Learning Semantic Interaction Among Indoor Objects

Thesis submitted in partial fulfilment
of the requirements for the degree of

Master of Science (by Research)
in
Electronics & Communications Engineering

by

Swagatika Panda
201032007
swagatika.panda@research.iiit.ac.in

Center for Visual Information Technology
International Institute of Information Technology
Hyderabad - 500 032, INDIA
July 2014

International Institute of Information Technology
Hyderabad, India

CERTIFICATE

It is certified that the work contained in this thesis, titled "Learning Semantic Interaction Among Indoor Objects" by Swagatika Panda, has been carried out under my supervision and is not submitted elsewhere for a degree.

_____ _____

Date Adviser: Prof. C. V. Jawahar

To My family and friends

Acknowledgments

My journey in IIIT-Hyderabad has been a journey worth it. It could not have been so without the support of many people. As I submit my MS thesis, I want to offer my gratitude to all those people who helped me in successfully completing this journey. First of all, I want to thank my guide Prof. C.V. Jawahar, for accepting me as a student and constantly guiding me. His guidance has helped me improve not only as a researcher but also as a person. I can never thank him enough for providing me support at my difficult times and helping me move forward. I offer sincere thanks to Dr. A.H. Abdul Hafez for his patience, support and encouragement. Many thanks to Dr. Hafez for identifying my strengths, dealing with my flaws and providing confidence to me. The discussions with you during our projects were crucial for my progress. I am grateful to Dr. K. Madhava Krishna for guiding me in many projects and being there for any advice I needed from him. I thank him for helping me with his positivity and patience. I am thankful to everyone in CVIT for providing such a positive work environment. Many thanks to Rajan, Praveen, Vikram, Rajvi, Natraj, Ayush, Anand, Jyothi, Yashaswi, Nisarg, Nagendar, Vijay, Vishal, Naveen, Ravi, Prabhu, Aniket, Sanchit, Udit, Ujjwal for all the help, discussions, study group sessions and support. Thanks to the students in RRC, Karthik, Laxit, Suryansh and Anusha for everything, starting from discussions, paper-reading, experimentations to encouraging me. In addition, I want to express gratitude to all the anonymous netizens with whom I had many healthy discussions and who had helped me through different online forums.

The transition from leaving job and becoming a researcher has not been easy. It had its overwhelming moments making me swing between extreme emotions. I could not have accomplished it without the support and understanding of my parents. I wish to thank my sisters for being my constant support and making me believe in myself. Nobody can know me better than you! Special thanks to Radhakrushna, Litu, Kuninani and Munabhaina for helping me during my stay at Hyderabad. Thanks to Deepti and Sumit, for remaining by my side rock steady throughout this journey and for being my friends through my ups and downs. Thanks Ragini, Ravishankar, Mahathi and Ruchi, for being my support and taking care of me like my own family. Thanks to Ravi, Nikhil, Sushma, Vikram, Praveen and Vasu, for their company, advice, help during courses, research and many other times. Thanks to Akhila, Niyati, Neelamma, Harsh, Priyanka, Jatin, Srk for the fun-filled moments in IIIT. Thanks to my best buddies Sreedhar, Aparajita, Pratyush, Himanshu and Satyabati for giving me moral support and being there whenever I needed you. Last, but not the least, thanks to IIIT community for giving me such a beautiful campus and environment to grow.

Abstract

Abstract

Robot manipulation in clutter with objects in physical contact remains a challenging problem till date. The challenge is posed by interaction involved among the objects at various levels of complexity. Understanding positional semantics of the environment plays an important role in such tasks. The interaction with surrounding objects in the environment must be considered in order to perform the task without causing the objects fall or get damaged. In our work, we learn the semantics in terms of support relationship among different objects in a cluttered environment by utilizing various photometric and geometric properties of the scene. To manipulate an object of interest, we use the inferred support relationship to derive a sequence in which its surrounding objects should be removed while causing minimal damage to the environment. We believe, this work can push the boundary of robotic applications in grasping, object manipulation and picking-from-bin, towards objects of generic shape and size and scenarios with physical contact and overlap.

In the first part of the thesis, we aim at learning semantic interaction among objects of generic shapes and sizes lying in clutter involving both direct and indirect physical contact. Three types of support relationships are inferred: "Support from below", "Support from side", and "Containment". Subsequently, the learned semantic interaction or support relationship is used to derive a sequence or order in which the objects surrounding the object of interest should be removed without causing damage to the environment. The generated sequence is called Support Order. We have proposed and analysed two alternative approaches for support inference. In the first approach "Multiple Object Support Inference", support relations between all possible pairs are inferred. In the second approach "Hierarchical Support Inference", given an object of interest, its support relationship with other graspable objects is inferred hierarchically. The support relationship is used to predict the "support order" or the order in which the surrounding objects need to be removed in order to manipulate the target object.

In the second part of the thesis, we attempt to learn the semantic interaction among different objects in clutter using multiple views. At first, support relationship among objects in each view is inferred. Then the inferred support relationships are combined to define support relationships across multiple views. The combined global support relationship is used to recover missing support relations and predict the support order. Support order is the order in which objects surrounding an object of interest should be removed. The support order predicted using global support relationship incorporates hidden objects and missing spatial support relations.

We have created two RGBD datasets consisting of various objects used in day-to-day life present in clutter. In "Indoor dataset for clutter", 50 cluttered scenes are captured from frontal view using 35 objects of different shapes and sizes. In "Indoor multiview dataset", 7 cluttered scene are captured.

Each scene each captured from multiple views. In this dataset, total 67 images are captured using 9 objects of different shapes and sizes. The dataset is made publicly available for the research community around the world. We explore many different settings involving different kind of object-object interaction. We successfully learn support relationships and predict support order in these settings. It can play significant role in extending the scope of manipulation to cluttered environment involving both direct and indirect physical contact, and generic objects.

Keywords: Robotic Vision, Support Relations, Support Order, RGBD, Semantic Interaction, Clutter, Multiple Views

Contents

Chapter Page

1 Introduction . . . 1
1.1 Motivation . . . 1
1.2 Problem Statement . . . 2
1.3 Contributions . . . 3
1.4 Related Work . . . 4
1.5 Thesis Organization . . . 7

2 Overview of our Approach . . . 8
2.1 Introduction . . . 8
2.2 Overview . . . 9
2.3 Segmentation based Approach . . . 10
2.4 Object Detection . . . 10
2.5 Geometric Features for Pairwise Support Relationship . . . 12
2.5.1 Proximity . . . 12
2.5.2 Depth Boundary . . . 12
2.5.3 Boundary Ratio . . . 13
2.5.4 Containment . . . 14
2.5.5 Relative Stability . . . 14
2.6 Datasets . . . 15
2.6.1 Indoor Dataset for Clutter . . . 16
2.6.2 Indoor Multi-view Dataset . . . 17
2.6.3 Annotation . . . 17
2.7 Summary . . . 18

3 Semantic Interaction in Clutter . . . 19
3.1 Learning Support Relationship . . . 19
3.1.1 MAP Inference . . . 19
3.1.2 Multiple Object Support Inference . . . 20
3.1.3 Hierarchical Support Inference . . . 21
3.2 Support Order Prediction . . . 23
3.2.1 Different Cases of Support . . . 24
3.2.2 Tree of Support . . . 25
3.3 Experiments & Results . . . 26
3.3.1 Segmentation . . . 26
3.3.2 Object Detection . . . 26

3.3.3 Structure Class Inference . . . 29
3.3.4 Multiple Object Support Inference & Support Order Prediction . . . 29
3.3.5 Hierarchical Support Inference & Support Order Prediction . . . 31
3.3.6 Evaluation Measure . . . 32
3.3.7 Discussions & Analysis of Results . . . 33
3.4 Summary . . . 34

4 Multiple View Support Order Prediction . . . 35
4.1 Introduction . . . 35
4.2 Overview of the Framework . . . 36
4.3 Region Mapping across Multiple Views . . . 37
4.3.1 Region Mapping between 2 Views . . . 37
4.3.2 Region Mapping between N Views . . . 38
4.4 Support Inference . . . 38
4.4.1 Support Inference for Single View . . . 38
4.4.2 Support Inference in N Views . . . 39
4.5 Support Order Prediction . . . 40
4.6 Experiments & Results . . . 40
4.6.1 Region Mapping . . . 40
4.6.2 Support Inference . . . 41
4.6.3 Support Order Prediction . . . 42
4.7 Summary . . . 43

5 Conclusions & Future Work . . . 45
5.1 Summary . . . 45
5.2 Directions for Future Work . . . 46

Bibliography . . . 48

List of Figures

Figure Page

1.1 Illustration of relevance of support order prediction 2

2.1 Illustration of different types of support relationships 9
2.2 Framework for support order prediction in single view 10
2.3 Results of segmentation and object detection 11
2.4 Demonstration of proximity 13
2.5 Demonstration of depth boundary concept 13
2.6 Demonstration of boundary ratio, containment, and relative stability 14
2.7 Indoor dataset for clutter 15
2.8 Template images used in Indoor dataset for Clutter 16
2.9 Indoor multi-view dataset 17
2.10 Template images used in Indoor multiview dataset 18

3.1 Illustration of Multiple Object Support Inference Method 22
3.2 Illustration of hierarchical support inference method 22
3.3 Illustration of hierarchical support inference classifier 23
3.4 Different cases of support: case 1 and case 2 24
3.5 Different cases of support: case 3 24
3.6 Example to demonstrate support order prediction 25
3.7 Demonstration of Results of Object Detection 27
3.8 Precision Recall of object detection for each object 28
3.9 True Positives, False Positives and False negatives of object detection 28
3.10 Performance of Multiple object inference vs MAP inference 30
3.11 Demonstration of support by multiple objects 31
3.12 Illustration of dependency on Structure class prediction 32
3.13 Results of Hierarchical Support Inference 33

4.1 Scenarios in which support relations needs multiple views 36
4.2 Framework for support order prediction using multiple views 37
4.3 Illustration of region mapping across 2 views 38
4.4 Result of Regionmapping 39
4.5 Result of support order prediction in multiple views 41

List of Tables

Table Page

3.1 Accuracy of Hierarchical Segmentation, measured as average overlap over ground truth regions for best-matching segmented region, either weighted by pixel area or unweighted. 26
3.2 Accuracy of Object Detection, measured as mean precision and recall, and standard deviation. 27
3.3 Accuracy Structure class Inference over ground truth regions and segmented regions. . 29
3.4 Accuracy of various approaches for Support Inference. Type aware accuracy penalizes incorrect support type whereas type agnostic accuracy does not. 29
3.5 Support Order using MAP inference and Multiple object support inference method: MAP inference method fails in case of side support and support by multiple objects. . 31
3.6 Support order using hierarchical support inference method. Support Order Prediction for different cases shown in Fig. 3.13. 32

4.1 Accuracy of Structure class Inference & Segmentation 42
4.2 Accuracy of Support Inference 42
4.3 Support Order using Single View vs Multiple Views 43

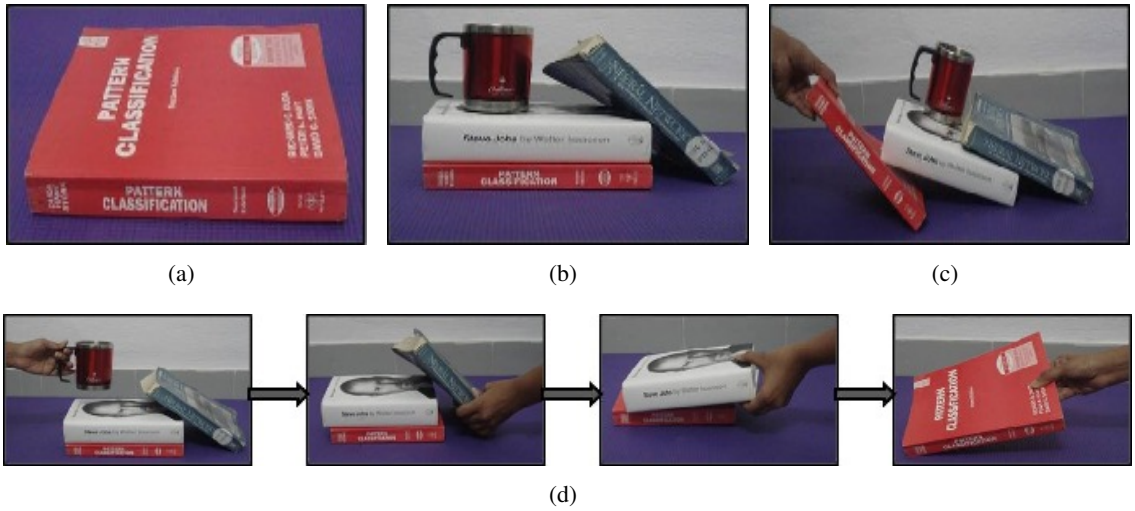
Chapter 1

Introduction

1.1 Motivation

Perception and scene understanding are challenging problems in computer vision and robotics. We perform countless daily chores involving object interaction like moving and placing utensils, grabbing a book from shelf, pick objects from piles, rearrange objects etc. We handle different objects differently. For example, we pick a cup directly without removing the spoon inside it, but carefully move aside other utensils before picking the one we want. Before picking a book from a pile of books on table, we move books on top of it whereas to pick a book from a book-shelf, we push and slide the books supported by it. However, such tasks are still a challenge for robots [43]. Most of the robotic manipulation tasks that involve clutter remain carefully restricted to objects in physical isolation and mostly lying on a planar surface [18, 44]. Learning the interaction among different objects in an environment can be of great benefit for robotic applications such as navigation [19, 26], grasping [12, 18] and object manipulation [19, 44]. In reality, cluttered environments are more complicated and involve physical contact, overlap and occlusion. Therefore, understanding of the interaction among different entities in the environment is important in order to facilitate for the robotic manipulation tasks in such settings. In this work, we attempt to learn the "object-object interaction" by answering the questions such as "Is this object graspable?", "What are the other entities it supports?" and "What are the entities it is supported by?".

Semantic interaction among objects in clutter is both essential and challenging for robotic manipulation. The knowledge of spatial layout, structural properties, physical contact of various types, direct and indirect support of the regions in a scene are useful in manipulation. At the same time, partial occlusion, complexity and diversity in physical contacts, presence of noisy background objects, wide range of shapes and sizes of indoor objects pose challenge to the problem of manipulation tasks. Grasping objects in a clutter is challenging not only due to the complex shapes of objects, but also due to the interaction they have with their surroundings. In order to grasp an object lying in clutter, the robots must take into account its interaction with its surroundings. Some recent works such as Dogar *et al.* [18, 19] and Gupta *et al.* [23] consider clutter for grasp manipulation tasks. However, most of the grasping and



(a) (b) (c)

(d)

Figure 1.1 Relevance of support order prediction: In order to manipulate an object, all objects it supports both directly and indirectly must be removed. (a) Object of Interest, (b) Clutter with physical contact and overlap, (c) Possible damage if semantic interaction is not considered, (d) Supported objects need to be removed sequentially to avoid damage.

grasp planning tasks are restricted to spatially isolated objects, thanks to complex interaction involved in objects in physical contact.

Recently, there has been many work on inference of pairwise support relationship: Rosman *et al.*'s [45] work on three types of support relationship among a pair of objects, Sjöö and Jensfelt's [50] work on four types of support relationships. Their work, however, limits to simulated environment only and not to real cluttered physical world, Silberman *et al.*'s [49] work on structure class and support inference, to name a few. However, these works either do not take into account the noisy background and real clutter, or do not consider interaction among multiple objects connected to each other. In a cluttered environment, an object is often simultaneously supported by multiple other objects apart from static entities such as wall, floor or furniture. We believe, extending the definition of support relationship to such "object-object interaction" will be of great benefit for robotic applications such as navigation, grasping and object manipulation.

1.2 Problem Statement

In clutter where objects lie overlapping on one another in many different ways, an object supports multiple other objects both directly and indirectly via other objects. In order to manipulate that object, it is important to identify the surrounding objects it supports (support relationship) and the order in which it supports them (support order) so that they can be removed prior to manipulating our object of interest without causing any damage (Fig. 1.1). The problem statement of our work is to understand the support relationship among objects placed in clutter and derive a support order in which surrounding objects

need to be removed in order to manipulate an object of interest. In our work, we aim at solving this problem by understanding different aspects of semantics of a scene such as graspability of an object, the entities supporting and (or) supported by the object and the kind of support. Learning the support relationships among different objects in the scene helps in understanding these aspects. We propose two approaches multiple object support inference and object-centric hierarchical support inference to infer support by multiple objects and inference of both direct and indirect support relationship. Then, based on the learned support relationships and given an object of interest, the order in which the surrounding objects should be removed, i.e., the "Support Order" is predicted.

There are many possible situations in clutter that can not be addressed using single view. It is not possible to detect hidden objects using single view. Often, spatial relation of partially occluded objects is inaccurate. Containment cannot always be correctly inferred using a frontal view. We believe that by exploring the scene from different views, different objects and different support relationships can be discovered, which otherwise is not possible using single views. Therefore, in our work, we have proposed an approach for support order prediction using multiple views to explore support relationships from different perspectives.

Exploiting multiple views for support order prediction is a challenging problem since it involves mapping across multiple regions. We move the Kinect [2] around the clutter in a sequence and capture the scene from different view points. With every consecutive view, some new objects are discovered, and some old objects are omitted. Different background entities create distraction and confusion. We deal with these complexities while mapping the regions across all the views. Using region mapping, a global mapping table is created that gives a one-to-one mapping among regions between different views. A scene from a particular view can be most similar to its previous and next views. Therefore, regions of one image are matched with only the regions of its previous and next views. Support inference is performed for each scene and a global support matrix is generated with the help of the global region mapping table. Using the global support matrix, support order prediction is done while taking into account different types of interaction among the objects in the clutter.

1.3 Contributions

In this research work, we focus on semantic interaction among objects in indoor setting. Some of our contributions are:

- We analyse spatial relationship or "support relationship" among objects in different levels of complexity: objects of varying shapes and size, noisy background with non-graspable objects present in it, clutter with varying level of complexity, both direct and indirect physical contact among objects, both pairwise support relations and hierarchical support relations among objects.

- We infer three types of support relations, "Support from Below", "Support from Side" and "Containment". We propose two alternative methods "Multiple Object Support Inference" and "Hier-

archical Support Inference" for support inference and compare to MAP inference adapted from Silberman *et al.* [49].

- We propose the idea of "Support Order Prediction" for objects present in clutter. Support Order is defined as the order or sequence in which surrounding objects need to be removed in order to grasp the object of interest. We consider different practical constraints in clutter and address them while predicting support order, in order to avoid damage.
- We extend support order prediction to multiple views. The cluttered scene is captured from multiple views. The regions from different views are mapped together and global support inference is generated from single views followed by support order prediction.
- We have created two RGBD datasets for our experimentation. "Indoor dataset for clutter" contains 50 images of 50 cluttered scenes. "Indoor multiview dataset" contains 67 images of 7 cluttered scenes. Manual annotation and ground truth generation are also done. For each image/view, its depthmap and point cloud are collected along with its RGB image using Kinect. We perform our experimentation in the same datasets. The datasets are also available for public usage.

1.4 Related Work

In this section, we discuss some of the state-of-the-art works in semantic interaction and applications like grasp-planning and manipulation using RGBD data. With availability of cheap depth sensors such as Kinect, it is easy to access depth information of an image. Depth data encodes significant geometric properties of the scene captured. Hence, it is advantageous to use both RGB and depth data in complimentary fashion for scene understanding purpose. RGBD is being increasingly used in many scene understanding tasks [38, 49] as well as object manipulation tasks [18, 19].

Dexterous Robots: Let us first throw light on different areas of applications where our idea can be utilized. Humanoids and dexterous Robots are developed as service robots for helping humans in various household work, HERB(Home Exploring Robotic Butler) developed by Srinivasa *et al.* [53] , STanford AI Robots(STAIR) developed by Stanford group [4], NASA Robonauts [3] built at NASA Johnson Space Center in Houston, Texas, Justin [1] developed by the German Aerospace Center (DLR) controlable through telepresence, to name a few. These robots have been extensively used for research in grasping. Providing them with the knowledge of how different objects interact with each other will only strengthen their ability to understand their environment.

Picking from Bin: Moving to industrial settings, automatically picking objects from a cluttered pile or bin is also an important operation where use of robots speeds up the performance. The main focus of such tasks is on finding a suitable position for the gripper and avoid collisions with the bin or other objects [52]. Lee *et al.* [35] perform surface segmentation and geometric primitive modelling followed by object recognition and pose estimation. Oh *et al.* [39] perform pick-and-place tasks for randomly piled parts in a bin through measuring the 3D pose of an object. But research in this application deal

with objects of only same pattern such as a pile of nuts or discs with specific primitive shapes and specific kind of machinery. Allowing objects of varied shape, size and amount of clutter will increase the scope of such applications to more generalized settings too.

Grasp and Grasp Manipulation: Now let us have a look at a few state-of-the-art research work relevant to our work. Varied shapes and sizes, and different kinds of semantic interaction among objects pose challenge to grasp-planning and manipulation tasks. Berenson *et al*'s work [9] is on autonomous grasping with dexterous hands in unstructured human environments. They propose a general algorithm for grasp synthesis in cluttered environments. They define a cost function for an object placed in a new scene. The cost function computes and encapsulates aspects of the object, the scene, and the force-closure of the ensuing grasp. However, their proposal is limited to the constraints like the robot hand must have a clear approach direction to the object and sufficient clearance around each contact point to allow the fingers to curl in and make contact. Dogar and Srinivasa's [19] grasp-planning framework in clutter aims at planning a grasp-strategy to displace the objects safely and efficiently in order to pick up an object. They propose four action plans, i.e., push-grasp, sweep, pickup and navigate. Their focus is on identifying occluding yet isolated objects and Dogar *et al.* [18] propose a planning method for grasping in cluttered environments where robot can reach for and grasp the object while simultaneously contacting and moving aside objects to clear a desired path. However, both the works avoid object-object interactions and assume that objects are located in spatially isolated manner. Physical contact among objects multiplies the difficulty level since it introduces complications in the kind of semantic interaction among the objects. We believe, understanding semantic interactions among different entities will enable such manipulation tasks to work in cluttered environment involving overlaps.

Pairwise Spatial Relationship: Rosman *et al.* [45] predict spatial relationships among different objects using stereo images. Their proposed algorithm predicts if an object A is "on" B or "adjacent to" B, or "both adjacent and on" B or "none", i.e., not related to B. However, they consider very simple objects and uncluttered settings and do not consider occlusion. Sjöö and Jensfelt [50] also work on understanding functional support relations among objects in a simulated environment. They find out four types of relations among each pair of object, i.e., casual support, support force, protection and constraint. They train a logistic regression classifier on a set of geometry based features on the simulated objects. Sjöö et al [51] aim at understanding the semantics of a scene in the context of searching for an object. They try to decide if an object is "on "or "in "another object and use this inference to decide the search space of the robot. Since, they use simulation, their work is restricted to the limitations imposed by the simulated environment such as the inherent imperfections of simulation, restrictions on different object geometries and scene configurations. Ye and Hua [57] infer 8 directional relationships between two different objects, but assume that one object does not contain or contact the other. Fichtl *et al.*'s work [20] is inspired by the perception of infants and explores two aspects of support relationship: support and obstruction and learning manipulation of the objects by interacting with them. However, their work is also limited to pairwise objects of basic geometric shapes and to a simulated environment. We wish to take into account more real objects with more generic shapes and sizes in typical clutter.

Interactive Grasp Manipulation: Recently, there have been couple of works that aim at handling objects in clutter with physical contact. Chang *et al.* [33] propose grasp strategies for piles of small objects on table. Katz *et al.* [15] provide an end-to-end framework for grasping in clutter with physical contact. They use interactive approach for segmentation of the objects in clutter. However, both approaches rely on push or poke techniques which may cause destabilization. Mojtahedzadeh *et al.* [42] apply spatial inference among objects in clutter and perform in-depth analysis of their support relations. However, they consider only cuboid shaped objects, thus are restricted to objects of a particular geometrical shape. Ornan and Degani [40] propose the concept of removability rank of each object in clutter with physical contact, however, they do not check for the effect of remaining objects if an object is removed. Aleotti and Caselli [6] propose non-destructive disassembly planning for objects in clutter, but are limited to simulations.

Indoor Scene Parsing: In the context of indoor scene parsing, Collet *et al.* [13] use range data and image data for structure recovery in a scene and use multiple segmentation approach with both RGB image and range data. Mishra *et al.* [38] perform contour-based segmentation on RGBD data exploiting the contact boundaries and depth boundaries of different objects present in the scene. Recently, Jia *et al.* [28] have proposed a 3D-Based Reasoning method. Their method is based on cluttered objects with blocks, spatial support and stability. They provide an improved segmentation of the indoor cluttered scene using these aspects.

Support Inference by Silberman et al.: Silberman *et al.* [49] have significant contribution to indoor scene parsing using RGBD data. They perform structure class inference and divide the objects present in the scene into four major classes such as floor, structure, furniture and props. In addition to this, they also infer support information, i.e., if a region is supported by another region or not and if yes, from which direction (from below or from behind). They use both depth and image cues to infer the structures and support relationships in heavily cluttered indoor environment. However, they do not exploit the spatial relationship among different objects that overlap onto each other. They only focus on finding major surfaces that support the objects in the scene. Due to this, they do not exploit more complex support relationships such as indirect support relationships and support by multiple objects. Hence, their work is not suitable for robotic applications where interaction among every object in contact with every other object is important. In our work, we attempt to address the above mentioned limitations by inferring support relations in clutter with noisy environment, physical contact, overlap, support by multiple objects and indirect support.

Our work focuses on learning the spatial support relationship among multiple objects in physical contact with each other. In addition, it predicts the support order in which the surrounding objects should be removed to pick up the object of interest without any damage to the environment. We believe, this extends the scope of different manipulation tasks as mentioned above to overlapping objects and occlusion cases.

1.5 Thesis Organization

This thesis is organized into five chapters. Brief overview of the content of each chapter is given below:

- The first and current chapter introduces the readers to the motivation behind our work. It also provides a brief overview of our work and explains our contribution in an attempt to solve the problem of semantic interaction among objects in clutter. In addition, different research work related to our idea and various applications to which our idea can be significantly useful are discussed.

- In Chapter 2, an overview of our work is presented with brief introduction to different modules of our framework.

- In Chapter 3, we discuss semantic interaction among objects in indoor settings. We propose a RGBD dataset of indoor objects of different shapes and sizes present in clutter of different levels of complexity. We infer support relationship between the objects and predict the support order for a given object of interest. We discuss two approaches for support inference and support order prediction.

- The support order prediction is extended into multiple views in Chapter 4 to address limitations of support inference in clutter using only single view. In this chapter, we discuss regions from different views are mapped to generate a global indexing for regions and how support inference is done in N views.

- Finally, we summarize our work and discuss future directions in the concluding chapter, i.e, Chapter 5.

Chapter 2

Overview of our Approach

2.1 Introduction

In clutter where objects lie overlapping on one another in many different ways, an object supports multiple other objects both directly and indirectly via other objects. In order to manipulate that object, it is important to identify the surrounding objects it supports (support relationship) and the order in which it supports them (support order) so that they can be removed prior to manipulating our object of interest without causing any damage. In this work, we learn spatial support relationship among different household objects in cluttered indoor environment. Three types of support relationships among different household objects present in clutter are learned: "Support from below", "Support from Side", and "Containment"(Fig. 2.1). We explore different approaches for learning support relationships. We explore the MAP inference method used in Silberman *et al.*'s work [49]. Then we discuss our multiple object inference method [41] to provide support by multiple objects. Then we discuss a hierarchical support inference method, where support inference is performed specific to an object of interest instead of inferring support relationship for all objects in the scene. A Tree of Support is then built keeping the object of interest at its root to encode the support relationship. The parent nodes of the tree represent supporting object and the child nodes represent supported objects. Tree of Support is traversed using Reverse Level Order Traversal in order to predict the Support Order for the object of interest. Special scenarios are identified and handled during Support Order Prediction so that minimal damage is caused to the environment.

Various geometric features are extracted using both RGB and depth data to robustly represent the physical properties of the objects. We demonstrate our results in RGBD dataset collected using Kinect in an indoor environment suitable for object manipulation. The dataset consists of various household objects in clutter with different kinds of support relationships. It captures scenes with different levels of complexity, occlusion, physical contact and noisy background, which makes it unique. We have also provided detailed annotation for the dataset. We have captured RGBD data for objects used in the scene at different orientations to aid object detection.

(a) (b) (c)

Figure 2.1 Illustration of different types of support relationships. Arrow heads point from supporting object to supported object. (a) Support from below. (b) Support from side. (c) Containment.

The rest of the chapter is structured as follows. An overview of the proposed framework is presented in Section 2.2. In Section 2.3, we discuss briefly about the segmentation approach used in our work. In Section 2.4, we discuss the object detection method used for detecting the object of interest. In Section 2.5, we discuss in detail about the features used for support inference. In Section 2.6, we discuss about Kinect sensor, RGBD data, our datasets and annotations. In the end, we provide a summary of the chapter in Section 2.7.

2.2 Overview

The various steps of our work are briefly explained in this section. As shown in the block diagram in Fig. 2.2, the framework includes segmentation module, object detection module, support inference module, and support order prediction module. The input images are segmented into different regions. These segmented regions are provided as input to object detection module and support inference module. Finally, given the region of interest and the support relationship among each pair of the objects, we predict the support order.

The support inference module infers the supporting regions and type of support for each region in the image, given the image regions and various geometric features. We first build on the MAP inference method presented in [49]. Then, we apply alternative methods to allow multiple object support inference and hierarchical support inference. We define 2D and 3D geometric features that exploit the pairwise support relationship among two objects. MAP inference method optimizes the pairwise support relation among objects, support type and structure classes using linear programming. However, it does not infer support by multiple objects. To infer the multiple object support relationship, we proposed rule-based inference method. In addition to this, we implement a hierarchical support inference approach. In this approach, the graspable objects are segmented out at first and then given an object of interest, support inference is performed between pair of objects hierarchically. The details of different geometric features used are discussed in Section 2.5 and the support inference methods can be found in Section 3.1.

A Tree of Support is built with our object of interest at the root of it using the inferred support relationship. The child nodes in the tree represent the objects supported by the object corresponding to the parent node(s). A detailed discussion on the approach for support order prediction and how different specific scenarios are handled is given in Section 3.2.

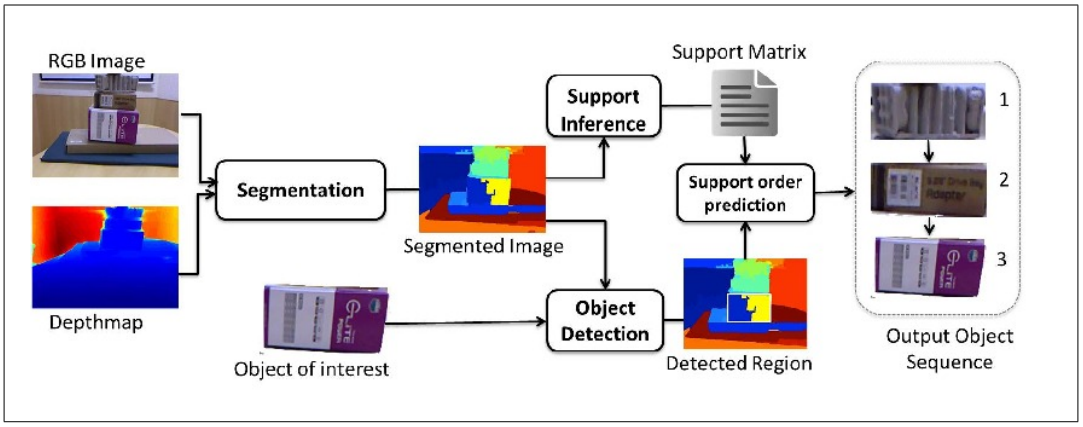


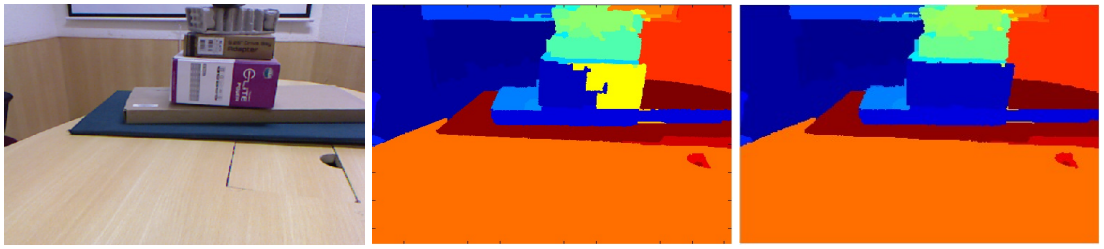
Figure 2.2 Block diagrammatic representation of our framework: Segmentation module takes RGB and depth images as input. Segmented image is provided as input to both Support Inference and Object Detection module. Object Detection module also takes the image of object of interest as input and outputs the detected region. Support Inference module gives the support relationship among each pair of regions. Support order prediction module uses the detected region and the support relationship to predict the order in which the objects should be picked.

2.3 Segmentation based Approach

The images are first segmented into superpixels using watershed algorithm applied to Pb boundaries proposed by Arbelaez *et al.* [8]. A 5-stage hierarchical segmentation method by Hoiem *et al.* [27] is used to segment the scene using the superpixels. Both 2D and 3D geometric and photometric features of the images are used for segmentation [49]. In this approach, a boosted decision tree classifier is trained to predict the boundary strengths of each image region and regions with boundary strength below a threshold are iteratively merged to give optimal segments. Both 2D and 3D features play important role in segmentation. 2D features distinguish between objects close to each other, whereas 3D features distinguish object edges from object textures. The segmented regions are provided as input to the object detection and support inference modules.

2.4 Object Detection

In the object detection module, dense SIFT descriptors [37, 55] of both the template image of the object of interest and the input image are extracted. Then SIFT feature matching is performed between the two images. The outliers among the matched points are discarded by applying RANSAC [21, 32]. The segmented regions corresponding to the matched points of the input image are merged into one region and chosen as the region corresponding to object of interest O, i.e., the object to be grasped (Refer Fig. 2.3). Before merging the segmented regions, they are geometrically as per rule shown in Algorithm 1. Insignificant segments are identified as the segments contributing to the matched points or inliers less than a threshold (say, 10%). If number of matched points are less than a threshold (say,



(a) Input Image (b) Segmented Image (c) Segmented Image after region merging

Figure 2.3 Result of 5-stage hierarchical segmentation method by Hoiem *et al.* [27] and object detection. The box in violet as shown in (a) is segmented into two regions (yellow and blue) in (b). After object detection, the two regions are merged together, as shown in (c).

7 points), it is concluded that the object is not found in the image. If the segmented regions do not share boundaries, then connected component analysis is performed to segregate them into blobs. The minimum distance of the blob with maximum number of inliers from other blobs is found out. The blob with distance greater than a threshold distance (say, 100 pixels) is considered as a false match and is discarded. The remaining segmented regions are merged together as the object of interest O.

Procedure 1 Conditions for Region Merging for Object Detection

---

Input: number of inliers **n**, inliers **M** = $\{M_i\}$, segmented image **I** = $\{S_i\}$, where S_i = ith segmented region.

Output: updated number of inliers **n**, inliers **M** = $\{M_i\}$, corrected segmented image **I_s** = $\{S_i\}$, where S_i = ith segmented region after merging.

if n $< TH\_INLIERS$ **then**

'No match found.'

EXIT

end if

$\{S_k\} \leftarrow$ segments containing $\{M_i\}$.

$ns_k \leftarrow$ number of inliers of kth segment S_k.

contribution of each segment $contri\{S_k\} = \frac{ns_k}{n}$

Discard S_k with $contri\{S_k\} < TH\_CONTRI$.

$\{CC_k\} \leftarrow$ connected components of remaining segments$\{S_k\}$.

$\{CC_{max}\} \leftarrow$ connected component or blob with maximum inliers.

$\{d_k\} \leftarrow$ minimum_distance($CC_{max}, \{CC_k\}$) for all k.

Discard the blob(s) whose $\{d_k\} > TH\_DISTANCE$.

Update **n**.

Merge the remaining $\{S_k\}$ into one region.

Update **I_s**.

return **n**, **I_s** = $\{S_i\}$

Multiple templates for an object are stored in the database. The same process is repeated for all the templates. The template with maximum number of matching points is taken as the most matching template and corresponding matching regions are decided as the regions corresponding to the object of interest.

The object of interest *O* is given as input to the support inference module. The regions supported by the object of interest both directly and indirectly are predicted in the support inference module. Then their order (Support Order) is predicted so that objects can be manipulated while causing minimal damage to the environment.

Through our approach of object detection, it is ensured that the entire object region is chosen for grasping. However, it is possible that an object is not uniquely represented by a single image segment, thanks to the limitations in segmentation. With the advent of more accurate segmentation techniques, this problem can be solved.

2.5 Geometric Features for Pairwise Support Relationship

Given the segments or regions that correspond to each object, a set of 2D and 3D geometric features which exploit the support relationship between each pair of objects are introduced for support inference. While 2D geometric features are (boundary ratio) extracted from the RGB image, 3D features (proximity, depth boundary, containment and relative stability) are extracted from the the depthmap and the point clouds. Thanks to devices like Kinect, the depth information is easily available and provides significant information about the geometry in 3-dimensions which cannot be extracted from 2D images. These features are described in the following subsections.

2.5.1 Proximity

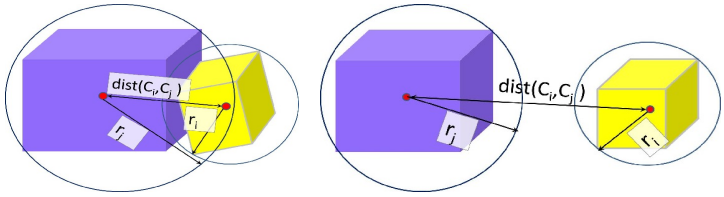
Two objects must be in each others' proximity in order to provide support to each other as shown in Fig. 2.4. Proximity f_p of two objects i and j can be measured by the ratio of the distance in 3D between their centroids C_i and C_j and the sum of radii r_i and r_j of the sphere circumscribing the two regions as described by the following equation:

$$f_p(i,j) = \frac{\text{dist}(C_i, C_j)}{(r_i + r_j)}. \tag{2.1}$$

Value of $f_p(i,j)$ is less than 1 for objects close to each other and greater than 1 for far-away objects.

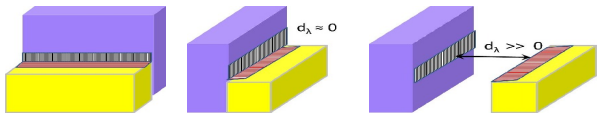
2.5.2 Depth Boundary

In case of visual occlusion, two objects may be either actually in contact or may be isolated from each other (Fig. 2.5). The feature "depth boundary" discriminates between these two situations [38]. Plane-fitting is done corresponding to two regions adjacent to the boundary between the two objects. If



(a) at close proximity, $f_p(i,j) < 1$ (b) at distance, $f_p(i,j) > 1$

Figure 2.4 Demonstration of proximity: lesser f_p implies closer proximity and higher f_p implies less proximity.



(a) Visual occlusion (b) Side View: actual contact (c) Side View: no contact

Figure 2.5 Demonstration of depth boundary concept: The regions in black and red imply two planes fitted along the boundaries of the two objects. (a) shows two objects in visual occlusion with two possibilities. (b) shows the side view where a contact boundary exists between the two objects. (c) shows the side view where a depth discontinuity exists.

the two objects are isolated, then the 3D planes of the objects do not intersect and a depth discontinuity or "depth boundary" exists between the two of them (Fig. 2.5(c)). Otherwise, they intersect at a certain angle and a "contact boundary " exists between the two of them (Fig. 2.5(b)). Let $d_\perp$ be defined as the average of the maximum 3D distance of the boundary pixels from the two planes measured in meters. $d_\perp$ tends to zero for contact boundaries and has higher values for depth boundaries. Depth boundary is measured by a logistic function f_{depth} as follows:

$$f_{depth}(i,j) = \frac{1}{1 + e^{-(\beta_1 d_\perp(i,j) + \beta_2)}}. \tag{2.2}$$

Here, f_{depth} tends to 0 for objects not in contact with each other and tends to 1 for objects in contact. β_1 and β_2 are learned using logistic regression with a few training examples.

2.5.3 Boundary Ratio

As shown in Fig. 2.6(a) and 2.6(b), a significant overlap exists at the object boundaries when two objects are in contact. The boundary lines are shown in black in the figures. In Fig. 2.6(a), there is significant boundary between the two objects showing greater chances of support, while in 2.6(b) smaller boundary implies less chance of support. The feature "boundary ratio" measures the overlap of a pair of objects over each other. Boundary ratio f_{br} is computed using the following:

$$f_{br}(i,j) = \frac{L(i,j)}{\text{perim}(i)}. \tag{2.3}$$

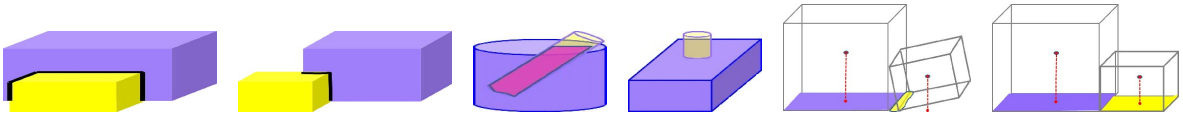


Figure 2.6 Demonstration of boundary ratio, containment, and relative stability. Boundary Ratio: (a) larger boundary, (b) smaller boundary. Containment: (c) Object contained inside another, (d) Object on top of another object. Stability: (e) unstable supported object, (f) stable objects.

Here, $L(i, j)$ is the length of visual boundary of the supported object i with the supporting object j, and perim(i) is the perimeter of supported object region in the RGB image. i.

2.5.4 Containment

If an object is contained inside another, we need not remove the supported object for picking up the supporting object. The feature "containment" measures how much volume of the supported object is contained inside the supporting object (Fig. 2.6(c), 2.6(d)). It is defined as the fraction of the number of points that belong to the supported object N_i contained inside the 3D convex hull Hull(j) of the supporting object j.

$$f_{cnt}(i, j) = \frac{N_i \cap \text{Hull}(j)}{N_i}. \tag{2.4}$$

In Fig. 2.6(c) and 2.6(d), the object in purple implies the supporting object and the object in yellow implies the supported object. Region in magenta shows the portion of the supported object contained inside the convex hull of the supporting object.

2.5.5 Relative Stability

Relative stability between two objects in contact is defined as:

$$f_{stab}(i, j) = \begin{cases} -1, & \text{if } i \text{ stable and } j \text{ unstable} \\ +1, & \text{if } i \text{ unstable and } j \text{ stable} \\ 0, & \text{otherwise} \end{cases} \tag{2.5}$$

A stable object has higher probability of supporting its neighbouring objects compared to an unstable object. An object is stable if its gravity line is in alignment with the baseline, otherwise it is unstable and needs support from side as depicted in Fig. 2.6(e) and 2.6(f). If the horizontal projection of the centroid of the object belongs to the 2D convex hull of horizontal projection of the baseline points of the object, then the object is considered as stable. In the figures, the region in violet shows the baseline of left object and the region in yellow shows the baseline of the right object. The lines in red show the gravity lines. In 2.6(e), the horizontal projection of the centroid of the right object does not belong to the baseline and hence the object is unstable. In 2.6(f), the horizontal projection of the centroids of both the objects lies in their baseline, hence both are stable.

Figure 2.7 Indoor dataset for clutter: Objects of various sizes and shapes with different kind of support relationship among each other.

The features discussed above are used in support inference methods as discussed in the next Chapter 3. In multiple object support inference (Section 3.1.2), the features are used explicitly. In MAP inference (Section 3.1.1) and hierarchical support inference (Section 3.1.3), the features are added to the support features proposed in [49].

2.6 Datasets

RGB cameras capture color images which give rich information about the colour, intensity and textures of the scene. However, they do not give direct information about geometry of the scene, since the images provide 2D information only. Depth sensors, on the other hand, provide accurate information about geometry of the scene in 3D. However, laser range sensors are expensive. With the development of cheap sensors such as Kinect [2], it is now possible to combine the advantage of both RGB and depth sensors. Kinect is a product of Microsoft and is a motion sensing input device, commercially used for Xbox 360 and Xbox one video game consoles [5]. Kinect sensor consists of a RGB camera, depth sensor and multi-array microphone. The depth sensor consists of an infrared laser projector and a monochrome CMOS sensor. It can capture video data in 3D under ambient light conditions. Using the RGB camera, the depth sensor and the microchip, Kinect uses a variant of image-based 3D reconstruction to capture the 3D information in the scene. The 3D information can be stored as point clouds(*.pcd* files) along with the RGB image (default resolution 640x480 pixels) and the depth map of the scene. The output video of the sensor is in the range of 9 to 30 Hz. Its field of view is 57 and 43 in horizontal and vertical direction respectively. It has a motorized pivot that can tilt the sensor up to 27 up or down. Due to less cost, and the ability to use both RGB camera and depth sensor in real-time, Kinect is increasingly being popular in the robotics community [14, 24] in various indoor applications such as object tracking and recognition, human activity analysis, hand gesture analysis, and indoor 3-D mapping. In our work, we have used both RGB and depth information so as to exploit both color and depth information. Therefore, we have used RGBD data and Kinect as the best choice.

Figure 2.8 Template images for the 35 objects used in Indoor dataset for Clutter. The dataset contains objects of different shapes and sizes. The objects are kept in row-wise order of indexing.

For object manipulation, it is desirable that the objects are in the vicinity of the camera, at a reachable distance from the robot arm and have overlap among one another. In the publicly available datasets for cluttered environment such as NYU depth dataset [48] and Cornell Scene Understanding dataset [7,31], the graspable objects are usually present in a far corner of the room instead of being in the center. In RGBD dataset proposed by Lai *et al.* [34], the objects are mostly placed in uncluttered settings. In addition, to our knowledge, multiple view datasets using RGBD are not available in public. This necessitated creation of our own datasets. We created two datasets "Indoor dataset for clutter" and "Indoor multi-view dataset" with objects in different types of clutter using Kinect.[1] Details about the datasets are discussed in the following subsections. The datasets capture different support relations among objects in clutter overlapping on one another.

2.6.1 Indoor Dataset for Clutter

We have collected a dataset consisting of 50 images with different levels of clutter along with their point clouds and depth images using Kinect. Some of the images of our dataset are shown in Fig. 2.7. In this dataset, one can find different kinds of object interactions such as support from below, support from side, containment, support in hierarchy, simultaneous support by multiple objects, partial occlusion, noisy background with furniture etc, objects lying on different planes. These aspects make our dataset unique and suitable for experimentation on clutter using RGBD data. In addition to this, the images of objects used in the dataset are also created for object detection. For each object, all possible views are captured. Dense labelling for each image is done manually. The dataset is divided into training and test data in 30 : 20 ratio for training and testing purpose.

[1]Datasets available at
`http://cvit.iiit.ac.in/projects/semantic_interaction_indoor_objects/`

Figure 2.9 Indoor multi-view dataset: Objects in clutter captured from different views. Each row corresponds to images of a single scene captured from different views.

Along with the RGBD dataset of cluttered images, we have also captured the individual objects used in our dataset. Total 35 number of objects of different shapes and sizes are used in our dataset. In Fig. 2.8, the objects are kept in row-wise order of their indexing. For each object, multiple templates are stored corresponding to different view points. During object detection, all the templates of the object of interest **O** are matched with the clutter image and the results corresponding to the best matching template are taken into account.

2.6.2 Indoor Multi-view Dataset

"Indoor Multi-view Dataset" consists of 7 scenes captured from multiple views. There are total 67 images and 9 objects along with corresponding depthmaps and point clouds. Some of the images used in the dataset are shown in Fig. 2.9. Each row in the image corresponds to images of a single scene captured from different views. In this dataset, one can find different kinds of object interactions such as support from below, support from side, containment captured from multiple views. These aspects make our dataset unique and suitable for experimentation on clutter using RGBD data. In addition to this, the images of objects used in the dataset are also captured for object detection. For each object, multiple templates all possible views are captured to be used for object detection. The 9 objects are shown in Fig. 2.10 in row-order of their indexing. We have created dense labelling for each image using LabelMe [46]. This dataset is used as test data while the data captured for support order prediction in single view [41] is used as training data.

2.6.3 Annotation

Four kinds of annotations are created for each image in both the datasets: (a)Object label, where each label represents each individual region. (b) Structure class label, where each region is assigned one of four labels each for floor, structure, furniture, prop. (c)Object category label, where each label

Figure 2.10 Template images for the 9 objects used in ndoor multiview dataset. The objects are kept in row-wise order of indexing.

corresponds one category of object. (d)Object instance label, where each instance of two objects of same category is given one label. Further, support matrix is created for the regions of each image. Support matrix encodes the ground truth support relationship among each pair of region in the form of a set of 3-tuples: $[R_i, S_i, T_i]$. The raw depthmaps are smoothened using an adaptation of colorization method by Levin *et al.* [36].

2.7 Summary

In this chapter, we presented an overview of our work. In addition, we discussed briefly about Segmentation, Object Detection and various geometric features proposed by us. These processes precede the main modules i.e., support inference and support order prediction discussed in detail in the next chapter. We also discussed about the datasets created by us in detail.

Chapter 3

Semantic Interaction in Clutter

3.1 Learning Support Relationship

In this section, we explore different approaches for learning support relationships. We explore the MAP inference method used in Silberman *et al.*'s work [49]. Then we discuss our multiple object inference method [41] to provide support by multiple objects. Then we discuss a hierarchical support inference method, where support inference is performed specific to an object of interest instead of inferring support relationship for all objects in the scene.

3.1.1 MAP Inference

The structure class of all regions in the images and the support relation among each pair of region is inferred using a probabilistic energy framework given in equation (3.1). A joint probability distribution is defined in terms of supporting regions, structure class and support type adapted from [49]. The random variable $\mathbf{S} \in \{S_1, \ldots, S_R\}$ represents the support regions corresponding to each of the R regions of the image. $S_i \in \{-1, 0, 1, \ldots, R\}$ represents support region for each region $i \in \{1, \ldots, R\}$ where, a hidden region is denoted by -1 and ground denoted by 0. The variable $\mathbf{T} \in \{1, 2, 3\}^R$ represents support type. $T_i = 1$ implies support from below, $T_i = 2$ implies support from a side and $T_i = 3$ implies containment. The variable $\mathbf{M} \in \{1, ..., 4\}^R$ represents four structure classes viz, floor, structure, furniture and props.

$$\begin{aligned} \{\mathbf{S}^*, \mathbf{T}^*, \mathbf{M}^*\} &= argmax_{\mathbf{S,\ T,\ M}} P(\mathbf{S,\ T,\ M}|I) \\ &= argmin_{\mathbf{S,\ T,\ M}} E(\mathbf{S,\ T,\ M}|I), \end{aligned} \tag{3.1}$$

where, $E(\mathbf{S,\ T,\ M}|I) = -logP(\mathbf{S,\ T,\ M}|I)$ is the energy of the joint probabilistic distribution. The energy is defined as

$$E(\mathbf{S,\ T,\ M}) = -\sum_{i=1}^{R} log(D_s(f_{su}|S_i, T_i)) + log(D_m(f_{st}|M_i)) + E_P(\mathbf{S,\ T,\ M}) \tag{3.2}$$

In this equation (3.2), f_{su} are the support features and f_{st} are the structure class features. D_s is support classifier trained to maximize $P(f_{su}|S_i, T_i)$ and D_m is structure class classifier trained to maximize $P(f_{st}|M_i)$. E_P(**S, T, M**) is the prior energy term which encodes various properties of scene elements such as probability of different structure classes supporting each other, proximity of two regions in support, enforcing lowest region to be floor. MAP inference is solved by using linear programming.

However, the work of Silberman *et al.* [49] is aimed for indoor scene parsing where a comparatively larger surface region is chosen as the region supporting a region. In robotic tasks, objects of similar and even smaller surface area may also support other objects and it is important to infer such support relationship. Moreover, the approach of MAP inference imposes a constraint that one object can be supported by only one other object. The support relation among multiple objects are not taken into account even if they support each other, which is inappropriate for most of the robotic tasks. To overcome this restriction, we developed a multiple object method to infer support by multiple objects as discussed in the next section.

3.1.2 Multiple Object Support Inference

In this approach [41], explicit use of the features discussed in Section 2.5 is done for support inference. An illustration of the function of this approach is shown in Fig. 3.1. A structure class classifier is trained to classify the structure classes of different regions using neural networks. If the classifier predicts any region as "floor", then vertical structures and furniture are decided to be supported directly by the floor. Otherwise it is assumed that floor is not visible in the scene. Identifying vertical structures such as walls and windows and furniture such as tables, chairs, cupboards and sofas plays a significant role to avoid infeasible support inference such as a small object supporting a wall or a table. For a prop or a graspable object, different types of support are inferred by considering its surrounding region. Objects lower to the current object whose centroids are closer to the current object are selected (Proximity, f_p) as potential candidates for providing "support from below". In case of conflict, the ones with higher boundary ratio (Boundary Ratio, f_{br}) are chosen as regions providing "support from below". If a significant portion of 3D convex hull of the current object belongs to the 3D convex hull of the supporting region (Containment, f_{cnt}), the support is termed as "containment". All regions in contact with the current object (Depth Boundary, f_{depth}) other than the regions below are considered as "support from side", if they are labelled as stable regions (Relative Stability, f_{stab}).

The algorithm of multiple object support inference method is given in Procedure 2. The symbols in bold text denote arrays instead of scalar values. The subscript notation ik implies the index $i + k$. After support inference is performed, the support order for a given object of interest is predicted as discussed in Section 3.2.

Procedure 2 Multiple Object Support Inference Method

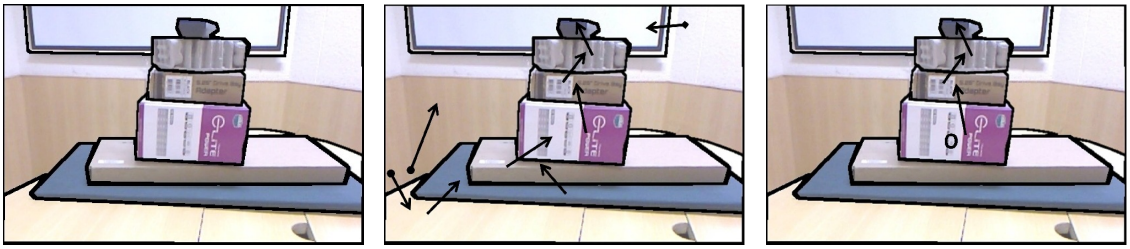
Input: Image regions **R**, structure class features $\mathbf{f}_{st}$

Output: the support relationship **S**, **T** among all the regions **R**. structure class of each region **M**.

$\mathbf{M} \leftarrow predictStructureClass(\mathbf{R}, \mathbf{f}_{st})$
if CanSeeFloor **then**
  $floorId \leftarrow findFloorRegion(\mathbf{R}, \mathbf{f}_{st})$
else
  $floorId \leftarrow -1$
end if
for each region i **do**
  if $M_i = FLOOR$ **then**
    $S_i \leftarrow 0, T_i \leftarrow 0$
  else if $M_i = STRUCTURE\ or\ FURNITURE$ **then**
    $S_i \leftarrow floorId, T_i \leftarrow 1$
  else
    $k \leftarrow 0$
    $B_i \leftarrow findSupportFromBelow(i, horzSupport(i, \mathbf{R}))$
    if $\mathbf{B}_i$ *not empty* **then**
      $\mathbf{B}_i \leftarrow findClosestSupport(i, \mathbf{B}_i, PROX(i, \mathbf{B}_i))$
      $S_{ik} \leftarrow findBestSupport(i, \mathbf{B}_i, BND(i, \mathbf{B}_i))$
      $T_{ik} \leftarrow findContainment(i, CNT(i, S_{ik}))$
    end if
    $k \leftarrow k + 1$
    $\mathbf{SS}_i \leftarrow findRegionsInContact(i, L_{\perp}(i, \mathbf{R}))$
    $\mathbf{SS}_i \leftarrow discardRegionsFromBelow(i, \mathbf{B}_i, \mathbf{SS}_i)$
    $\mathbf{SS}_i \leftarrow discardRegionsSupported(i, horzSupport(\mathbf{SS}_i, i))$
    $\mathbf{SS}_i \leftarrow chooseStableRegions(i, STAB(i, \mathbf{SS}_i))$
    for $each\ j\ in \mathbf{SS}_i$ **do**
      $S_{ik} \leftarrow \mathbf{SS}_i(j)$
      $T_{ik} \leftarrow 2$
      $k \leftarrow k + 1$
    end for
  end if
end for
return **S**, **T**, **M**.

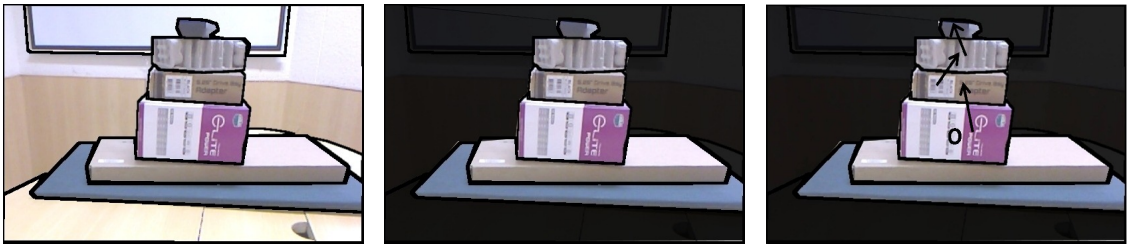
3.1.3 Hierarchical Support Inference

Unlike the above methods, in hierarchical support inference method, support inference is performed w.r.t the object of interest instead of inferring support relations for all objects. The process of hierarchical support inference is illustrated in Fig. 3.3 and an illustration of the function of this approach is shown



(a) (b) (c)

Figure 3.1 Illustration of Multiple Object Support Inference Method: (a) Input image with boundaries of different regions marked in black. (b) Support relations among all regions. (c)Support Order Prediction: Object of Interest *O* and the objects supported by it.



(a) (b) (c)

Figure 3.2 Illustration of Hierarchical Support Inference Method: (a) Input image with boundaries of different regions marked in black. (b) Structure class inference: Structure classes such as floor, wall, and furniture are identified and discarded. (c)Support Inference & Order Prediction: Object of Interest *O* and the objects supported by it are inferred hierarchically.

in Fig. 3.2. Given the segmented regions and the target object, we apply a cascade of classifiers for support inference. Given a cluttered scene to grasp objects from, it is important to first determine which objects are graspable. An indoor environment typically consists of entities with distinct structural properties such as floor, walls as well as entities with highly diverse structural properties such as furniture. A logistic regression classifier is trained to predict the structure class (floor/wall/furniture/graspable objects) of each region. The output of the classifier is interpreted as probability of the region of belonging to a particular structure class. The structure class with maximum probability is assigned to the region. The regions predicted as graspable objects are selected for support inference while other regions are discarded as background.

In order to access any object *O*, all objects that *O* supports must be identified and removed. Our goal is to predict these supported objects. Pairwise support relationship among these objects is inferred using a 3-layer feed-forward neural network based support classifier. We perform support inference hierarchically instead of comparing all regions with each other for efficiency. Given a pair of regions (A, B), the support class classifier predicts if B supports A "from below", "from side", "contains" it or "not related" to it.

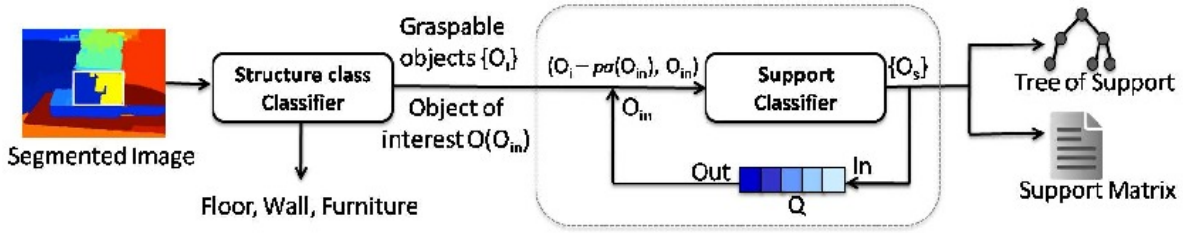


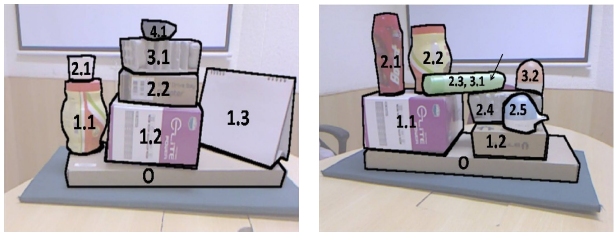
Figure 3.3 Hierarchical Support Inference Module: The graspable object regions are filtered using structure class classifier. Support inference is performed hierarchically. Tree of Support and Support Matrix are generated as final output.

The support features for the regions corresponding to only the graspable objects are extracted and normalized for training the neural network. Sigmoidal activation function is used for the four output units to limit all the outputs to a fixed range([0 1]) so that the outputs can be interpreted as probabilities. The number of hidden nodes is kept as one tenth of the size of the training data to avoid over-fitting. Stochastic gradient descent algorithm is used to minimize the cross-entropy loss function which is appropriate for dealing with probabilities. The most probable support class is assigned to the pair of regions given as input.

Support inference is performed hierarchically beginning with the target object. Suppose, $S = \{O_i | i = 1, \ldots n\}$ is the set of n graspable objects/regions in the scene and $O \in S$ is the target object. A Tree of Support T is built with the target object in the root to encode the predicted support relationship. Instead of inferring for all possible pairs of objects which will require $n(n-1)/2$ comparisons, support relations of all the objects that O directly supports are inferred at first. Every object is paired with O and support relations of each of the pairs $\{(O_i, O) | O_i \in S - \{O\}\}$ are inferred from the support classifier. The objects $S_s = \{O_s\}$ predicted as objects supported by O are stored in a FIFO queue Q. They are also inserted into the tree T as child nodes of O. The object pulled from the queue Q is again fed to the support classifier to predict the objects that it supports in turn. All other regions except the supporting regions(s) or parent(s) $pa(O_s)$ and grandparents $pa(pa(O_s))$ of O_s, are paired with O_s for the prediction. Note that special care is taken to discard all the parents and grandparents of the object O_s in order to avoid loops which may lead to damage in practical scenario. The support inference is performed hierarchically until all the outcomes of the classifier are negative, i.e., all the support relationships are predicted as "not related"and consequently the queue Q is empty. The advantage of performing support inference hierarchically instead of inferring for all pairs of objects is that this approach significantly reduces the number of comparisons required from O(n^2) to O($nlogn$).

3.2 Support Order Prediction

The objects supported by an object *O* both through direct and indirect contact need to be removed prior to grasping *O*. So it is necessary to recursively find out the objects which are supported by *O*, and



(a) Case 1 (b) Case 2

Figure 3.4 Case 1: Support in hierarchy in (a). The objects are supported by one another in hierarchical manner. Therefore, in order to pick up the desired object, all the objects in the hierarchy need to be picked up one by one.Case 2: Simultaneous support in multiple hierarchy in (b). The green bottle (pointed by an arrow) is supported by objects in multiple hierarchy. So it should be treated as an object in layer 3 and removed before removing other objects in layer 2.

(a) (b) (c)

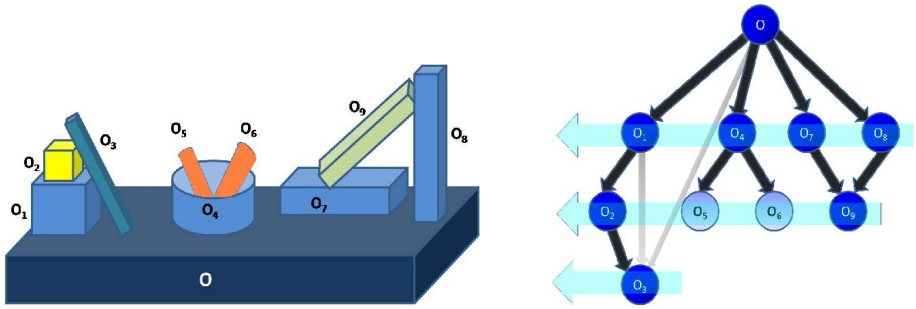
Figure 3.5 Case 3: Containment. In case of containment, the contained object need not be removed while removing the containing object. In (a) the basket can be directly grasped alongwith object contained in it. In (b), the plastic bottle can be directly picked up since it does not support any other object. In (c), the basket can be directly removed for grasping the board without removing the bottles.

the objects that these objects support in turn. In this section, we discuss our approach for determining the support order by means of "Tree of Support" of the objects surrounding our object of interest. Before that, we explore the different possible cases of support.

3.2.1 Different Cases of Support

In this section, we discuss different possible cases while we do support order prediction. Trying to provide a generalized solution to handle all the cases, categorize them into three different cases. We treat each case differently and provide solution to them. The first and most generic case is illustrated in Fig. 3.4(a) where one object supports the other in hierarchical fashion. There can be possibility that one object is supported by multiple objects. Therefore, we should remove all 4.* (all objects in layer 4) first, then 3.* and so on by adopting reverse level order traversal.

In the second case, one object may be supported by objects at two different hierarchies. For example, in Fig. 3.4(b), the green bottle is supported by two objects object 1.1 and 2.4. It gets two labels 2.3 and 3.1. During such conflict, label 3.1 is kept and the label 2.3 is discarded. So the green bottle is removed prior to removing any other object in layer 2 of the hierarchy, i.e., the object labeled 2.4.



(a) (b)

Figure 3.6 Example to demonstrate support order prediction. (a)$\{O, O_1, O_2, O_3\}$ represent support in multiple hierarchy; $\{O_4, O_5, O_6\}$ represent containment and $\{O_7, O_8, O_9\}$ represent simultaneous support by multiple objects. (b)Tree traversal is done from leaf nodes towards the root node. O_3 is connected to O_2, O_1 as well as O which implies O_3 is supported by O_2, O_1 and O. In this case the edges connecting to parent nodes at all the higher hierarchy are pruned (edges shown in gray). Nodes O_5 and O_6 (shown in light blue) are contained in node O_4. These nodes are skipped during reverse level order traversal.

The third case arises when one object is contained in another, instead of merely supporting, for example the plastic bottles in the basket as shown in Fig. 3.5. If the object *O* is the basket as shown in Fig. 3.5(a) , the basket is directly grasped without any need to remove the plastic bottles present in it. If the object *O* is one of the plastic bottles, i.e, the object which lies inside some other object as shown in Fig. 3.5(b), it can be picked directly since it does not support any other object. Now, suppose the object *O* is the board which supports the basket. In that case, it is tested if objects 2.* are inside the object 1.1. If yes (the case of Fig. 3.5(c)), then 1.1 is removed directly. Otherwise, all the objects 2.* are removed before removing 1.1. This idea is implemented using reverse level order traversal as explained in detail in the next Section 3.2.2.

3.2.2 Tree of Support

In order to determine the "support order", a tree of support is built with the object of interest placed at the root of the tree. The parent node in the tree represents supporting object and the child node represents supported object. Tree traversal is performed using reverse level order traversal. The objects present at the leaf nodes are the ones not providing support to any other object. So they are picked up first and then, the upper layer is traversed and the process repeats until we reach the root node that is our object of interest.

The cases discussed in Section 3.2.1 are taken care of during tree traversal to ensure minimal damage while manipulation. In case of support by multiple hierarchy (Fig. 3.4(b)), the child node corresponding to the supported object is connected to multiple parent nodes from different layers. It is not feasible to retain all edges connecting to the child node. Retaining any of the edges in the upper layer(s) implies that the object corresponding to the child node will be searched even after its removal. If the edge to the

parent node(s) at lower layer is pruned, then while picking the object corresponding to this parent node, the presence of the supported object will be ignored which may cause damage. Therefore, the edge(s) between the child node and the parent node(s) at the lowest layer are retained while pruning off edges connected to parent node(s) in the upper layer(s). During tree traversal, prior to retrieving any node, if the support type for a node is found to be "containment", then, this node is not retrieved since we do not need to pick it up for grasping the object containing it, as discussed in case 3 in Section 3.2.1 and shown in Fig. 3.5.

Graphical demonstration of the tree traversal is depicted in Fig. 3.6(b) for objects shown in Fig. 3.6(a). The dark edges represent valid connections. Lighter edges denote the connections removed in case of support by objects of multiple layers. The nodes in light color denote objects contained in the objects corresponding to their parent nodes. We traverse from the leaf nodes towards the root node. The support order is predicted as

$$O_3 \rightarrow O_9 \rightarrow O_2 \rightarrow O_8 \rightarrow O_7 \rightarrow O_4 \rightarrow O_1 \rightarrow O.$$

The system is advised to pick the objects from scene in the predicted order.

3.3 Experiments & Results

In this section, we validate the inferred support relationship and support order on various images from our RGBD dataset.

3.3.1 Segmentation

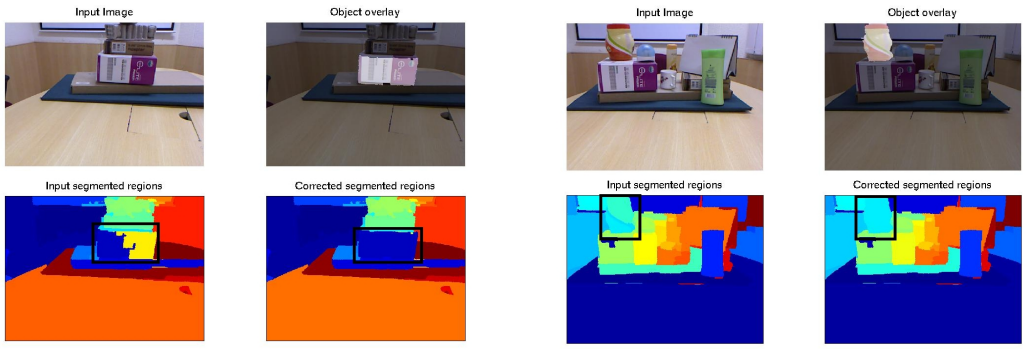
A 5-stage hierarchical segmentation approach proposed by Arbelaez *et al.* [8] was used for segmenting the images. RGB and depth features used in [49] are used for segmentation. Achieving accurate segmentation is important since performance of segmentation has direct impact on subsequent stages of inference. Segmentation accuracy is measured as average overlap of segmented regions over groundtruth regions as defined in [27]. The unweighted average overlap score and the score weighted by pixel area are given in Table 4.1(b).

Table 3.1 Accuracy of Hierarchical Segmentation, measured as average overlap over ground truth regions for best-matching segmented region, either weighted by pixel area or unweighted.

Type	Training Accuracy	Test Accuracy
Weighted	87.1	75.4
Unweighted	74.3	60.4

3.3.2 Object Detection

Some of the results of object detection are shown in Fig. 3.7. In 3.7(a), the white purple box is originally segmented into two segments. After object detection, however, the segments are merged



(a) (b)

Figure 3.7 Demonstration of Results of Object Detection: In 3.7(a) and 3.7(b), the multiple segments are merged together to define a single object region.

together to accurately represent one object (highlighted in black box). Similarly in Fig. 3.7(b), the yellow-orange amla bottle is initially divided into multiple segments. However, after object detection, the segments are merged together to represent one whole object.

Precision and Recall for each object are defined as

$$Precision = \frac{no.\ of\ positive\ images\ retrieved}{total\ no.\ of\ images\ retrieved} \tag{3.3}$$

$$Recall = \frac{no.\ of\ positive\ images\ retrieved}{total\ no.\ of\ images\ in\ dataset} \tag{3.4}$$

As shown in the Table 3.2, the mean precision for all the 35 objects is 38.1% and mean recall is 29%. The objects have standard deviation of 36.0% from mean precision and 30.1% from mean recall.

Table 3.2 Accuracy of Object Detection, measured as mean precision and recall, and standard deviation.

Type	Precision	Recall
Mean	0.381	0.290
Std. Dev	0.360	0.301

Precision and recall for each object in the dataset is plotted in Fig. 3.8. As shown in the figure, objects such as white purple box, grey box, amla candy bottle, white porcelain cup, allout and paper plate have maximum precision, while the basket and brown folder have maximum recall. Small objects like steel spoon, steel plate, steel glass (shining objects, low texture), paper cup, bubble bottle, pears cover have zero precision and recall.

True positives, false positives and false negatives for each object is also plotted as shown in Fig. 3.9. Basket and Brown Folder have highest true positives and least false negatives whereas white porcelain cup and paper plate have low false positive.

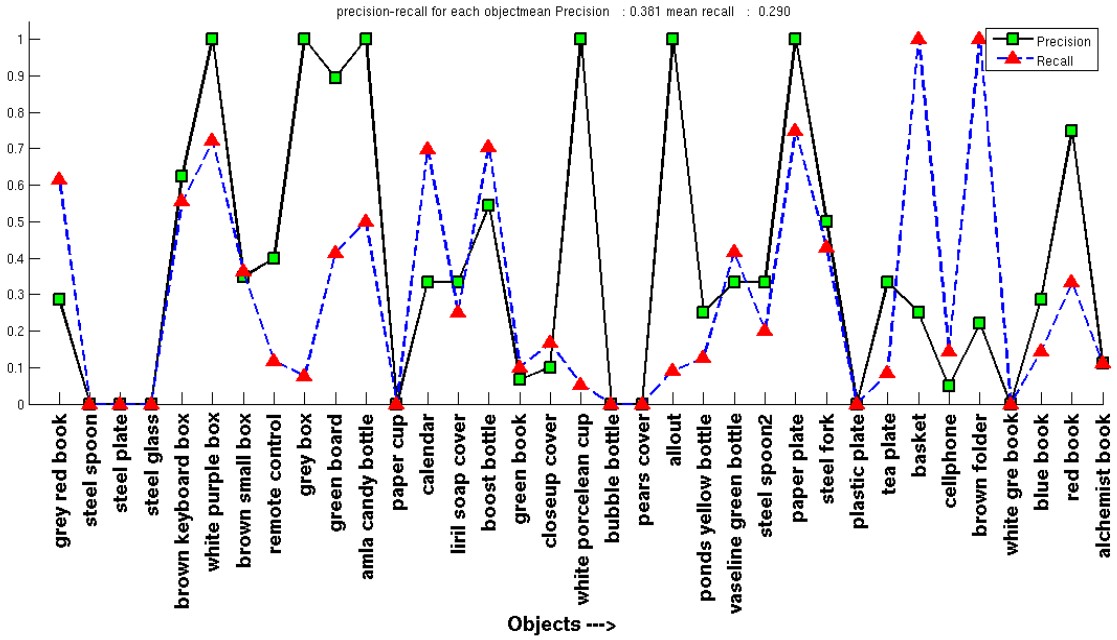


Figure 3.8 Precision Recall of object detection for each object

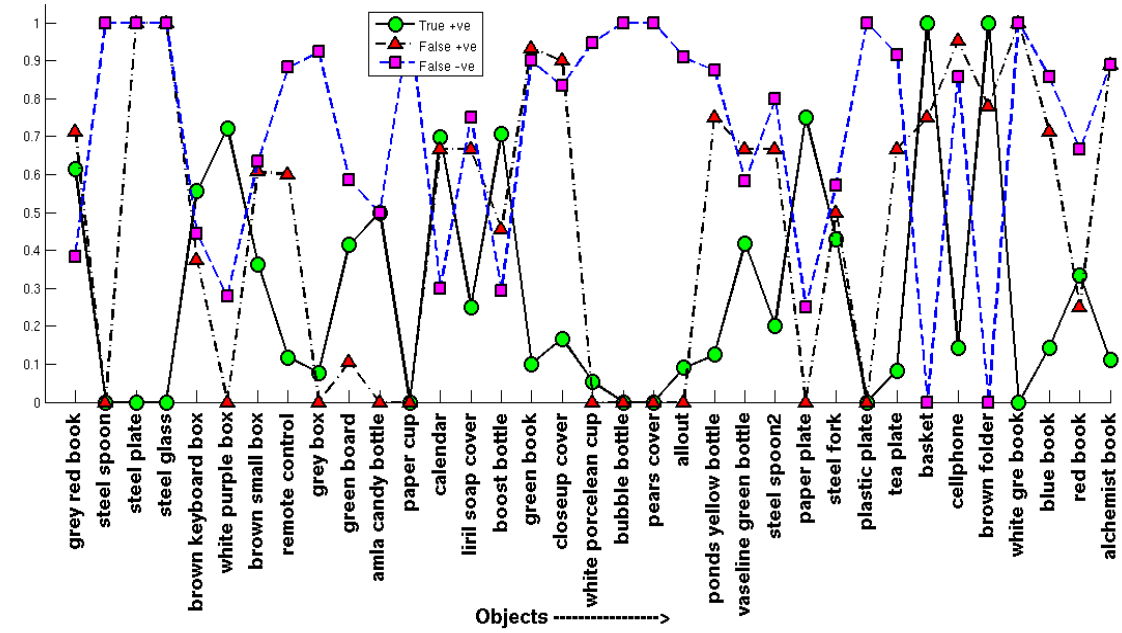


Figure 3.9 True Positives, False Positives and False negatives of object detection for each object

3.3.3 Structure Class Inference

We observe that the support inference gets affected by the inaccuracies of structure class prediction. Incorporating explicit structure class information in support inference helps avoiding infeasible support relations such as an object supporting the walls or furniture to a significant extent. The accuracy of structure class prediction is shown in Table 4.1(a). Since the images in our dataset are taken in similar environment, the accuracy is reported to be high both for groundtruth and segmented regions.

Table 3.3 Accuracy Structure class Inference over ground truth regions and segmented regions.

Type	Training Accuracy	Test Accuracy
Ground Truth Regions	100	97.02
Segmented Regions	97.79	83.88

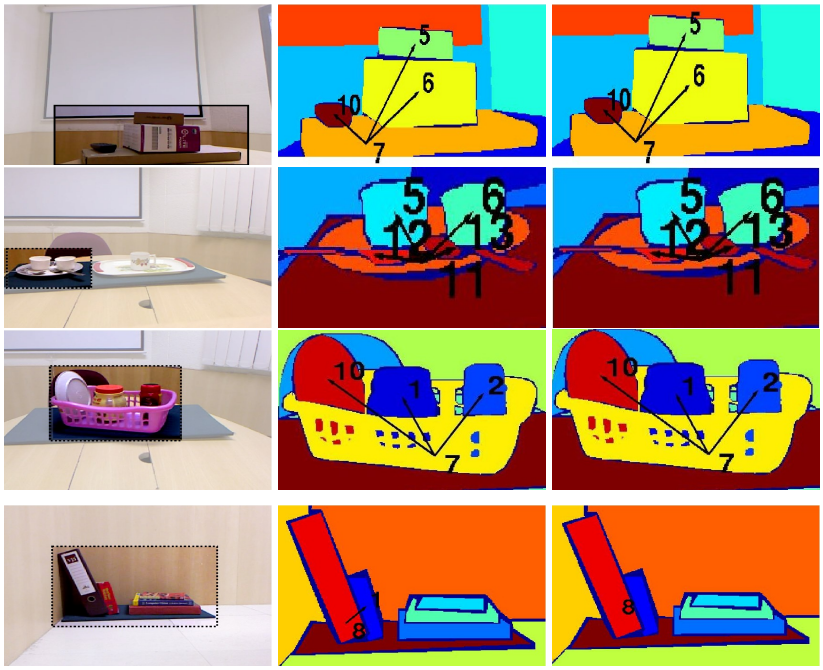
3.3.4 Multiple Object Support Inference & Support Order Prediction

Accuracy of support inference directly impacts the accuracy of support order determination. Hence the support inference accuracy is measured both on ground truth regions and on segmented regions. For each case, both "type aware"and "type agnostic" accuracies are evaluated similar to [49]. In case of type agnostic accuracy, the support type is not considered while comparing support relation with ground truth. But in case of type aware accuracy, both support relation and support type are taken into account. The accuracy of support inference using groundtruth regions and segmented regions are given in Table 4.2.

Table 3.4 Accuracy of various approaches for Support Inference. Type aware accuracy penalizes incorrect support type whereas type agnostic accuracy does not.

Region Source	Ground Truth		Segmentation	
Inference Type	Type Agnostic	Type Aware	Type Agnostic	Type Aware
Multiple object support inference	66.2	56.1	35.1	32.4
MAP Inference	65.8	48.0	32.1	30.5

The comparative results of support inference for a selected set of images from our dataset using multiple object support inference method and MAP inference method are shown in Fig. 3.10. The support relationship is shown by pointing arrows from the object of interest to objects supported by it. The support order prediction for Fig. 3.10 is given in Table 4.3. The images in row 1 show the support from below. Both multiple object and MAP inference method do well in such cases. The images in row 2 show that both the methods can successfully infer the support relation between the plate and all the other objects on it. The images in row 3 show the support by the basket to the objects contained in it. However, since they are contained inside the basket (label 7), the basket is supposed to be picked up



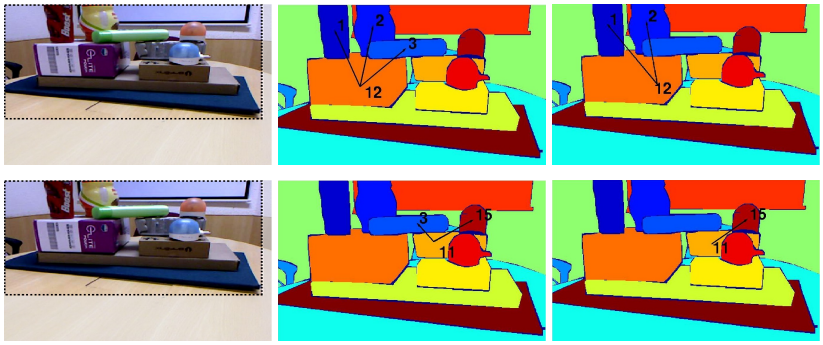
(a) Input Images (b) Multiple object Inference (c) MAP Inference

Figure 3.10 Performance of Multiple object inference vs MAP inference: The highlighted section in input images in column (a) are zoomed in columns(b) and (c) for better view. The arrows point from object of interest to objects directly and indirectly supported by it. In Row 4, MAP inference fails to provide side support while Multiple object inference successfully provides side support.

as it is. Hence the support order prediction does not generate the labels of the objects contained in the basket as given in Table 4.3. The images in row 4 show support from side. MAP inference fails to infer side support of book 1 by the folder 8, but multiple object support inference successfully infers the side support.

Fig. 3.11 demonstrates the scenario where an object is supported by multiple objects and comparison of the performance of our Multiple object inference method and MAP inference in this challenging situation. The green bottle (shown by pointing an arrow in Fig. 3.4(b)) is supported by two boxes labelled 12 and 11 simultaneously as shown in Fig. 3.11. Therefore, if our object of interest is either of the two, we must pickup the green bottle labelled 3 prior to picking them up. Our method takes such a situation into account and infers that both box 11 and box 12 support the green bottle 3. But the MAP inference method fails to do that since it discards the possibility of support by both boxes 11 and 12.

The impact of structure class inference is evident in the figures in Fig. 3.10 and 3.11. The wall, projector screen and chair are clearly not inferred as supported objects. However, sometimes, due to error in structure class prediction, some of the vertical structures and furniture are shown as supported by objects. In addition to that, in some cases, objects are predicted as furniture due to which the desirable support relation can not be achieved. Some of such results are shown in Fig. 3.12 and their corresponding



(a) Input Images (b) Multiple object Inference (c) MAP Inference

Figure 3.11 Demonstration of support by multiple objects.The highlighted section in input images in column (a) are zoomed in columns(b) and (c) for better view. The arrows point from object of interest to objects directly and indirectly supported by it. Multiple object inference provides support by multiple objects while MAP inference method does not.

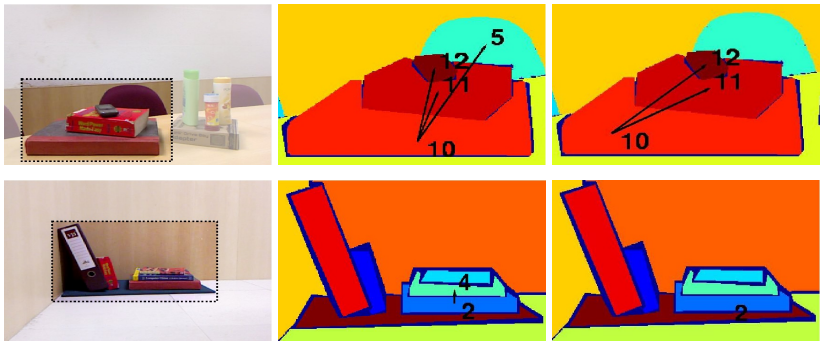
Table 3.5 Support Order using MAP inference and Multiple object support inference method: MAP inference method fails in case of side support and support by multiple objects.

Img No.	Object of interest	Support Order Multiple object method	Support Order MAP inference
3.10.1	7	5 10 6	5 10 6
3.10.2	11	6 13 12 5	6 13 12 5
3.10.3	7	-	-
3.10.4	8	1	-
3.11.1	12	3 2 1	2 1
3.11.2	11	15 3	15
3.12.1	10	5 12 11	12 11
3.12.2	2	4	-

support order are given in Table 4.3. In the image in 1st row, the chair labelled 5 is treated as an object and is shown as supported by the closest object that is the book labelled 11. Using MAP inference, these errors were eliminated. On the other hand, in row 2, the book on the top is predicted as furniture and the true support by the books below it are missed both by multiple object and MAP inference methods.

3.3.5 Hierarchical Support Inference & Support Order Prediction

The result of hierarchical support inference is shown in Fig. 3.13. It demonstrates different types of support relationship, support by multiple objects and support in hierarchy using hierarchical support inference method. The predicted support order corresponding to the images are shown in Table 3.6. The image are indexed in row major order in Table 3.6. The results in row 1, 2 and 3 depict support from below, support from side and containment respectively. The results in row 1 show support relation from below to multiple objects. In row 2, we can observe the hierarchical support relationship. Book 7



(a) Input Images (b) Multiple object Inference (c) MAP Inference

Figure 3.12 Dependency on Structure class prediction. The highlighted section in input images in column (a) are zoomed in columns(b) and (c) for better view. The arrows point from object of interest to objects directly and indirectly supported by it.

supports book 1 directly and book 1 supports books 2 and 3. Therefore, book 7 also indirectly supports books 2 and 3. The results in row 3 show containment. All the objects contained in the basket are shown as supported by the basket. But due to containment, they need not be removed in order to remove the basket, as evident in the support order shown for images in row 3.

3.3.6 Evaluation Measure

There are many feasible correct support orders for an object of interest in a clutter. Therefore, in order to validate the predicted support order, it needs to compared against more than groundtruths. In order to achieve this, two data-structures are created and stored for all the graspable objects in all the images. An 'Offset Table' stores the number of graspable objects in each image. A 'Hashtable' stores the multiple feasible support orders for each graspable object. The index is the object index. The image is traversed along the hash table using the numbers in offset table.

Table 3.6 Support order using hierarchical support inference method. Support Order Prediction for different cases shown in Fig. 3.13.

Img No.	Object of interest	Support Order hierarchical support inference
3.13.1	16	15 14 13 5 4 3 2 1
3.13.2	7	3 2 1 6
3.13.3	1	3 2
3.13.4	7	3 2 1
3.13.5	11	-
3.13.6	7	-

(a) Input Images (b) Results (c) Input Images (d) Results

Figure 3.13 Results of Hierarchical Support Inference: The highlighted section in input images in column (a) & (c) are zoomed in columns(b) & (d) respectively for better view. The arrows point from target object to objects directly and indirectly supported by it.

The predicted sequence is compared with corresponding objects ground truth sequences. If at least one sequence matches completely, a score of 1 is given, otherwise a score of 0 is given. In our case, we achieved an accuracy of 66.5893 % for support order prediction in "Indoor dataset for clutter".

3.3.7 Discussions & Analysis of Results

We observe that, support inference fails in a few situations. Support from side is not correctly inferred in cases when baseline of supporting object is not visible or when supporting object is also unstable. Often in frontal view, the entire surface area of the supporting object is not visible. In these cases, support to objects lying on top of it are not inferred, especially if they are partially occluded and contact to the supporting surface is not visible.

Due to noise in depth values, sometimes false contact boundary is created between two isolated objects and false support is inferred. Accuracy of support inference using segmented regions is lower than that using ground truth regions. In many situations, the segmented regions do not uniquely represent an object. An object region may comprise of more than one segments. A segment may also represent parts of more than one object region. This imposes limitation on the practicality of our approach. With improvement in segmentation methods, the performance of support inference and support order prediction can be improved and also can be practically more feasible. Recently, many interactive segmentation methods have been developed [10, 16, 29, 30, 54] to support robotic manipulation tasks where user input is taken as initial input for segmentation. Incorporating user input using such methods can also help in achieving more accurate segmented regions.

We have verified different scenarios of support in our experiment such as support by multiple objects, support in multiple hierarchy and containment. We plan to learn support relationship and support order

in more complex and varied settings with objects of more diversity. Exploring combinations of the three types of support such as the situations when an object contained inside another also supports other objects from below or side, will help in learning more complex support relationships. Subcategories of containment like complete containment and partial containment can also be considered. We have experimented on images captured from frontal view. By incorporating images from an elevated view and top view will increase the diversity in support inference.

3.4 Summary

In this chapter, we inferred support relationship among objects present in cluttered environment in terms of "support from below", "support from side" and "containment". This support relationship is utilized to predict the support order of different objects with respect to our object of interest, i.e., the order in which the surrounding objects need to be removed to be able to manipulate our object of interest. We represented the support relationship in a tree data structure and performed reverse level order traversal to predict support order of the objects. We created a dataset consisting of different objects used in household and office environment and performed our experimentation on the same. Our work extends the scope for different applications such as grasping, manipulation and picking from bin towards cluttered environments consisting of objects of generic shape and size that overlap on one another.

Chapter 4

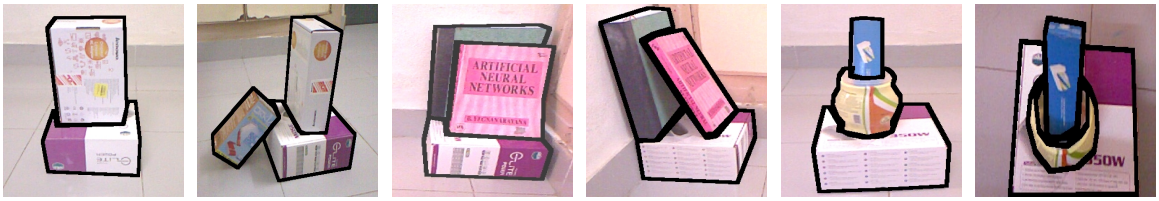
Multiple View Support Order Prediction

4.1 Introduction

In the previous chapter, we have attempted to address the issues mentioned in cluttered scenarios when objects lie with physical contact and spatial support. Three types of support relationships, i.e., support "from below", "from side" and "containment"are defined and support relationships are inferred using different approaches. Finally, support order for an object of interest is derived. Support order is the order or sequence in which surrounding objects should be removed to access the object of interest without causing any damage to the surroundings. Recently, Jia *et al.* [28] have proposed a 3D-Based Reasoning method. Their method is based on cluttered objects with blocks, spatial support and stability. They provide an improved segmentation of the indoor cluttered scene using these aspects. However, these works approach the scene from single view.

There are many possible situations in clutter that can not be addressed using single view. It is not possible to detect hidden objects using single view. Often, spatial relation of partially occluded objects is inaccurate. Containment cannot always be correctly inferred using a frontal view (Refer Fig. 4.1). We aim at finding a solution to such problems by exploring the scene from different views. This will help in discovering different objects and different support relationships, which otherwise is not possible using single views. Therefore, in our work, we have proposed an approach for support order prediction using multiple views to explore support relationships from different perspectives. In addition to this, we collect an RGBD dataset of 7 scenes. Images are captured for each scene from multiple views, number of views varying from 4 to 18. Total 67 images along with their depthmaps and point clouds are captured. The dataset is also available for public usage. To our knowledge, this is first RGBD dataset with multiple views.

This chapter is organized as follows. In Section 4.2, we provide an overview of multiple view support order prediction. In Section 4.3, we discuss how regions across different views are mapped to obtain a global reference across views. Then, in Section 4.4, we discuss approach for support inference for N views in consideration. The inferred support relationship among objects in N views is used for support



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

Figure 4.1 Scenarios in which support relations needs multiple views: (a) and (b): Hidden object discovered in another view, (c) and (d): Side Support cannot be determined from front view, (e) and (f): Containment cannot be detected from frontal view.

order prediction as discussed in Section 4.5. The experiments are discussed in Section 4.6. Finally, we conclude the chapter in Section 4.7.

4.2 Overview of the Framework

The overall framework of our work is explained through the block diagram shown in Fig. 4.2. The images are first over-segmented into superpixels using Arbelaez's method [8], then segmented using hierarchical segmentation method of Hoiem *et al.* [27]. Both 2D and 3D features of images are used for segmentation. The segmented regions are provided as input to the object detection and support inference modules. In the object detection module, SIFT feature matching [25, 37, 55] between the template image of the object of interest and the input images is performed. The outliers are discarded by applying RANSAC [21, 32]. The segmented regions corresponding to the matched points of the input image(s) are merged into one region and chosen as the region corresponding to object of interest O, i.e., the object to be grasped. This approach ensures that the entire object region is chosen for grasping.

After segmentation of each region, region mapping is performed beginning with first view till last view. For each view, it is matched with its previous view and next view. Region mapping is found for every pair of images corresponding to adjacent views. In scenes where images are captured by moving Kinect in circular fashion around the clutter, regions are mapped from last view with first view and cross checking is done to correct any inconsistencies. After region mapping is done with every pair of images, a global region mapping table is created to capture the region mapping across all the views.

Given the image regions and various geometric features, the support inference module infers the supporting regions and type of support for each region in the image. Support relationship is inferred by applying rule-based inference method proposed in [41]. After support inference is performed for each scene, the support matrices are combined to form a global support matrix with the help of the global region mapping table.

Given the object of interest and the inferred support relationship, a Tree of Support is built with the object of interest as the root. The tree is traversed for support order prediction using reverse level order traversal. During the construction and traversal of the tree, different scenarios are considered to avoid damage and are taken care of.

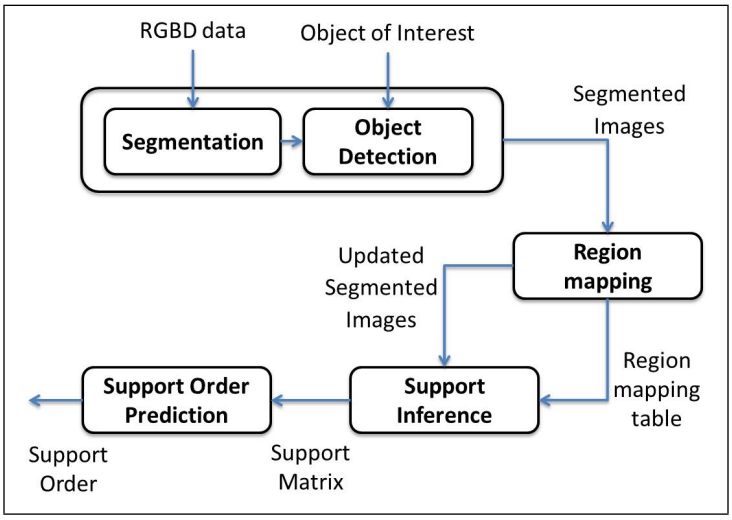


Figure 4.2 Block diagrammatic representation of our framework for support order prediction using multiple views.

Region mapping is discussed in detail in Section 4.3.2. The details of the support inference approach can be found in Section 4.4. A detailed discussion on the approach for support order prediction and how different specific scenarios are handled is given in Section 4.4 while the analysis of the results on various images from our RGBD dataset is given in Section 4.6.

4.3 Region Mapping across Multiple Views

Prior to support inference across multiple views, it is essential to find correspondence among regions across different views. The scene is captured from different views by moving the Kinect around the clutter. Starting from the first view, each region of a view is mapped with regions in two adjacent views. Finally, the last view is mapped with the first view along with the last but one view. The mapping is captured in a region mapping table, which is updated and rectified as mapping proceeds from one view to another. The region mapping table T_R is initialized with zeros, with each row corresponding to each view and each column corresponding to each region. The first row of T_R is initialized to the regions in 1^{st} view and number of regions are initiated to number of regions in 1^{st} view.

4.3.1 Region Mapping between 2 Views

Regions of every ith view are mapped to its previous and next view. The images are captured in a sequence. The regions in the last view (nth view) are mapped with $(n-1)th$ view and $1st$ view, if the scene is captured in a circular fashion around the clutter.

For the image pair (A, B), first, for each region in image A, corresponding region is found in image B by point correspondence. SIFT feature matching [25, 37, 55] between each region of image A is

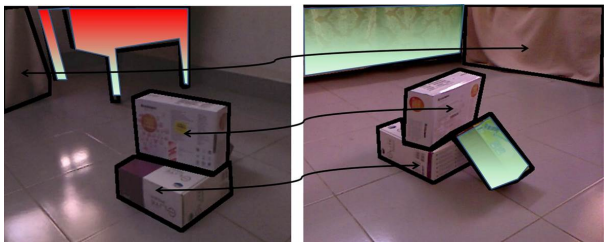


Figure 4.3 Illustration of Region mapping across 2 views. Mutual matches are shown in black arrows. New objects are marked as green regions and omitted objects are marked in red region.

performed with image B. The outliers are discarded by applying RANSAC. The matches are further filtered by discarding the ones with number of matches and distance ration below empirically chosen thresholds. The same process is repeated for the pair (B, A).

After matches are found in both directions, i.e., from A to B ($match_{A\rightarrow B}$), and from B to A ($match_{B\rightarrow A}$), the bidirectional match is captured. At first, the mutual matches are chosen as the robust matches and updated as $match_{AB}$. Then mismatches are resolved by addressing different cases. If a region i in A does not have any match in B in $match_{A\rightarrow B}$, but has a match in $match_{B\rightarrow A}$, then the latter mapping is trusted and appended to $match_{AB}$. Otherwise, $match_{AB}$ is updated with the pair $(i, 0)$, where 0 represents that no match is found in the other view. On the other hand, if i in A has a match j in B in $match_{A\rightarrow B}$, but j matches to another region k in A in $match_{B\rightarrow A}$, it is no more a case of mutual mapping. This conflict is resolved by first checking if two regions i and k in A are neighbouring regions or not. In that case, they are merged.match with maximum number of matching points is selected and updated to $match_{AB}$. The same process is repeated for mismatches in $match_{B\rightarrow A}$. Every region in A for which $match_{A\rightarrow B}$ does not find a region in B is the region that gets omitted in the next view B. Every region in B for which $match_{B\rightarrow A}$ does not produce any match in A, is a new region discovered in B. An illustration of region mapping across two views is shown in Fig. 4.3.

4.3.2 Region Mapping between N Views

After $match_{AB}$ is found out for each pair of adjacent views, region mapping table T_R is updated for view B. For new objects discovered in B, new columns are added to T_R. If A is the nth view that corresponds to the end of a cycle, then cross-check is done with corresponding regions in $1st$ view in T_R and those in $match_{AB}$.

4.4 Support Inference

4.4.1 Support Inference for Single View

For each view of the scene, support inference is performed using the multiple object support inference approach Section 2. In this method, the features proposed in Section 2.5 are used for support

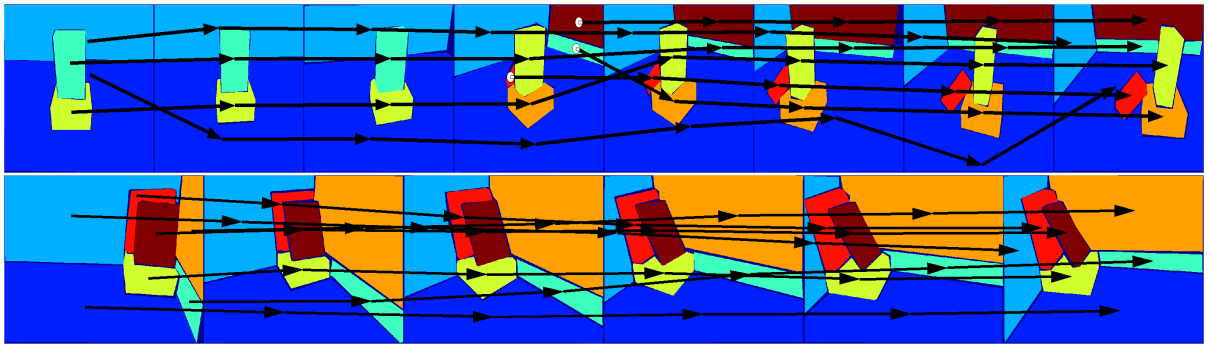


Figure 4.4 Result of Regionmapping: Region correspondence across views is shown using arrow marks. New objects discovered in a view are marked by white circle.

inference. Three types of support relationships are inferred: support "from below", "from side" and "containment".

A structure class classifier is trained to classify the structure classes of different regions using neural networks. If the classifier predicts any region as "floor", then vertical structures and furniture are decided to be supported directly by the floor. Otherwise it is assumed that floor is not visible in the scene. In some scenarios, a vertical structure such as window or a furniture such as projector screen is surrounded by another structure such as wall. Such cases are identified where a vertical structure or furniture is completely surrounded by another region. Here, the it is inferred that the region is supported by its surrounding region from side. Identifying vertical structures like walls and windows, and furniture like tables, chairs, cupboards and sofas plays a significant role to avoid infeasible support inference such as a small object supporting a wall or a table. For a prop or a graspable object, different types of support are inferred by considering its surrounding region. Objects lower to the current object whose centroids are closer to the current object are selected (Proximity, f_p) as potential candidates for providing "support from below". In case of conflict, the ones with higher boundary ratio (Boundary Ratio, f_{br}) are chosen as regions providing "support from below". If a significant portion of 3D convex hull of the current object belongs to the 3D convex hull of the supporting region (Containment, f_{cnt}), the support is termed as "containment". All regions in contact with the current object (Depth Boundary, f_{depth}) other than the regions below are considered as "support from side", if they are labelled as stable regions (Relative Stability, f_{stab}). The algorithm for the support inference method is given in Procedure 2.

4.4.2 Support Inference in N Views

Using the region mapping table T_R, regions in support matrices of individual matrices are updated to the global notation of regions. After that, the support matrices are merged together to form a global support matrix. Redundant support relations are removed while keeping note of the frequency of occurrence f_o of each support relation. The frequency of occurrence of each support relation is used to resolve potential conflicts in support relations.

We have identified different kinds of conflicting scenarios. The first case, when a region R is supported by another region S but with two different types of support. This is resolved using frequency of occurrence f_o of support relations. The relation with maximum frequency is retained while discarding the other one(s). In case of equal frequencies, preference is given to the support relations in the following order: support from side, support from below, containment. Containment is given least priority since it is not harmful to put extra effort to remove a contained object. Support from side is given higher priority since support from side is usually associated with instability. In the second scenario, the region R is supported by multiple other regions through same support type. To resolve this, f_o is used to find the relation with maximum frequency and discard the other. However, care is taken if the relations have equal (nearly equal) frequency. In this case, it turns out to be a case of partial support. Therefore, none of the relations are discarded. In the fourth and last scenario, loops are detected where region R is supported by S, and S also supports R in turn. This is an infeasible condition and is resolved using f_o.

4.5 Support Order Prediction

Support Order is the sequence in which an objects surrounding an object of interest should be removed with minimal damage to the environment. Support Order Prediction is performed on the global Support Matrix using the approach of reverse level order traversal discussed in Section 3.2. The pairwise support relations are translated to a Tree of Support. The object of interest denotes the root. Parent nodes denote the supporting objects and child nodes denote the supported objects. Edges are directed from parent nodes to the child nodes. When a child node receives edges from parent nodes from different layers of the tree, this implies that the object at child node is supported by multiple objects at different hierarchies. The edges corresponding to upper layers are pruned to avoid any damage. Contained objects are skipped during tree traversal, since they need not be removed in order to access the supporting objects. The global region index of the object of interest O is found using region mapping table T_R. Support Order Prediction results in the support order in terms of the global notation. Later, it is mapped back to respective views.

4.6 Experiments & Results

In this section, we validate the inferred support relationship and support order on various images from our "Indoor multi-view dataset".

4.6.1 Region Mapping

The results of region mapping across N views for some of our data are shown in Fig. 4.4. It can be seen in the figure that the regions are correctly matched from each view to the subsequent view through black arrows. New objects appearing for the first time in the scene are marked with white circle. In

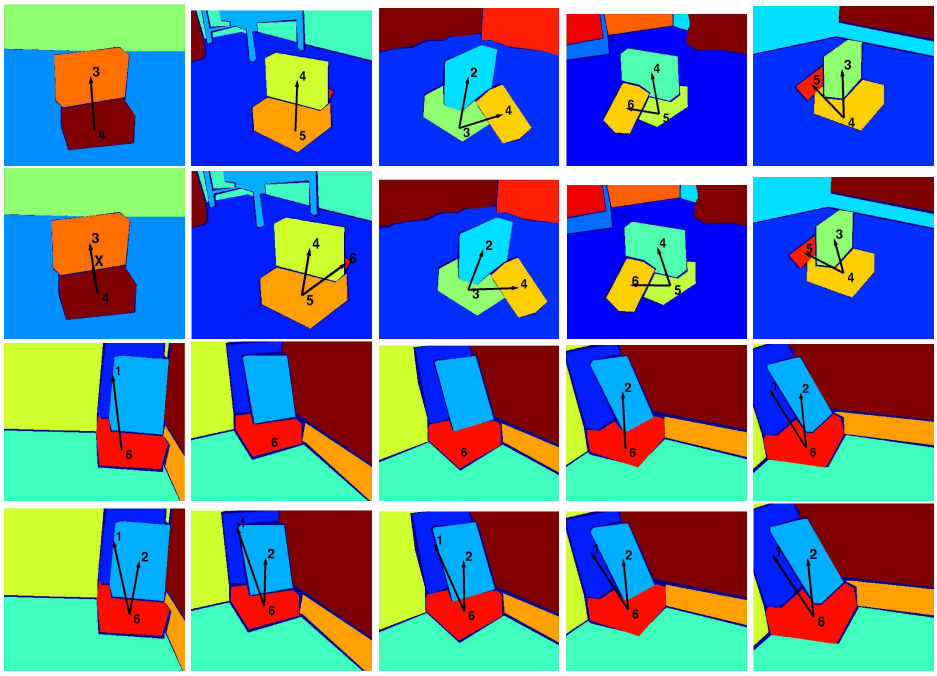


Figure 4.5 Result of Support Order Prediction: Row1, Row3: Support order prediction using single view. Row2, Row4: Support order prediction using multiple views. In Row2, presence of hidden objects is shown using a cross mark. In Row4, support order that was not accurately inferred using single view is recovered.

the first row, a sequence of images are taken around a set of boxes. A box is discovered in (marked in red) in view 4 along with cupboard and another wall. In the second row, region mapping is shown for a sequence of images captured in an attempt to capture support from side more accurately.

A better resolution of the results of different images in our dataset is available for view at our website.[1]

4.6.2 Support Inference

A 5-stage hierarchical segmentation approach proposed by Arbelaez *et al.* [8] was used for segmenting the images. RGB and depth features used in [49] are used for segmentation. Segmentation accuracy is measured as average overlap of segmented regions over groundtruth regions as defined in [27]. The unweighted average overlap score and the score weighted by pixel area are given in Table 4.1(b). Structure class of different image regions is inferred using a logistic regression classifier [49]. It predicts four types of structure classes: floor, vertical structures, furniture and graspable objects. The accuracy of structure class prediction is shown in Table 4.1(a). Since the images are taken in similar environment, the accuracy is reported to be high. The accuracy of support inference by multiple object support infer-

[1]Results with better resolution available at
`http://researchweb.iiit.ac.in/~swagatika.panda/IROS14_Results.html`

ence (refer Section 3.1.2) is shown in Table 4.2. Type agnostic accuracy is calculated while ignoring the support type. On the other hand, type aware accuracy is calculated considering support type such that a support relation with wrong support type is penalized.

Table 4.1 Accuracy of Structure class Inference & Segmentation

(a) Accuracy Structure class Inference

Type	Training Accuracy	Test Accuracy
Ground Truth Regions	100	94.79
Segmented Regions	97.79	83.88

(b) Accuracy of Hierarchical Segmentation

Type	Training Accuracy	Test Accuracy
Weighted	87.1	75.4
Unweighted	74.3	60.4

Table 4.2 Accuracy of Support Inference

Region Source	Ground Truth		Segmentation	
Inference Type	Type Agnostic	Type Aware	Type Agnostic	Type Aware
Multiple Object Support Inference	85.0	82.0	65.0	59.4

4.6.3 Support Order Prediction

The results of support order prediction are shown in Fig. 4.5. As shown in the figure, missing support relations and hidden objects are identified by exploiting multiple views. The figures in first row and second row show the support order prediction in single view and multiple views, respectively. As shown in the first column, using multi-view support order prediction, presence of a hidden object is known (shown by cross mark). Similarly, the missing support relation in second column is restored. Similarly, inaccurate support relations are rectified using multiple views as shown in the third and fourth rows in the Fig. 4.5. As shown in third row, side support is not inferred correctly using frontal view. Similarly, in some images, support from below is not inferred correctly since the surface of the supporting object (object 6) is not clearly visible. However, using multiple views, the support relationships and support order are rectified properly. The support orders of the respective images are shown in Table 4.3.

Table 4.3 Support Order using Single View vs Multiple Views

Img No.	Object of Interest	Support Order Single View	Support Order Multiple Views
1	4	3	x 3
2	5	4	6 4
3	3	4 2	4 2
4	5	6 4	6 4
5	4	5 3	5 3
6	6	1	2 1
7	6	-	2 1
8	6	-	2 1
9	6	2	2 1
10	6	2 1	2 1

4.7 Summary

In this work, we discuss support order prediction for manipulation in clutter using multiple views. The limitations of support inference in single view such as missing/wrong support relations, partial(complete) occlusion, hidden contained objects are addressed by exploring the scene from multiple views. Our work extends the previous work on support order prediction proposed in the previous chapter. In future, we are planning to extend our work to optimize the best view(s) for support order prediction.

Chapter 5

Conclusions & Future Work

5.1 Summary

Semantic interaction among objects has significant applications in grasping, grasp-manipulation, scene understanding. Understanding semantic interaction among various entities in a scene helps in extending such tasks to more and more complex settings. The more complex environment the objects are in, the more complicated semantic interaction needs to be identified and inferred for successful use in desired application.

In our thesis, we have considered cluttered objects in indoor. Such clutters have following properties: (1) objects with both direct and indirect physical contact, i.e., one object indirectly supporting one or more objects, (2) noisy background with non-graspable entities present in it, (3) objects with varied shapes and sizes, (4) objects placed in non-planar fashion. We study the semantic interaction among each pair of objects by inferring the support relationship among the pair. "Support relation" among a pair of object (A, B) is defined as how an object A physically supports B such that removing A will make B unstable or cause it to fall down. We identify three kinds of support relationships: "support from below", "support from side" and "containment". These pairwise support relationship are used to generate "Support Order" for a given object of interest. we define "Support Order" as the order in which the surrounding objects need to be removed before accessing the object of interest, while causing minimal damage to the environment. Support Order extends understanding of semantic interaction from pairwise support relationships to hierarchical support relationships.

At first, we compare an adaptation of the state-of-the-art method MAP inference by Silberman *et al.* [49] with our proposed methods "Multiple Object Support Inference" and "Hierarchical Support Inference". The former method tries to infer support relationships among all possible pairs of objects while the latter is object-centric. It establishes support inference w.r.t the object of interest. We propose support order prediction method while addressing different conditions which may cause damage to the environment while removing the objects.

In the second part of thesis, we extend the support order prediction into multiple views. The aim of this extension is to overcome limitations of single view such as partial and complete occlusion,

incorrect inference of support relationship and missing support relations. The challenges of multiple views are: (1) correct mapping between the objects seen from different views. (2) combining the support inference from all the views while handling redundancies and removing errors. We attempt to address the challenges by finding a region-mapping table that establishes a mapping between all views. This region-mapping table is used to combine support inference from all the views into one global support matrix. This global support matrix is used to predict the support order for the scene.

For our experimentations, we have created two RGBD datasets: "Indoor dataset for clutter" and "Indoor multiview dataset." Both datasets contain images, point clouds and depthmaps for each scene. In addition, the same are captured for individual objects used in the dataset for object detection. The former contains 50 images from 50 different cluttered scenes using 35 objects. The latter contains 67 images from 7 different cluttered scenes using 9 objects. Both datasets are made available for public use in the area of scene understanding, RGBD data, segmentation etc.

5.2 Directions for Future Work

Our work can be extended in the following directions, evolving into more accurate and efficient system.

- **Segmentation:** Accuracy of segmentation method affects the performance of the entire system. Therefore, an efficient segmentation algorithm for RGBD data can be beneficial to the system as a whole.
- **Object Detection:** More sophisticated methods for object detection can be used [22, 47, 56]. Interactive object detection methods in clutter can be also explored.
- **Real-time:** Currently, our system works off-line. It can be optimized to work in real-time.
- **Next Best View Point:** In multiple view support order prediction, we capture the clutter from multiple view points and combine the inferred support relations to find the global support relation. This can be optimized by finding the next best view point from current view. By finding the next best view point, the need for capturing from all the view points can be avoided.
- **Multi-view object segmentation:** Multi-view object segmentation is simultaneous segmentation of a group of images that share common objects. Since in multiple view support order prediction, one single scene is captured multiple view points, most of the objects in the images are common. This can be exploited to co-segment the images to find improved segmentation for each object. The problem, of course, poses some inherent challenges such as limited number of views and far apart view points. There are works in this direction using RGB images by Djelouah *et al.* [17] and Campbell *et al.* [11]. However, the assumption is that the scene consists of a single foreground object located at significant distance from its background. This is not the case in the scenario of clutter using RGBD data. In this scenario, multiple objects consist of the foreground, making the problem even more challenging.

Related Publications

- **Learning Support Order for Manipulation in Clutter**, Swagatika Panda, A.H. Abdul Hafez and C.V. Jawahar,
  IROS 2013, IEEE International Conference on Intelligent Robots and Systems, Tokyo, Japan, 3-7 November, 2013.

- **Learning Semantic Interaction among Graspable Objects**, Swagatika Panda, A.H. Abdul Hafez and C.V. Jawahar,
  PReMI 2013, International Conference on Pattern Recognition and Machine Intelligence, Kolkata, India, 2-5 December, 2013.

Bibliography

[1] Justin. Available from:<http://www.dlr.de/rm/en/desktopdefault.aspx/tabid-5471/>.

[2] Microsoft kinect. Available from: <http://www.xbox.com/en-us/kinect>.

[3] Nasa robonaut. Available from:<http://robonaut.jsc.nasa.gov/>.

[4] Stair: Stanford artificial intelligence robot. Available from:<http://stair.stanford.edu/index.php>.

[5] Wikipedia: Kinect. Available from: <http://en.wikipedia.org/wiki/Kinect>.

[6] J. Aleotti and S. Caselli. Efficient planning of disassembly sequences in physics-based animation. IROS, 2009.

[7] A. Anand, H. S. Koppula, T. Joachims, and A. Saxena. Contextually guided semantic labeling and search for three-dimensional point clouds. *The International Journal of Robotics Research*, 2012.

[8] P. Arbelaez, M. Maire, C. Fowlkes, and J. Malik. Contour detection and hierarchical image segmentation. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 33(5), 2011.

[9] D. Berenson and S. Srinivasa. Grasp synthesis in cluttered environments for dexterous hands. In *IEEE-RAS International Conference on Humanoid Robots (Humanoids08)*, December 2008.

[10] M. Bjorkman and D. Kragic. Active 3d scene segmentation and detection of unknown objects. In *ICRA*. IEEE, 2010.

[11] N. Campbell, G. Vogiatzis, C. Hernndez, and R. Cipolla. Automatic 3d object segmentation in multiple views using volumetric graph-cuts. *Image and Vision Computing*, 28(1).

[12] L. Y. Chang, S. Srinivasa, and N. Pollard. Planning pre-grasp manipulation for transport tasks. In *ICRA*, 2010.

[13] A. Collet Romea, S. Srinivasa, and M. Hebert. Structure discovery in multi-modal data : a region-based approach. In *ICRA*, 2011.

[14] L. Cruz, D. Lucio, and L. Velho. Kinect and rgbd images: Challenges and applications. In *Graphics, Patterns and Images Tutorials (SIBGRAPI-T), 2012 25th SIBGRAPI Conference on*, Aug 2012.

[15] J. B. D. Katz, M. Kazemi and A. Stentz. Clearing a pile of unknown objects using interactive perception. ICRA, 2013.

[16] A. Delong, L. Gorelick, F. R. Schmidt, O. Veksler, and Y. Boykov. Interactive segmentation with super-labels. In *EMMCVPR*, 2011.

[17] A. Djelouah, J.-S. bastien Franco, and E. Boyer. Multi-view object segmentation in space and time. In *Proceedings of the International Conference on Computer Vision*, 2013.

[18] M. Dogar, K. Hsiao, M. Ciocarlie, and S. Srinivasa. Physics-based grasp planning through clutter. In *RSS VIII*, July 2012.

[19] M. Dogar and S. Srinivasa. A framework for push-grasping in clutter. In *RSS VII*, 2011.

[20] S. Fichtl, J. Alexander, D. Kraft, J. Jrgensen, N. Krger, and F. Guerin. Learning object relationships which determine the outcome of actions. *Paladyn*, 2012.

[21] M. A. Fischler and R. C. Bolles. Random sample consensus: a paradigm for model fitting with applications to image analysis and automated cartography. *Communications of the ACM*, 1981.

[22] D. Fouhey, A. Collet, M. Hebert, and S. Srinivasa. Object recognition robust to imperfect depth data. In *Computer Vision ECCV 2012. Workshops and Demonstrations*, volume 7584 of *Lecture Notes in Computer Science*. Springer Berlin Heidelberg, 2012.

[23] M. Gupta, T. Ruhr, M. Beetz, and G. S. Sukhatme. Interactive environment exploration in clutter. In *Intelligent Robots and Systems (IROS), 2013 IEEE/RSJ International Conference on*. IEEE, 2013.

[24] J. Han, L. Shao, D. Xu, and J. Shotton. Enhanced computer vision with microsoft kinect sensor: A review. *Cybernetics, IEEE Transactions on*, 2013.

[25] C. Harris and M. Stephens. A combined corner and edge detector. In *Alvey vision conference*. Manchester, UK, 1988.

[26] K. Hauser. Cutting through the clutter: Identifying minimally disturbed subsets. In *RSS Workshop on Robots in Clutter: Manipulation, Perception and Navigation in Human Environments*, 2012.

[27] D. Hoiem, A. A. Efros, and M. Hebert. Recovering occlusion boundaries from an image. *International Journal of Computer Vision*, 91(3), 2011.

[28] Z. Jia, A. Gallagher, A. Saxena, and T. Chen. 3d-based reasoning with blocks, support, and stability. CVPR, 2013.

[29] D. Katz, M. Kazemi, J. A. Bagnell, and A. Stentz. Clearing a pile of unknown objects using interactive perception. Technical report, DTIC Document, 2012.

[30] D. Katz, M. Kazemi, J. A. D. Bagnell, and A. T. Stentz. Semi-autonomous manipulation of natural objects. Technical report, Robotics Institute, 2012.

[31] H. S. Koppula, A. Anand, T. Joachims, and A. Saxena. Semantic labeling of 3d point clouds for indoor scenes. In *NIPS*, 2011.

[32] P. D. Kovesi. MATLAB and Octave functions for computer vision and image processing. Available from: <http://www.csse.uwa.edu.au/∼pk/research/matlabfns/>.

[33] D. F. L. Chang, J. Smith. Interactive singulation of objects from a pile. ICRA, 2012.

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

[35] S. Lee, J. Kim, M. Lee, K. Yoo, L. G. Barajas, and R. Menassa. 3d visual perception system for bin picking in automotive sub-assembly automation. In *Automation Science and Engineering (CASE), 2012 IEEE International Conference on*, 2012.

[36] A. Levin, D. Lischinski, and Y. Weiss. Colorization using optimization. *ACM Transactions on Graphics (TOG)*, 23(3), 2004.

[37] D. G. Lowe. Distinctive image features from scale-invariant keypoints. *International journal of computer vision*, 60(2), 2004.

[38] A. K. Mishra, A. Shrivastava, and Y. Aloimonos. Segmenting "simple" objects using RGB-D. In *ICRA*, 2012.

[39] J.-K. Oh, S. Lee, and C.-H. Lee. Stereo vision based automation for a bin-picking solution. *International Journal of Control, Automation, and Systems*, 2012.

[40] O. Ornan and A. Degani. Toward autonomous disassembling of randomly piled objects with minimal perturbation. IROS, 2013.

[41] S. Panda, A. H. Abdul Hafez, and C. V. Jawahar. Learning support order for manipulation in clutter. In *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 2013.

[42] A. B. R. Mojtahedzadeh and A. Lilienthal. Automatic relational scene representation for safe robotic manipulation tasks. IROS, 2013.

[43] A. Ramisa, D. Aldavert, S. Vasudevan, R. Toledo, and R. Lopez de Mantaras. Evaluation of three vision based object perception methods for a mobile robot. *Journal of Intelligent & Robotic Systems*, 2011.

[44] B. Rasolzadeh, M. Björkman, K. Huebner, and D. Kragic. An active vision system for detecting, fixating and manipulating objects in the real world. *The International Journal of Robotics Research*, 29(2-3), 2010.

[45] B. Rosman and S. Ramamoorthy. Learning spatial relationships between objects. *The International Journal of Robotics Research*, 30(11), 2011.

[46] B. C. Russell, A. Torralba, K. P. Murphy, and W. T. Freeman. Labelme: a database and web-based tool for image annotation. *International journal of computer vision*, 77(1-3), 2008.

[47] A. Shrivastava and A. Gupta. Building part-based object detectors via 3d geometry. In *Computer Vision (ICCV), 2013 IEEE International Conference on*, 2013.

[48] N. Silberman and R. Fergus. Indoor scene segmentation using a structured light sensor. In *Proceedings of the International Conference on Computer Vision - Workshop on 3D Representation and Recognition*, 2011.

[49] N. Silberman, D. Hoiem, P. Kohli, and R. Fergus. Indoor segmentation and support inference from rgbd images. In *ECCV 2012*. Springer, 2012.

[50] K. Sjoo and P. Jensfelt. Learning spatial relations from functional simulation. In *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), 2011*. IEEE, 2011.

[51] K. Sj, A. Aydemir, and P. Jensfelt. Topological spatial relations for active visual search. *RAS*, 2012.

[52] F. Spenrath, A. Spiller, and A. Verl. Gripping point determination and collision prevention in a bin- picking application. *Robotics; Proceedings of ROBOTIK 2012; 7th German Conference on*, 2012.

[53] S. Srinivasa, D. Ferguson, C. Helfrich, D. Berenson, A. Collet, R. Diankov, G. Gallagher, G. Hollinger, J. Kuffner, and M. VandeWeghe. Herb: A home exploring robotic butler. *Autonomous Robots*, 2009.

[54] K. Varadarajan, E. Potapova, and M. Vincze. Attention driven grasping for clearing a heap of objects. In *Intelligent Robots and Systems (IROS), 2012 IEEE/RSJ International Conference on*, 2012.

[55] A. Vedaldi and B. Fulkerson. VLFeat: An open and portable library of computer vision algorithms. `http://www.vlfeat.org/`, 2008.

[56] E. S. Ye and J. Malik. Object detection in rgbd indoor scenes. Technical report, Electrical Engineering and Computer Sciences, University of California at Berkeley, 2012.

[57] J. Ye and K. A. Hua. Exploiting depth camera for 3d spatial relationship interpretation. In *Proceedings of the 4th ACM Multimedia Systems Conference*, MMSys '13, 2013.