Learning Support Order for Manipulation in Clutter

Swagatika Panda[1], A.H. Abdul Hafez[1,2] and C.V. Jawahar[1]

***Abstract*— Understanding positional semantics of the environment plays an important role in manipulating an object in clutter. 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 this paper, 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. We have created an RGBD dataset that consists of various objects used in day-to-day life present in clutter. We explore many different settings involving different kind of object-object interaction. We successfully learn support relationships and predict support order in these settings.**

I. INTRODUCTION

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 [1]. Most of the robotic manipulation tasks that involve clutter remain carefully restricted to objects in physical isolation and mostly lying on a planar surface [2], [3]. Learning the interaction among different objects in an environment can be of great benefit for robotic applications such as navigation [4], [5], grasping [3], [6] and object manipulation [2], [5]. 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?".

In this work, we propose a framework in which the support relationship among different entities in a scene is inferred in terms of "support from below", "support from side", or "containment"(Fig. 1). Then a hierarchical tree of support is built and traversed to derive the sequence of objects or "support order" for our object of interest. Special situations are identified and addressed during tree traversal so that minimal damage occurs when objects are removed. 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.

(a) (b) (c)

Fig. 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.

II. RELATED WORK AND APPLICATIONS

Due to complimentary properties of RGB and depth features and due to availability of low cost RGBD sensors like Kinect, RGBD is being increasingly used in many scene understanding [7], [8] and object manipulation tasks [3], [5]. Dogar and Srinivasa [5] and Dogar *et al.* [3] work on grasping and grasp-panning in clutter. However, they assume that objects are spatially isolated. Understanding semantic interaction among objects in contact will enable such manipulation tasks in clutters involving overlap. Recently, there has been work on inferring support relationship between a pair of objects [8]–[10]. Rosman *et al.* [9] predict spatial relationships among different objects using stereo images. However, their work deals with simple objects without occlusion and static background. Sjöö and Jensfelt [10] find four types of relations between each pair of objects, viz. casual support, support force, protection and constraint, but only in a simulated environment and are restricted to limitations imposed by simulated environment. Silberman *et al.* [8] consider cluttered indoor environment and predict support relations for each object, i.e., the region supporting a region and the type of support. However, their work does not consider support relationship among different objects overlapping onto each other.

Inference of support relationship among objects in cluttered environment gives information about the objects supported by an object and the type of support. This information can be used to manipulate an object of interest while causing minimal damage to the environment. In order to achieve this,

[1] International Institute of information Technology, Gachibowli, Hyderabad, India swagatika.panda@research.iiit.ac.in

[2] Dept. of Computer Engineering, Hasan Kalyoncu University, Gaziantep, Turkey. ah.abdulhafez@gmail.com

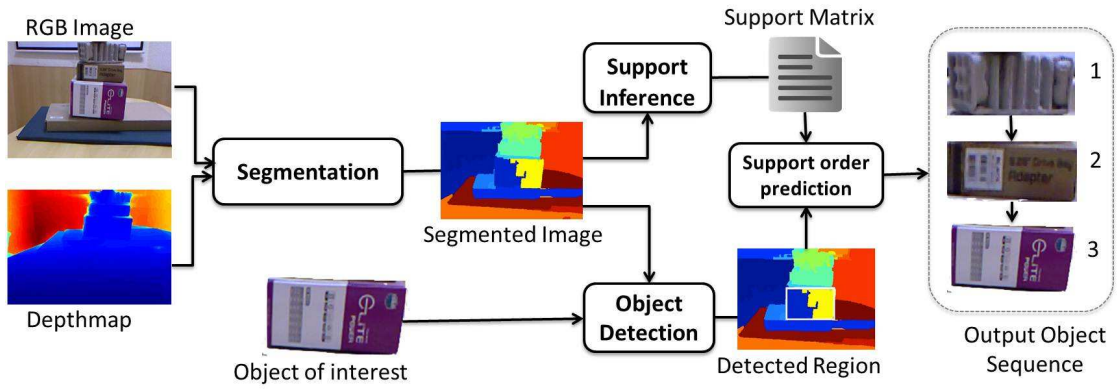


Fig. 2. Block diagrammatic representation of our framework: Segmentation module takes RGB and depth images as input. Segmented image is provided as input to both Support Inference and Object Detection module. Object Detection module also takes the image of object of interest as input and outputs the detected region. Support Inference module gives the support relationship between 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.

we use the inferred support relationship to derive the order in which we need to remove the surrounding objects from the clutter to enable access to our object of interest.

III. OVERVIEW OF FRAMEWORK

The overall framework of our work is explained through the block diagram shown in Fig. 2. The images are first over-segmented into superpixels using Arbelaez's method [11], then segmented using hierarchical segmentation method of Hoiem *et al.* [12]. 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 [13]–[15] between the template image of the object of interest and the input image is performed. The outliers are discarded by applying RANSAC. 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. This approach ensures that the entire object region is chosen for grasping.

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 a MAP inference method adapted from [8] as well as our rule-based inference method. MAP inference method optimizes the pairwise support relation between objects, support type and structure classes using linear programming. However, it does not infer support by multiple objects. In the proposed rule-based inference method, we infer support by multiple objects too. The details of different geometric features used and both the support inference methods can be found in Section IV.

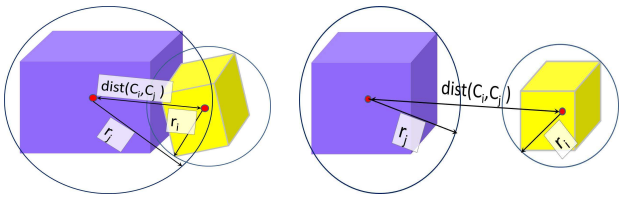
Given the object of interest and the inferred support relationship, a tree is built with the object of interest as the root and it is traversed for support order prediction. A detailed discussion on the approach for support order prediction and how different specific scenarios are handled is given in Section V while the analysis of the results on various images from our RGBD dataset is given in Section VI.

IV. SUPPORT INFERENCE

Given the segmented regions in the image, the object-object interaction among different regions in the image can be inferred. Note that we use the term "object" and "region" interchangeably, since we assume each segmented region corresponds to an object. Our goal is to infer the pairwise relationship between each pair of objects (i, j) where object i is supported by j "from below", "from side" or "contained in it" (Fig. 1). Once this support relationship is inferred, we can derive how to manipulate an object in a clutter by removing other surrounding objects, which we discuss in Section V.

A. Feature Extraction

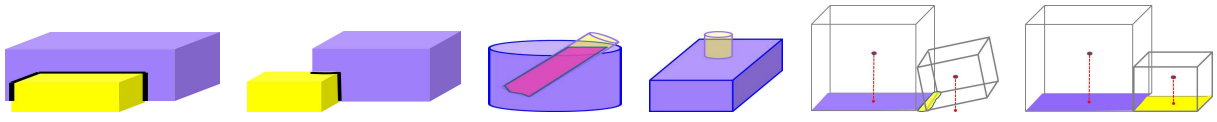
We are interested in finding the support between each pair of objects. Hence, a set of geometric features which exploit the support relationship between each pair of object are introduced for support inference. These features are described as follows:



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

Fig. 3. Demonstration of proximity: lesser f_p implies closer proximity and higher f_p implies less proximity.

Proximity: Two objects must be in each others' proximity in order to provide support to each other as shown in Fig. 3. Proximity f_p of two objects i and j can be measured by the ratio of the distance 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:



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

Fig. 4. (a)(b)Boundary Ratio: The boundary lines are shown in black. (a)There is significant boundary between the two objects showing greater chances of support. (b)Smaller boundary implies less chance of support. (c)(d) containment: Object in purple implies the supporting object and 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.(e)(f)Stability: 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 (e), the horizontal projection of the centroid of the right object does not belong to the baseline and hence the object is unstable. In (f), the horizontal projection of the centroids of both the objects lies in their baseline, hence both are stable.

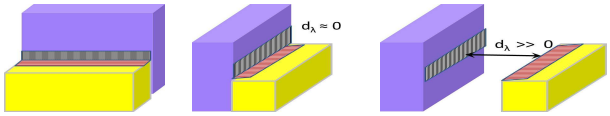
$$f_p(i,j) = \frac{\text{dist}(C_i, C_j)}{(r_i + r_j)}. \quad (1)$$

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

Boundary Ratio: When two objects are in contact, a significant overlap between them exists at their boundaries as shown in Fig. 4(a) and 4(b). The feature "boundary ratio" measures the overlap of a pair of objects over each other. Boundary ratio f_{br} is computed using the following:

$$f_{br}(i,j) = \frac{L(i,j)}{\text{perim}(i)}. \quad (2)$$

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 i.



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

Fig. 5. Demonstration of depthBoundary: 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.

Depth Boundary: In case of visual occlusion, two objects may be either actually in contact or may be isolated from each other (Fig. 5). The feature "depth boundary" discriminates between these two situations [7]. Plane-fitting is done corresponding to two regions adjacent to the boundary between the two objects. If 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. 5(c)). Otherwise, they intersect at a certain angle and a "contact boundary " exists between the two of them (Fig. 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:

$$f_{depth}(i,j) = \frac{1}{1 + e^{-(\beta_1 d_\perp(i,j) + \beta_2)}}. \quad (3)$$

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.

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. 4(c), 4(d)). It is defined as the fraction of the number of points that belong to the supported object N_i contained inside the convex hull Hull(j) of the supporting object j.

$$f_{cnt}(i,j) = \frac{N_i \cap \text{Hull}(j)}{N_i}. \quad (4)$$

Relative Stability A stable object has higher probability of supporting its neighboring 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. 4(e) and 4(f). If the horizontal projection of the centroid of the object belongs to the convex hull of horizontal projection of the baseline points of the object, then the object is considered as stable. Relative stability is defined as:

$$f_{stab}(i,j) = \begin{cases} -1, & \text{if } i \text{ stable and } j \text{ unstable} \\ +1, & \text{if } i \text{ unstable and } j \text{ stable} \\ 0, & \text{otherwise} \end{cases} \quad (5)$$

B. MAP *Inference*

The structure class of all regions in the images and the support relation between each pair of region is inferred using a probabilistic energy framework given in equation (6). A joint probability distribution is defined in terms of supporting regions, structure class and support type adapted from [8]. 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.

$$\begin{aligned} \{\mathbf{S}^*, \mathbf{T}^*, \mathbf{M}^*\} &= argmax_{\mathbf{S,\ T,\ M}} P(\mathbf{S,\ T,\ M}|I) \\ &= argmin_{\mathbf{S,\ T,\ M}} E(\mathbf{S,\ T,\ M}|I), \end{aligned} \quad (6)$$

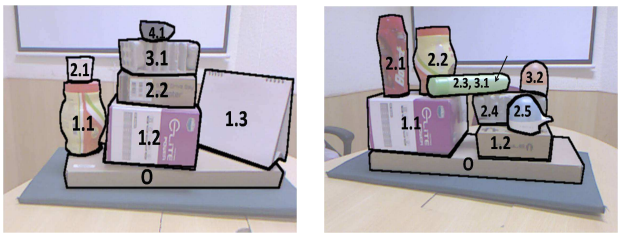
where, $E(\mathbf{S, T, M}|I) = -logP(\mathbf{S, T, M}|I)$ is the energy of the joint probabilstic distribution. Then MAP inference solved by using linear programming.

However, this 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 robotics tasks. To overcome this restriction, we developed a rule based method to infer support by multiple objects as discussed in the next section.

C. Rule Based Method

In this approach, explicit use of the features discussed in Section IV-A is done for support inference. 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 furnitures are decided to be supported directly by the floor. Otherwise it is assumed that floor is not visible in the scene. 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 labeled as stable regions (Relative Stability, f_{stab}). After support inference is performed, the support order for a given object of interest is predicted as discussed in next section.

V. SUPPORT ORDER PREDICTION



(a) (b)

Fig. 6. (a) Case 1: Support in hierarchy. 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.(b)Case 2: Simultaneous support in multiple hierarchy. 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)

Fig. 7. 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 supported by an object *O* need to be removed prior to gasping *O*. So it is necessary to recursively find the objects which are supported by *O*, and the objects that these objects support in turn. In this section, we discuss our approach for determining the "support order" of the objects surrounding our object of interest.

A. Different Cases of Support

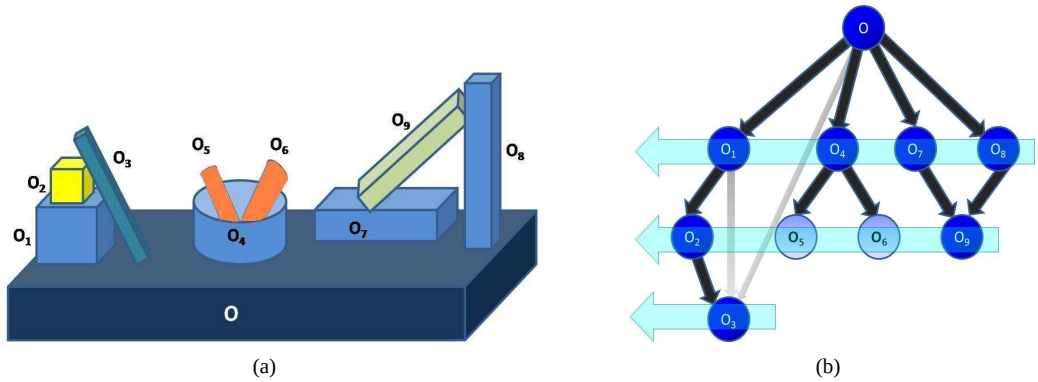
In this section, we discuss different possible cases while we do support order prediction. It is not possible to provide a generalized solution to handle all the cases. Therefore, we treat each case differently and provide a well-tailored solution to each case. The first and most generic case is illustrated in Fig. 6(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. 6(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.

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. 7. If the object *O* is the basket as shown in Fig. 7(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. 7(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 7(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 Section V-B.

B. Hierarchy 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.



(a) (b)

Fig. 8. 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.

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 special cases discussed in Section V-A are taken care of during tree traversal to ensure minimal damage while manipulation. In case of support by multiple hierarchy (Fig. 6(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 V-A and shown in Fig. 7.

Fig. 8(b) graphically demonstrates the tree traversal and support order determination for objects shown in Fig. 8(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

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

VI. EXPERIMENTS AND RESULTS

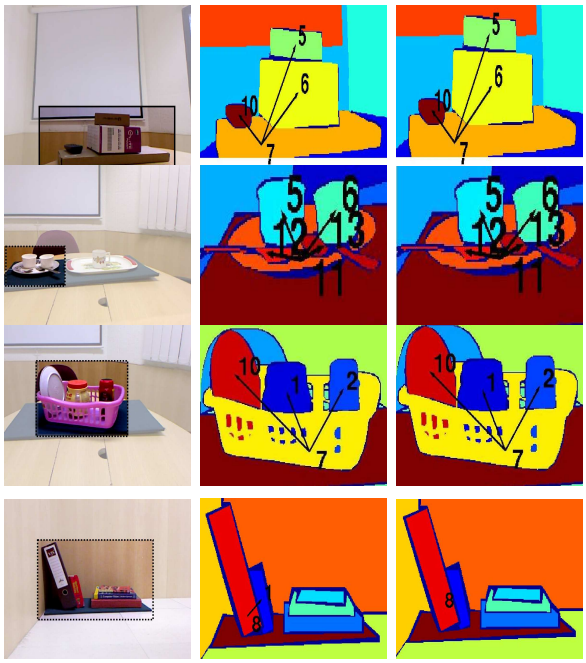
A. Experimental Setup and Dataset Collection

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 between one another. In the publicly available datasets for cluttered environment such as NYU depth dataset [16] and Cornell Scene Understanding dataset [17], the graspable objects are usually present in a far corner of the room instead of being in the center. This necessitated creation of our own dataset. We have collected a dataset consisting of 50 images with different levels of clutter along with their point clouds and depth images using Kinect. We manually create dense labeling and a support matrix for each image. Support matrix encodes the ground truth support relationship between each pair of region in the form of a set of 3-tuples: $[R_i, S_i, T_i]$. The raw depth maps are smoothened using an adaptation of colorization method by Levin *et al.* [18]. The dataset is divided into training and test data in 30 : 20 ratio.

B. Results and Discussion

The results of support inference for a selected set of images from our dataset using rule based method and MAP inference method are shown in Fig. 9. The support relationship is shown by pointing arrows from the object of interest to objects supported by it. The support order prediction for Fig. 9 is given in Table III. The images in row 1 show the support from below. Both rule based 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 as it is. Hence the support order prediction does not generate the labels of the objects contained in the basket as given in Table III. The images in row 4 show support from side. MAP inference fails to infer side support of book 1 by the folder 8, but rule based inference successfully infers the side support.

An object can be supported by multiple objects as shown in Fig. 6(b). The green bottle (shown by pointing an arrow) is supported by two boxes labeled 12 and 11 simultaneously as shown in Fig. 10. Therefore, if our object of interest is



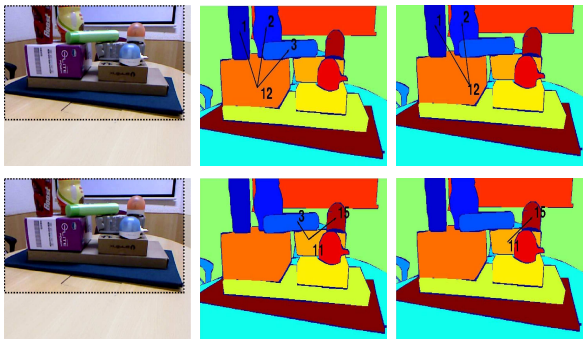
(a) Input Images (b) Rule based Inference (c) MAP Inference

Fig. 9. Results of inference: The highlighted section in input images in col.(a) are zoomed in col. (b) and (c) for better view. The arrows point from object of interest to objects directly and indirectly supported by it.

either of the two, we must pickup the green bottle labeled 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.

We observe that the support inference gets affected by the inaccuracies of structure class prediction. Incorporating explicit structure class information in rule based inference helps avoiding infeasible support relations such as an object supporting the walls or furnitures to a significant extent. As evident in Fig. 9 and 10, 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 furnitures are shown as supported by objects. In addition to that, in some cases, objects are predicted as furnitures due to which the desirable support relation can not be achieved. Some of such results are shown in Fig. 11 and their corresponding support order are given in Table III. In the image in 1st row, the chair labeled 5 is treated as an object and is shown as supported by the closest object that is the book labeled 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 rule based and MAP inference methods. The accuracy of structure class prediction is shown in Table I(a). Since the images are taken in similar environment, the accuracy is reported to be high.

A 5-stage hierarchical segmentation approach proposed by Arbelaez *et al.* [11] was used for segmenting the images.



(a) Input Images (b) Rule based Inference (c) MAP Inference

Fig. 10. 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.

RGB and depth features used in [8] are used for segmentation. Segmentation accuracy is measured as average overlap of segmented regions over groundtruth regions as defined in [12]. The unweighted average overlap score and the score weighted by pixel area are given in Table I(b).

TABLE I

ACCURACY OF STRUCTURE CLASS INFERENCE & SEGMENTATION

(a) Accuracy Structure class Inference

Type	Training Accuracy	Test Accuracy
Ground Truth Regions	100	97.02
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

Accuracy of support inference directly impacts the accuracy of support order determination. Hence the support inference accuracy on two scenarios: using ground truth regions and segmented regions. For each scenario, both “type aware ”and “type agnostic” accuracies are evaluated similar to [8]. 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 II.

TABLE II

ACCURACY OF SUPPORT INFERENCE

Region Source	Ground Truth		Segmentation	
Inference Type	Type Agnostic	Type Aware	Type Agnostic	Type Aware
Rule Based	66.2	56.1	35.1	32.4
MAP Inference	65.8	48.0	32.1	30.5

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 [19], [20] 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 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.

TABLE III

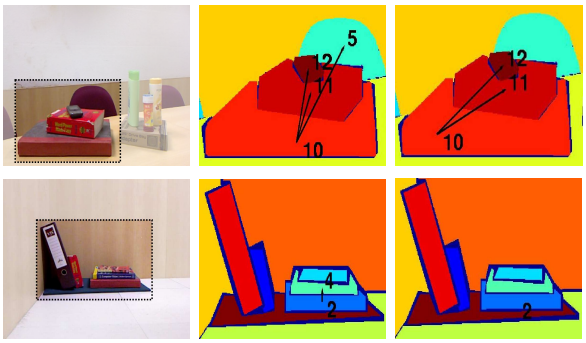
ORDER OF PICKING OF SURROUNDING OBJECTS

Img No.	Object of interest	Order of picking Rule based method	Order of picking MAP inference
9.1	7	5 10 6	5 10 6
9.2	11	6 13 12 5	6 13 12 5
9.3	7	-	-
9.4	8	1	-
10.1	12	3 2 1	2 1
10.2	11	15 3	15
11.1	10	5 12 11	12 11
11.2	2	4	-

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.

VII. CONCLUSIONS

In this paper, 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 used to predict the support order, 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 datastructure 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.



(a) Input Images (b) Rule based Inference (c) MAP Inference

Fig. 11. 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.

REFERENCES

[1] 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.

[2] B. Rasolzadeh, M. Björkman, K. Huebner, and D. Kragic, “An active vision system for detecting, fixating and manipulating objects in the real world,” *IJRR*, 2010.

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

[4] 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.

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

[6] L. Y. Chang, S. Srinivasa, and N. Pollard, “Planning pre-grasp manipulation for transport tasks,” in *ICRA*, 2010.

[7] A. K. Mishra, A. Shrivastava, and Y. Aloimonos, “Segmenting ”simple” objects using RGB-D,” in *ICRA*, 2012.

[8] P. K. Nathan Silberman, Derek Hoiem and R. Fergus, “Indoor segmentation and support inference from rgbd images,” in *ECCV*, 2012.

[9] B. Rosman and S. Ramamoorthy, “Learning spatial relationships between objects,” *IJRR*, 2011.

[10] K. Sjöö and P. Jensfelt, “Learning spatial relations from functional simulation,” in *IROS*, 2011.

[11] P. Arbelaez, M. Maire, C. Fowlkes, and J. Malik, “Contour detection and hierarchical image segmentation,” *PAMI*, 2011.

[12] D. Hoiem, A. Efros, and M. Hebert, “Recovering occlusion boundaries from an image,” *IJCV*, 2011.

[13] D. G. Lowe, “Distinctive image features from scale-invariant keypoints,” *IJCV*, 2004.

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

[15] C. Harris and M. Stephens, “A combined corner and edge detector.” in *Alvey vision conference*. Manchester, UK, 1988.

[16] Nyu depth datasets. [Online]. Available: http://cs.nyu.edu/ silberman/datasets/

[17] Cornell scene understanding datasets. [Online]. Available: http://pr.cs.cornell.edu/sceneunderstanding/data/data.php

[18] A. Levin, D. Lischinski, and Y. Weiss, “Colorization using optimization,” in *TOG*, 2004.

[19] M. Bjorkman and D. Kragic, “Active 3d scene segmentation and detection of unknown objects,” in *ICRA*. IEEE, 2010.

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