms and missing detections, respectively.

Figure 3.6: Detection examples of our approach. See text for details.

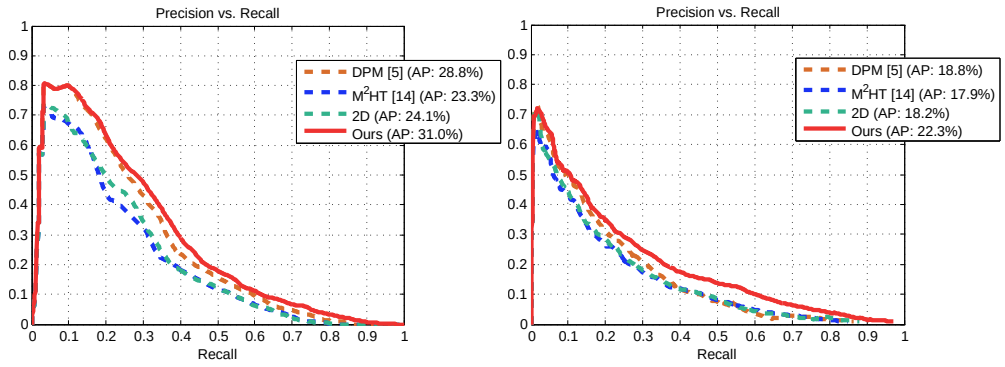


Figure 3.7: Detection precision-recall curves on the Berkeley 3D Object dataset (left) and the NYU Depth dataset (right). The solid curve corresponds to our approach (**Ours**). The dashed curves correspond to baseline methods: Deformable Parts Model (**DPM**) [46], Max-margin Hough transform (**M^{2}HT**) [129], and Max-margin Hough transform with 2D geometric context (**2D**). See details in text.

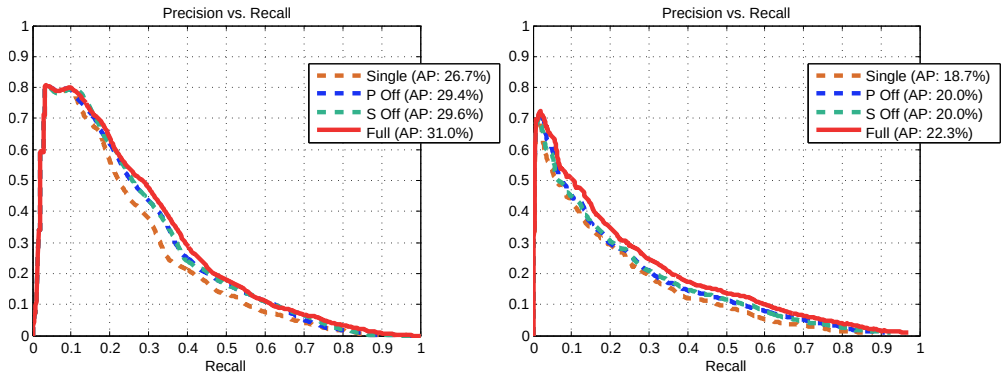


Figure 3.8: Detection precision-recall curves on the Berkeley 3D Object dataset (left) and the NYU Depth dataset (right). The solid curve corresponds to our full model (**Full**). The dashed curves correspond to diagnostic results with various components in our full model turned off, i.e., single layer context (**Single**), patch pair term off (**P Off**), and segmentation off (**S Off**). See details in text.

(DPM) [46] and max-margin Hough transform (M^{2}HT) [129]. Note that both baseline methods use 2D image cues only, without encoding contextual cues. Furthermore, we include a comparison with Hough voting using additional 2D geometric context, which uses 2D offsets only in generating a single-layered contextual codebook. For modeling the object itself with a depth-encoded codebook, we also tried M^{2}HT with a codebook learned with 3D offsets, which did not work well due to noisy labels of 3D object centers. It is clear from the results that our method outperforms all baselines on both datasets. For results on each object category, see Figures 3.11 and 3.12.

In addition, Figure 3.8 reports results from an ablation study on the contributions from three components in our approach. Specifically, we run three diagnostic tests with one of the

following components in our full model turned off: (1) multi-layer context (i.e., use single-layer context instead), (2) patch pair term (second-order features), and (3) segmentation (alternating inference). The resultant performance drops suggest all these components boost the performance of our approach. In particular, the multi-layer context has the largest impact. See Figures 3.13 and 3.14 for results on each object category. We note that on 11 out of 13 object categories, our full model performs better than without the three components in term of average precision.

Finally, Table 3.1 summarizes the per-class and mean average precision (mAP) values for all experiments above.

We also make the following observations in relation to the abovementioned results:

- For baseline results, DPM [46] outperforms M^2HT [129], on which our method is based. The latent training process based on discriminative learning allows DPM to more effectively capture object parts in the presence of heavy deformation and occlusion. We note that the object layer in our method may be detached from the rest of the model, and we may potentially improve over our current results by combining our context representation with more powerful object detectors.

- 2D geometric context contributes to baseline detection performance slightly on some object categories. Further with the depth-encoded context, the performance of our structured Hough voting model is improved. This suggests context is properly modeled in our method and suppresses object activations at unlikely locations within an image. It may worth to note that, in general, we observe a higher precision in high-recall regimes when context is modeled. This perhaps relates to the fact that context cues play a more important role in objects that have lesser visual cue support from themselves, in line with our intuition mentioned at the beginning of this chapter. An alternative view to this precision characteristics at higher recall is that context layers essentially narrow down the spatial search space for objects softly. In light of this, the linear addition of object and context cues currently used in our method may be improved. In our implementation we currently use log-scale context scores but a finer relation may be learned from data.

3.4.4 Segmentation performance analysis

Next, we present a segmentation performance analysis with different mask terms enabled. We present the precision-recall of the visibility mask at the point of 50% recall in object detection. For each object hypothesis, we obtain a soft segmentation score, which is used to compute the segmentation precision-recall curve in Figure 3.9. We can see that both local and global mask features help improve the segmentation performance. It is also clear that simultaneously voting for the local mask position and the whole object mask yields best segmentation performance.

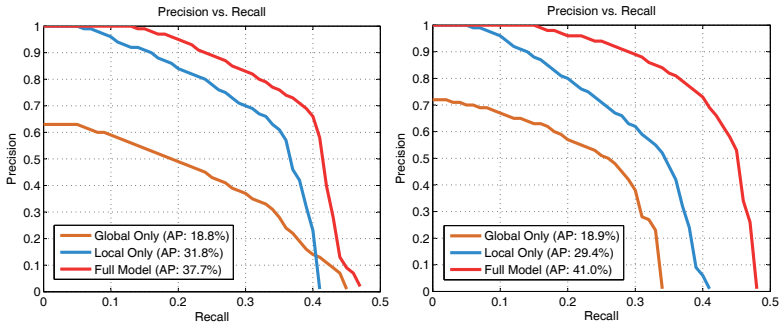


Figure 3.9: Precision-recall curves on the Berkeley 3D Object dataset (left) and the NYU Depth dataset (right) for segmentation at 50% recall rate in Figure 5.5. Simultaneously voting for local feature position and whole object hypothesis yields the best segmentation results.

3.4.5 More detailed examples

Finally, we present some more detailed results in Figure 3.10. Each row from (a) to (f) corresponds to one specific object instance on a test image. From left to right, we present (1) the RGB frame with ground-truth labelings as available in training. Specifically, these are two bounding boxes marked in green and red respectively. The green bounding box indicates visible parts of the instance, while the red one indicates the whole object including both visible and invisible regions. Note that we use a separate pixelwise labeling for evaluating segmentation performance. The pixelwise labeling was manually generated on the Berkeley 3D Object Dataset [77], while on NYU Depth [145] it is readily available. Then, we show (2) votes from different layers for the object centroid. From the upper-left corner, we show votes from the object layer (red), nearby context layer (green), occluder layer (yellow), and faraway context layer (blue) in the clockwise direction. In (3), the next column, the aggregated votes for the object centroid are shown. After that, we show (4) results with our alternating inference algorithm. The whole object hypothesis is shown as a red bounding box, with image cells inferred as visible highlighted in green. Next, we show (5) the corresponding mask prediction. Finally, (6) the segmentation results based on GrabCut are presented.

The examples presented in Figure 3.10 include some of the most representative results on both datasets, and reflect various aspects of our model.

Firstly, we can see the multi-layer representation helps build a more discriminative centroid voting codebook by suppressing false alarms in the object layer. This can be easily observed from examples (a), (b) and (e). Our model allows the object layer to generate concentrated peaks while raising or lowering the underlying terrain using the smeared votes from contextual layers. If a local peak from the object layer lacks support from its surrounding context, the vote will be weakened. On the other hand, if all layers have a consensus, the peak will be strengthened.

Secondly, our model captures the appearance of some occluders and use that information to

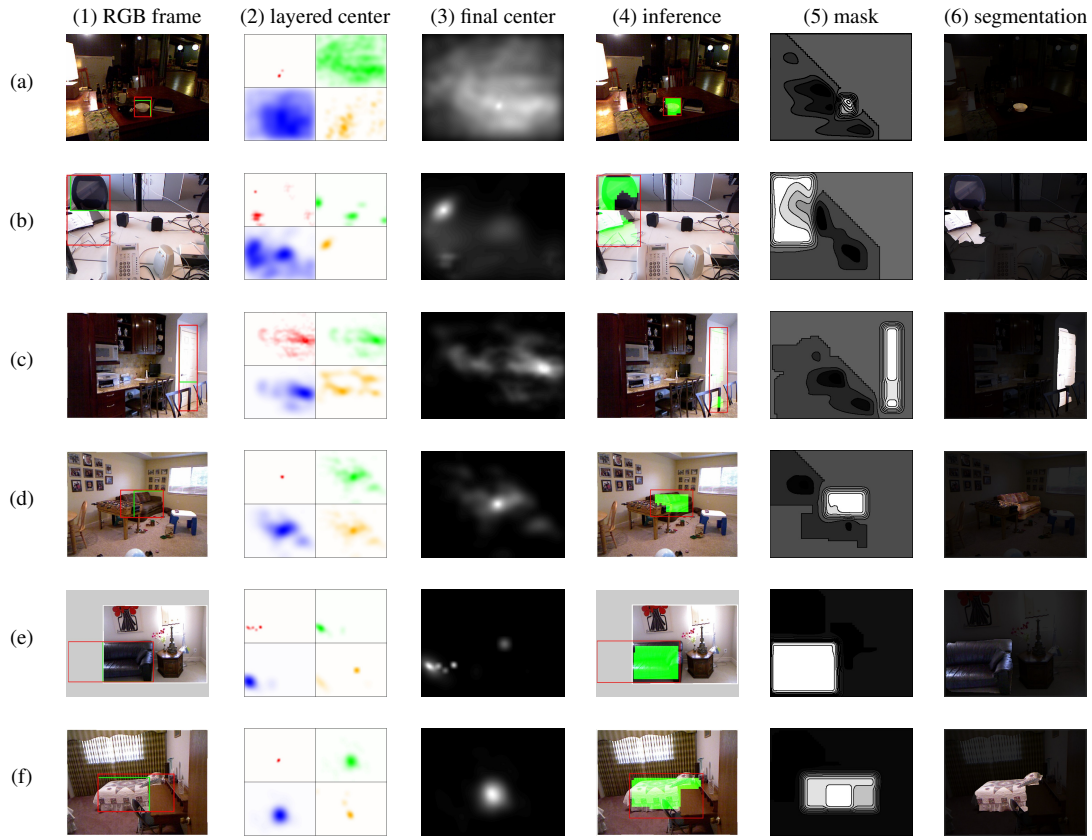


Figure 3.10: More experimental results of the proposed approach on Berkeley 3D Object Dataset [77] and NYU Depth Dataset [145]. Each row corresponds to a specific instance on a test image. See text for detailed discussion.

strengthen local centroid peaks, as well as carving out the shape of an object. This is inherently a very challenging task because the appearance of occluders varies greatly, and our model learns their appearances from only coarse-level labels. Successful examples include (d) and (f). In contrast, although the occluder layer gives roughly correct vote positions in (b), the shape voting breaks down on the desktop occluding the chair. In (c), the chair occluding the door is ambiguous and our model fails to fully recover the correct occlusion pattern.

Finally, our model is also capable of localizing truncated objects, as shown in (e) and there are some similar examples in the previous sections.

3.5 Conclusion

In this chapter, we have presented a novel structured Hough voting model for indoor object detection and occlusion prediction. We extend the original Hough voting based detection model by introducing a joint Hough space of object locations and visibility patterns. The structured

Hough model can naturally incorporate both the object and its context information, which is especially important for cluttered indoor scenes. In addition, we utilize depth information at the training stage to build a multi-layer contextual model so that a better visual codebook is learned and more detailed object-context relationships can be captured. The efficacy of our approach has been demonstrated on two publicly available RGBD datasets, and our experiments show we achieve improvements over the state-of-the-art 2D object detection approaches.

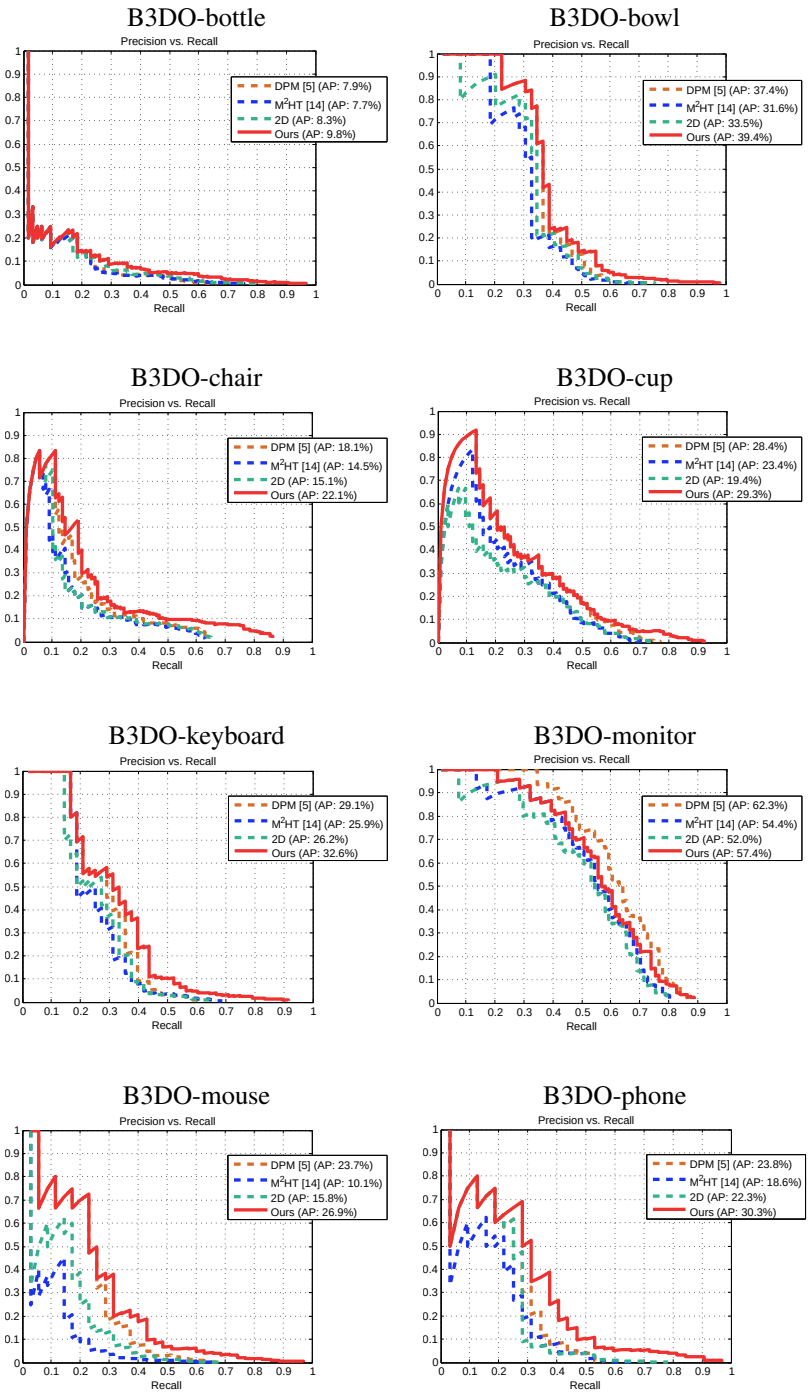


Figure 3.11: Per-class detection precision-recall curves on the Berkeley 3D Object dataset (B3DO). The solid curve corresponds to our approach (**Ours**). The dashed curves correspond to baseline methods: Deformable Parts Model (**DPM**) [46], Max-margin Hough transform (**M^{2}HT**) [129], and Max-margin Hough transform with 2D geometric context (**2D**). See details in text.

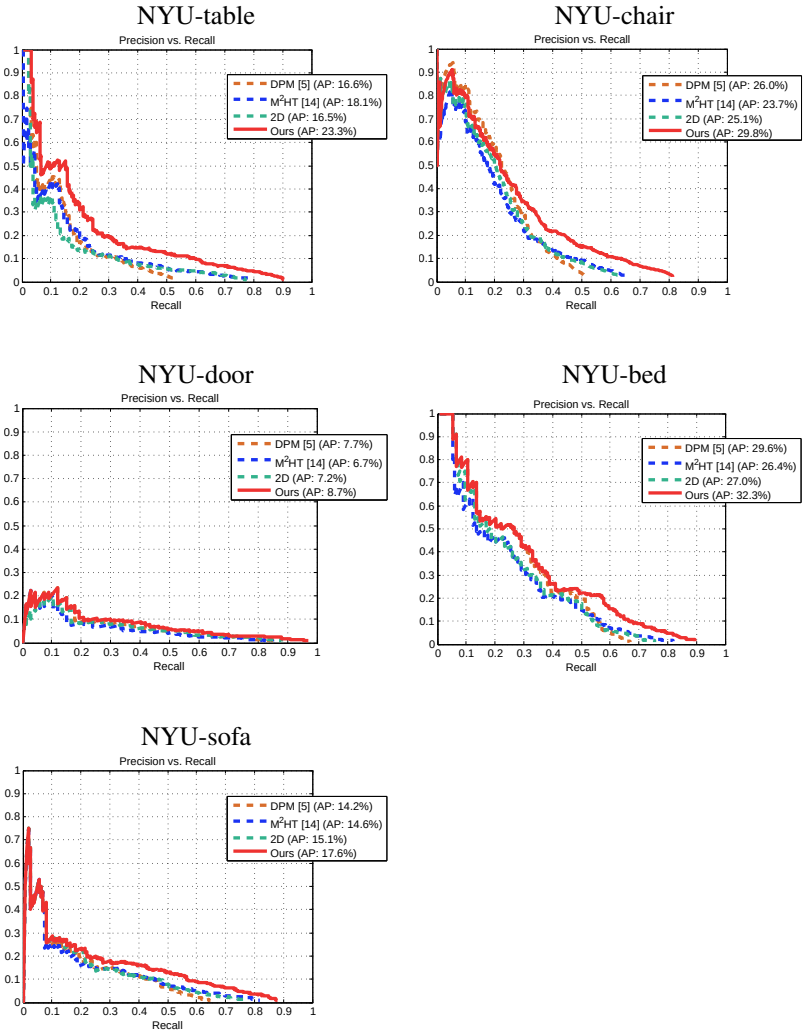


Figure 3.12: Per-class detection precision-recall curves on the NYU Depth dataset (NYU). The solid curve corresponds to our approach (**Ours**). The dashed curves correspond to baseline methods: Deformable Parts Model (**DPM**) [46], Max-margin Hough transform (**M^{2}HT**) [129], and Max-margin Hough transform with 2D geometric context (**2D**). See details in text.

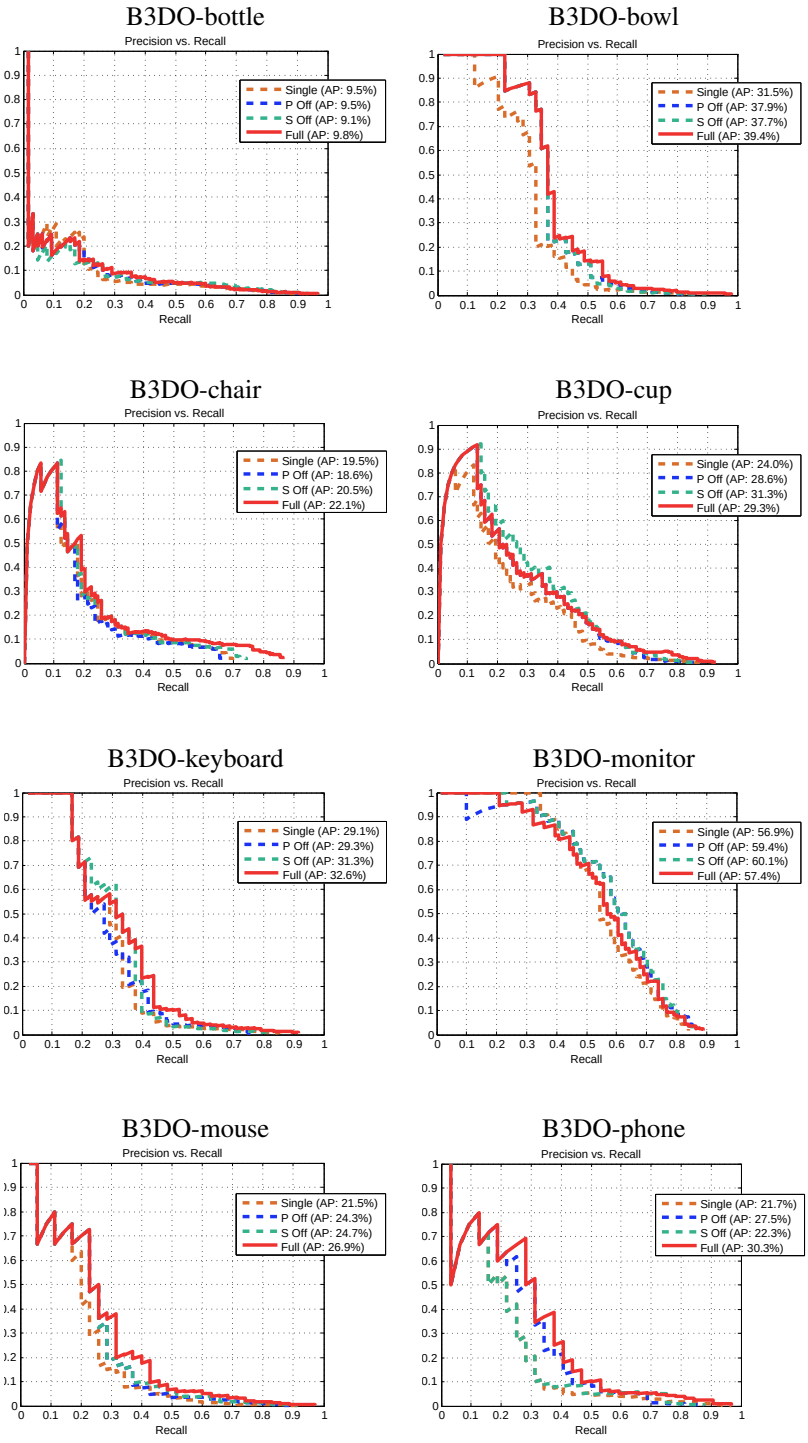


Figure 3.13: Per-class detection precision-recall curves on the Berkeley 3D Object dataset (B3DO). The solid curve corresponds to our full model (**Full**). The dashed curves correspond to diagnostic results with various components in our full model turned off, i.e., single layer context (**Single**), patch pair term off (**P Off**), and segmentation off (**S Off**). See details in text.

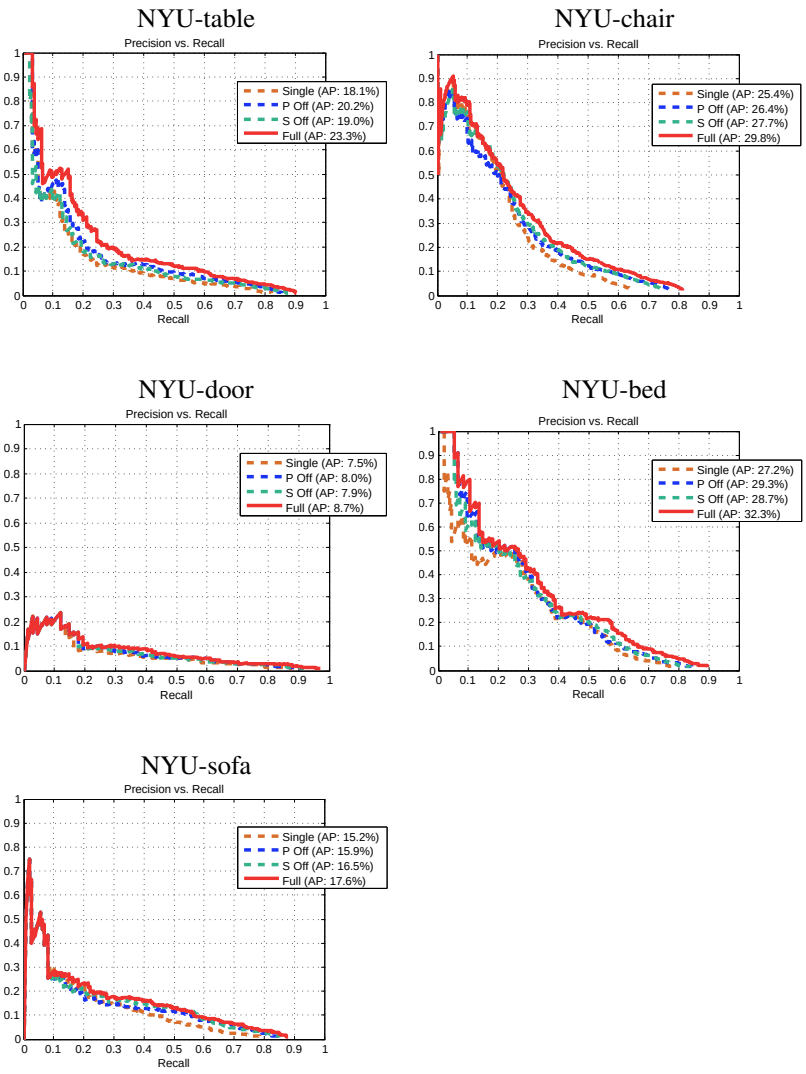


Figure 3.14: Per-class detection precision-recall curves on the NYU Depth dataset (NYU). The solid curve corresponds to our full model (**Full**). The dashed curves correspond to diagnostic results with various components in our full model turned off, i.e., single layer context (**Single**), patch pair term off (**P Off**), and segmentation off (**S Off**). See details in text.

Table 3.1: Per-class average precision on the Berkeley 3D Object dataset and the NYU Depth dataset. Mean average precision values are calculated separately for each dataset.

Object Categories	B3DO-bottle	B3DO-bowl	B3DO-chair	B3DO-cup	B3DO-keyboard	B3DO-monitor	B3DO-mouse	B3DO-phone	NYU-table	NYU-chair	NYU-door	NYU-bed	NYU-sofa	B3DO-mAP	NYU-mAP
Our Approach															
2D Context	8.3	33.5	15.1	19.4	26.2	52.0	15.8	22.3	16.5	25.1	7.2	27.0	15.1	24.1	18.2
Single Layer	9.5	31.5	19.5	24.0	29.1	56.9	21.5	21.7	18.1	25.4	7.5	27.2	15.2	26.7	18.7
Patch Pair Off	9.5	37.9	18.6	28.6	29.3	59.4	24.3	27.5	20.2	26.4	8.0	29.3	15.9	29.4	20.0
Segmentation Off	9.1	37.7	20.5	**31.3**	31.3	**60.1**	24.7	22.3	19.0	27.7	7.9	28.7	16.5	29.6	20.0
Full Model	**9.8**	**39.4**	**22.1**	29.3	**32.6**	57.4	**26.9**	**30.3**	**23.3**	**29.8**	**8.7**	**32.3**	**17.6**	**31.0**	**22.3**
Baseline Approaches															
DPM [46]	7.9	37.4	18.1	28.4	29.1	62.3	23.7	23.8	16.6	26.0	7.7	29.6	14.2	28.8	18.8
M^2HT [129]	7.7	31.6	14.5	23.4	25.9	54.4	10.1	18.6	18.1	23.7	6.7	26.4	14.6	23.3	17.9

Chapter 4

Glass Object Segmentation by Joint Inference of Boundary and Depth

4.1 Introduction

Semi-transparent objects are commonly found in indoor environments such as household or office scenes, and play a key role in daily human activities. As such, it is important for scene understanding and visual recognition systems to be able to localize them. Although the detection and segmentation for generic objects are well studied, localizing semi-transparent objects from RGB cameras is much more challenging due to lack of locally discriminative visual features and homogeneity of surface appearance [135, 51].

Most previous work on glass object detection and segmentation focused on detecting special properties of the glass surfaces and their interaction with the opaque environment in images [151, 3, 144]. In particular, McHenry, Ponce and Forsyth [135] design a classifier which attempts to find a glass/non-glass boundary based on a combination of cues, such as color and intensity distortion, blurring and specularity. In addition, contextual [134] or categorical [51] information is employed to integrate a variety of local features into a coherent surface or object model. Despite those efforts, glass object detection and segmentation still remain unsatisfactory in practice due to the ambiguity and lack of cues in 2D RGB images.

Recently, range (depth) cameras have been employed to detect transparent objects, in which the attenuation of signal intensities is exploited. Wallace and Csakany [209] develop a time-of-flight laser sensor based on photon counts. Klank, Carton and Beetz [84] use two images from a time-of-flight camera to detect and reconstruct transparent objects. The popularity of RGBD sensors has allowed researchers to utilize both intensity and depth to localize glass objects. Lysenkov, et al. [123] have proposed a model taking into account both silhouette and surface edges, and a CAD-based pose estimation method with a robotic grasping pipeline.

In this work, we aim to localize semi-transparent surfaces more precisely by exploring multi-mode sensors and incorporating depth information as a novel contextual cue. In particular, we seek to exploit low cost RGBD consumer cameras, such as the structured-light

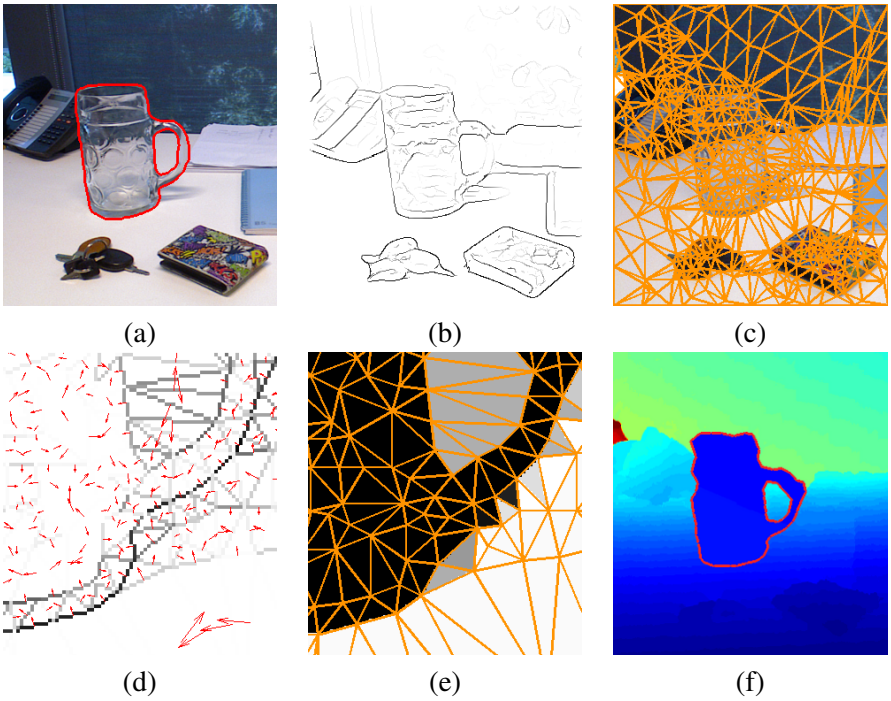


Figure 4.1: Illustration of the proposed approach. **(a)** Intensity image with ground truth foreground mask overlaid. **(b)** Edge detector output. **(c)** Triangulation result. **(d)** Boundary classifier output (magnified). **(e)** Superpixel classifier output (magnified). **(f)** Reconstructed depth with joint inference result overlaid.

PrimeSense device (e.g., Kinect), to fuse the intensity and depth information from a single view point for indoor environments. While recent work with RGBD cameras is mainly addressing generic object detection [98, 99, 49], here our goal is joint detection, segmentation and depth inference, which can facilitate many interactive tasks such as robotic manipulation. As discussed in Section 2.2.4, there has been some work exploiting range devices to detect or reconstruct semi-transparent objects (e.g., [209, 84]). Unlike those methods, we rely on a single view RGBD image and combine both intensity and depth cues.

In particular, we exploit the refraction and attenuation that will be experienced by an active signal passing through glass objects. This physical process is difficult to model, but it provides a distinctive missing-vs-nonmissing pattern in the depth channel. See Figure 2.8 for examples and note the irregular nature of the pattern. We integrate boundary cues from RGB channel with region cues from depth to build a glass boundary and region detector. In addition, we incorporate spatial cues by constructing a Markov Random Field on triangularized contour fragments and the corresponding superpixels [15]. A joint inference is designed to predict the glass boundary and region simultaneously. Furthermore, we perform a plane segmentation of the 3D scene in non-glass regions, and fill in the missing depth values caused by glass refraction and other factors. Note that this step would be difficult without the glass boundary/region

information. For the glass region, however, due to lack of depth measurement, we approximate its depth by assuming a cardboard cut-out standing on its (non-transparent) supporting surface, similar to the scene layout in [174].

The rest of this chapter is organized as follows. Section 4.2 describes the setup of our Markov Random Field model. This is followed by experimental evaluation in Section 4.3 and concluding remarks in Section 4.4.

4.2 Our approach

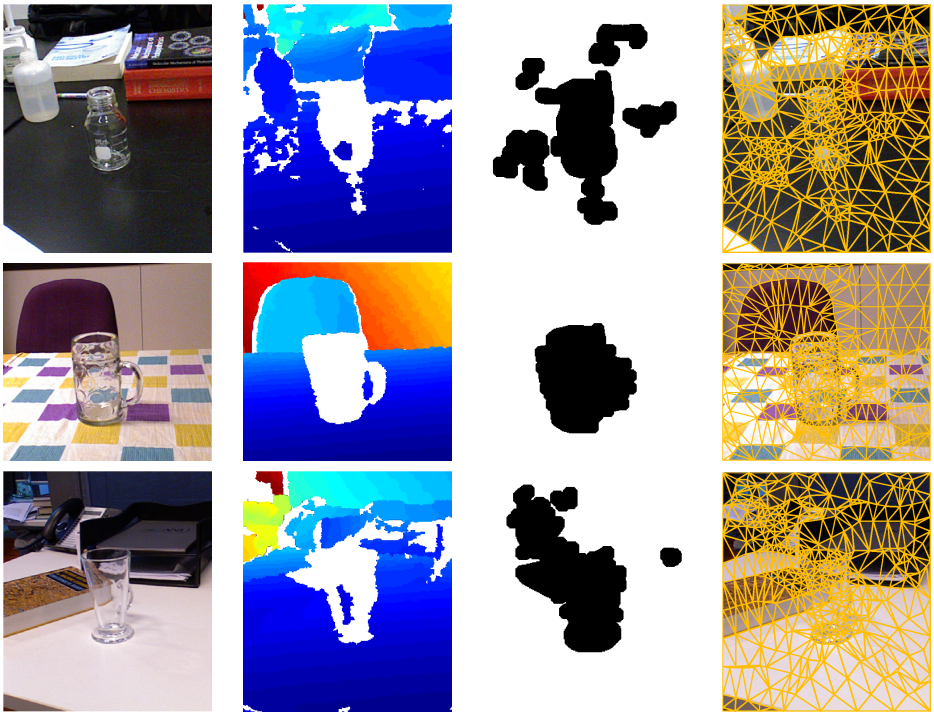
We address the glass object segmentation problem with a single view RGBD image, in which we combine intensity and depth cues and jointly reason about image boundaries and regions. Our main focus is to model the spatial context by constructing a boundary-region graph and design effective constraints that help resolve local ambiguities. This is achieved by building a Markov Random Field (MRF) on image boundaries and regions, and formulate the segmentation as an MAP inference problem of the random field.

To this end, we first propose potential glass regions and boundaries which help our graph construction process to create more detailed image partitions where glass objects may be present. Given the boundary-region graph, we combine intensity and depth cues for our local glass/non-glass estimates, and design an MRF model to encode the spatial dependency and label all image boundary fragments and regions. This also allows us to partially correct artefacts in depth readings and improve depth reconstruction of the scene.

4.2.1 Boundary and region graph

Glass region proposal. To facilitate glass segmentation, we first propose potential regions that may contain glass boundaries. Our boundary and region graph will then focus on these regions (i.e., creating more detailed image partitions to make accurate glass boundary localization possible). As images are usually dominated by non-glass objects and surfaces, this preprocessing allows us to maintain a relatively small number of image partitions, while having a high enough resolution in regions near glass boundaries. We make use of the distinctive missing pattern in the depth channel, as it is a good indication of the approximate location of a glass object.

Because the missing pattern is usually misaligned with ground-truth glass boundaries due to varying local refractive properties, and in many cases there may be incorrect depth readings in glass regions, it is unreliable to directly use missing depth regions as our glass region proposal. See Figure 4.2 (a) and (b) for some image examples. In this work, we use a heuristic approach based on image morphological processing to propose potential glass regions. We begin with removing small missing regions in the depth image as they are more likely a result from occlusion boundaries and other random noise. Next, we dilate edge fragments detected



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

Figure 4.2: Examples of the boundary and region graph construction. **(a)** Input intensity image. **(b)** Input depth image (missing readings are shown in white). **(c)** Glass region proposal with proposed glass regions in black. **(d)** Triangulation result.

near the remaining depth missing regions so that large connected components can be formed. We use a disk-shaped structuring element with radius r for this dilation. In addition, we fill in any holes to avoid hollow regions. See Figure 4.2 (c) for examples of glass region proposals. Note that our goal in this step is to recall as many regions near glass boundaries as possible, while keeping the radius r of the disk reasonably small, and we are less concerned about precision. We report quantitative evaluation results on the glass region proposal step in Section 4.3.2.

Boundary proposal. We would like our image partitions to follow glass boundaries where possible, so that a segmentation similar to the shape of the glass object can be obtained by assuming a subset of image partitions as glass regions, and the remaining as non-glass. However, the challenge of glass boundary detection is evident: glass boundaries are often weak and exhibit large local appearance variations. Therefore, detecting glass boundaries may require multiple types of information to deal with different local appearances. In addition, we would like the partitions to follow depth discontinuities. This would allow us to fit a plane in 3D to each image partition in order to reconstruct the depth of the scene. It should be noted that, as shown in Figure 2.8, the irregular nature of the missing patterns on depth maps renders it a

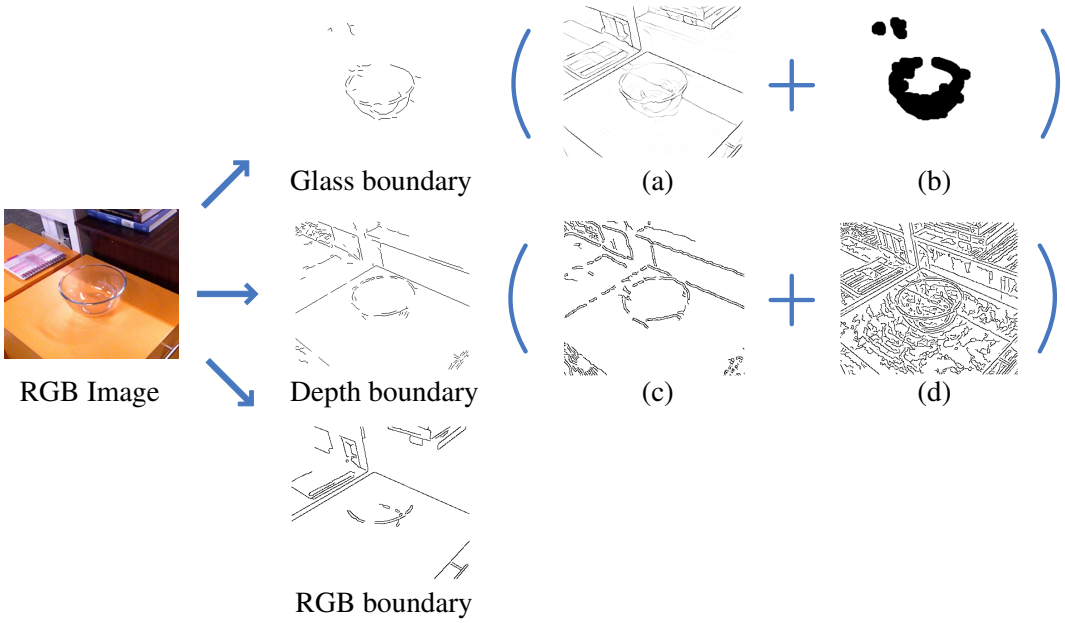


Figure 4.3: An example of boundary proposal including glass, depth and RGB boundary. **(a)** BGTG boundary detector output. **(b)** Glass region proposal results. **(c)** depth boundary detector output before alignment. **(d)** low-threshold RGB edge detector output. See text for details.

non-trivial task to proposal glass boundaries.

In order to detect glass boundaries and depth discontinuities, we combine boundary cues from multiple sources as follows:

$$gPb\_rgbd = \alpha_1 \cdot gPb\_glass + \alpha_2 \cdot gPb\_depth + (1 - \alpha_1 - \alpha_2) \cdot gPb\_rgb \tag{4.1}$$

where gPb_rgbd is the boundary map we use to partition an input image, gPb_glass a *glass boundary* map, gPb_depth a *depth boundary* map, and gPb_rgb an *RGB boundary* map. α_1 and α_2 are weighting coefficients. See Figure 4.3 for an example.

Firstly, for glass boundary we empirically evaluated some popular edge detectors, in particular boundary detectors from [132], and found that the BGTG boundary detector is generally good at recovering glass boundaries. Figure 4.3 (a) shows the output from a BGTG boundary detector. It is then thresholded and filtered by our glass region proposal discussed earlier in this section. The filtering ensures that BGTG is not applied to most non-glass image regions, and helps reduce the overall number of detected edge fragments significantly. We link edge fragments where possible to partially recover disconnected detections; then remove short, isolated fragments [93]. Again, depth information is not used here as it is highly noisy near glass boundaries and the missing patterns can either be dilated or corroded depending on the local

refractive properties.

Secondly, we detect the depth boundary by computing a local depth orientation map a smoothed depth image with missing regions filled in by a median filter [98], as shown in Figure 4.3 (c). To address the misalignment between RGB and depth image pairs, we run a Canny edge detector on RGB image with a very low threshold as in Figure 4.3 (d), then use the benchmark suite that comes with [132] to compute a minimum-cost correspondence between the depth boundary map and the RGB Canny edge map. The final depth boundary we use are the correspondences of the depth boundary on the RGB Canny edge map, discarding the original (noisy) depth boundary. In this way, any drifted depth boundaries can be realigned to their correspondences on the RGB Canny edge map.

Finally, we supplement the glass and depth boundary maps with an *RGB boundary* map, which again is a Canny edge map on the RGB image but with a higher threshold. This captures any weak glass and depth boundaries that co-occur with strong intensity changes. We found this necessary to recover some depth boundaries between adjacent regions with different orientations in 3D (e.g., the boundary between the brown desktop and the blue book in Figure 4.3).

In Section 4.3.2, we quantitatively examine the effectiveness of the three distinctive boundary maps, and show that their combination gives the best recall rate for glass boundary proposal.

Graph construction. To model the spatial context, we construct a graph on proposed boundaries and planar regions as follows. We first break the linked boundaries into shorter lines and perform Delaunay triangulation on their end points. To control the resolution of the graph, we set the maximum length of these shorter lines to 50 px in the proposed glass regions and 100 px elsewhere. The triangulation generates two types of nodes and their connectivity: boundary fragment nodes connected with their end points, and triangular superpixel nodes partitioned by boundary fragments. Triangulation allows for a straightforward local neighborhood relationship among boundary fragments and and regions. In particular, each boundary fragment has exactly two region neighbors in our graph. As we will show, this allows us to design our energy function based on local cliques involving two neighboring superpixels and the boundary fragment in between without double counting. As a side effect, the graph construction process creates small artefacts along glass boundaries due to the linear nature of the sides of triangles. However, as the resolution of the graph is higher near the glass regions in our work, it only slightly affects the glass boundary recall rate. See Section 4.3.2 for a quantitative evaluation. As shown in Figure 4.2, most glass boundaries and depth discontinuities are followed by our partition, and this process partially recovers broken/missing boundary detections.

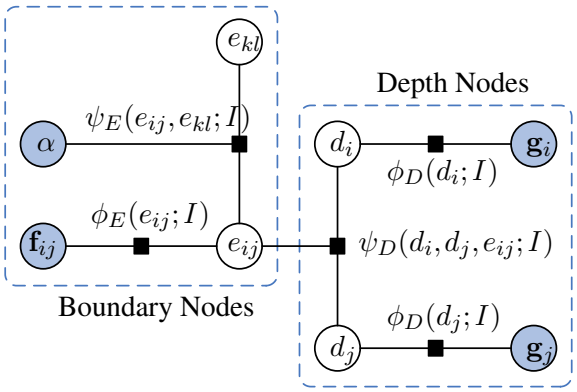


Figure 4.4: The factor graph of the MRF model for our glass detector. Each black square represents a term in Equation 5.2. Each circular node represents a random variable. Shaded nodes are observations.

4.2.2 A Markov Random Field on boundaries and superpixels

We build a Markov Random Field model [15] on the boundary fragments and superpixels w.r.t. the graph in Section 4.2.1, which defines a joint distribution over the glass labeling given an RGBD image input. Note that our output includes both boundary and region labeling – with which we are able to encode the spatial dependency in a more expressive way. We first introduce the energy function of our model and then describe its components in detail.

Let the boundary fragments be $\mathbf{E} = \{e_{ij}\}$ and its subgraph be (V_E, G_E). Similarly we have $\mathbf{D} = \{d_i\}$ and (V_D, G_D) for superpixels. We define the state space of d_i as $\mathcal{D}_i = \{0, 1\}$, indicating glass and non-glass. For boundary variable e_{ij}, we first assign a direction to it and define its left and right side. e_{ij} is 0 if it is not a glass-vs-nonglass boundary, $+1$ if the glass region lies at left side and -1 otherwise. Therefore the state space for e_{ij} is $\mathcal{E}_{ij} = \{0, +1, -1\}$. The energy function we propose can be written as follows:

$$E = \underbrace{\sum_{ij \in V_E} \phi_E(e_{ij}; I)}_{\text{boundary unary}} + \beta \underbrace{\sum_{(ij,kl) \in G_E} \psi_E(e_{ij}, e_{kl}; I)}_{\text{boundary pairwise}} + \gamma \underbrace{\sum_{i \in V_D} \phi_D(d_i; I)}_{\text{superpixel unary}} + \lambda \underbrace{\sum_{(i,j) \in G_D} \psi_D(d_i, d_j, e_{ij}; I)}_{\text{superpixel pairwise}} \tag{4.2}$$

where I is the input image, and β, γ and λ are weighting coefficients. The factor graph is shown in Figure 4.4.

Boundary unary potentials. The boundary unary potential is the negative log-probability

from a classifier based on local cues:

$$\phi_E(e_{ij}; I) = -\log(P(e_{ij}|\mathbf{f}_{ij})) \tag{4.3}$$

where $\mathbf{f}_{ij} \in R^N$ is the local feature vector for the boundary fragment e_{ij}. We evaluate two different local classifiers: a Support Vector Machine (SVM) with a Radial Basis Function (RBF) kernel and a Random Forest (RF) classifier. The classifier input consists of features extracted from both sides of a boundary fragment. In particular, we extract features from multiple pairs of feature windows, each on either side of the boundary fragment. See Section 4.3.1 for details. The features used for training the boundary unary classifier include:

- Hue and Saturation [135]: This feature is designed to measure the color similarity between both sides of a boundary fragment, as color on both sides of a glass boundary tends to be similar. A twenty-bin histogram is constructed for hue and saturation values on both feature windows. The histograms are then normalized and the Euclidean distance between them are used as our feature.

- Blurring [135]: This feature quantifies the relative smoothness between both sides of a boundary fragment, as glass surface could have a blurring effect on the background. The discrete cosine transform is used and the mean of frequency coefficients is chosen as an indication of smoothness. After this, we use the difference of mean frequencies on both sides as our feature, which reflects the relative smoothness. As this measure can be less reliable on highly textured regions, the measured frequency difference is normalized by a texture entropy measure (i.e., the standard deviation of intensity values on the smoother side of the boundary).

- Blending and Emission: The feature is based on the overlay assumption of glass surfaces [3] and particularly, the linear model for the intensity of a transparent surface:

$$I = \alpha I_B + e \tag{4.4}$$

  where I_B is the intensity of the background, α is a blending factor, and e is the emission of the semi-transparent surface. We follow the method in [135] by clustering the intensities on both sides of the boundary and solving for α and e as a linear least square problem.

- Texture distortion [135]: The feature measures the similarity of texture between both sides of a boundary fragment. In particular, texture can be magnified or skewed when

observed through glass. We use a filter bank as described in [130] to obtain a distribution of filter outputs. The texture similarity is then measured by the Euclidean distance between the distributions observed on both sides of the boundary.

- Missing depth: This feature exploits the fact that depth readings of an RGBD sensor tend to be missing on the glass side of the glass boundary while being valid on the other side. Therefore, we can firstly compute the depth missing ratio of a feature window:

$$missing\ ratio = \frac{No.\ of\ pixels\ with\ missing\ depth\ reading}{No.\ of\ pixels\ in\ the\ feature\ window} \tag{4.5}$$

  Once the missing ratios of a pair of feature windows are obtained (one on each side of a boundary fragment), we use their difference as the feature value. The underlying assumption is that, for non-glass regions the missing ratios from both sides should be low, while for glass regions they should be both high. Generally, a large difference in missing ratio may only be observed near glass/non-glass boundaries.

For boundary fragment orientation we train a separate SVM classifier. We use only two features: saturation and depth missing ratio. The two features were found to be quite robust in identifying boundary orientation. We assign an associated direction to each boundary fragment (i.e., viewed as a vector on the 2D image plane) so we can unambiguously define its left and right. We compute features on the left and right patches respectively, and then subtract the right from the left.

The boundary unary potential is illustrated in Figure 4.1 (d). Each fragment is assigned with a probability for glass object contour (i.e., the darker the more possible), and the orientation is marked with red arrows pointing towards detected glass regions.

Boundary pairwise potentials. The boundary pairwise potential imposes a direction-sensitive smoothness prior. Note that for each boundary fragment e_{ij} there are three possible states. The model prefers configurations where connected boundary fragments have the glass region on the same side. More formally, we define the smoothness prior for two connected boundary fragments e_{ij} and e_{kl} as:

$$\begin{aligned}\psi_E(e_{ij}, e_{kl}) &= 1 - \delta(e_{ij} = e_{kl} \neq 0) \\ &+ C_1\delta(e_{ij} = e_{kl} = 0) + C_2\delta(e_{ij} \neq e_{kl})\end{aligned} \tag{4.6}$$

where $\delta(\cdot)$ is the indicator function, and we choose $C_1 = 0.3 * \delta(\frac{\pi}{2} < \alpha \leq \pi)$, and $C_2 = (1 - \cos\alpha)^3\delta(\frac{\pi}{2} < \alpha \leq \pi)$ empirically. Here α is the angle between two fragments.

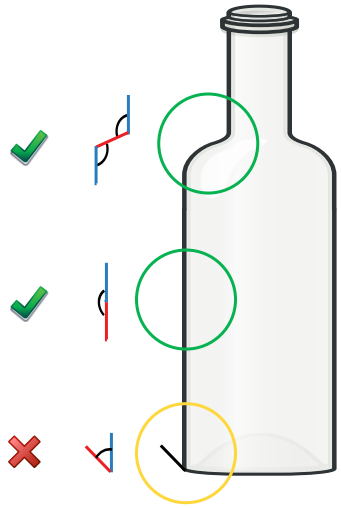


Figure 4.5: Illustration of our angle preference for boundary pairwise term. In the top and middle examples, the angles between connected boundary fragments are obtuse and straight respectively. These are commonly found in ground-truth glass boundaries. In the bottom example, however, the angle is acute and is more likely a result from incorrectly identified glass boundary.

We prefer configurations where the angle between two neighboring boundary fragments are obtuse, so additional penalty terms are added if there is no glass boundary (i.e., $e_{ij} = e_{kl} = 0$) or the boundary orientation is incompatible (i.e., $e_{ij} \neq e_{kl}$). If the angle is acute, we simply treat all states equally except if the orientation is compatible (i.e., $e_{ij} = e_{kl} \neq 0$). See Figure 4.5 for an illustration of our angle preference.

Superpixel unary potentials. This term is similar to the boundary unary term except that features are extracted from triangular superpixels. Similar to the boundary orientation classifier, only saturation and depth missing ratio are used. This is because other features we experimented with are less effective, particularly when compared to the depth missing cues. As shown in Figure 4.4, we denote the local feature vector for superpixel d_i with $\mathbf{g}_i$. The result is illustrated in Figure 4.1 (e).

Superpixel pairwise potentials. This pairwise term specifies valid configurations of a boundary fragment and its neighboring superpixels. Any incompatible state will be penalized. Specifically, for boundary fragment e_{ij} let d_i be the superpixel that resides to its left and d_j to the right. We set the pairwise potential as:

$$\begin{aligned}\psi_D(d_i, d_j, e_{ij}) = \quad & \delta(d_i \neq d_j, e_{ij} = 0) \\ & - \delta(d_i = 0, d_j \neq 0, e_{ij} = +1) \\ & - \delta(d_i \neq 0, d_j = 0, e_{ij} = -1).\end{aligned} \tag{4.7}$$

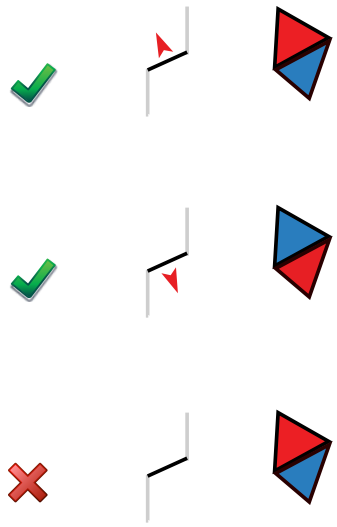


Figure 4.6: Illustration of our superpixel pairwise term. Assume the arrow points towards glass regions in red, and the non-glass regions are in blue. See text for details.

The three terms correspond to the bottom, top and middle examples in the illustration in Figure 4.6. The top and middle examples in Figure 4.6 are two scenarios where the orientation of the boundary fragment (middle column) and the neighboring superpixels (right column) are consistent, so we encourage them by adding negative energy terms. In the bottom example, the labels of two neighboring superpixels are different, but the boundary fragment in between is not part of the glass boundaries (i.e., $e_{ij} = 0$, illustrated as without an orientation arrow). This is in violation with our graph consistency assumption so a penalty (positive) term is added.

4.2.3 Joint prediction

We greedily search for the global parameters β,γ and λ using a small held-out validation set, and use $\beta = 0.25$, $\gamma = 50$ and $\lambda = 20$ in our work. To predict the boundary and region labels jointly, we adopt an alternating inference approach to compute the marginals of the boundary nodes and superpixel nodes. We start with no depth terms and use Loopy Belief Propagation (LBP) [15] to compute an initial guess of the marginals of boundary nodes. In each iteration, we first use mean field approximation [81, 215] to marginalize out the boundary variables and compute the marginals on depth nodes. Then we update the marginals on boundary nodes in a similar way. This procedure is repeated until there is no change on the marginals. Usually, convergence can be obtained within 5 such iterations and the inference takes a few seconds with our MATLAB implementation on an Intel i7 desktop.

We now describe some details on the mean field approximation. In particular, we approximate the distribution $p(\mathbf{E}, \mathbf{D}|I)$ with a fully factorized distribution $q(\mathbf{E}, \mathbf{D})$:

$$q(\mathbf{E},\mathbf{D}) = \prod_{ij\in V_E} \mu_{ij}(e_{ij}) \cdot \prod_{i\in V_D} \nu_i(d_i) \tag{4.8}$$

where μ_{ij} and ν_i are variational parameters corresponding to marginal probabilities over boundary nodes and superpixel nodes respectively. μ_{ij} and ν_i are obtained in an alternating fashion by minimizing the KL divergence between $p(\mathbf{E},\mathbf{D}|I)$ and $q(\mathbf{E},\mathbf{D})$ which is also equivalent to minimizing the mean field free energy:

$$F_\mu(\{\mu_{ij}\}) = -\sum_{(ij,kl)\in G_E} \sum_{\substack{e_{ij}\in\mathcal{E}_{ij}\\ e_{kl}\in\mathcal{E}_{kl}}} \mu_{ij}(e_{ij})\mu_{kl}(e_{kl}) \log\psi_E + \sum_{ij\in V_E}\sum_{e_{ij}\in\mathcal{E}_{ij}} [\log\mu_{ij}(e_{ij}) - \log\phi_E] \tag{4.9}$$

$$F_\nu(\{\nu_i\}) = -\sum_{(i,j)\in G_D} \sum_{\substack{d_i\in\mathcal{D}_i\\ d_j\in\mathcal{D}_j}} \nu_i(d_i)\nu_j(d_j) \log\psi_D + \sum_{i\in V_D}\sum_{d_i\in\mathcal{D}_i} [\log\nu_i(d_i) - \log\phi_D] \tag{4.10}$$

where the variables in the energy terms are omitted for notation simplicity.

Setting the derivatives with respect to μ_{ij} and ν_i equal to zero gives the fixed-point equations for mean field approximation:

$$\mu_{ij}(e_{ij}) = \frac{1}{Z_E} \cdot \phi_E(e_{ij};I) \cdot \exp\left(\sum_{kl\in N_{ij}} \sum_{e_{kl}\in\mathcal{E}_{kl}} \mu_{kl}(e_{kl}) \log\psi_E(e_{ij},e_{kl};I)\right) \tag{4.11}$$

$$\nu_i(d_i) = \frac{1}{Z_D} \cdot \phi_D(d_i;I) \cdot \exp\left(\sum_{j\in N_i} \sum_{d_j\in\mathcal{D}_j} \nu_j(d_j) \log\psi_D(d_i,d_j,e_{ij};I)\right) \tag{4.12}$$

where Z_D and Z_E are normalization constants chosen so that $\sum_{ij\in G_E}\mu_{ij}(e_{ij}) = 1$ and $\sum_{i\in G_D}\nu_i(d_i) = 1$. We use the value of e_{ij} from the previous iteration in $\psi_D(d_i,d_j,e_{ij};I)$. The mean field approximation for boundary nodes is done by iterating Equation 4.11, then approximating the marginal probability $p(e_{ij}|I)$ by the steady state $\mu_{ij}(e_{ij})$. Similarly, we use $\nu_i(d_i)$ to substitute $p(d_i|I)$ after iteratively updating Equation 4.12. It should be noted that the update for fixed point equations can also be seen as a message-passing algorithm where every node sends a message μ_{ij} (or ν_i) to its neighbors. The message is, in turn, based on the message it received from its neighbors in the previous iteration.

4.2.4 Depth reconstruction

Given the segmentation, we can reconstruct the depth of the scene in a post-processing step. First, we perform a plane segmentation of the scene directly in 3D by fitting each superpixel with a plane. We assume a parametric planar form for each superpixel, i.e., $-a_i(x_{ir} - x_{i0}) + b_i(y_{ir} - y_{i0}) + z_{ir} - c_i = 0$ so the parameters for each superpixel can be expressed as a triplet $\mathbf{p_i} = (a_i, b_i, c_i)$. We then identify major planes in the scene by running K-means clustering on the plane parameters with an increasing number of clusters, or equivalently, planes. We begin with 2 planes and use the plane parameters of the cluster centroid to reconstruct its member superpixels, then measure the reconstruction error. We repeat with one more plane at a time until the decrease in reconstruction error is small, or reaching a maximum of 20 planes for a scene. Each glass object is modeled as a cardboard cut-out standing on, and perpendicular to, its supporting plane. We use a simple assumption that the plane adjacent to the bottom of a glass object is the supporting plane, and it works well for most glassware in our experiments. See Figure 4.1 (f) for an example.

4.3 Experimental evaluation

4.3.1 Dataset and setup

We collected an RGBD Glass Dataset that contains 171 RGB and depth image pairs of 43 distinct glass objects taken from multiple views and with different levels of background clutter. We manually generated a pixelwise ground-truth segmentation mask for each object. In the experiment that follows, we randomly split the dataset into training and testing subsets, including 92 and 79 RGBD image pairs respectively.

For the local classifiers on boundary fragments, we extract features from multiple pairs of image patches at the two sides (i.e., left and right) of the boundary. The locations of those pairs are defined by a triplet $l_i = (d_i, r_{1i}, r_{2i})$, where $d_i \in \{3, 5, 10\}$ is the pixel distance from the patches to the boundary, and $r_{1i}, r_{2i} \in \{5, 10, 15, 20\}$ are the lengths of two adjacent sides. For the Random Forests classifiers, we use a three-fold cross-validation process which resulted in 500 trees with 16 predictors sampled for splitting at each node. The superpixel unary potentials are given by an SVM with an RBF kernel. We do not use Random Forest classifier for superpixels as the feature dimension is small.

4.3.2 Recall statistics for glass proposal

As images are usually dominated by non-glass regions and surfaces, glass region and boundary proposals are important preprocessing steps to ensure we have a manageable number of image partitions in the construction of our boundary-region graph. In this section, we quantitatively

Table 4.1: The overall glass region recall rate, near-boundary glass region recall rate, and the proposed glass area under different dilation disk radii r. Setting $r = 15$ px gives a good tradeoff between recall rates and the proposed area. See text for details.

Dilation (pixels)	$r = 5$	$r = 10$	$r = 15$	$r = 25$
Overall Recall	0.967	0.987	1.000	1.000
Boundary Recall	0.892	0.970	0.995	0.996
Proposed Area	25.7%	36.5%	42.2%	57.9%

evaluate the recall statistics for our glass region and boundary proposals using the method described in Section 4.2.1.

Glass region proposal. For glass region proposal, we would like to recall as many glass regions as possible, particularly those near glass boundaries. We can therefore create detailed image partitions in these regions to facilitate accurate glass boundary localization. However, the area of the proposed regions should be relatively small, as a coarse segmentation would suffice for non-glass regions. In our experiment, we tune the radius r of the dilation disk described in Section 4.2.1 and observe the changes to relevant statistics. When r is small, we dilate the depth missing pattern conservatively and may miss some ground-truth glass regions. When r is large, we include areas around missing patterns more aggressively but run the risk of including too many non-glass regions. Table 4.1 reports the overall glass region recall rate, near-boundary glass region recall rate, and proposed glass area (out of the entire image area, shown in percentages). The near-boundary glass region is created by dilating the ground-truth glass boundaries by 5 px. The proposed glass area, shown in the last row, grows with r. As we can see from the first two rows, the overall and near-boundary glass region recall rates roughly saturate at $r = 15$ px and we use this value in our succeeding experiments.

Glass boundary proposal. We now evaluate the effectiveness of the three distinctive boundary cues in Equation 4.1. The three cues capture different aspects of the image contours. gPb_glass is the output from a thresholded BGTG boundary detector. In our experiments, we set the threshold to the 50% quantile of the BGTG detector output in proposed glass region to make it image adaptive. gPb_depth is the realigned depth boundary. gPb_rgb is the output from a Canny edge detector to capture only strong intensity edges. We set the high and low sensitivity thresholds to 0.40 and 0.16 empirically. Table 4.2 shows the boundary recall rates measured with the benchmark utility from [132]. As we can see, although the BGTG boundary detector alone performs well, the other two boundary maps are complementary and the final combined result gPb_rgbd gives the best recall. In addition, we measure the boundary recall after graph construction, to quantitatively measure the loss of recall due to triangulation. It turns out that the loss is only 1.6% which suggests most glass boundaries are still followed by our partition after triangulation.

Table 4.2: The glass boundary recall rates from various boundary cues. The first three columns give the recall rates for the three boundary cues in Equation 4.1. The last two columns give the recall rates using the combination of the three cues, before and after triangulation. See text for details.

	gPb_glass	gPb_depth	gPb_rgb	gPb_rgbd (before)	gPb_rgbd (after)
Recall	0.853	0.301	0.257	0.975	0.959

Table 4.3: F-measures at 50% recall for boundary and region accuracy metrics, respectively.

	Intens.+ SVM	Intens.+ Depth	Detached Inference	Joint Inference
Boundary	19.52	44.38	54.08	62.27
Region	28.06	55.84	61.85	65.96

4.3.3 Segmentation results and comparisons

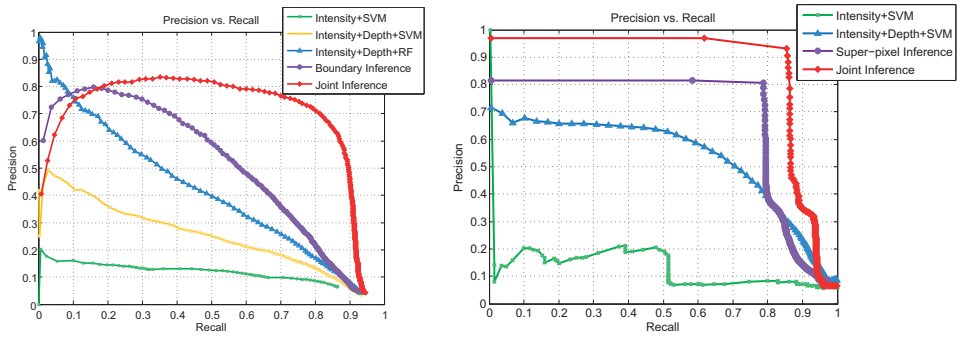


Figure 4.7: The precision-recall curves based on boundary matching (left panel) and pixelwise region matching (right panel).

The glass segmentation results are summarized in Figure 4.7, which shows the precision-recall curves of our glass detector under two metrics: boundary pixel accuracy and region pixel accuracy. For boundary accuracy, we use the benchmark utility from [132] and the matching procedure. We compute a list of correspondences below a distance threshold between the boundary estimate and the ground-truth boundary map. We also report the F-measure computed at 50% recall rate in Table 4.3. Here the F-measure is the harmonic mean of the precision and recall rates, i.e., $F = 2/(1/Pr + 1/Rc)$. Where we use both the SVM and the Random Forest classifiers, we report the better performance from the two.

We can see that our method achieves much better performance than the baselines. For the methods that use features from RGB images only, the performance is poorest due to the challenging nature of our dataset. We have tested the same set of features on the dataset in [135] and achieved similar results as theirs. The performance is greatly improved by using

Table 4.4: Comparison of average runtime per image (in seconds) between detached and joint inference. The numbers report here are a comparison of MRF inference times (not including feature extraction and local classification).

	Boundary Inference	Region Inference	Joint Inference
Runtime (sec)	2.350	2.927	5.617

depth cues, and by almost 40% precision on average. For boundary fragments, the Random Forest classifier with features extracted at multiple locations further increases the accuracy, which provides around 20% precision increase at 50% recall.

The MRF model further improves the performance, particularly in maintaining high precision into high recall regime. We observe a 10% precision gap between local classifier performance and results from the MRF. Joint inference is the most effective method of all. The precision for both boundary fragments and pixelwise matching sustained at a high level until around 80% recall. Our method is able to recall over 80% of glass boundaries and regions with a boundary matching precision over 70% and region matching precision over 90% respectively.

We present some examples where our method performs well in Figure 4.8, and a few failure scenarios in Figure 4.9. As shown in Figure 4.8, our method is able to deal with background clutter and texture variations with the help of unary classifiers trained both on boundaries and regions, and the joint inference. Although the most important cue for localizing glass objects, the depth missing pattern, exhibits large variations we can still successfully identify glass regions accurately in most cases. The piece-wise planar assumption for glass object gives a reasonable depth reconstruction in most cases. Indeed, the planar assumption is not sufficient for certain applications such as robotic manipulation with a gripper. We leave detailed shape reconstruction as our future work. In Figure 4.9, we show examples of failure cases on our dataset. Most failure cases are due to weak RGB cues, or strong local deformation of depth missing pattern, or background texture incorrectly identified as glass boundaries. These usually lead to protrusion or erosion in our segmentation and inconsistent boundary and region inference results. As a future direction, we may extend our energy model to encourage contour closure [140] and the consistency in the prediction for boundary and region.

In terms of computation time, we are interested in the extra runtime costs for our joint inference. We compare the average runtime per image between detached and joint inference on an Intel i7 desktop in Table 4.4. In our experiment we observe that the first one or two rounds of the alternating inference are the most time-consuming, as values of many random variables may change. The latter rounds generally take much less time, and the increase in computation time w.r.t. rounds of alternating inference is sub-linear.

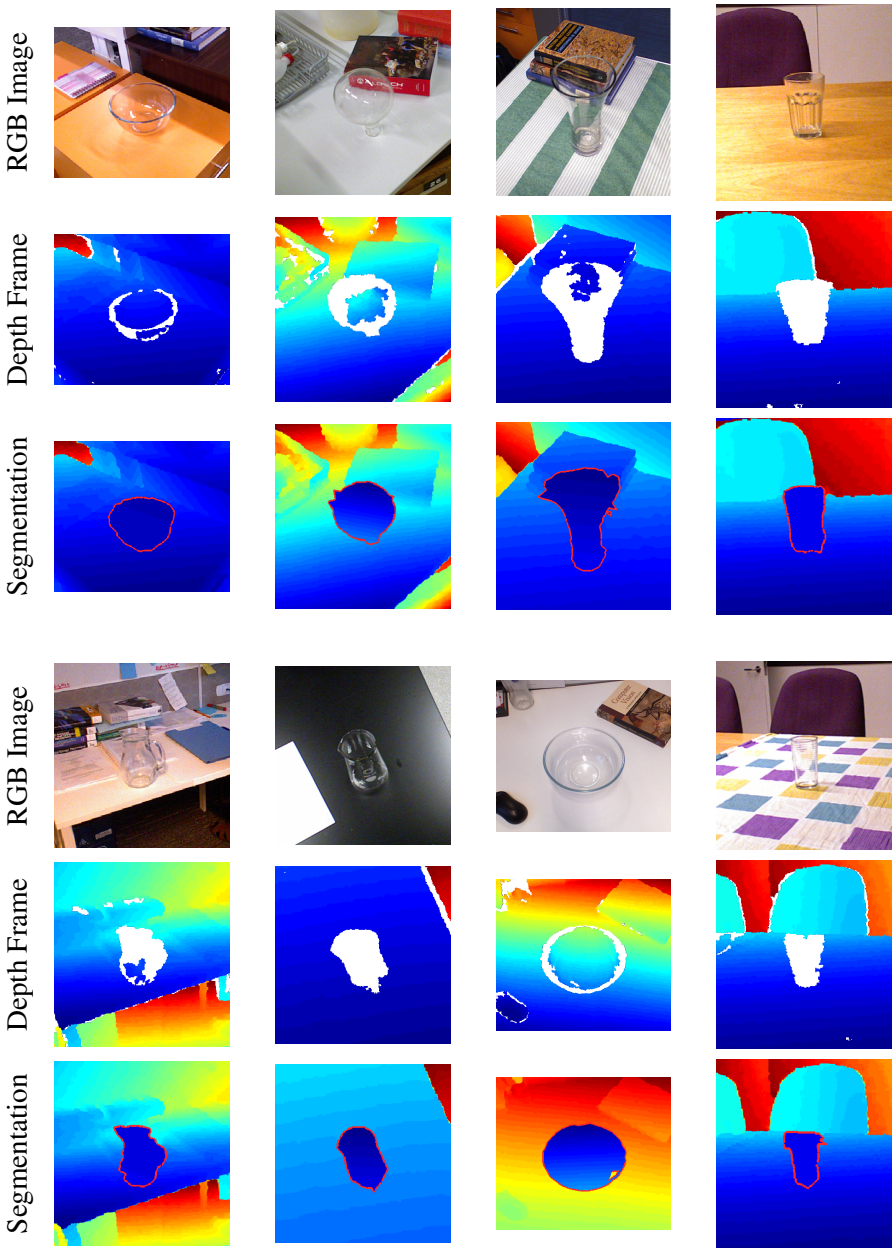


Figure 4.8: Examples of glass detection results on our new RGBD Glass dataset. Note that missing areas are shown in white, and depth readings are recovered by a piece-wise planar model.

4.3.4 Qualitative analysis for joint inference

One key contribution in our work is to jointly reason about boundary and region. Reasoning about boundary and region jointly allows us to combine local features from a boundary and superpixel perspective simultaneously. More importantly, the boundary and region graph can capture more detailed interactions locally, such as the interplay between boundary orientation

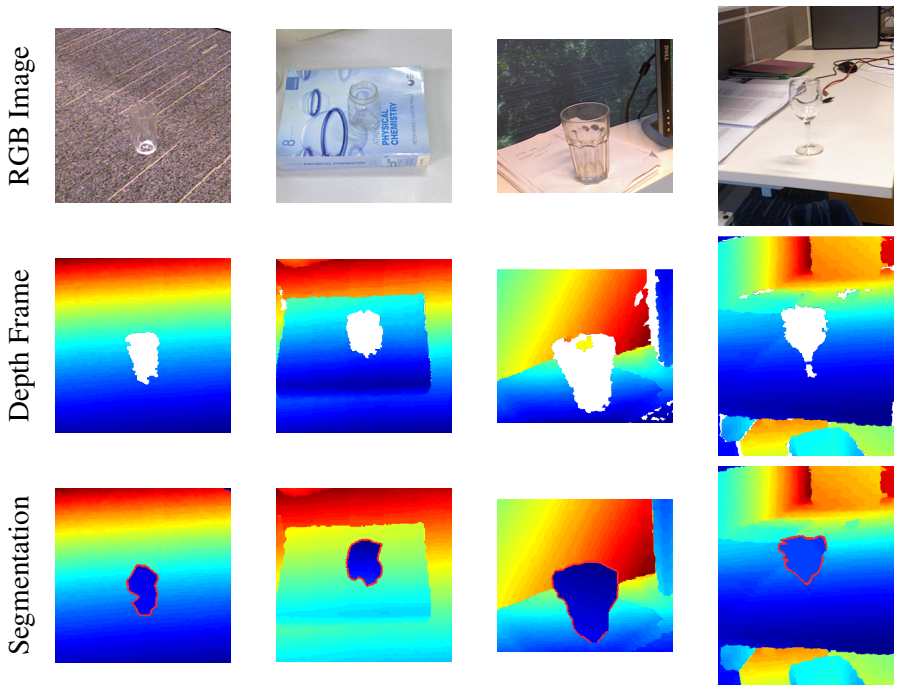


Figure 4.9: Failure examples of glass detection on our RGBD Glass dataset. See text for details.

and neighboring regions. In Figure 4.7 and Table 4.3, we have shown that quantitatively this results in a better glass segmentation performance; in this section we show some qualitative examples to justify our design choice. We look at two aspects of our joint inference: the unary terms and the iterative inference process.

Figure 4.10 shows some examples of boundary and region unary classifier outputs. In the first three examples, the boundary classifiers do a better job at identifying local glass boundary in general. The region classifiers in these examples give some spurious protrusions and erosions. If we follow the region classifiers, incompatible boundary orientations will be derived and penalties in our energy terms will apply to these configurations. In the remaining examples, however, region classifiers are more reliable and this can guide us find the correct boundary configuration.

Figure 4.11 shows some examples of comparisons among the boundary unary classifier output, the boundary marginals with the initial LBP inference involving boundary potentials only, and the boundary marginals with joint prediction after 5 iterations. Although the initial boundary inference helps in strengthening some weak glass boundaries, it is not powerful enough to identify true glass boundaries particularly near noisy predictions. Also, in the last two examples, we have spurious glass boundary detections well outside the glass region. Joint inference helps suppress these boundaries mainly because it is otherwise difficult to find a valid configuration with our constraints on both boundary and region.

4.4 Conclusion

In this chapter, we have proposed a novel approach to glass segmentation with consumer RGBD cameras. By setting up an MRF which jointly encodes boundary fragment and superpixel properties and constraints, we proposed a global optimization procedure for glass detection, segmentation and recovery of the noisy depth maps. We validated the efficacy of this approach on our new RGBD Glass dataset, which shows the superior performance of our method.

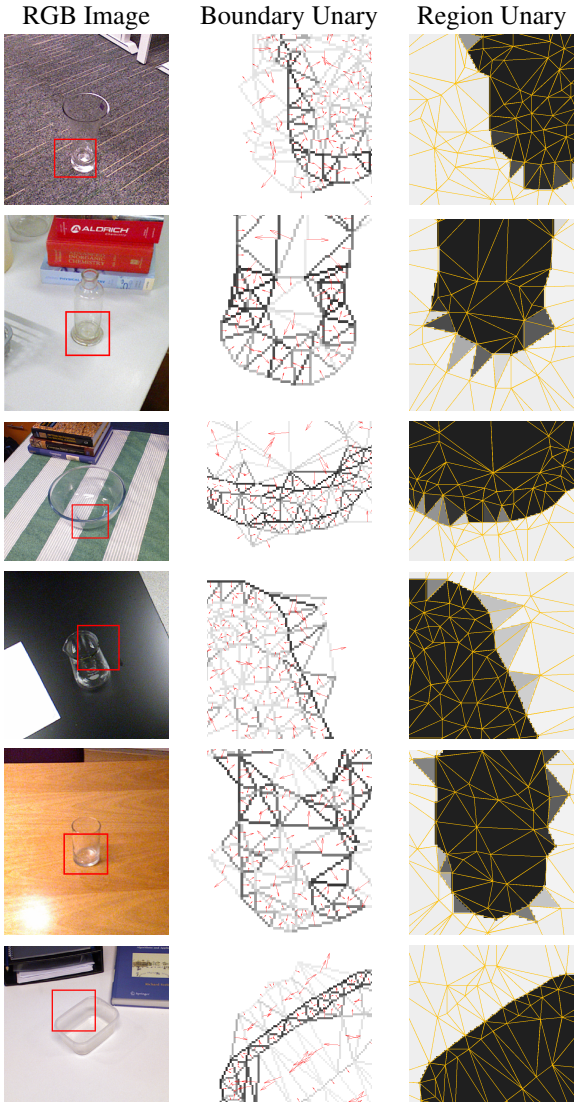


Figure 4.10: Examples of boundary and region unary terms (magnified, the viewing window is marked as a red bounding box in the RGB images). The boundary orientation is shown as a red arrow pointing towards glass regions. Local boundary and region classifiers provide complementary information for glass object segmentation. See text for details.

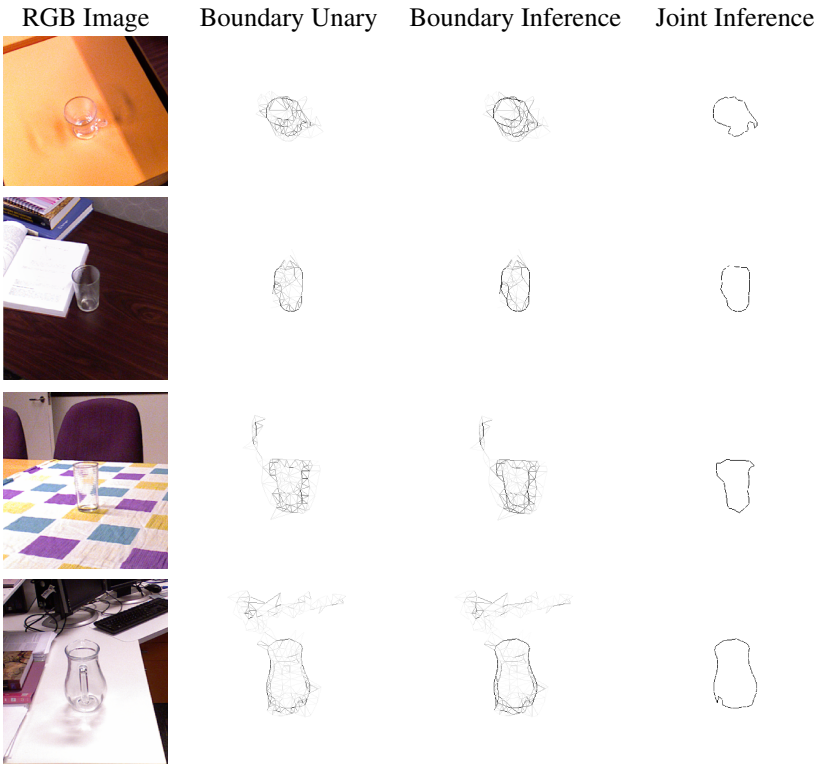


Figure 4.11: Examples of iterative joint inference. While the initial boundary inference smoothes the unary classifier output, we obtain much cleaner boundary inference results with the joint inference. See text for details.

Chapter 5

Glass Object Segmentation by Label Transfer on Joint Depth and Appearance Manifolds

5.1 Introduction

In this chapter, we continue our effort to localize glass objects with RGBD images. In Chapter 4 we proposed a joint inference algorithm for glass object segmentation. We exploited the missing-vs-nonmissing pattern in the depth channel which can be used as an effective feature to approximately localize glass objects. Despite our ability to produce high quality segmentation from the local estimates through constraints on the joint configurations of the boundary and region, this method has difficulty in handling glass objects with weak RGB cues or strong local deformation of depth missing patterns, as shown in Figure 4.9 and 5.6. One main issue in these cases is that the local estimates are too noisy due to the very large appearance variations at glass boundaries, as shown in a few image patch examples in Figure 5.1. Although relative features focusing on the difference between image patches on both sides of the boundary can reduce feature variation, it is still difficult to train a generic classifier because glass overlays can introduce many different effects such as blurring, highlights, texture distortion, depth missing, etc. The local effects with an individual object instance may be selective and depend on a number of factors including the glass material, illumination, viewpoint, etc. It is therefore difficult to single out each effect and extract more expressive features associated with it.

As a result, we move our focus to methods that are able to deal with large feature variations. Particularly, we propose an image adaptive approach to predicting glass boundaries. Our focus is still on the scenario in which inputs are captured with an RGBD camera. The main idea of our method is to generate boundary proposals based on a nonparametric feature model. Our model is represented by a joint depth and appearance feature manifold, on which each point is the glass boundary feature of an image patch pair. The boundary label of any pair of neighboring patches is predicted by a weighted voting of its nearest neighbors on the feature

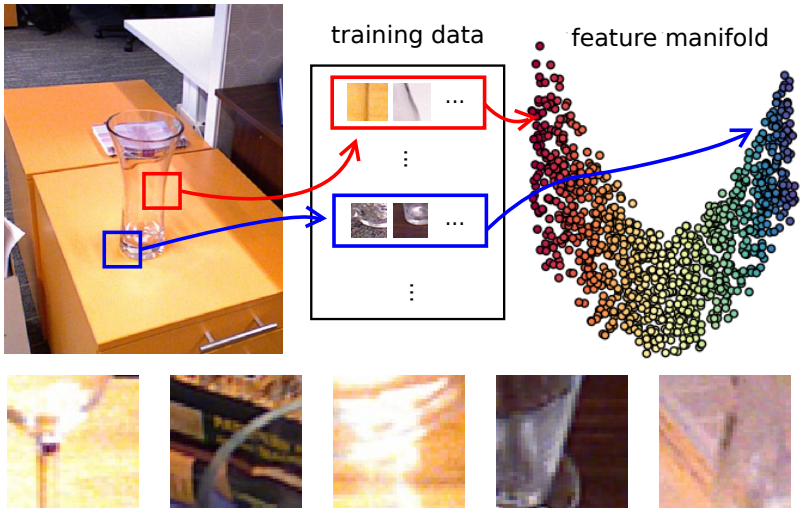


Figure 5.1: **Top:** Illustration of feature manifold based glass boundary classification. We use a learned feature manifold to match boundary fragments in a test scene (shown as image patches) to a training set in order to predict their labels. **Bottom:** Large variation on glass boundaries: patches examples.

manifold. The distance metric on the manifold is learned in a supervised manner.

We then integrate the locally adapted glass boundary predictor into a superpixel-based pairwise Markov Random Field (MRF) for glass object detection and segmentation. The MRF labels every superpixel as glass vs non-glass, in which our boundary prediction is used to modulate the smoothing terms in random fields. As we will show in the experiments, our approach generates more accurate glass boundary predictions, which simplifies the overall model structure and the inference algorithm.

Our work is inspired by the recent progress in nonparametric, data-driven approaches on label transfer and propagation (e.g., [199, 115]). These methods first retrieve a subset of training images based on global image statistics, and use the retrieved images for label transfer on the superpixel level for dense image parsing. In particular, Fathi et al. [44] take a semi-supervised learning approach to learn a metric for label propagation in videos.

Our contributions in this chapter are threefold. Firstly, we propose novel features for glass object segmentation and a flexible feature pool for improving performance. Secondly, our work is the first to explore nonparametric label transfer within the context of glass detection, and exploit a joint depth-appearance manifold for transductive learning. Lastly, we integrate our locally adapted glass boundary detector into an MRF framework for glass object detection and segmentation, achieving a clear improvement to the state-of-the-art on a challenging RGBD Glass dataset in terms of accuracy and speed.

The rest of this chapter is organized as follows. We describe the proposed approach in details in Section 5.2, followed by experimental evaluation and analysis in Section 5.3 and a

brief conclusion in Section 5.4 .

5.2 Our approach

The main idea of our method is to treat every pair of neighboring superpixels as a data unit, and build a feature manifold of such pairs for transferring boundary labels. We design a relative feature for the superpixel pairs in a joint appearance and depth feature space to capture the difference caused by glass overlay. The transferred boundary label predictions are then integrated into a pairwise MRF to generate spatially coherent glass object segmentation.

5.2.1 Superpixels and features

Superpixels. Our first step is to run SLIC [2] and partition image into superpixels. We choose SLIC as it better follows glass and depth boundaries overall compared to alternatives (e.g., edge detector and triangulation used in Chapter 4). Note that superpixel boundaries should follow depth boundaries to facilitate depth reconstruction as a post-processing step. We compare SLIC with our triangulation-based method in Section 5.3.2.

Boundary features. Suppose we have an input image I and denote each superpixel with a single letter (e.g., i), then any boundary fragment can be indexed by two letters (e.g., ij, indicating i and j are neighbors and ij is the shared boundary between them). The local boundary feature vector $\mathbf{f}_{ij}$ includes: (i) Hue and saturation [135]; (ii) Blurring [135]; (iii) Blending and emission [3]; (iv) Texture distortion [135, 130]; (v) Missing depth (same as described in Chapter 4). See Section 4.2 from the previous chapter for details on these features. Note that the above features are extracted from a pair of windows on either side of a boundary fragment, and we use the non-oriented relative ratios in our feature vector.

In addition, we add three more depth-aware features in this work:

- (vi) Color histogram on boundary: a histogram of 30 bins with 10 each for red, green and blue channels respectively.
- (vii) HOG [38] on depth data: for each feature window we extract HOG features on depth maps with 2×2 or 3×3 cells depending on the scale of feature windows.
- (viii) Range (depth) histogram [98]: a histogram of 20 bins with each bin having a range of 0.15m.

Note that feature (vii) captures surface orientations and depth discontinuities which may be repeating in visually similar local structures. This feature has been proven effective in other object detection tasks, and we refer readers to Section 2.1.3 for details. We augment the image cues by sampling features on multiple scales and at multiple locations. Specifically, we augment the feature set in the following two ways:

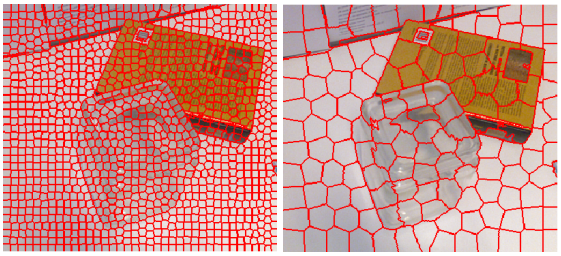


Figure 5.2: Example of SLIC [2] superpixels with initial region sizes of 10 px (left) and 30 px (right) respectively.

(A) We run superpixelization at a coarse scale and a fine scale, as shown in Figure 5.2. Label transfer was performed separately on each scale (see details in Section 5.2.2). Afterwards, we merge the local glass boundary proposals from the coarse into the fine scale. Merging is based on the image spatial location, subject to a fixed pixel error tolerance.
(B) Multi-scale and pattern-based features are extracted for each boundary fragment. The multi-scale extraction involves features within windows at 2-times and 3-times the default feature window size, while the pattern-based feature sampling further augments the features with randomly selected rectangular patterns, at both sides of a boundary fragment, similar to TextonBoost [185].

5.2.2 Boundary label transfer

The main challenge of glass object segmentation lies in boundary detection, as the refractive properties of glass lead to large variations in the relative features (i.e., features computed on the difference at both sides of glass boundaries). Instead of building a single classifier in the feature space, we explore the local feature manifold, and label transfer based on local matches on the feature manifold.

More formally, let e_{ij} be a binary variable associated with boundary fragment ij, and $e_{ij} = 1$ if the fragment is part of glass boundary and 0 otherwise. In order to reason about the label for e_{ij}, we denote $\mathcal{K} = \{uv\}$ as a set of boundary fragments from training data with known labels $\mathbf{L}_{\mathcal{K}} = \{l_{uv}\}$. A weighted voting scheme is adopted to estimate $P(e_{ij}|I)$ using $\mathcal{K}$:

$$\begin{aligned} P(e_{ij}|I) &\propto \sum_{uv \in \mathcal{K}} w_{ij,uv} \cdot \delta(e_{ij} = l_{uv}) \\ &= \sum_{uv \in \mathcal{K}} e^{-(\mathbf{f}_{ij} - \mathbf{f}_{uv})^T \Sigma (\mathbf{f}_{ij} - \mathbf{f}_{uv})} \cdot \delta(e_{ij} = l_{uv}) \end{aligned} \tag{5.1}$$

where $\mathbf{f}_{ij} \in \mathbf{R}^{\mathbf{N}}$ and $\mathbf{f}_{uv} \in \mathbf{R}^{\mathbf{N}}$ are local feature vectors for boundary fragments ij and uv, Σ is a diagonal matrix with diagonal elements being the distance between $\mathbf{f}_{ij}$ and $\mathbf{f}_{uv}$, and $\delta(\cdot)$

is an indicator function. The weight $w_{ij,uv} = \exp\big(-(\mathbf{f}_{ij} - \mathbf{f}_{uv})^T \Sigma (\mathbf{f}_{ij} - \mathbf{f}_{uv})\big)$ is based on a distance metric learned on the feature manifold. Since we assume Σ is diagonal, we can rewrite the above equation into $w_{ij,uv} = \exp\big(-\sum_{d=1}^{N} \sigma(d)(\mathbf{f}_{ij} - \mathbf{f}_{uv})^2\big)$ where $\sigma(d)$ is the d-th diagonal element of Σ. Therefore, $|\sigma(d)|$ essentially indicates the "importance" of the d-th dimension of $\mathbf{f}_{ij}$ for boundary label transfer. We visualize the accumulated values of $|\sigma(d)|$ in Section 5.3.4 in an attempt to decode the relative importance of various features used in our model. We only estimate $P(e_{ij}|I)$ with k-nearest neighbors, i.e., $|\mathcal{K}| = k$, and members in $\mathcal{K}$ have the k highest weights $w_{ij,uv}$. We set $k = 10$ in our experiments.

The weight $w_{ij,uv}$ is learned with manually labeled samples, by adopting the strategy proposed in [44] which casts a distance metric learning problem as a binary classification task. Let uv and $u'v'$ be two boundary fragments from training data. We define a target metric as $w_{uv,u'v'} = 1$ if $l_{uv} = l_{u'v'}$, and $w_{uv,u'v'} = 0$ otherwise. Learning of Σ is performed with linear regression on training data. Intuitively, we prefer the similarity weight $w_{uv,u'v'}$ to be high if both fragments are part of glass boundary, or both are not. In Section 5.3.4, we compare learning a single dataset-wide target metric and a set of subset-specific metrics.

5.2.3 Object model and inference

Our glass object model follows a pairwise MRF [15] formulation with unary and pairwise terms on superpixel nodes. Denote the set of all image sites (i.e., superpixels) as $\mathcal{S}$. Let $\mathcal{G}$ be the neighborhood graph on $\mathcal{S}$ based on the spatial relationship. Denote $\mathbf{D} = \{d_i\}$ as a set of binary variables associated with superpixels, and we assume a binary state space $\{0, 1\}$ for d_i, with 1 indicating glass regions. Our energy function can be written as follows:

$$E(\mathbf{D}) = \sum_{i \in \mathcal{S}} \phi_D(d_i; I) + \beta \sum_{(i,j) \in \mathcal{N}} \psi_D(d_i, d_j; I) \tag{5.2}$$

where β is the weighting coefficient between unary and pairwise terms, and $\mathcal{N}$ is the neighborhood. The unary term $\phi_D(d_i; I)$ is the negative log-likelihood given by a local SVM classifier:

$$\phi_D(d_i; I) = -\log(P(d_i|\mathbf{g}_i)) \tag{5.3}$$

where $\mathbf{g}_i \in \mathbf{R}^{\mathbf{M}}$ are features extracted for superpixel d_i. The features we use for superpixels only include (i), (v), (vii), and (viii) of those used for boundary (see Section 5.2.1 for all boundary features). We also extract multi-scale image features for each superpixel. The implementation of this unary term is similar to the superpixel unary potential in Section 4.2.

For the pairwise term $\psi_D(d_i, d_j; I)$, we utilize $P(e_{ij}|I)$ estimated by boundary label transfer to modulate the smoothing prior. We set the pairwise potential between neighboring super-

pixels d_i and d_j as follows:

$$\psi_D(d_i, d_j; I) = \delta(d_i \neq d_j) P(e_{ij} = 0|I) \\ + \alpha\delta(d_i = d_j) P(e_{ij} \neq 0|I) \tag{5.4}$$

where $P(e_{ij}|I)$ is estimated by the locally adapted k-nearest neighbor voting described in Section 5.2.2. This pairwise term is a simplified version of the superpixel pairwise potential we used in Section 4.2 as we remove the orientation estimates of boundary fragments. It penalizes two scenarios: (1) where labels of two adjacent superpixels are different and the boundary fragment in between is not a glass boundary and (2) where labels of two adjacent superpixels are the same but there is a glass boundary in between. In the experiments that follows, we use Loopy Belief Propagation (LBP) [15] to compute the marginals for MRF inference. Model parameters α and β were learned through cross-validation.

5.3 Experimental evaluation

5.3.1 Data specifications and setup

We test our approach on the RGBD Glass dataset used in Chapter 4, which contains 171 RGBD image pairs with 43 distinct glass objects. We follow the training/test data split in Chapter 4. As shown in Figure 5.3, the dataset was collected in various scene categories and many of the glass objects are challenging to localize due to background clutter. The dataset consists of three subsets: floor, laboratory and office, each contains images taken from a different environment.

We use SLIC [2] to generate superpixels, with initial region sizes 10 and 30 px. The pixel error tolerance for merging the boundary proposals from the coarse superpixel layer is set to 5 px. For local boundaries, we extract features on 3 different scales, and each scale consists of 50 randomly selected rectangular patterns on both sides of the detected boundary, resulting in 300 feature windows. The local superpixel feature set is also generated at 3 scales, and we use an SVM with an RBF kernel for the unary potential in our MRF. The model parameters α and β chosen by cross validation were 0.5 and 0.25 respectively.

5.3.2 Ablation studies

In order to verify the efficacy of the various improvements on superpixels, features and label transfer we proposed, we present our findings from three ablation studies in this section.

Firstly, we compare the glass boundary recall rates for SLIC [2] superpixels against triangulation-based image partitioning we used in Chapter 4. The recall rates presented in Table 5.1 cap the maximum attainable recall for the rest of the system. If a glass boundary segment is not captured by superpixelization, it is impossible to rectify it later using our method. Therefore, it is

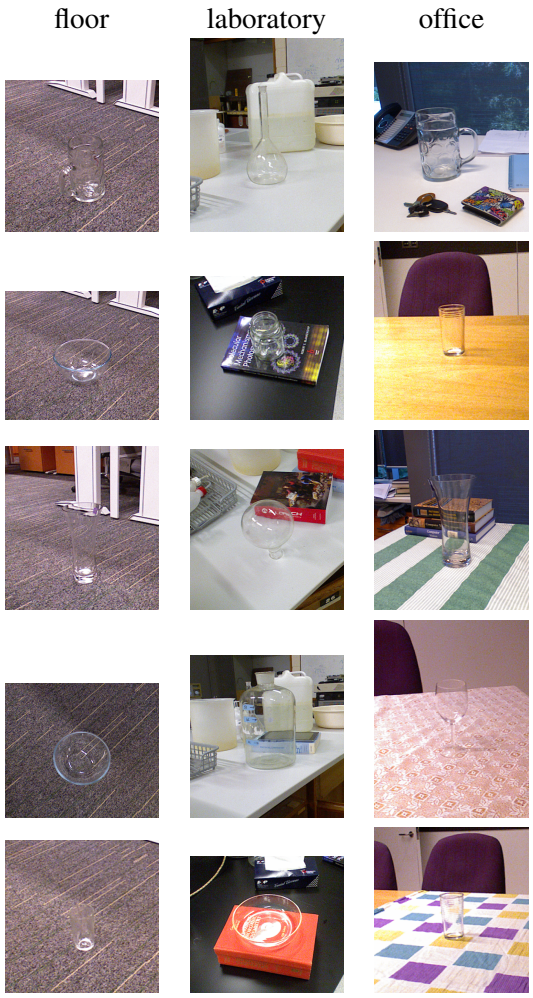


Figure 5.3: Example images from the three subsets of our RGBD Glass dataset. See text for details.

sensible to choose a method which gives the highest glass boundary recall rate. As shown in Table 5.1, SLIC performs slightly better than triangulation particularly when the pixel error tolerance e_{max} is small. We note that although we choose $e_{max} = 5$ px in our other experiments, higher recall at an even lower tolerance (e.g., $e_{max} = 3$ px) means SLIC follows boundaries more closely, generating more visually pleasing results in general. We show some qualitative examples in Figure 5.4. Note that clustering of pixels locally help SLIC achieve better results where the intensity gradient is weak.

Secondly, we wanted to justify our design of the feature pool. More specifically, we report the boundary label transfer performance improvements obtained from (1) image partitioning at multiple scales, (2) sampling features on multiple scales, and (3) sampling features at multiple locations against a baseline without any of these components. Table 5.2 summarizes our results. As can be seen from the table, all three components contribute to the glass boundary

Table 5.1: Glass boundary recall rates for triangulation-based method used in Chapter 4 versus SLIC [2] used in this chapter. e_{max} denotes the pixel error tolerance. See text for details.

e_{max}	3 px	5 px	10 px
Triangulation	0.8923	0.9592	0.9897
SLIC [2]	0.9211	0.9757	1.0000

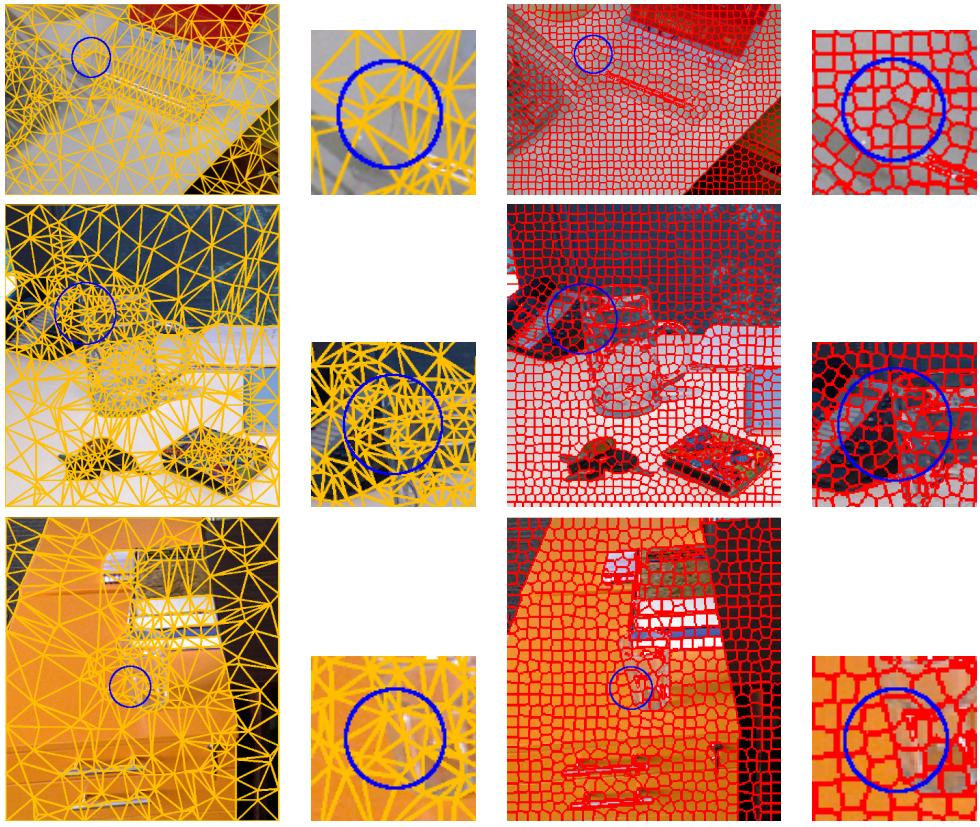


Figure 5.4: Qualitative comparisons between triangulation-based image partitioning method (left two columns, partitions shown in orange) used in Chapter 4 and SLIC [2] superpixels (right two columns, partitions shown in red). Note how SLIC superpixels more closely follow glass boundaries, especially in regions highlighted with blue circles. The SLIC initial region size shown here is 10 px.

classification performance, with sampling features at multiple locations being the most effective. In particular, the three components combined provide a large feature pool for distance metric learning, which yield superior results compared to the baseline with a smaller fixed-sized feature pool. In fact, a large and flexible feature pool is essential for both our models in Chapter 4 and in this chapter as it allows the model learning process to pick up the most effective features when appearance variations at glass boundaries are large.

Table 5.2: Precision (in percentage %) and F-measures at 25%, 50% and 75% recall for glass boundary label transfer. Column Base refers to baseline performance without the feature pool. Columns (1) through (3) refer to (1) image partitioning at multiple scales, (2) sampling features on multiple scales, and (3) sampling features at multiple locations. Column Full refers to our full model with all of the three components. See text for details.

	Precision (%)					F-measures				
	Base	(1)	(2)	(3)	Full	Base	(1)	(2)	(3)	Full
25% Recall	75.7	79.3	79.0	84.4	93.5	37.59	38.01	37.98	38.58	39.45
50% Recall	53.6	56.8	55.8	59.2	61.9	51.74	53.17	52.75	54.23	55.31
75% Recall	25.5	26.9	27.7	30.2	31.8	38.07	39.60	40.48	43.07	44.64

Table 5.3: Precision (in percentage %) and F-measures at 25%, 50% and 75% recall for glass boundary label transfer. The first three columns refer to scenarios in which we remove certain depth-aware features. Specifically, they refer to No Color histogram (NC), No HOG on depth data (NH) and No Range histogram (NR), respectively. The fourth column, kNN, refers to the case where we disable the distance metric learning. The final column, Full, refers to our full model. See text for details.

	Precision (%)					F-measures				
	NC	NH	NR	kNN	Full	NC	NH	NR	kNN	Full
25% Recall	89.5	91.2	88.1	67.9	93.5	39.08	39.24	38.95	36.55	39.45
50% Recall	60.6	61.5	59.9	43.4	61.9	54.80	55.15	54.51	46.48	55.31
75% Recall	30.7	31.4	31.4	16.5	31.8	43.57	44.31	44.23	27.02	44.64

Finally, we show that the choice of depth-aware features and the distance metric learning for label transfer are also important to our performance. In particular, we note that our method is equivalent to a $k-$nearest neighbor classifier if we assign uniform values to the weight coefficients $\sigma(d)$ in Σ. In this case, we will be disabling the distance metric learning step and working in the original high dimensional feature space instead of the joint depth and appearance manifold. Table 5.3 reports precision and F-measure values when we disable the depth-aware features or the distance metric learning, compared to the performance of our full model. As can be observed from the results, all three depth-aware features contribute to precision rates slightly. Perhaps more important is the distance metric learning, as it provides a way to "select" more important axes from a high dimensional feature space. We will look into the feature selection mechanism with another experiment in Section 5.3.4.

5.3.3 Results and discussion

The quantitative and qualitative results using our method are shown in Figure 5.5 and Figure 5.6, respectively. We compare our approach with the joint inference approach proposed in Chapter 4, referred as "Joint". We also show the performance based on the boundary classifier

Table 5.4: F-measures at 50% recall for boundary and region accuracy metrics. The final row (Bound Region) is based on region pixel accuracy in the glass boundary neighborhoods (i.e., regions within 10 px of ground-truth glass boundaries).

	Joint Unary	Ours Unary	Joint Inference	Ours Inference
Bound	44.38	55.31	62.27	64.02
Region	55.84	57.27	65.96	66.49
Bound Region	-	-	46.98	62.33

Table 5.5: Per-image runtime statistics for the method in Chapter 4 and the proposed method. On average the proposed method is about 8 times faster. See text for details.

	Local (s)	Inference (s)	Total (s)
Joint	0.257	14.542	14.799
Ours	0.928	0.898	1.826

output, and see why our method is capable of producing superior results with a simpler MRF model. These local boundary classifier outputs are referred to as "Unary" in the figures.

The overall precision and recall on the RGBD Glass dataset is shown in Figure 5.5. The left and middle plots present the precision-recall figures under two metrics: boundary pixel accuracy and region pixel accuracy. For boundary accuracy, we use the benchmark utility from [132] and follow the matching procedure. We compute a list of correspondences below a distance threshold between the boundary estimate and the ground-truth boundary map. As both the method from Chapter 4 and our method are capable of recovering major glass surfaces (as a result of using depth features), region pixel accuracy can be less sensitive to noise at glass boundaries as it measures pixelwise accuracy over the entire image. Therefore we additionally present another region pixel accuracy based result in the right plot which only considers pixels within 10 px of ground-truth glass boundaries. This metric directly reflects the region recovery quality near glass boundaries, which is vital to accurately recovering the shape of glass objects. We achieved superior results on both glass boundary detection and final inference results. While joint inference is able to boost the performance of noisy unary responses, having cleaner boundary proposals allows us to adopt simple and more efficient inference algorithms. F-measures corresponding to Figure 5.5 at 50% recall rate are reported in Table 5.4, where $F = 2/(1/Pr + 1/Rc)$.

Figure 5.6 presents some hard examples for comparison between both methods. Note that the noisy boundary estimate is the main reason for failure cases of the joint inference method. The proposed method, on the other hand, shows reliable and accurate prediction results. Our method eliminates circumstances where predictions on the boundary nodes and superpixel nodes are inconsistent (e.g., the second example in Figure 5.6). As we can see, the success of

the proposed method is primarily due to cleaner glass boundary proposals based on the learned feature manifold. Even sophisticated inference is unlikely to recover the glass boundary if the initial estimates are too weak or severely contaminated by their neighbors.

Finally, we compare the runtime of both methods with our mixed MATLAB and C (mex) implementation. The runtime was broken down into two major components: local boundary estimation and inference. The local part includes pre-processing, feature extraction and local classification. The proposed method takes longer as we need to extract more features. The inference part for the method in Chapter 4 requires up to 20 runs for LBP or mean-field approximations, while ours only requires one. The post-processing (i.e., plane segmentation and depth recovery) takes only a fraction of the total runtime, and therefore is not timed. We report the average runtime per image on an Intel i3 laptop in Table 5.5. Note that with a native implementation, our method may be further accelerated for real-time applications due to the simple nature of the inference process.

5.3.4 Building subset-specific manifolds

So far we showed how to create a large, flexible feature pool for distance metric learning. Intuitively, not all features are equally important; in fact, many of them may be less effective due to the large appearance variation issue we discussed at the beginning of the chapter. Therefore, we need to adopt a learning technique to determine their relative importance. More importantly, working with a learned manifold can be more effective than working in the original feature space. In this section, we investigate the benefits of working with a learned feature manifold more closely. In particular, we show that learning subset-specific distance metrics can further improve our achieved results in certain scenes.

An alternative view to the large appearance variation issue is that, for a specific scene setup (e.g., images with similar objects, background, viewpoint, illumination, etc.) we only have a limited amount of training data available. Fortunately, linear regression used in our distance metric learning works well with a limited amount of data, which has also been found in [44]. In addition, it is difficult to find a single set of feature weight coefficients $\sigma(d)$ that can work for a variety of scenes. Therefore, in the followings we learn subset-specific distance metrics and compare their performance to learning a single metric across all scenes.

More specifically, the glass dataset we use in this work contains three subsets: floor, laboratory and office, containing 16, 29 and 126 images respectively. Figure 5.3 shows some example images from each subset. The floor subset contains images of different glass objects with an identical background observed from similar viewpoints. The laboratory subset contains pictures taken at a university chemistry laboratory. The office subset contains common glass objects in an office environment, which has the most diverse scenes. As we will discuss shortly, the different scene characteristics of these subsets also affect glass boundary

Table 5.6: Precision (in percentage $\%$) at 25%, 50% and 75% recall for glass boundary label transfer on the three subsets of our RGBD Glass dataset. Columns under "Single manifold" refer to results from our proposed approach with a single manifold built on the entire dataset. Columns under "Subset-specific manifold" refer to results obtained with subset-specific manifolds.

	Single manifold			Subset-specific manifold		
	Floor	Lab	Office	Floor	Lab	Office
25% Recall	78.5	88.4	**95.9**	**83.6**	**88.7**	95.4
50% Recall	49.3	**58.6**	**63.3**	**53.7**	58.4	61.0
75% Recall	22.1	31.5	**32.5**	**24.5**	**31.6**	32.2

classification performance.

Table 5.6 reports the precision values under different glass boundary recall rates on each subset of our RGBD Glass dataset. As a baseline, results from our full model are listed under "Single manifold". We use a single distance metric learned from the entire dataset and apply it to each of the subsets. On the contrary, results under "Subset-specific manifold" are obtained with subset-specific distance metrics. For the smaller and the more visually homogeneous Floor subset, building a subset-specific manifold clearly is the better choice, as it provides an average of 4% precision gain. This is similar for the Laboratory subset, with the subset-specific manifold performs slightly better than building a single manifold across the entire dataset. However, for the largest subset Office the subset-specific manifold does not offer a performance improvement, perhaps due to the diverse nature of scenes contained in the subset. In addition, we report the normalized accumulated weight $|\sigma(d)|$ in Figure 5.7 to visualize the differences between dataset-wide and subset-specific manifolds. As we can see, not all features have comparable weights. For example, the missing depth feature is the most effective in boundary label transfer, in line with our findings in Chapter 4. It should be noted that subset-specific distance metrics capture some of the features which get less attention under a single manifold setup.

Conceptually, it is more preferable to build scene-specific instead of subset-specific manifolds, as the scene setups from within a subset can be large. In fact, the idea of building scene-specific manifolds is conceptually similar to the scene retrieval step in prior literature on nonparametric image parsing such as [199] and [115]. In this work we focus on validating that (1) the distance metric learning used in our work is suitable for learning from a limited amount of training data, and (2) using a subset-specific manifold produces superior results for certain scenes. We would like to more thoroughly explore scene retrieval as a future work which would potentially produce superior results on difficult scenes from our dataset.

5.4 Conclusion

In this chapter, we explored a feature based label transfer approach to glass object segmentation. We propose a novel depth and appearance feature representation for glass boundary and surface detection, and learn a distance metric on the relative feature manifold for glass boundary label transfer. By integrating our glass boundary proposals into a pairwise MRF model, we obtained a significant improvement to the state-of-the-art on challenging examples in an RGBD Glass dataset. Our method can be used as a starting point for more sophisticated algorithms that involve glass surface reconstruction. For future directions, we would like to explore scene retrieval with our method, and learning depth-encoded feature manifolds with weakly labeled data.

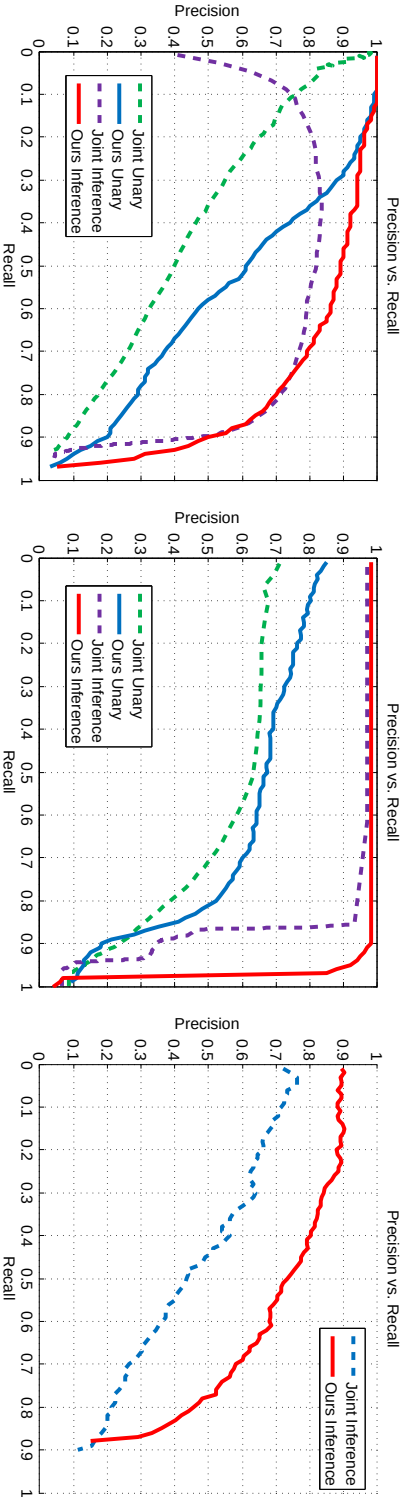
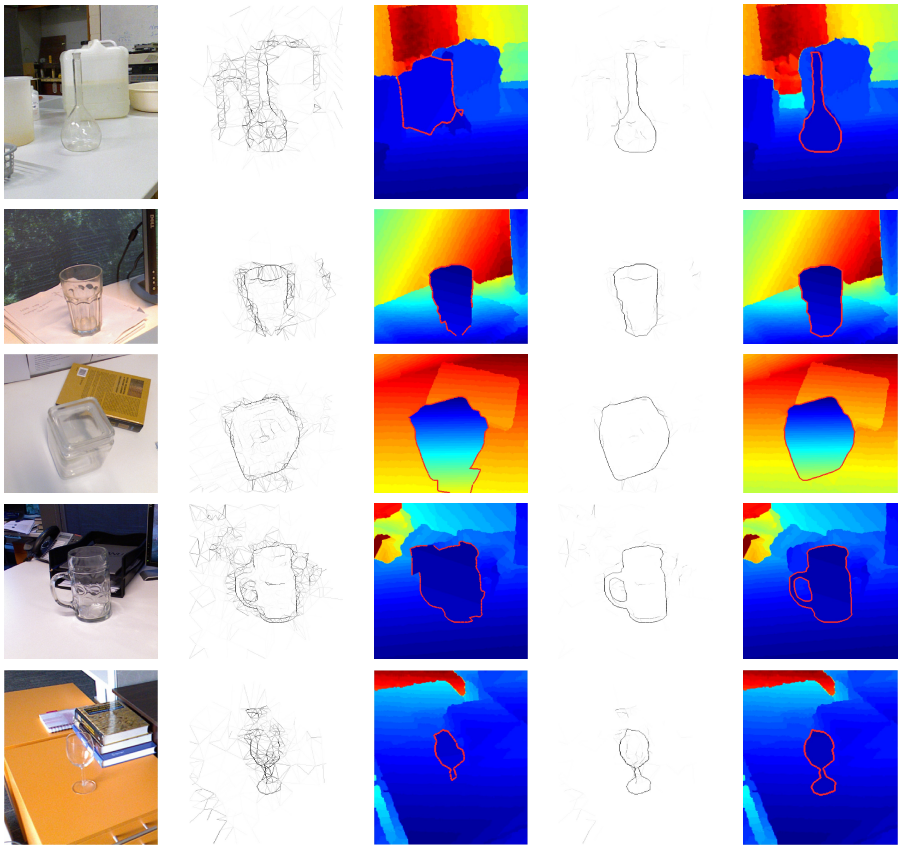


Figure 5.5: The overall precision and recall on RGBD Glass dataset for various methods. **Left**: Performance based on boundary pixel accuracy. **Middle**: Performance based on region pixel accuracy on the whole dataset. **Right**: Performance based on region pixel accuracy in the glass boundary neighborhoods (i.e., regions within 10 px of ground-truth glass boundaries).



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

Figure 5.6: Hard examples of glass detection results on the RGBD glass dataset. Column **(a)**: RGB image frame. **(b)**: Unary responses from local glass boundary classifiers in Chapter 4. **(c)**: Joint inference and depth recovery results in Chapter 4. **(d)**: Glass boundary label transfer results. **(e)**: Inference and depth recovery results with the proposed method. Note that missing depth readings are recovered by a piece-wise planar model for glass region and smoothed out using a median filter elsewhere.

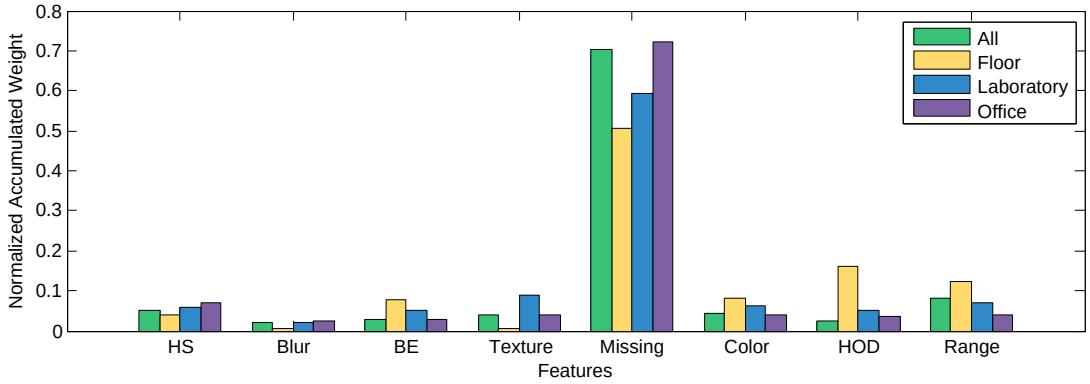


Figure 5.7: Normalized accumulated weight for different features on the RGBD Glass dataset (All) and its subsets (Floor, Laboratory and Office). We add up the absolute values of weights for feature dimensions belong to specific types of features. The resulting bar graph illustrates the relative accumulated "importance" of various types of features used in our method. The feature types are: Hue and Saturation (HS), Blurring (Blur), Blending and Emission (BE), Texture Distortion (Texture), Missing Depth (Missing), Color histogram (Color), HOG on depth data (HOD), Range histogram (Range). See text for details.

Chapter 6

Laplacian Margin Distribution Boosting for Learning from Sparsely Labeled Data

6.1 Introduction

In previous chapters we addressed object detection and segmentation with partial object visibility and limited sensory data availability. We continue our discussion in this chapter by looking at another issue in relation to partial information. In many real-world applications complete and accurate ground-truth annotations are difficult and expensive to obtain, usually requiring extensive human effort. Although some previously impractical large-scale labeling tasks have been made possible by online crowdsourcing services such as the Amazon Mechanical Turk [1], the monetary cost involved scales with the number of images in a dataset. Also, for annotators without domain knowledge, the quality of their labelings varies. Semi-supervised learning algorithms that seek to make use of unlabeled data for training are an appealing alternative to supervised learning in these scenarios.

In this chapter, we propose a semi-supervised version of a margin distribution-based variant of boosting algorithms. We choose to base our work on boosting algorithms because they have achieved great popularity in a spectrum of computer vision problems due to their good generalization, robust performance, and intrinsic feature selection mechanism. In particular, they have been an integral part of many object detection and segmentation systems (e.g., [207, 203, 72, 185, 73, 42]). Despite their success, the classic AdaBoost and its variants suffer from two disadvantages in real world applications. First, the exponential loss and greedy nature of its learning algorithm tend to generate a classifier with many weaker learners, which can be inefficient and prone to overfitting. Also, boosting usually requires a large number of training examples to achieve high accuracy. As discussed, ground truth labeling is usually scarce and difficult to obtain in practice.

Our work aims to address those issues within a unified framework based on the margin

distribution theory of boosting [178, 168, 183]. One key observation is that the appealing properties of boosting are closely related to the *margin distribution* (MD) instead of solely the minimum margin [168] – which are commonly used in margin-based classification. It has been shown that the margin distribution seems to play a more important role in attaining better overall performance empirically and provides a tighter generalization bound in theory [54, 168]. Therefore, several papers advocate optimizing MD-based criteria to improve the test accuracy of boosting-like algorithms [120, 54, 182]. Notably, Shen and Li [182] proposed a totally corrective boosting, termed MDBoost, to maximize the average margin while minimizing margin variance. The new boosting method achieves competitive performance and faster convergence (i.e., fewer weak learners) on several classification tasks.

While the additional margin variance provides a better measure of the margin distribution, the overall criterion is based on the second-order statistics only, thus lacks capacity to capture finer-scale structure of the distribution. Manifold learning refers to a collection of algorithms for non-linear dimension reduction. Laplacian Eigenmap is an important manifold learning algorithm that finds a low dimensional representation of a dataset using a spectral decomposition of the graph Laplacian. The graph Laplacian can be considered as a discrete approximation of the low dimensional manifold in the high dimensional space. More importantly, the graph ensures that points close to each other on the manifold are mapped close to each other in the low dimensional space, preserving local distances [9]. Inspired by this, we propose to improve MDBoost by incorporating a local representation of margin variance, in which only neighboring points on the data manifold contribute to the variance computation. Intuitively, the data-dependent margin variance may give a better description of the margin distribution. Due to its resemblance to the Laplacian Eigenmap [10], we refer to this new boosting approach as *Laplacian MDBoost*.

More importantly, our learning criterion can be naturally generalized to the semi-supervised learning scenario. Given both labeled and unlabeled data, we augment the supervised learning criterion with a graph Laplacian-based regularization term, which encourages the classifier outputs on unlabeled data to satisfy the data manifold constraint. This combined learning criterion provides a coherent framework and admits a simple convex quadratic dual formulation such as MDBoost. We employ a column-generation (CG) based optimization procedure to incrementally add informative weak learners, yielding a boosting-like algorithm.

We empirically demonstrate that the supervised Laplacian MDBoost is better than or comparable to AdaBoost(-CG) [183], LPBoost [39] and MDBoost in terms of classification performance on most UCI datasets [162]. In addition, we design a set of semi-supervised learning tasks based on UCI datasets, the YouTube Celebrities Face datasets [83], and our RGBD Glass dataset. We compare the Semi-supervised Laplacian MDBoost with two recent approaches to learning from partially labeled data: LLGC [236] and SemiBoost [131]. The results show the Semi-supervised Laplacian MDBoost outperforms the baseline methods on most of these

datasets.

We organize the rest of this chapter as follows. In the next section, we derive the supervised and Semi-supervised Laplacian MDBoost based on the dual formulation of optimizing a novel margin distribution cost. We demonstrate the performance of our approach by comparing with several recent (semi-)supervised boosting methods on UCI datasets, a video segmentation task, and an RGBD glass object segmentation problem in Section 6.3. Finally, Section 6.4 summarizes our conclusion and discusses future work.

6.2 Our approach

6.2.1 Margin distribution and Laplacian MDBoost

We first review the key ideas of the margin distribution boost (MDBoost) in [182] and introduce some notation for formulating our Laplacian MDBoost. Let $\mathcal{D}_l = \{(\mathbf{x}_i, y_i)\}_{i=1,\cdots,M}$ be the training dataset, where $\mathbf{x}_i \in \mathcal{X}$ is the input feature vector and $y_i \in \{-1, +1\}$ is the output label. Given the training data, our goal is to train a classifier to assign binary label to any input vector $\mathbf{x}$. In the setting of boosting methods, the classifier consists of a weighted combination of weak learners.

More specifically, denote $h(\cdot) \in \mathcal{H}$ as a weak learner that maps an input vector $\mathbf{x}$ to binary output. We assume we choose K weak learners from the set $\mathcal{H}$ in our boosted classifier, and define the matrix $H \in \mathbb{Z}^{M\times K}$ to be all the possible predictions of the training data using weak classifiers. That is, $H_{ij} = h_j(\mathbf{x}_i)$ is the label ($\{+1, -1\}$) given by weak classifier $h_j(\cdot)$ on the training example $\mathbf{x}_i$. We also use $H_{i:} = [H_{i1}\ H_{i2}\cdots H_{iK}]$ to denote the i-th row of H, which constitutes the output of all the weak classifiers on the training example $\mathbf{x}_i$. Let α be the weight vector for the weak learners. We can write the output of the final classifier on any training data $\mathbf{x}_i$ as $H_{i:}\alpha$, and the so-called (unnormalized) *margin* at data $\mathbf{x}_i$ is defined as $y_i H_{i:}\alpha$.

Based on the margin distribution theory of boosting, MDBoost directly maximizes the average margin and minimizes the margin variance. Specifically, let ρ_i denote the unnormalized margin for the i-th example datum, i.e., $\rho_i = y_i H_{i:}\alpha, \forall i = 1, \cdots, M$. The cost function and the learning problem in MDBoost can be written as follows:

$$\begin{aligned} &\min_{\alpha}\ \frac{1}{2(M-1)}\sum_{i>j}(\rho_i - \rho_j)^2 - \sum_{i=1}^{M}\rho_i \\ &\text{s.t. } \alpha \succcurlyeq 0, 1^\top\alpha = D, \end{aligned} \tag{6.1}$$

where D is a regularization parameter. By defining a matrix $A \in \mathbb{R}^{M\times M}$, where

$$A = \begin{bmatrix} 1 & -\frac{1}{M-1} & \cdots & -\frac{1}{M-1} \\ -\frac{1}{M-1} & 1 & \cdots & -\frac{1}{M-1} \\ \vdots & \vdots & \ddots & \vdots \\ -\frac{1}{M-1} & -\frac{1}{M-1} & \cdots & 1 \end{bmatrix},$$

the optimization problem can be rewritten into the following form:

$$\begin{aligned} &\min_{\alpha} \ \tfrac{1}{2}\rho^\top A\rho - 1^\top\rho, \\ &\text{s.t. } \alpha \succcurlyeq 0, 1^\top\alpha = D, \\ &\qquad \rho_i = y_i H_{i:}\alpha, \forall i = 1,\cdots,M. \end{aligned} \tag{6.2}$$

It has been shown [183] the problem in (6.2) can be efficiently solved by considering its dual form, i.e.,

$$\begin{aligned} &\min_{r,u} \ r + \tfrac{1}{2D}(u-1)^\top A^{-1}(u-1), \\ &\qquad \text{s.t.} \sum_{i=1}^{M} u_i y_i H_{i:} \preccurlyeq r1^\top. \end{aligned} \tag{6.3}$$

The form of the dual problem allows us to incrementally search the solution space by the column generation technique. At each iteration, we obtain a new weak classifier through searching for the most violated constraint:

$$h'(\cdot) = \underset{h(\cdot)}{\operatorname{argmax}} \ \textstyle\sum_{i=1}^{M} u_i y_i h(\mathbf{x}_i). \tag{6.4}$$

While the MDBoost learning cost incorporates the margin variance information, the global variance can be restrictive and cannot describe the finer structure of the distribution beyond the second order statistics. We propose to use the "local" version of variance that considers the geometric properties of the data manifold. Specifically, we adapt the concept of graph Laplacian of data manifold [10], and use a data-dependent margin variance in the MDBoost learning criterion:

$$\begin{aligned} &\min_{\alpha} \ \frac{1}{2(M-1)} \sum_{i>j} w_{ij}(\rho_i - \rho_j)^2 - \sum_{i=1}^{M} \rho_i \\ &\text{s.t. } \alpha \succcurlyeq 0, 1^\top\alpha = D, \end{aligned} \tag{6.5}$$

where w_{ij} is an edge weight defined on a neighborhood graph that measures the adjacency

between $\mathbf{x}_i$ and $\mathbf{x}_j$. The *heat kernel*, given by $w_{ij} = \exp(\frac{-||\mathbf{x}_i - \mathbf{x}_j||^2}{t})$, is a typical choice for this weight that preserves local information optimally when we consider a certain graph mapping problem [10, 35]. Another common choice is to use a simple truncation function, i.e., $w_{ij} = 1$ if and only if $||\mathbf{x}_i - \mathbf{x}_j||^2 < \epsilon$, or $\mathbf{x}_i$ are among the k nearest neighbors (kNN) of $\mathbf{x}_j$. See Figure 6.1 for examples of different choices for w_{ij} for the graph Laplacian. Note that in [10] ϵ-or-kNN truncation is also combined with the heat kernel. We choose the heat kernel without truncation in our work as it yielded best results in our initial experiments. We refer to the new learning problem in (6.5) as Laplacian MDBoost.

Note that if we define the matrix $A = \{A_{ij}\}$ by the following terms,

$$A_{ij} = \begin{cases} w_{ij}, & \text{if } i \neq j, \\ \sum_{k=1, k\neq i}^{M} w_{ik}, & \text{if } i = j, \end{cases} \tag{6.6}$$

then we can derive new primal and dual problems with the same form as in (6.2) and (6.3). The dual problem can be solved with a column generation method such as in MDBoost. We notice that both MDBoost and Laplacian MDBoost in their dual form are regularized hard-margin LPBoost, but have different types of regularizer.

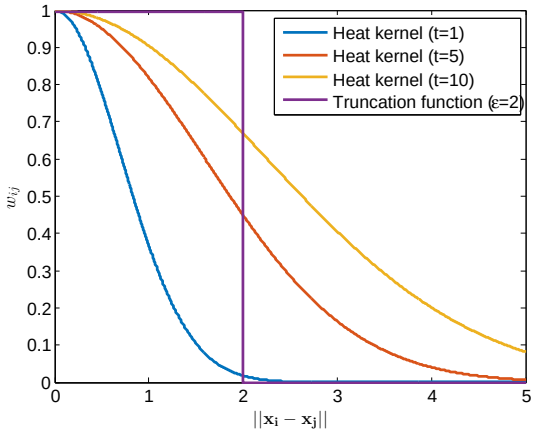


Figure 6.1: Examples of different choices of the edge weights w_{ij} for the graph Laplacian. See text for details.

6.2.2 Semi-supervised Laplacian MDBoost

The main idea in Laplacian MDBoost, which makes use of the geometric properties of data distribution, can be naturally extended to a semi-supervised learning setting. Assume we have an additional unlabeled dataset $\mathcal{D}_u = \{\mathbf{x}_i, i = M+1, \cdots, N\}$ and would like to use it to help improve the classification performance. Similar to [10], we incorporate a graph Laplacian-based regularization term into our objective function, which imposes a smoothness constraint over the class output on the unlabeled data w.r.t. the empirical estimate of data manifold structure.

Given a neighborhood graph defined on the dataset, we can define the graph Laplacian as $L = D - W$ where W is a $N \times N$ matrix and $w_{ij} = \exp(\frac{-||\mathbf{x}_i - \mathbf{x}_j||^2}{t})$, if x_i and x_j are adjacent and zero otherwise. D is a diagonal degree matrix given by $D_{ii} = \sum_i w_{ij}$. A smoothness regu-

larization term on the class output $f(\mathbf{x})$ can be written as $f^t Lf = \sum_{i,j=1}^{n}(f(\mathbf{x}_i) - f(\mathbf{x}_j))^2 w_{ij}$.

In Laplacian MDBoost, the class prediction $f(\mathbf{x}_i)$, denoted by f_i, is the combined prediction of all weak classifiers for the i-th example datum, i.e., $f_i = H_{i:}\alpha, \forall i = 1, \cdots, M$. By adding a smoothness term into the primal objective function, we derive the following learning criterion for Semi-supervised Laplacian MDBoost:

$$\begin{aligned} &\min_{\alpha} \frac{\sum_{i>j} w_{ij}(\rho_i - \rho_j)^2}{2(M-1)} + C\sum_{i>j} w_{ij}(f_i - f_j)^2 - \sum_{i=1}^{M}\rho_i \\ &\text{s.t. } \alpha \succcurlyeq 0, 1^\top \alpha = D, \end{aligned} \tag{6.7}$$

where D is also a regularization parameter as in (6.1). Here we have two quadratic terms: the first one corresponds to the margin variance of labeled data, while the second is the smoothness penalty on all data (including the labeled and unlabeled). C is the tradeoff parameter between the two terms.

Denote A_1 as the matrix defined in (6.6) on all the data points (including labeled and unlabeled), and A_2 as the $M \times M$ upper left corner of A_1 (suppose the data is sorted so that the labeled data are the first M elements when defining the graph Laplacian), our optimization problem can be rewritten into a concise form:

$$\begin{aligned} &\min_{\alpha} \ \tfrac{C'}{2} f^\top A_1 f + \tfrac{1}{2}\rho^\top A_2 \rho - 1^\top \rho, \\ &\text{s.t. } \alpha \succcurlyeq 0, 1^\top \alpha = D, \\ &\quad \rho_i = y_i H_{i:}\alpha, \forall i = 1, \cdots, M, \\ &\quad f_i = H_{i:}\alpha, \forall i = 1, \cdots, N. \end{aligned} \tag{6.8}$$

where M refers to the number of labeled examples, while N is the number of all (labeled and unlabeled) examples. C' is equivalent to C up to a constant.

Notice that the new Semi-supervised Laplacian MDBoost objective has a similar form to the supervised version, thus we can derive its dual formulation as follows. The Lagrangian of the convex optimization problem in (6.8) is written as

$$\begin{aligned} &L(\alpha, \rho, f, u, v, r, q) \\ &= \tfrac{C'}{2} f^\top A_1 f + \tfrac{1}{2}\rho^\top A_2 \rho - 1^\top \rho + r(1^\top \alpha - D) - q^\top \alpha \\ &+ \textstyle\sum_{i=1}^{M} u_i(\rho_i - y_i H_{i:}\alpha) + \sum_{i=1}^{N} v_i(f_i - H_{i:}\alpha), \end{aligned} \tag{6.9}$$

with $q \succcurlyeq 0$. The infimum of L w.r.t. to the primal variable can be computed as

$$\inf_{\rho, f, \alpha} L = \inf_{f} \left[\tfrac{C'}{2} f^\top A_1 f + v^\top f\right]$$

$$+\inf_{\rho}\left[\tfrac{1}{2}\rho^\top A_2\rho+(u-1)^\top\rho\right]-Dr \tag{6.10}$$
$$+\inf_{\alpha}\left[(r1^\top-q^\top-\textstyle\sum_{i=1}^{M}u_iy_iH_{i:}-\sum_{i=1}^{N}v_iH_{i:})\alpha\right].$$

Clearly, $r1^\top-q^\top-\sum_{i=1}^{M}u_iy_iH_{i:}-\sum_{i=1}^{N}v_iH_{i:}=0$ must hold in order to have a finite infimum. Therefore, we have

$$\textstyle\sum_{i=1}^{M}u_iy_iH_{i:}+\sum_{i=1}^{N}v_iH_{i:}\preccurlyeq r1^\top. \tag{6.11}$$

For the first and second term in (6.10), the gradient must vanish at the optimum:

$$\frac{\partial\left[\frac{C'}{2}f^\top A_1f+v^\top f\right]}{\partial f_i}=0,\forall i=1,\cdots,N. \tag{6.12}$$
$$\frac{\partial\left[\frac{1}{2}\rho^\top A_2\rho+(u-1)^\top\rho\right]}{\partial\rho_i}=0,\forall i=1,\cdots,M. \tag{6.13}$$

This leads to $f=-A_1^{-1}v$; and $\rho=-A_2^{-1}(u-1)$ and the infimum is $-\frac{C'}{2}v^\top A_1^{-1}v-\frac{1}{2}(u-1)^\top A_2^{-1}(u-1)$.

By substituting the results back to (6.10), we can write the dual problem as:

$$\max_{r,u,v}\ -r-\tfrac{1}{2D}(u-1)^\top A_2^{-1}(u-1)-\tfrac{C'}{2}v^\top A_1^{-1}v,$$
$$\text{s.t. (6.11).} \tag{6.14}$$

We employ a similar column generation strategy to induce weak learners incrementally. At each iteration, we choose a weak learner that violates the constraint most:

$$h'(\cdot)=\operatorname*{argmax}_{h(\cdot)}\sum_{i=1}^{M}u_iy_ih(\mathbf{x}_i)+\sum_{i=1}^{N}v_ih(\mathbf{x}_i). \tag{6.15}$$

We summarize the proposed algorithm in Algorithm 2.

6.3 Experimental evaluation

In this section, we evaluate the performance of Laplacian MDBoost and Semi-supervised Laplacian MDBoost by conducting a set of experiments on real world datasets. We first present a comparison between the proposed Laplacian MDBoost and several most widely-used supervised boosting algorithms. Following that, we design a benchmark of semi-supervised inductive inference tasks by removing a certain ratio of training data labels in UCI datasets. We test the proposed Semi-supervised Laplacian MDBoost against two baseline approaches, including LLGC [236] combined with MDBoost and SemiBoost [131]. Finally, we apply our

Algorithm 2: Column generation based Semi-supervised Laplacian MDBoost.

Input: labeled training data $(\mathbf{x}_i, y_i), i = 1 \cdots M$; unlabeled training data $\mathbf{x}_i, i = M+1 \cdots N$; termination threshold $\varepsilon > 0$; regularization parameter D; maximum number of iterations $T_{\max}$.

Initialization: $N = 0$; $\alpha = \mathbf{0}$; $u_i = \frac{1}{M}$, $i = 1 \cdots M$; and $v_i = \frac{1}{N}$, $i = 1 \cdots N$.

for iteration $= 1 : T_{\max}$ **do**

1. Obtain a new base $h'(\cdot)$ by solving (6.15);
2. Check for optimal solution:
   if $\sum_{i=1}^{M} u_i y_i h'(\mathbf{x}_i) + \sum_{i=1}^{N} v_i h'(\mathbf{x}_i) < r + \varepsilon$,
   then break and the problem is solved;
3. Add $h'(\cdot)$ to the restricted master problem, which corresponds to a new constraint in the dual problem;
4. Solve the dual problem (6.14) and update r, u_i ($i = 1 \cdots M$) and v_i ($i = 1 \cdots N$).
5. Count weak classifiers $T = T + 1$.

end

Output:

1. Compute the primal variable α from the optimality conditions and the last solved dual problem (primal-dual interior point methods [23] produce α in the meantime);
2. The final strong classifier is $H(\mathbf{x}) = \mathbf{sign}\big(\sum_{j=1}^{N} \alpha_j h_j(\mathbf{x})\big)$.

semi-supervised method and two other baselines to an object segmentation in video task, as well as an RGBD glass object segmentation problem as discussed in Chapter 4.

6.3.1 Datasets and setup

The first set of our experiments is based on the 13 UCI benchmark datasets from [162]. For the supervised learning setting, we randomly split each of the UCI datasets into 3 subsets. 60% of the samples are used for training; 20% for cross validation and the rest for testing. For the larger datasets (**ringnorm**, **twonorm** and **waveform**), we randomly select 10% for training, 30% for cross validation and 60% for testing. All experiments are run 30 times for accuracy.

We choose the model hyperparameters by cross validation. The parameter D for AdaBoost-CG and all algorithms in the MDBoost family are chosen from $\{2, 5, 10, 20, 40, 70, 100, 150\}$. The search range of coefficient C for Semi-supervised Laplacian MDBoost and combining LLGC with MDBoost are set to $\{-3, -2, -1, -0.75, -0.5, -0.25, 0, 0.25, 0.5, 0.75, 1, 2, 3\}$ in negative log scale. The trade-off parameter C for LPBoost [39] is chosen similarly. For the graph Laplacian, we let t be proportional to the variance of data and normalize all feature

values to $[-10, +10]$. We set parameters of LLGC and SemiBoost to their respective optimal values given by [236] and [131]. For simplicity, we use decision stumps as weak learners in all tests and limit the maximum number of iterations $T_{\max}$ to 1000. The convergence threshold ε are uniformly set to 10^{-5}.

To evaluate the performance of Semi-supervised Laplacian MDBoost on real-world applications, we also choose a subset of the YouTube Celebrities Face Tracking and Recognition Dataset [83], which includes 6 sequences, and apply our method to a semi-supervised object segmentation task. In addition, we validate the efficacy of our approach with an RGBD glass segmentation problem as discussed in Chapter 4.

6.3.2 Laplacian MDBoost for supervised learning

To demonstrate the effectiveness of the new Laplacian MDBoost learning criterion, we first test our algorithm in a fully-supervised learning setting. The performance of Laplacian MDBoost is compared with four other boosting algorithms, namely AdaBoost, AdaBoost-CG, LPBoost and MDBoost. The experiments are run on 13 UCI benchmark datasets for 30 times, and average test error with standard deviation are reported in Figure 6.2. As we can see, Laplacian MDBoost outperforms its opponents in most cases. This result confirms our intuition and shows that local variance is effective in representing the margin distribution. In addition, we have the following observations in the comparison among the tested variants of boosting algorithms:

- LPBoost converges most quickly among all 5 algorithms followed by AdaBoost-CG, MDBoost and Laplacian MDBoost. These four algorithms are totally corrective, meaning weights of every weak learner in α could change at each training iteration. AdaBoost, on the other hand, does not change the weight of weak classifiers in previous iterations, resulting in a slower coordinate descent rate and a larger number of weak learners. In our experiments, all totally corrective boosting algorithms converge in 100 iterations.

- As extensively studied in the literature (e.g., [182]), LPBoost has the lowest average training error yet its test error is weaker than other totally corrective variants on almost all datasets. Once again this confirms that a lower training error does not necessarily lead to a lower test error. We have similar finindings in our experiments in Section 6.3.4.

- The proposed Laplacian MDBoost improves test error of the otherwise best-performing MDBoost at the cost of a small computational overhead. As the pairwise distance among input feature vectors w_{ij} can be efficiently pre-computed before the column generation procedure, Laplacian MDBoost is not significantly slower in training compared to MDBoost. Although LPBoost converges in the least number of iterations on most datasets, its generalization errors are higher than other totally corrective variants.

Table 6.1: Test error and standard deviation (in percentage $\%$) of Laplacian MDBoost (using only labeled data), Semi-supervised Laplacian MDBoost (SemiLap-MDBoost), Learning with Local and Global Consistency combined with MDBoost (LLGC+MDBoost), and SemiBoost on UCI datasets.

	Laplacian MDBoost	**SemiLap-MDBoost**	**LLGC+ MDBoost**	**SemiBoost**
banana	57.1 ± 4.8	$\mathbf{41.6 \pm 3.2}$	51.5 ± 7.4	41.7 ± 2.3
b-cancer	38.5 ± 14.2	$\mathbf{31.4 \pm 9.1}$	34.7 ± 9.2	33.3 ± 9.4
diabetes	36.7 ± 14.6	$\mathbf{30.1 \pm 4.8}$	30.7 ± 4.5	32.9 ± 11.7
f-solar	46.3 ± 9.3	44.5 ± 7.9	49.0 ± 9.6	$\mathbf{43.9 \pm 8.6}$
german	39.5 ± 16.1	31.6 ± 3.4	$\mathbf{31.4 \pm 3.4}$	32.4 ± 3.3
heart	$\mathbf{29.5 \pm 8.7}$	32.5 ± 8.1	35.6 ± 8.8	40.4 ± 9.1
image	34.2 ± 10.4	$\mathbf{28.5 \pm 1.9}$	35.7 ± 2.7	34.0 ± 3.4
ringnorm	51.9 ± 10.0	38.0 ± 1.7	38.6 ± 2.3	$\mathbf{40.1 \pm 5.3}$
splice	36.5 ± 28.1	$\mathbf{25.8 \pm 3.7}$	26.4 ± 3.9	26.2 ± 5.8
thyroid	$\mathbf{22.8 \pm 7.3}$	23.5 ± 5.1	25.3 ± 5.4	25.0 ± 7.4
titanic	52.0 ± 12.2	$\mathbf{49.7 \pm 13.3}$	53.3 ± 14.0	50.7 ± 16.4
twonorm	$\mathbf{18.1 \pm 5.1}$	29.8 ± 5.7	30.0 ± 5.5	33.4 ± 5.3
waveform	$\mathbf{19.7 \pm 2.6}$	23.4 ± 3.5	25.1 ± 3.7	25.8 ± 3.7

6.3.3 Semi-supervised Laplacian MDBoost

We first evaluate the Semi-supervised Laplacian MDBoost on a set of partially labeled datasets derived from the UCI benchmark. In this experiment, we followed the setup in Section 6.3.1 and choose randomly 10% of the original training data to keep their labels, while manually removing the labels of the other 90%. Our approach is compared with two other state-of-the-art semi-supervised algorithms: LLGC and SemiBoost. LLGC is widely used in different applications as a transductive algorithm [210, 158]. In contrast, SemiBoost is an inductive yet effective alternative [61, 110]. Note that LLGC is transductive so it does not by default offer the capability for predicting labels unseen during training. Therefore we combine it with MDBoost, by using LLGC first to predict the “fill-in" labels of unlabeled training data, then cascading with MDBoost as if all training data are labeled. For data with “fill-in" labels, we use a cross-validated coefficient during reweight sampling to limit their impact. This method effectively uses LLGC as a mean of manifold regularization while Laplacian MDBoost uses a Laplacian Eigenmap instead.

The results are summarized in Table 6.1. In 9 out of 13 datasets, utilizing unlabeled data helps to improve test performance, among which Semi-supervised Laplacian MDBoost is leading in 6 cases, showing the superior inductive inference performance.

Another interesting problem which will naturally arise is the performance gain under different ratios of labeled data. We present the results in Figure 6.3, where the labeled data ratio

changes from 10% to 100% with a step of 10%. We can see from the figure that, with limited labeled data and abundant unlabeled data, Semi-supervised Laplacian MDBoost significantly outperforms Laplacian MDBoost. However, with more unlabeled data turned into labeled, the performance gain decreases and the error rates converge at a same level. This is reasonable if we look at the objective function in Equation 6.7. When there is little (or no) unlabeled data, the value of the second term will approach (or equal to) zero, making it close (or equal) to Equation 6.5.

6.3.4 Video segmentation with Semi-supervised Laplacian MDBoost

In this section, we apply our Semi-supervised Laplacian MDBoost to an object segmentation in video problem. We randomly choose 6 video sequences from the YouTube Celebrities Face Tracking and Recognition Dataset [83]. For each sequence, we extract 15 consecutive frames. The first 10 frames are used for training and the last 5 frames for testing. The overall task is to accurately detect and label human face in each frame in a pixelwise manner.

To facilitate the labeling task, we first apply a frontal face detector [207] to find a bounding box for human face as in Figure 6.4. This would approximately guarantee that the face is in the center of the box while non-face located at the edges. Within the box we perform a segmentation [192] for superpixels. Each superpixel is then considered a basic input vector (datum) for the semi-supervised algorithms. Next, an automated training strategy was adopted to train the semi-supervised algorithms. The superpixels in the center of the bounding box (within a 20 pixel range) are labeled positive (face) while the superpixels on the brim are labeled negative (non-face). Two examples are shown in Figure 6.4. The green areas are labeled positive in training while the blue ones are negative. All other superpixels in between are treated as unlabeled training data. This automated training process eliminates the need for manually labeling the ground-truth (which can be a tedious task in real world applications), while it also generates a more challenging task for classification. We use color and position histograms as feature vectors, as faces are typically at the center of the face detector output, and have similar color distributions.

Figure 6.4 visualizes the test results of Semi-supervised Laplacian MDBoost, LLGC+MDBoost and SemiBoost on the two datasets. The performance difference is greater in the second case because the test frames involve a pose change which is likely to cause failure to the baseline classifiers. In both examples, Semi-supervised Laplacian MDBoost presents the best labeling performance visually. Full test results are reported in Table 6.2. In all 6 video sequences, Semi-supervised Laplacian MDBoost is the best in 5 cases in terms of test error, although SemiBoost is better at training error. This may imply that the baseline is prone to overfitting on these datasets.

Table 6.2: Average test and training error (in percentage %) of Semi-supervised Laplacian MDBoost (SemiLap-MDBoost), Learning with Local and Global Consistency combined with MDBoost (LLGC+MDBoost), and SemiBoost on the YouTube Celebrities Face Tracking and Recognition Datasets over 10 tests.

		test error	training error
0146 Al Pacino	SemiLap-MDBoost	$\mathbf{13.7 \pm 2.1}$	5.9 ± 1.2
	LLGC+MDBoost	15.4 ± 2.4	5.5 ± 1.1
	SemiBoost	19.8 ± 3.2	$\mathbf{4.2 \pm 0.6}$
0370 Bill Clinton	SemiLap-MDBoost	$\mathbf{11.1 \pm 1.6}$	10.5 ± 1.7
	LLGC+MDBoost	16.8 ± 2.0	$\mathbf{8.5 \pm 1.0}$
	SemiBoost	22.5 ± 2.2	10.7 ± 1.3
0564 Donald Trump	SemiLap-MDBoost	$\mathbf{7.2 \pm 2.1}$	4.3 ± 0.6
	LLGC+MDBoost	16.8 ± 3.2	3.9 ± 0.7
	SemiBoost	18.5 ± 4.7	$\mathbf{3.5 \pm 0.3}$
0727 Harrison Ford	SemiLap-MDBoost	12.6 ± 2.4	$\mathbf{4.9 \pm 0.3}$
	LLGC+MDBoost	15.3 ± 2.3	6.1 ± 0.5
	SemiBoost	$\mathbf{11.5 \pm 1.9}$	5.5 ± 0.3
0935 Jennifer Lopez	SemiLap-MDBoost	$\mathbf{16.5 \pm 3.2}$	11.5 ± 2.4
	LLGC+MDBoost	16.9 ± 2.9	10.2 ± 1.7
	SemiBoost	20.2 ± 4.1	$\mathbf{6.4 \pm 1.0}$
0950 Jennifer Lopez	SemiLap-MDBoost	$\mathbf{19.8 \pm 2.1}$	9.8 ± 2.2
	LLGC+MDBoost	29.1 ± 3.8	14.2 ± 2.9
	SemiBoost	28.3 ± 3.5	$\mathbf{7.9 \pm 1.2}$

6.3.5 RGBD glass object segmentation with Semi-supervised Laplacian MDBoost

We further validate the efficacy of our algorithm on the dataset for RGBD glass segmentation used in Chapter 4 and Chapter 5. In Chapter 4, we demonstrated that we can substantially improve glass segmentation performance by adding depth cues into the feature set for local glass boundary and region classification. In addition, in Chapter 5 we showed that segmentation performance can be further improved by building a classification model based on label transfer.

However, both methods require extensive human effort to label the exact glass regions and boundaries in every image from the training set. In this section, we aim to reduce the required labeling effort in this task by assuming only coarse or partial ground-truth annotation being available for training, and compare the performance of our algorithm against other semi-supervised learning schemes, and also with baseline methods from previous chapters which use the fully labeled dataset.

Glass region classification. For the glass regions, we use coarse labelings similar to what is used in Section 6.3.4, by creating a bounding box for glass objects as shown in Figure 6.5. As

the bounding box approximately guarantees that pixels at the center of the box belong to glass objects and pixels near the brim and outside are non-glass, we label superpixels at the center of the bounding box as glass (within a smaller, center-aligned bounding box which covers one fourth the area of the original bounding box), and label superpixels on the brim and outside as non-glass. All superpixels in between are treated as unlabeled training data. In our experiment, we have 4429 positive, 46789 negative and 13388 unlabeled superpixels in our training set. As the number of positive and negative training examples are unbalanced, we use a two level classification cascade to mine hard negatives. We follow the experiment settings in Chapter 4 to ensure a direct comparison can be made, except that we extract features from rectangles randomly sampled at multiple scales to capture local context, similar to TextonBoost [185] and also similar to what we did in Chapter 5 for boundary features. Examples of the superpixel labelings are shown in Figure 6.5.

During test, as the number of negative examples are approximately 10 times the number of positive examples, we report the precision at 25%, 50% and 75% recall instead of test error as in previous sections as the test error could be biased by the large proportion of negatives. As reported in Table 6.3 (a), although all semi-supervised algorithms using the partially labeled dataset suffer from a loss in precision compared to fully labeled baselines, our Semi-supervised Laplacian MDBoost performs on a par with alternative semi-supervised learning schemes. In addition, semi-supervised learning algorithms improve the results from Laplacian MDBoost, which uses labeled data only under the partially labeled setting. These results suggest that we are able to substantially relieve the labeling effort at a modest cost of glass segmentation precision.

Glass boundary classification. For the glass boundary, we take a more straight-forward approach by assuming only a subset of images in the training set are labeled. Since the 92 training images are unevenly distributed in 8 scenes, we randomly choose 30% of images from each scene as labeled data, and assume the remaining images unlabeled. The results are reported in Table 6.3 (b).

Similar to our observations on glass region classification, under the partially labeled setting we experience up to 17% loss in classification precision when using only the labeled data. However, the performance gap can be reduced with the help of unlabeled data, especially in the low-recall regime. This suggests that unlabeled data is particularly helpful in avoiding mistakes with large negative margins. Again, our proposed algorithm performs on a par with other semi-supervised alternatives.

It should be noted again that in both experiments, although semi-supervised learning algorithms improve the segmentation precision over Laplacian MDBoost which does not use unlabeled data, the performance of these algorithms is still inferior to baseline methods under the fully labeled setting. This is expected as we only assume only 9% and 30% of training data are labeled respectively. The same is true to the results on UCI datasets from Section 6.3.3.

Table 6.3: Precision (in percentage %) at 25%, 50% and 75% recall for glass region and boundary classification using fully labeled dataset and partially labeled dataset, respectively. Methods using fully labeled dataset include SVM and Random Forest (RF) from Chapter 4 and MDBoost. Methods using partially labeled dataset include Laplacian MDBoost (without using unlabeled data), Semi-supervised Laplacian MDBoost (SemiLap-MDBoost), Learning with Local and Global Consistency combined with MDBoost (LLGC+MDBoost), and SemiBoost.

(a) Glass region classification

	Fully Labeled		**Partially Labeled**			
	SVM	**MDBoost**	**Laplacian MDBoost**	**SemiLap-MDBoost**	**LLGC+ MDBoost**	**SemiBoost**
25% Recall	65.7	$\mathbf{68.0 \pm 5.2}$	50.6 ± 3.9	$\mathbf{59.7 \pm 4.4}$	52.3 ± 4.1	58.0 ± 3.3
50% Recall	62.9	$\mathbf{65.0 \pm 3.0}$	35.6 ± 5.1	$\mathbf{43.9 \pm 5.3}$	38.1 ± 5.5	40.1 ± 2.8
75% Recall	44.9	$\mathbf{52.8 \pm 4.6}$	23.2 ± 4.0	25.7 ± 6.1	23.3 ± 5.4	$\mathbf{27.8 \pm 3.5}$

(b) Glass boundary classification

	Fully Labeled			**Partially Labeled**			
	RF	**SVM**	**MDBoost**	**Laplacian MDBoost**	**SemiLap-MDBoost**	**LLGC+ MDBoost**	**SemiBoost**
25% Recall	**59.3**	33.5	53.2 ± 5.4	35.2 ± 4.2	$\mathbf{42.2 \pm 3.8}$	37.8 ± 2.7	39.9 ± 3.3
50% Recall	**39.7**	25.2	36.3 ± 3.3	26.4 ± 2.9	28.9 ± 3.5	26.2 ± 3.4	$\mathbf{30.2 \pm 3.5}$
75% Recall	21.5	15.9	$\mathbf{25.7 \pm 2.1}$	10.2 ± 3.1	$\mathbf{11.0 \pm 2.4}$	9.2 ± 1.8	10.3 ± 2.2

In our experiments, the performance of semi-supervised algorithms gets close to fully labeled baselines only when we assume more than 80% training data are labeled. To maximize the performance gain obtained by using unlabeled data, we usually need to assume less than 40% data from the training set are labeled. This trend suggests that we may potentially improve the results from the fully labeled setting in Table 6.3 if we collect more unlabeled data in addition to all labeled data we have in our dataset.

6.4 Conclusion

In this chapter, we have proposed a novel semi-supervised boosting algorithm based on the margin distribution boosting. Inspired by Laplacian Eigenmaps, we use the graph Laplacian as an effective means of manifold regularization on both labeled and unlabeled data. Like MDBoost, the algorithm is totally-corrective and a column generation based optimization technique is used to facilitate minimizing the objective function.

The proposed Semi-supervised Laplacian MDBoost, along with its supervised version, exhibits promising inductive performance in a variety of tasks including classification on real-world data, video segmentation and glass object segmentation. Our experiments show that

Semi-supervised Laplacian MDBoost outperforms LLGC and SemiBoost in terms of classification accuracy. In particular, we show that we can relieve the labeling effort at a modest segmentation precision cost in the glass object segmentation problem discussed in Chapters 4 and 5. This is achieved by either assuming only a coarse labeling is available or only a subset of images in the training set are labeled. In addition, the proposed algorithm is a generic inductive semi-supervised learning method that can be applied to many more object detection and segmentation problems with partial labelings.

Like many other semi-supervised classification algorithms, Semi-supervised Laplacian MDBoost is currently a two-class algorithm. We are exploring the possibility to a multiple class extension by introducing new similarity measures. We also want to test our algorithm on more practical applications to further explore the strength of graph Laplacian on different intrinsic geometric structures. One possible extension is to add more methods for manifold regularization to adapt to different manifold assumptions.

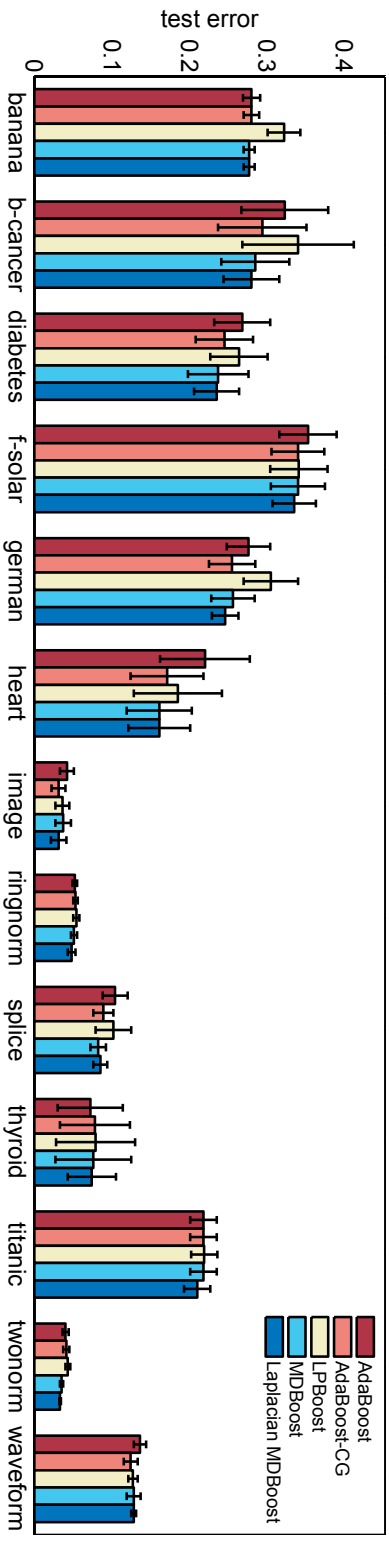


Figure 6.2: Average test errors (with standard deviations) of AdaBoost, AdaBoost-CG, LPBoost, MDBoost and Laplacian MDBoost on 13 UCI benchmark datasets.

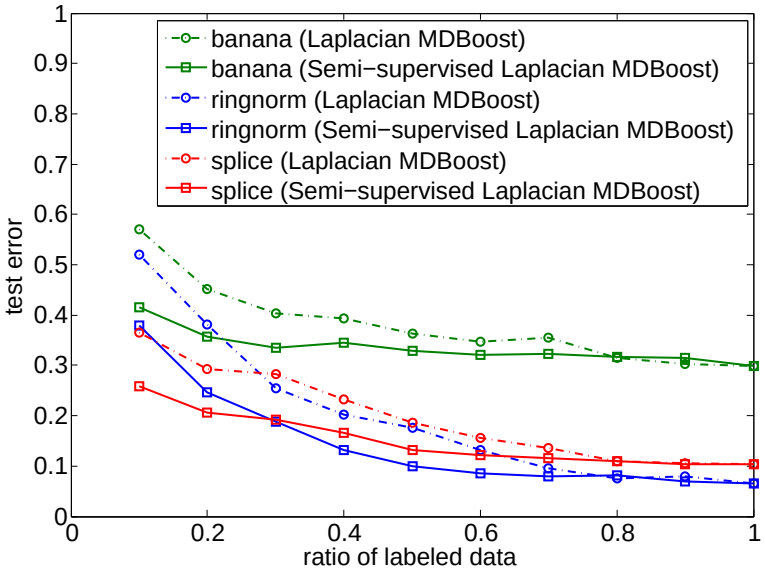


Figure 6.3: Performance of Laplacian MDBoost (dash-dot line) and Semi-supervised Laplacian MDBoost (solid line) on UCI datasets **banana** (green), **ringnorm** (blue) and **splice** (red).

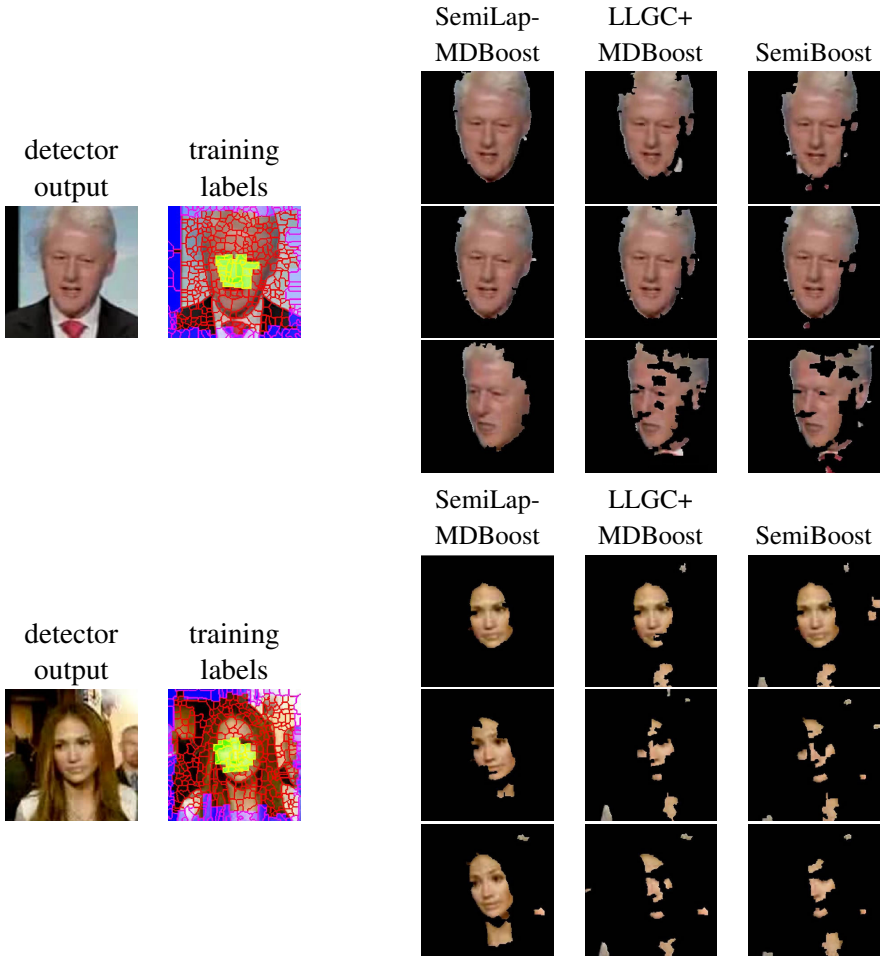


Figure 6.4: Examples of video segmentation with three different semi-supervised algorithms: Semi-supervised Laplacian MDBoost (SemiLap-MDBoost), Learning with local and global consistency combined with MDBoost (LLGC+MDBoost) and SemiBoost. The video data are sequences 0370 and 0950 from the Youtube Celebrity Face Tracking and Recognition Dataset [83].

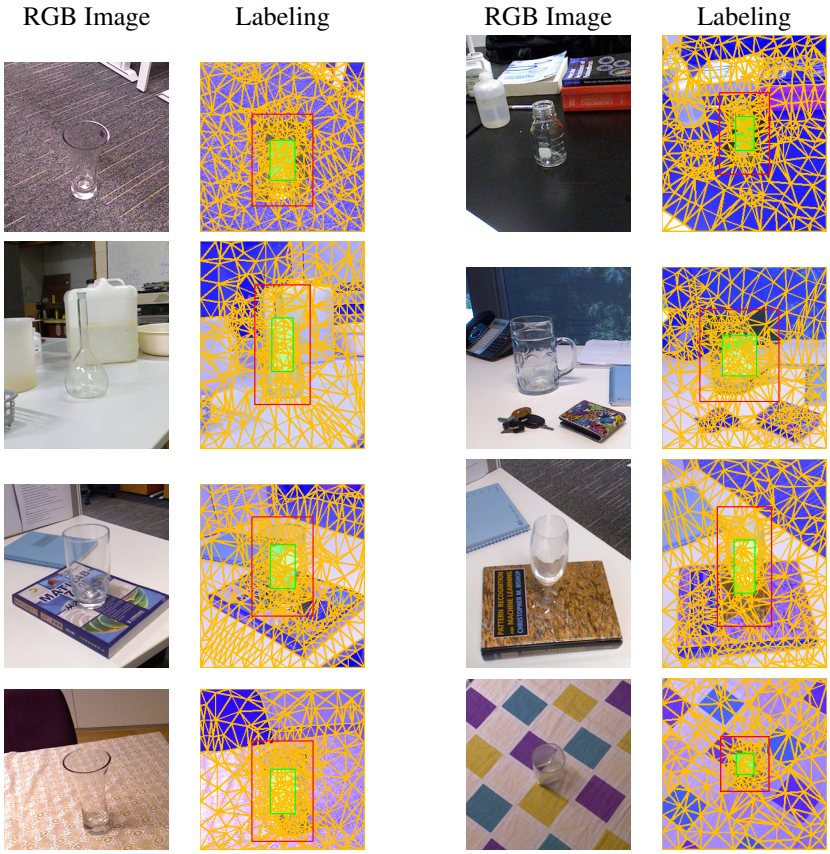


Figure 6.5: Examples of coarse ground-truth superpixel labelings used for our glass region classification experiment. Each red bounding box covers a ground-truth glass object. The center-aligned green bounding boxes cover one fourth the area of the red bounding boxes. A superpixel is labeled as glass if it has 50% or more overlap with the green bounding box, and non-glass if it has 50% or more overlap with the region outside the red bounding box. All superpixels inside the red bounding box but outside the green bounding box are treated as unlabeled data. See text for details.

Chapter 7

Conclusion

This thesis has proposed and implemented a series of context-aware object detection and segmentation models with varying degrees of auxiliary information availability. This final chapter summarizes the main contributions of the thesis, and closes with possible future directions of our work.

7.1 Primary contributions

Object detection and segmentation have wide application in computational vision, and it is one of the most essential steps towards understanding a scene. Both object detection and segmentation study the problem of localizing objects of interest in an image. The main difference is the definition of the object pose space with different levels of details. For object detection, the object pose is described by a set of parameters including the object center location, scale and an aspect ratio. For object segmentation, the detailed pose is inferred with a pixelwise segmentation mask. A key issue in these problems concerns exploiting the spatial context, as local evidence is often insufficient to determine object pose in the presence of partial object visibility, varying sensory data modality, and limited annotation availability. This thesis addresses the object detection and segmentation problems in such adverse conditions with auxiliary information such as depth maps and unlabeled data, focusing on four main issues in context-aware object detection and segmentation: 1) the effective context representations, 2) inference with imperfect depth data, 3) depth-aware features and label transfer, and 4) the relaxation of the labeling requirements for training data.

We discuss three object detection and segmentation scenarios based on varying degrees of auxiliary information availability. In Chapter 3, we propose a structured Hough voting method for detecting objects with heavy occlusion in indoor environments. First, we extend the Hough hypothesis space to include both the object's location, and its visibility pattern. We design a new score function that accumulates votes for object detection and occlusion prediction. In addition, we explore the correlation between objects and their environment, building a depth-encoded object-context model based on RGBD data. Particularly, we design a layered context

representation and allow image patches from both objects and backgrounds to vote for the object hypotheses. We demonstrate that using a data-driven 2.1D (layered) representation we can learn visual codebooks with better quality, and obtain more interpretable detection results in terms of spatial relationship between objects and the viewer. We test our algorithm on two challenging RGBD datasets with significant occlusions and intraclass variations, and demonstrate the superior performance of our method.

In Chapters 4 and 5, we move our focus to the segmentation of glass objects, which are commonly found in indoor environments and play a key role in daily human activities. Yet, localizing glass objects is much more challenging due to lack of locally discriminative visual features and homogeneity of surface appearance. Therefore, we seek to exploit low cost RGBD consumer cameras to incorporate depth information as a novel contextual cue. In Chapter 4, we developed a method for localizing glass objects with a multimodal RGBD camera. Our method integrates the intensity and depth information from a single view point, and builds a Markov Random Field that predicts glass boundary and region jointly. Based on the segmentation, we also reconstruct the depth of the scene and fill in the missing depth values. The efficacy of our algorithm is validated on an RGBD Glass dataset of 43 distinct glass objects. Following this, in Chapter 5 we propose an approach that uses a nonparametric, data-driven label transfer scheme for local glass boundary estimation. A weighted voting scheme based on a joint feature manifold is adopted to integrate depth and appearance cues, and we learn a distance metric on the depth-encoded feature manifold. Local boundary evidence is then integrated into an MRF framework for spatially coherent glass object detection and segmentation. The efficacy of this approach is verified on the same RGBD Glass dataset where we obtained a clear improvement over the state-of-the-art approaches using statistical learning based classifiers for local estimation, both in terms of accuracy and speed.

In Chapter 6, we propose a semi-supervised boosting algorithm to address the annotation availability issue in object detection and segmentation. We choose boosting algorithms as they attract much attention in computer vision and image processing because of their strong performance in a variety of applications. Recent progress on theory of boosting algorithms suggests a close link between good generalization and the margin distribution of the classifier w.r.t. a dataset. Therefore, we propose a novel data-dependent margin distribution learning criterion for boosting, termed Laplacian MDBoost, which utilizes the intrinsic geometric structure of datasets. One key aspect of our method is that it can seamlessly incorporate unlabeled data by including a graph Laplacian regularizer. We derive a dual formulation of the learning problem that can be efficiently solved by column generation. Experiments on various datasets validate the effectiveness of the new graph Laplacian based learning criterion on both supervised and unsupervised learning settings. We also show that the performance of our algorithm performs on a par with the state-of-the-art semi-supervised learning algorithms on a variety of inductive inference tasks, including real world video segmentation and RGBD glass object segmentation.

7.2 Future work

The discussion in the previous chapters has suggested directions in which we would like to extend our work based on the specific problem settings in each chapter. In this section, we describe more general directions in context-aware object detection and segmentation with auxiliary information.

7.2.1 3D scene structure reasoning

The work in Chapter 3 suggests a data-driven 2.1D (layered) representation can help us learn a visual codebook with better quality. In Chapters 4 and 5, we also reported improved segmentation performance with context-driven features and joint reasoning on glass boundary and depth. Yet, the depth-augmented context representation described in this thesis is still coarse, as we do not explicitly reason about the scene structures and surfaces in 3D. The problem of understanding the underlying 3D scene structure from a single 2D image has been well studied. For example, Lee, Hebert and Kanade [104] generate plausible interpretations of a scene from a collection of line segments automatically extracted from a single indoor image. Liu, Gould and Koller [114] studies the problem of single image depth estimation by exploiting the fact that semantic class prediction strongly informs depth perception. Gupta, Efros and Hebert [65] uses a qualitative physical representation of an outdoor scene with geometry and mechanics to recover 3D scene structures. Recent progress in scene structure reasoning (e.g. [78, 233, 211, 240, 118]) allows us to recover the location and orientation of major structures and scene layout with varying input information. This opens up the possibility of designing a detailed object-context representation in 3D to facilitate object detection and segmentation.

A natural extension to our object-context model is to incorporate the 3D scene structure. In particular, occlusion can be viewed as an integral part of the scene structure reasoning process. Currently, our model in Chapter 3 learns the appearance of occluders separately from the rest of the spatial context to infer a visibility pattern of an object. In fact, the occluders can be seamlessly merged with the rest of the context in a 3D object-context model, as the location and orientation of 3D structures naturally informs their distance to the viewer, hence the occlusion relationships can be straightforwardly inferred. In addition, physical relations between an object and its context can also be inferred, providing a more detailed pose description of detected objects. For example, pictures are commonly attached to a wall, while tables and chairs are usually supported by the ground plane. On the other hand, an unlikely physical relation, such as an object floating above the ground with no support or attachment, may indicate a false positive in object detection.

One key challenge in this direction is to reliably recover scene structures and layout in 3D, particularly when depth data is not present during model evaluation. Conceptually, it is

preferable to jointly solve for object detection/segmentation and scene structure reasoning, as the cues inferred from one task can be beneficial for the others. Efficient inference could also be a challenge in real-world applications.

7.2.2 Holistic scene understanding

Apart from jointly solving for object detection/segmentation and scene structure reasoning, other related tasks in scene understanding may also provide important cues for object detection and segmentation. As discussed in Section 2.1.5, there has been literature in holistic scene understanding which involves object detection and segmentation in both 2D and 3D. Holistic scene understanding aims to solve problems such as scene classification, object detection, semantic labeling, depth reconstruction and geometric layout estimation in a unified framework.

In this thesis, we address either object detection or pixelwise object segmentation as a standalone problem. We wanted to explore the potential of considering these problems in a detached manner although they could be integral to a more complete scene understanding framework. This means that although our methods encode contextual cues, it is either from an object level view (for detection) or a local feature level view (for segmentation). In fact, a visual story told by an image contains a hierarchy of information, and the various scene attributes can be best described at different levels in the hierarchy. For instance, it would be difficult for our models to encode the scene-level information. It is an interesting question to explore which subproblems of scene understanding are most relevant to object detection and segmentation, hence could positively impact localization performance.

It should be noted that not all tasks in scene understanding may help improve localization performance. In particular, certain tasks may be redundant in certain scenarios, i.e., they use similar visual cues so their outputs can be highly correlated to an inherent component in object detection/segmentation systems. From a holistic scene understanding perspective, we need to make informed decisions on the scopes of object detection and segmentation systems and the other scene understanding problems.

7.2.3 Other types of auxiliary information

In this thesis we discussed a few object detection and segmentation scenarios with partial information. In fact, in many real-world problems there are alternative information sources we can look into to address the partial information issues. In other words, the ambiguities induced by the missing information may be resolved with information sources beyond static images and the auxiliary information discussed in this thesis.

1) Video sequences. Compared to static images, video sequences provide more information about scenes and the objects within. In particular, with additional temporal and spatial cues we are able to identify moving and static objects (e.g., [27]) which may help resolve the appear-

ance variations induced by occlusion, and build a complete and high quality context model. However, the problem is also more challenging as we need to consider additional temporal and spatial priors.

2) Descriptive text. Recent work by Fidler, Sharma and Urtasun [48] suggests text in the form of complex sentential descriptions can help improve the semantic parsing performance for an image. In fact, many images from the Internet are accompanied by text tags, descriptive descriptions, and sometimes questions and answers. Particularly, contextual information can be inferred from descriptive text (e.g., "the chair is *behind* the table"). Therefore, it is an interesting direction to incorporate textual information into a context-aware object detection and segmentation system.

3) Application-specific sensors. In some specific applications such as satellite imaging and autonomous navigation, we may be supplied with application-specific sensors. For example, spectral cameras provide multispectral imaging data beyond the visible spectrum. The problems are usually also highly domain-specific, meaning that additional domain knowledge can be integrated into the localization task. In practice, more efficient feature extraction and inference algorithms are usually necessary for real-time processing. It is an interesting direction to explore some specific applications and make use of additional sensory data to address the partial information issues discussed in this thesis.

References

1. Amazon mechanical turk. http://www.mturk.com. 115
2. R. Achanta, A. Shaji, K. Smith, A. Lucchi, P. Fua, and S. Süsstrunk. Slic superpixels. *École Polytechnique Fédéral de Lausssanne (EPFL), Tech. Rep*, 2010. xix, xxi, 101, 102, 104, 106
3. E. Adelson and P. Anandan. Ordinal characteristics of transparency. In *AAAI*, 1990. 9, 45, 77, 84, 101
4. S. Albrecht and S. Marsland. Seeing the unseen: Simple reconstruction of transparent objects from point cloud data. In *workshops. acin. tuwien. ac. at*. 47
5. B. Alexe, T. Deselaers, and V. Ferrari. Measuring the objectness of image windows. *IEEE Trans. PAMI*, 34(11):2189–2202, 2012. 20
6. P. Arbelaez, M. Maire, C. Fowlkes, and J. Malik. Contour detection and hierarchical image segmentation. *IEEE Trans. PAMI*, 33(5):898–916, 2011. 35
7. D. Ballard. Generalizing the hough transform to detect arbitrary shapes. *Pattern recognition*, 13(2):111–122, 1981. 20, 55
8. S. Y. Bao, M. Sun, and S. Savarese. Toward coherent object detection and scene layout understanding. In *CVPR*, 2010. 6, 33, 53
9. M. Belkin. Problems of learning on manifolds. 2003. 116
10. M. Belkin and P. Niyogi. Laplacian eigenmaps for dimensionality reduction and data representation. *Neural computation*, 15(6):1373–1396, 2003. 11, 48, 116, 118, 119
11. M. Belkin, P. Niyogi, and V. Sindhwani. Manifold regularization: A geometric framework for learning from labeled and unlabeled examples. *JMLR*, 7:2399–2434, 2006. 48
12. K. Bennett, A. Demiriz, et al. Semi-supervised support vector machines. In *NIPS*, 1999. 48, 49
13. K. P. Bennett, A. Demiriz, and R. Maclin. Exploiting unlabeled data in ensemble methods. In *KDD*, 2002. 49
14. I. Biederman, R. J. Mezzanotte, and J. C. Rabinowitz. Scene perception: Detecting and judging objects undergoing relational violations. *Cognitive psychology*, 14(2):143–177, 1982. 32

15. C. Bishop. *Pattern recognition and machine learning*. Springer, 2006. 9, 17, 25, 48, 78, 83, 87, 103, 104
16. M. Blaschko and C. Lampert. Object localization with global and local context kernels. In *BMVC*, 2009. 6, 32, 53
17. A. Blum and T. Mitchell. Combining labeled and unlabeled data with co-training. In *COLT*, 1998. 48
18. L. Bo, X. Ren, and D. Fox. Depth kernel descriptors for object recognition. In *IROS*, 2011. 27
19. L. Bo, X. Ren, and D. Fox. Unsupervised feature learning for rgb-d based object recognition. In *Experimental Robotics*, pages 387–402, 2013. 27
20. U. Bonde, V. Badrinarayanan, and R. Cipolla. Robust instance recognition in presence of occlusion and clutter. In *ECCV*. Springer, 2014. 30
21. E. Borenstein and J. Malik. Shape guided object segmentation. In *CVPR*, 2006. 35
22. L. Bourdev and J. Malik. Poselets: Body part detectors trained using 3d human pose annotations. In *ICCV*, 2009. 3, 19
23. S. Boyd and L. Vandenberghe. *Convex optimization*. Cambridge Univ Pr, 2004. 122
24. Y. Boykov, O. Veksler, and R. Zabih. Fast approximate energy minimization via graph cuts. *IEEE Trans. PAMI*, 23(11):1222–1239, 2001. 42
25. Y. Y. Boykov and M.-P. Jolly. Interactive graph cuts for optimal boundary & region segmentation of objects in nd images. In *ICCV*, 2001. 40
26. T. Brox, L. Bourdev, S. Maji, and J. Malik. Object segmentation by alignment of poselet activations to image contours. In *CVPR*, 2011. 30, 35
27. T. Brox and J. Malik. Large displacement optical flow: descriptor matching in variational motion estimation. *IEEE Trans. PAMI*, 33(3):500–513, 2011. 138
28. F. Buc, Y. Grandvalet, and C. Ambroise. Semi-supervised marginboost. In *NIPS*, 2002. 49
29. V. Caselles, R. Kimmel, and G. Sapiro. Geodesic active contours. *IJCV*, 22(1):61–79, 1997. 46
30. S. Chandra, G. Chrysos, and I. Kokkinos. Surface based object detection in rgbd images. In *BMVC*, 2016. 27
31. O. Chapelle, B. Scholkopf, and A. Zien, editors. *Semi-Supervised Learning*. MIT Press, 2006. 48
32. J. Chen, X. Liu, and S. Lyu. Boosting with side information. In *ACCV*, 2012. 3
33. K. Chen and S. Wang. Regularized boost for semi-supervised learning. In *NIPS*, 2007. 49

34. W. Choi, Y.-W. Chao, C. Pantofaru, and S. Savarese. Understanding indoor scenes using 3d geometric phrases. In *CVPR*, 2013. 28

35. F. R. Chung. Spectral graph theory. In *CBMS Regional Conference Series in Mathematics, No. 92*, 1997. 119

36. M. Collins, R. Schapire, and Y. Singer. Logistic regression, AdaBoost and Bregman distances. *Machine Learning*, 48(1):253–285, 2002. 49

37. G. Csurka, C. Dance, L. Fan, J. Willamowski, and C. Bray. Visual categorization with bags of keypoints. In *ECCV Workshops*, 2004. 23

38. N. Dalal and B. Triggs. Histograms of oriented gradients for human detection. In *CVPR*, 2005. 16, 17, 27, 101

39. A. Demiriz, K. Bennett, and J. Shawe-Taylor. Linear programming boosting via column generation. *Machine Learning*, 46(1):225–254, 2002. 50, 116, 122

40. S. K. Divvala, D. Hoiem, J. H. Hays, A. A. Efros, and M. Hebert. An empirical study of context in object detection. In *CVPR*, 2009. 4, 32

41. I. Endres and D. Hoiem. Category-independent object proposals with diverse ranking. *IEEE Trans. PAMI*, 36(2):222–234, 2014. 20

42. I. Endres, K. J. Shih, J. Jiaa, and D. Hoiem. Learning collections of part models for object recognition. In *CVPR*, 2013. 19, 115

43. M. Everingham, L. Van Gool, C. K. I. Williams, J. Winn, and A. Zisserman. The PASCAL Visual Object Classes Challenge 2012 (VOC2012) Results. http://www.pascal-network.org/challenges/VOC/voc2012/workshop/index.html. 16, 27

44. A. Fathi, M. Balcan, X. Ren, and J. Rehg. Combining self training and active learning for video segmentation. In *BMVC*, 2011. 39, 100, 103, 109

45. L. Fei-Fei and P. Perona. A bayesian hierarchical model for learning natural scene categories. In *CVPR*, 2005. 23

46. P. Felzenszwalb, R. Girshick, D. McAllester, and D. Ramanan. Object detection with discriminatively trained part-based models. *IEEE Trans. PAMI*, 32(9):1627–1645, 2010. xvii, 18, 19, 29, 58, 66, 67, 71, 72, 75

47. P. F. Felzenszwalb and D. McAllester. Object detection grammars. In *ICCV Workshops*, 2011. 18

48. S. Fidler, A. Sharma, and R. Urtasun. A sentence is worth a thousand pixels. In *CVPR*, 2013. 139

49. B. Frank, R. Schmedding, C. Stachniss, M. Teschner, and W. Burgard. Learning the elasticity parameters of deformable objects with a manipulation robot. In *IROS*, 2010. 8, 78

50. R. Fransens, C. Strecha, and L. Van Gool. A mean field em-algorithm for coherent occlusion handling in map-estimation prob. In *CVPR*, 2006. 30

51. M. Fritz, M. Black, G. Bradski, and T. Darrell. An additive latent feature model for transparent object recognition. In *NIPS*, 2009. 44, 46, 77

52. J. Gall and V. Lempitsky. Class-specific hough forests for object detection. In *CVPR*, 2009. 22, 23, 55

53. T. Gao, B. Packer, and D. Koller. A segmentation-aware object detection model with occlusion handling. In *CVPR*, 2011. 30, 53

54. A. Garg and D. Roth. Margin distribution and learning algorithms. In *ICML*, 2003. 116

55. A. Geiger, P. Lenz, and R. Urtasun. Are we ready for autonomous driving? the kitti vision benchmark suite. In *CVPR*, 2012. 27

56. S. Geman and D. Geman. Stochastic relaxation, gibbs distributions, and the bayesian restoration of images. *IEEE Trans. PAMI*, (6):721–741, 1984. 43

57. R. Girshick. Fast r-cnn. In *ICCV*, 2015. 17

58. R. Girshick, J. Donahue, T. Darrell, and J. Malik. Rich feature hierarchies for accurate object detection and semantic segmentation. In *CVPR*, 2014. 17, 28

59. R. B. Girshick, P. F. Felzenszwalb, and D. A. McAllester. Object detection with grammar models. In *NIPS*, 2011. 19, 30

60. S. Gould, J. Zhao, X. He, and Y. Zhang. Superpixel graph label transfer with learned distance metric. In *ECCV*, 2014. 39

61. H. Grabner, C. Leistner, and H. Bischof. Semi-supervised on-line boosting for robust tracking. In *ECCV*, 2008. 124

62. D. Greig, B. Porteous, and A. H. Seheult. Exact maximum a posteriori estimation for binary images. *Journal of the Royal Statistical Society. Series B (Methodological)*, pages 271–279, 1989. 41

63. C. Gu, J. J. Lim, P. Arbeláez, and J. Malik. Recognition using regions. In *CVPR*, 2009. 26

64. X. Guo, X. Wang, L. Yang, X. Cao, and Y. Ma. Robust foreground detection using smoothness and arbitrariness constraints. In *ECCV*, 2014. 35

65. A. Gupta, A. A. Efros, and M. Hebert. Blocks world revisited: Image understanding using qualitative geometry and mechanics. In *ECCV*, 2010. 137

66. S. Gupta, R. Girshick, P. Arbeláez, and J. Malik. Learning rich features from rgb-d images for object detection and segmentation. In *ECCV*, 2014. 27, 28

67. K. Han, K.-Y. K. Wong, and M. Liu. A fixed viewpoint approach for dense reconstruction of transparent objects. In *CVPR*, 2015. 47

68. K. He, X. Zhang, S. Ren, and J. Sun. Deep residual learning for image recognition. In *CVPR*, 2016. 17

69. X. He, R. S. Zemel, and M. Carreira-Perpindn. Multiscale conditional random fields for image labeling. In *CVPR*, 2004. 38

70. X. He, R. S. Zemel, and D. Ray. Learning and incorporating top-down cues in image segmentation. In *ECCV*, 2006. 40

71. G. E. Hinton. Training products of experts by minimizing contrastive divergence. *Neural computation*, 14(8):1771–1800, 2002. 43

72. D. Hoiem, A. A. Efros, and M. Hebert. Geometric context from a single image. In *ICCV*, 2005. 115

73. D. Hoiem, A. A. Efros, and M. Hebert. Putting objects in perspective. *IJCV*, 80(1):3–15, 2008. 32, 115

74. E. Hsiao and M. Hebert. Occlusion reasoning for object detection under arbitrary viewpoint. In *CVPR*, 2012. 30

75. I. Ihrke, K. N. Kutulakos, H. P. Lensch, M. Magnor, and W. Heidrich. State of the art in transparent and specular object reconstruction. In *EUROGRAPHICS 2008 STAR–STATE OF THE ART REPORT*, 2008. 44

76. S. D. Jain and K. Grauman. Predicting sufficient annotation strength for interactive foreground segmentation. In *ICCV*, 2013. 35

77. A. Janoch, S. Karayev, Y. Jia, J. Barron, M. Fritz, K. Saenko, and T. Darrell. A category-level 3-d object dataset: Putting the kinect to work. In *ICCV Workshops*, 2011. xvii, 27, 53, 63, 68, 69

78. H. Jiang and J. Xiao. A linear approach to matching cuboids in rgbd images. In *CVPR*, 2013. 137

79. A. E. Johnson and M. Hebert. Using spin images for efficient object recognition in cluttered 3d scenes. *IEEE Trans. PAMI*, 21(5):433–449, 1999. 27

80. P. Jolicoeur, M. A. Gluck, and S. M. Kosslyn. Pictures and names: Making the connection. *Cognitive psychology*, 16(2):243–275, 1984. 15

81. M. I. Jordan, Z. Ghahramani, T. S. Jaakkola, and L. K. Saul. An introduction to variational methods for graphical models. *Machine learning*, 37(2):183–233, 1999. 87

82. M. Kass, A. Witkin, and D. Terzopoulos. Snakes: Active contour models. *IJCV*, 1(4):321–331, 1988. 35

83. M. Kim, S. Kumar, V. Pavlovic, and H. Rowley. Face tracking and recognition with visual constraints in real-world videos. In *CVPR*, 2008. xx, 116, 123, 125, 132

84. U. Klank, D. Carton, and M. Beetz. Transparent object detection and reconstruction on a mobile platform. In *ICRA*, 2011. 8, 46, 77, 78

85. J. Kleinberg and E. Tardos. Approximation algorithms for classification problems with pairwise relationships: Metric labeling and markov random fields. *Journal of the ACM*, 49(5):616–639, 2002. 42

86. G. J. Klinker, S. A. Shafer, and T. Kanade. A physical approach to color image understanding. *IJCV*, 4(1):7–38, 1990. 45

87. S. Kluckner, T. Mauthner, P. Roth, and H. Bischof. Semantic image classification using consistent regions and individual context. In *BMVC*, 2009. 32

88. P. Kohli, P. H. Torr, et al. Robust higher order potentials for enforcing label consistency. *IJCV*, 82(3):302–324, 2009. 40

89. D. Koller and N. Friedman. *Probabilistic graphical models: principles and techniques*. MIT press, 2009. 36

90. V. Koltun. Efficient inference in fully connected crfs with gaussian edge potentials. In *NIPS*, 2011. 41, 58

91. V. Kompella and P. Sturm. Detection and avoidance of semi-transparent obstacles using a collective-reward based approach. In *ICRA*, 2011. 45

92. S. Konishi and A. L. Yuille. Statistical cues for domain specific image segmentation with performance analysis. In *CVPR*, 2000. 38

93. P. D. Kovesi. Matlab and octave functions for computer vision and image processing. *Online: http://www. csse. uwa. edu. au/˜ pk/Research/MatlabFns/# match*, 2000. 81

94. A. Krizhevsky, I. Sutskever, and G. E. Hinton. Imagenet classification with deep convolutional neural networks. In *NIPS*, 2012. 17

95. M. P. Kumar, P. Ton, and A. Zisserman. Obj cut. In *CVPR*, 2005. 35

96. S. Kumar and M. Hebert. Discriminative random fields. *IJCV*, 68(2):179–201, 2006. 38, 40

97. L. Ladickỳ, P. Sturgess, K. Alahari, C. Russell, and P. H. Torr. What, where and how many? combining object detectors and crfs. In *ECCV*, 2010. 35

98. K. Lai, L. Bo, X. Ren, and D. Fox. A large-scale hierarchical multi-view rgb-d object dataset. In *ICRA*, 2011. 8, 27, 78, 82, 101

99. K. Lai, L. Bo, X. Ren, and D. Fox. Sparse distance learning for object recognition combining rgb and depth information. In *ICRA*, 2011. 8, 78

100. K. Lai, L. Bo, X. Ren, and D. Fox. Detection-based object labeling in 3d scenes. In *ICRA*, 2012. 28

101. C. H. Lampert, M. B. Blaschko, and T. Hofmann. Beyond sliding windows: Object localization by efficient subwindow search. In *CVPR*, 2008. 20

102. D. Larlus and F. Jurie. Combining appearance models and markov random fields for category level object segmentation. In *CVPR*, 2008. 35

103. S. Lazebnik, C. Schmid, and J. Ponce. Beyond bags of features: Spatial pyramid matching for recognizing natural scene categories. In *CVPR*, 2006. 39

104. D. C. Lee, M. Hebert, and T. Kanade. Geometric reasoning for single image structure recovery. In *CVPR*, 2009. 137

105. S. Lee and H. Shim. Skewed stereo time-of-flight camera for translucent object imaging. *Image and Vision Computing*, 43:27–38, 2015. 46

106. A. Lehmann, B. Leibe, and L. Van Gool. Fast prism: Branch and bound hough transform for object class detection. *IJCV*, 94(2):175–197, 2011. 22

107. A. D. Lehmann, B. Leibe, and L. J. Van Gool. Prism: Principled implicit shape model. In *BMVC*, 2009. 16

108. Z. Lei, K. Ohno, M. Tsubota, and E. Takeuchi. State of the art in transparent and specular object reconstruction. In *ROBIO*, 2011. 47

109. B. Leibe, A. Leonardis, and B. Schiele. Combined object categorization and segmentation with an implicit shape model. In *ECCV Workshops*, 2004. 22, 23, 35, 55, 57

110. C. Leistner, H. Grabner, and H. Bischof. Semi-supervised boosting using visual similarity learning. In *CVPR*, 2008. 124

111. B. Li, T. Wu, and S.-C. Zhu. Integrating context and occlusion for car detection by hierarchical and-or model. In *ECCV*, 2014. 30

112. S. Z. Li and S. Singh. *Markov random field modeling in image analysis*, volume 26. Springer, 2009. 42

113. D. Lin, S. Fidler, and R. Urtasun. Holistic scene understanding for 3d object detection with rgbd cameras. In *ICCV*, 2013. 28

114. B. Liu, S. Gould, and D. Koller. Single image depth estimation from predicted semantic labels. In *CVPR*, 2010. 5, 137

115. C. Liu, J. Yuen, and A. Torralba. Nonparametric scene parsing via label transfer. *IEEE Trans. PAMI*, 33(12):2368–2382, 2011. 38, 39, 100, 110

116. D. Liu, X. Chen, and Y.-H. Yang. Frequency-based 3d reconstruction of transparent and specular objects. In *CVPR*, 2014. 46

117. L. Liu and S. Sclaroff. Region segmentation via deformable model-guided split and merge. In *ICCV*, 2001. 34

118. M. Liu, X. He, and M. Salzmann. Building scene models by completing and hallucinating depth and semantics. In *ECCV*, 2016. 137

119. W. Liu, R. Ji, and S. Li. Towards 3d object detection with bimodal deep boltzmann machines over rgbd imagery. In *CVPR*, 2015. 28

120. H. Lodhi, G. Karakoulas, and J. Shawe-Taylor. Boosting the margin distribution. In *IDEAL*, 2000. 116

121. D. G. Lowe. Object recognition from local scale-invariant features. In *ICCV*, 1999. 17, 27

122. R. C. Luo, P.-J. Lai, and V. W. S. Ee. Transparent object recognition and retrieval for robotic bio-laboratory automation applications. In *IROS*, 2015. 47

123. I. Lysenkov, V. Eruhimov, and G. Bradski. Recognition and pose estimation of rigid transparent objects with a kinect sensor. In *RSS*, 2012. 47, 77

124. I. Lysenkov and V. Rabaud. Pose estimation of rigid transparent objects in transparent clutter. In *ICRA*, 2013. 47

125. C. Ma, X. Lin, J. Suo, Q. Dai, and G. Wetzstein. Transparent object reconstruction via coded transport of intensity. In *CVPR*, 2014. 46

126. S. Mahamud, L. R. Williams, K. K. Thornber, and K. Xu. Segmentation of multiple salient closed contours from real images. *IEEE Trans. PAMI*, 25(4):433–444, 2003. 35

127. M. Maire, S. Yu, and P. Perona. Object detection and segmentation from joint embedding of parts and pixels. In *ICCV*, 2011. 6, 32, 53

128. S. Maji, A. C. Berg, and J. Malik. Classification using intersection kernel support vector machines is efficient. In *CVPR*, 2008. 18

129. S. Maji and J. Malik. Object detection using a max-margin hough transform. In *CVPR*, 2009. xvii, 21, 25, 54, 62, 66, 67, 71, 72, 75

130. J. Malik, S. Belongie, T. Leung, and J. Shi. Contour and texture analysis for image segmentation. *IJCV*, 43(1):7–27, 2001. 85, 101

131. P. Mallapragada, R. Jin, A. Jain, and Y. Liu. Semiboost: Boosting for semi-supervised learning. *IEEE Trans. PAMI*, 31(11):2000–2014, 2009. 49, 116, 121, 123

132. D. Martin, C. Fowlkes, and J. Malik. Learning to detect natural image boundaries using local brightness, color, and texture cues. *IEEE Trans. PAMI*, 26(5):530–549, 2004. 81, 82, 90, 91, 108

133. P. McCullagh, J. A. Nelder, and P. McCullagh. *Generalized linear models*, volume 2. Chapman and Hall London, 1989. 38

134. K. McHenry and J. Ponce. A geodesic active contour framework for finding glass. In *CVPR*, 2006. 9, 46, 77

135. K. McHenry, J. Ponce, and D. Forsyth. Finding glass. In *CVPR*, 2005. 9, 44, 45, 77, 84, 91, 101

136. D. Meger, C. Wojek, J. J. Little, and B. Schiele. Explicit occlusion reasoning for 3d object detection. In *BMVC*, 2011. 30

137. F. Mériaudeau, R. Rantoson, D. Fofi, and C. Stolz. Review and comparison of non-conventional imaging systems for three-dimensional digitization of transparent objects. *Journal of Electronic Imaging*, 21(2):021105–1, 2012. 44

138. F. Metelli. The perception of transparency. *Scientific American*, 1974. 45

139. K. Mikolajczyk, B. Leibe, and B. Schiele. Multiple object class detection with a generative model. In *CVPR*, 2006. 23, 26

140. Y. Ming, H. Li, and X. He. Connected contours: A new contour completion model that respects the closure effect. In *CVPR*, 2012. 92

141. T. Minka. The summation hack as an outlier model. *Tutorial note*, 2003. 21

142. G. Mori, X. Ren, A. A. Efros, and J. Malik. Recovering human body configurations: Combining segmentation and recognition. In *CVPR*, 2004. 35

143. R. Mottaghi, X. Chen, X. Liu, N.-G. Cho, S.-W. Lee, S. Fidler, R. Urtasun, et al. The role of context for object detection and semantic segmentation in the wild. In *CVPR*, 2014. 32

144. H. Murase. Surface shape reconstruction of an undulating transparent object. In *ICCV*, 1990. 9, 45, 77

145. P. K. Nathan Silberman, Derek Hoiem and R. Fergus. Indoor segmentation and support inference from rgbd images. In *ECCV*, 2012. xvii, 27, 53, 63, 68, 69

146. T. Ojala, M. Pietikainen, and T. Maenpaa. Multiresolution gray-scale and rotation invariant texture classification with local binary patterns. *IEEE Trans. PAMI*, 24(7):971–987, 2002. 17

147. A. Oliva and A. Torralba. Modeling the shape of the scene: A holistic representation of the spatial envelope. *IJCV*, 42(3):145–175, 2001. 39

148. B. Ommer and J. Malik. Multi-scale object detection by clustering lines. In *ICCV*, 2009. 23, 26

149. A. Opelt, A. Pinz, and A. Zisserman. Learning an alphabet of shape and appearance for multi-class object detection. *IJCV*, 80(1):16–44, 2008. 25

150. V. Ordonez, J. Deng, Y. Choi, A. C. Berg, and T. L. Berg. From large scale image categorization to entry-level categories. In *ICCV*, 2013. 15

151. M. Osadchy, D. Jacobs, and R. Ramamoorthi. Using specularities for recognition. In *ICCV*, 2003. 9, 45, 46, 77

152. P. Ott and M. Everingham. Shared parts for deformable part-based models. In *CVPR*, 2011. 19

153. J. Pan and T. Kanade. Coherent object detection with 3d geometric context from a single image. In *ICCV*, 2013. 32

154. S. N. Parizi, A. Vedaldi, A. Zisserman, and P. Felzenszwalb. Automatic discovery and optimization of parts for image classification. In *ICLR*, 2015. 19

155. O. M. Parkhi, A. Vedaldi, C. Jawahar, and A. Zisserman. The truth about cats and dogs. In *ICCV*, 2011. 35

156. J. Pearl. *Probabilistic reasoning in intelligent systems: networks of plausible inference*. Morgan Kaufmann, 1988. 42

157. B. Pepikj, M. Stark, P. Gehler, and B. Schiele. Occlusion patterns for object class detection. In *CVPR*, 2013. 30

158. B. Pfahringer, C. Leschi, and P. Reutemann. Scaling up semi-supervised learning: An efficient and effective LLGC variant. In *KDD*, 2007. 124

159. C. Phillips, K. Derpanis, and K. Daniilidis. A novel stereoscopic cue for figure-ground segregation of semi-transparent objects. In *ICCV Workshops*, pages 1100–1107, 2011. 46

160. M. Prasad, A. Zisserman, A. Fitzgibbon, M. P. Kumar, and P. H. Torr. Learning class-specific edges for object detection and segmentation. In *Computer Vision, Graphics and Image Processing*, pages 94–105. Springer, 2006. 35

161. A. Rabinovich, A. Vedaldi, C. Galleguillos, E. Wiewiora, and S. Belongie. Objects in context. In *ICCV*, 2007. 32

162. G. Ratsch, T. Onoda, and K. Muller. Soft margins for AdaBoost. *Machine Learning*, 42(3):287–320, 2001. 116, 122

163. N. Razavi, J. Gall, P. Kohli, and L. Van Gool. Latent hough transform for object detection. In *ECCV*, 2012. 23, 26

164. N. Razavi, J. Gall, and L. Van Gool. Scalable multi-class object detection. In *CVPR*, 2011. 20, 22

165. J. Redmon, S. Divvala, R. Girshick, and A. Farhadi. You only look once: Unified, real-time object detection. In *CVPR*, 2016. 18

166. K. Rematas and B. Leibe. Efficient object detection and segmentation with a cascaded hough forest ism. In *ICCV Workshops*, 2011. 22

167. S. Ren, K. He, R. Girshick, and J. Sun. Faster r-cnn: Towards real-time object detection with region proposal networks. In *NIPS*, 2015. 17

168. L. Reyzin and R. E. Schapire. How boosting the margin can also boost classifier complexity. In *ICML*, 2006. 11, 50, 116

169. G. Rogez, M. Khademi, J. Supančič III, J. M. M. Montiel, and D. Ramanan. 3d hand pose detection in egocentric rgb-d images. In *ECCV Workshops*, 2014. 27

170. G. Rogez, J. S. Supancic, and D. Ramanan. Understanding everyday hands in action from rgb-d images. In *ICCV*, 2015. 27

171. S. Romdhani, P. Torr, B. Scholkopf, and A. Blake. Computationally efficient face detection. In *ICCV*, 2001. 18

172. E. Rosch, C. B. Mervis, W. D. Gray, D. M. Johnson, and P. Boyes-Braem. Basic objects in natural categories. *Cognitive psychology*, 8(3):382–439, 1976. 15

173. C. Rother, V. Kolmogorov, and A. Blake. Grabcut: Interactive foreground extraction using iterated graph cuts. *ACM Trans. Graphics*, 23(3):309–314, 2004. 35, 40, 47, 64

174. B. C. Russell and A. Torralba. Building a database of 3d scenes from user annotations. In *CVPR*, 2009. 79

175. A. Saffari, H. Grabner, and H. Bischof. Serboost: Semi-supervised boosting with expectation regularization. In *ECCV*, 2008. 49

176. L. K. Saul and M. I. Jordan. Exploiting tractable substructures in intractable networks. In *NIPS*, pages 486–492, 1996. 43

177. A. Saxena, S. H. Chung, and A. Y. Ng. 3-d depth reconstruction from a single still image. *IJCV*, 76(1):53–69, 2008. 5

178. R. E. Schapire and Y. Freund. Boosting the margin: a new explanation for the effectiveness of voting methods. *The Annals of Statistics*, 26:322–330, 1998. 50, 116

179. E. Seemann, B. Leibe, and B. Schiele. Multi-aspect detection of articulated objects. In *CVPR*, 2006. 23, 26

180. P. Sermanet, D. Eigen, X. Zhang, M. Mathieu, R. Fergus, and Y. LeCun. Overfeat: Integrated recognition, localization and detection using convolutional networks. In *ICLR*, 2014. 17

181. L. Shams, D. R. Wozny, R. Kim, and A. Seitz. Influences of multisensory experience on subsequent unisensory processing. *Frontiers in Psychology*, 2:27–38, 2011. 3

182. C. Shen and H. Li. Boosting through optimization of margin distributions. *IEEE Trans. Neural Networks*, 21(4):659–666, 2010. 11, 48, 50, 116, 117, 123

183. C. Shen and H. Li. On the dual formulation of boosting algorithms. *IEEE Trans. PAMI*, 32(12):2216–2231, 2010. 50, 116, 118

184. J. Shotton, A. Blake, and R. Cipolla. Multiscale categorical object recognition using contour fragments. *IEEE Trans. PAMI*, 30(7):1270–1281, 2008. 25

185. J. Shotton, J. Winn, C. Rother, and A. Criminisi. Textonboost: Joint appearance, shape and context modeling for multi-class object recognition and segmentation. In *ECCV*, 2006. xvi, 32, 38, 40, 58, 102, 115, 127

186. A. Shrivastava and A. Gupta. Building part-based object detectors via 3d geometry. In *ICCV*, 2013. 3, 28

187. K. Simonyan and A. Zisserman. Very deep convolutional networks for large-scale image recognition. In *ICLR*, 2015. 17

188. M. Singh and B. L. Anderson. Toward a perceptual theory of transparency. *Psychological review*, 109(3):492, 2002. 45

189. J. Sivic, B. C. Russell, A. A. Efros, A. Zisserman, and W. T. Freeman. Discovering objects and their location in images. In *ICCV*, 2005. 23

190. L. Spinello and K. O. Arras. People detection in rgb-d data. In *IROS*, 2011. 27

191. N. Srivastava and R. R. Salakhutdinov. Multimodal learning with deep boltzmann machines. In *NIPS*, 2012. 3

192. A. Stein and M. Hebert. Occlusion boundaries from motion: low-level detection and mid-level reasoning. *IJCV*, 82(3):325–357, 2009. 125

193. E. Sudderth, A. Torralba, W. Freeman, and A. Willsky. Depth from familiar objects: A hierarchical model for 3d scenes. In *CVPR*, 2006. 6, 33, 53

194. M. Sun, Y. Bao, and S. Savarese. Object detection with geometrical context feedback loop. In *BMVC*, 2010. 33

195. M. Sun, G. Bradski, B. Xu, and S. Savarese. Depth-encoded hough voting for joint object detection and shape recovery. In *ECCV*, 2010. 28, 54

196. P. Sundberg, T. Brox, M. Maire, P. Arbeláez, and J. Malik. Occlusion boundary detection and figure/ground assignment from optical flow. In *CVPR*, 2011. 35

197. S. Tang, M. Andriluka, and B. Schiele. Detection and tracking of occluded people. *IJCV*, 110(1):58–69, 2014. 30

198. A. Tejani, D. Tang, R. Kouskouridas, and T.-K. Kim. Latent-class hough forests for 3d object detection and pose estimation. In *ECCV*, 2014. 31

199. J. Tighe and S. Lazebnik. Superparsing: scalable nonparametric image parsing with superpixels. In *ECCV*, 2010. 39, 100, 110

200. J. Tighe and S. Lazebnik. Finding things: Image parsing with regions and per-exemplar detectors. In *CVPR*, 2013. 39

201. A. Torralba. Contextual priming for object detection. *IJCV*, 53(2):169–191, 2003. 6, 32, 53

202. A. Torralba, K. P. Murphy, and W. T. Freeman. Contextual models for object detection using boosted random fields. In *NIPS*, 2004. 32

203. A. Torralba, K. P. Murphy, and W. T. Freeman. Sharing features: efficient boosting procedures for multiclass object detection. In *CVPR*, 2004. 115

204. J. Uijlings, K. van de Sande, T. Gevers, and A. Smeulders. Selective search for object recognition. *IJCV*, 104(2):154–171, 2013. 20

205. P. J. Van Laarhoven and E. H. Aarts. *Simulated annealing*. Springer, 1987. 43

206. A. Vedaldi and A. Zisserman. Structured output regression for detection with partial truncation. In *NIPS*, 2009. 30, 53

207. P. Viola and M. J. Jones. Robust real-time face detection. *IJCV*, 57(2):137–154, 2004. 16, 17, 18, 26, 49, 115, 125

208. C. Vondrick, A. Khosla, T. Malisiewicz, and A. Torralba. Hoggles: Visualizing object detection features. In *ICCV*, 2013. xv, 17, 29

209. A. Wallace, P. Csakany, G. Buller, A. Walker, and S. Edinburgh. 3d imaging of transparent objects. In *BMVC*, 2000. 8, 46, 77, 78

210. M. Wang, T. Mei, X. Yuan, Y. Song, and L. Dai. Video annotation by graph-based learning with neighborhood similarity. In *ACM-MM*, 2007. 124

211. S. Wang, S. Fidler, and R. Urtasun. Holistic 3d scene understanding from a single geo-tagged image. In *CVPR*, 2015. 137

212. T. Wang, X. He, and N. Barnes. Learning structured hough voting for joint object detection and occlusion reasoning. In *CVPR*, 2013. 64

213. X. Wang, T. X. Han, and S. Yan. An hog-lbp human detector with partial occlusion handling. In *ICCV*, 2009. 30

214. X. Wang, M. Yang, S. Zhu, and Y. Lin. Regionlets for generic object detection. In *ICCV*, 2013. 19

215. Y. Weiss. Comparing the mean field method and belief propagation for approximate inference in mrfs. *Advanced Mean Field Methods–Theory and Practice*, pages 229–240, 2001. 43, 87

216. M. Welling and C. Sutton. Learning in markov random fields with contrastive free energies. In *AISTATS*, pages 397–404, 2005. 43

217. T. Werner. A linear programming approach to max-sum problem: A review. *IEEE Trans. PAMI*, 29(7):1165–1179, 2007. 42

218. C. Wojek, S. Walk, S. Roth, and B. Schiele. Monocular 3d scene understanding with explicit occlusion reasoning. In *CVPR*, 2011. 30

219. L. Wolf and S. Bileschi. A critical view of context. *IJCV*, 69(2):251–261, 2006. 6, 32, 53

220. B. Wu and R. Nevatia. Detection of multiple, partially occluded humans in a single image by bayesian combination of edgelet part detectors. In *ICCV*, 2005. 30

221. J. Xiao, J. Hays, K. A. Ehinger, A. Oliva, and A. Torralba. Sun database: Large-scale scene recognition from abbey to zoo. In *CVPR*, 2010. 39

222. Y. Xu, H. Nagahara, A. Shimada, and R.-i. Taniguchi. Transcut: Transparent object segmentation from a light-field image. In *ICCV*, 2015. 47

223. Y. Yang, S. Hallman, D. Ramanan, and C. Fowlkes. Layered object detection for multi-class segmentation. In *CVPR*, 2010. 33, 35

224. C. Yanover, T. Meltzer, and Y. Weiss. Linear programming relaxations and belief propagation–an empirical study. *JMLR*, 7:1887–1907, 2006. 42

225. J. Yao, S. Fidler, and R. Urtasun. Describing the scene as a whole: Joint object detection, scene classification and semantic segmentation. In *CVPR*, 2012. 33

226. P. Yarlagadda, A. Monroy, and B. Ommer. Voting by grouping dependent parts. *ECCV*, 2010. 22

227. M. Ye, Y. Zhang, R. Yang, and D. Manocha. 3d reconstruction in the presence of glasses by acoustic and stereo fusion. In *CVPR*, 2015. 47

228. J. J. Yebes, L. M. Bergasa, and M. García-Garrido. Visual object recognition with 3d-aware features in kitti urban scenes. *Sensors*, 15(4):9228–9250, 2015. 28

229. J. S. Yedidia, W. T. Freeman, and Y. Weiss. Understanding belief propagation and its generalizations. *Exploring artificial intelligence in the new millennium*, 8:236–239, 2003. 42

230. Z. Yu, W. Zhang, B. Kumar, and D. Levi. Structured hough voting for vision-based highway border detection. In *WACV*, 2015. 26

231. M. D. Zeiler and R. Fergus. Visualizing and understanding convolutional networks. In *ECCV*, 2014. 17

232. H. Zhang, J. Xiao, and L. Quan. Supervised label transfer for semantic segmentation of street scenes. In *ECCV*, 2010. 38

233. J. Zhang, C. Kan, A. G. Schwing, and R. Urtasun. Estimating the 3d layout of indoor scenes and its clutter from depth sensors. In *ICCV*, 2013. 137

234. Q. Zhang, G. Hua, W. Liu, Z. Liu, and Z. Zhang. Can visual recognition benefit from auxiliary information in training? In *ACCV*, 2014. 3, 28

235. Y. Zhang and T. Chen. Weakly supervised object recognition and localization with invariant high order features. In *BMVC*, 2010. 59

236. D. Zhou, O. Bousquet, T. Lal, J. Weston, and B. Scholkopf. Learning with local and global consistency. In *NIPS*, 2004. 48, 116, 121, 123

237. L. Zhu, Y. Chen, A. Yuille, and W. Freeman. Latent hierarchical structural learning for object detection. In *CVPR*, 2010. 19

238. X. Zhu and Z. Ghahramani. Learning from labeled and unlabeled data with label propagation. *Carnegie Mellon Univ., CS Tech. Rep. CMUCALD-02-107*, 2002. 48

239. Y. Zhu, R. Urtasun, R. Salakhutdinov, and S. Fidler. segdeepm: Exploiting segmentation and context in deep neural networks for object detection. In *CVPR*, 2015. 32

240. W. Zhuo, M. Salzmann, X. He, and M. Liu. Indoor scene structure analysis for single image depth estimation. In *CVPR*, 2015. 137

241. M. Z. Zia, M. Stark, and K. Schindler. Explicit occlusion modeling for 3d object class representations. In *CVPR*, 2013. 30

242. C. L. Zitnick and P. Dollár. Edge boxes: Locating object proposals from edges. In *ECCV*, 2014. 20, 64