

# Mobile Manipulation in Domestic Environments Using A Low Degree of Freedom Manipulator

Jacob Huckaby, Carlos Nieto-Granda, John G. Rogers III,  
 Alex J. B. Trevor, Akansel Cosgun, and Henrik I. Christensen  
 Robotics & Intelligent Machines  
 Georgia Institute of Technology  
 Atlanta, GA 30332, USA

huckaby@gatech.edu, carlos.nieto@gatech.edu, {jgrogers, atrevor}@cc.gatech.edu,  
 akanselcosgun@gatech.edu, hic@cc.gatech.edu

**Abstract**—We present a mobile manipulation system used by the Georgia Tech team in the RoboCup@Home 2010 competition. An overview of the system is provided, including the approach taken for manipulation, SLAM, object detection, object recognition, and system integration. We focus on our manipulation strategy, which utilizes a low-degree of freedom manipulator and makes use of the robot’s differential drive as part of the manipulation strategy. Empirical results demonstrating our platform’s ability to detect and grasp a variety of tabletop objects are presented.

## I. INTRODUCTION

Mobile manipulation is a popular and growing area of interest in robotics research. One of the great promises of mobile manipulation is the ability to reliably function in domestic environments along with humans, enabling an increased quality of life for those who might not otherwise be able to function by themselves in a home environment. This quality of life improvement through mobile manipulation can only be achieved through the development of robust methods for dealing with the highly dynamic nature of domestic environments, such as navigation and mapping, as well as efficient and cost effective solutions for manipulating objects in the home. This paper presents work by researchers in the Cognitive Robotics Lab at Georgia Tech on a mobile manipulation platform that is being developed to address these issues of operating in a domestic environment. An approach is presented that demonstrates manipulating household objects using an inexpensive, low degree of freedom manipulator. Through experimental results, it will be shown that this approach is effective at manipulating objects for some subset of scenarios that could be encountered in a typical home environment. The scenario descriptions and solution metrics are given in the context of the RoboCup@Home contest, where this platform was demonstrated at the 2010 RoboCup competition.

This paper will first address previous work done in the area of mobile manipulation, especially mobile manipulation in domestic environments, in section II. Section III will then discuss the different hardware and software systems

utilized in the Jeeves mobile manipulation platform. This will include a description of the issues involved in grasping objects from planar surfaces such as tables and counters, as well as the approach used by the Jeeves platform. The experiments conducted to demonstrate and test this approach will be presented in section V, and a brief discussion of the results of the experiments and future work will follow in sections VI and VII.

## II. RELATED WORKS

One example of a successful implementation of a manipulation platform based upon limited degrees of freedom and simple control is due to Brooks *et. al.* [4], [3]. Brooks *et. al.* developed a robot called Herbert which searches offices at MIT and steals empty soda cans that it finds. Herbert has a limited DOF manipulator and a simplified control system based upon the subsumption architecture [2]. The subsumption architecture is a layered, coupled set of finite state machines which exhibits sophisticated high level behavior. The performance of Herbert and other robots using the subsumption architecture is robust to unexpected changes in the environment since each component relies on sensor information from the environment rather than an environmental model.

## III. SYSTEM DESIGN

This section will present a description of the system that was designed to perform tasks in the RoboCup@Home contest. First the physical description of the platform will be given, including specific hardware selection and purpose. An overview of hardware and software modules involved in the manipulation of simple domestic objects will also be presented.

### A. Hardware

The Jeeves mobile manipulation platform, shown in figure 1, is based on a Segway RMP-200 dynamically balancing mobile unit. The platform has been augmented with four

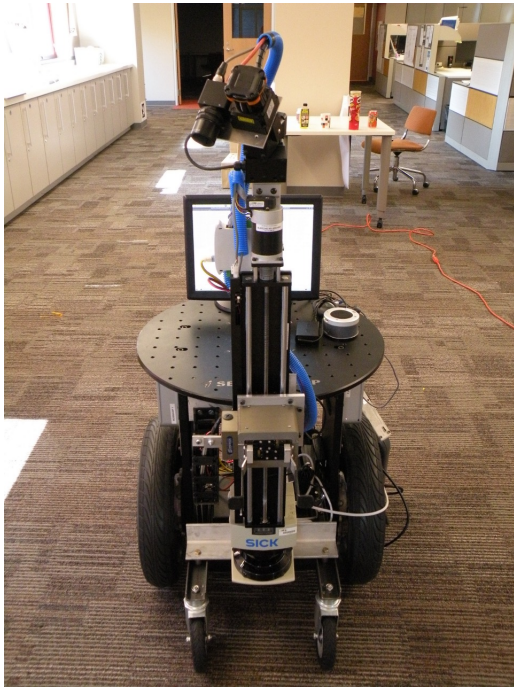


Fig. 1: Jeeves mobile manipulation platform.

caster wheels to make it statically stable, which simplifies the manipulation problem considerably.

Grasping and manipulation is performed by a Schunk PG-70 parallel gripper module. The custom steel fingers are sized to interact with small household objects, and are coated with rubber to reduce sliding. The gripper is mounted to our manipulator, which consists of an Anaheim 18-inch linear actuator mounted to the front of the Segway top plate. The linear actuator is positioned to allow a range of motion that would cover most usable surfaces in home settings, such as low coffee tables or high counters.

The current setup includes the use of two Mac Minis as the main computers. A Sony handheld PC, equipped with a wireless headset, serves as the main computer for human-robot interaction tasks, such as speech recognition.

The main sensor package on the Jeeves platform consists of a Prosilica gigabit ethernet monocular camera and a Hokuyo UTM-30 scanning laser range sensor attached to a Directed Perception D46-70 pan-tilt unit. This is used for both object detection and object recognition in manipulation tasks.

The Segway is also equipped with a SICK LMS-291 laser range finder which serves as the main navigation sensor, and is used primarily for localization/mapping and object avoidance. A Phidgets interface board is used to communicate between the hardware e-stop on top of the platform and the navigation computer controlling the platform motion.

### B. Software Architecture

Jeeves utilizes the popular Robot Operating System (ROS) [6] for inter-process communication and module support where possible. There are five main software components

in our system. The navigation components will be used in virtually all @Home tasks and they consist of *Localization*, *Motion Planning*, and *Obstacle Avoidance*. These represent the basic competencies for making a mobile robot operate in a known static environment amongst dynamic obstacles, such as people. For tasks where the environment is not known prior to the robot run, we will use the custom SLAM implementation in the *Mapping* module.

The Perception components consist of the *Object Recognition* module. Since this module uses monocular vision, it is coupled with an *Object Pose Estimation* module which uses the laser scanner to get more information about the pose of the manipulation target. Additionally, there are a *People Tracking* and *Face Recognition* modules which enable person following and tracking based on laser and visual data.

High level task coordination is handled by the *Task Planner* which is configured for task-specific events.

### C. Manipulation and Grasping

This section will discuss the manipulation capabilities of the Jeeves mobile manipulation platform. Jeeves is equipped with a Schunk PG-70 parallel-plate gripper. The gripper is mounted to an 18-inch linear actuator that has been mounted vertically on the front of the mobile base. This allows Jeeves to achieve a grasping range of motion that covers many common human grasping surfaces. The gripper utilizes a set of custom steel fingers that are long enough to allow for manipulation up to 20 centimeters into a surface. This provides us with a single degree of freedom for manipulation, along with the three degrees of freedom provided by the differential drive mobile base.

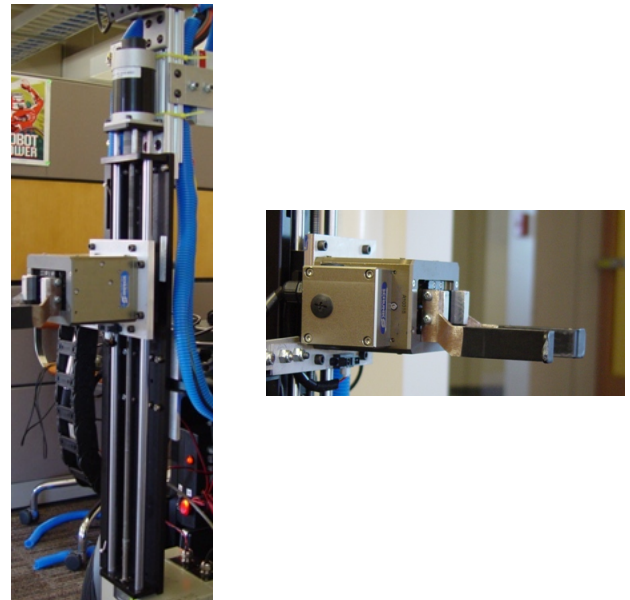


Fig. 2: Manipulator and gripper used on the Jeeves platform.

The design decision of using a low degree of freedom manipulator over the more common 5- or 6-DOF manipulators has a few hidden advantages. While the single degree of freedom limits the orientation of certain types of objects

to be considered manipulable, at least within the context of the RoboCup@Home event, this still leaves available a wide range of objects commonly used by humans, such as cups and containers. There are a number of advantages of this simple manipulator over more complex manipulators. Certainly it is more simple. This implies that controller design can be simplified, along with the computational overhead for computing complicated inverse kinematics and scenarios for dealing with singularity avoidance. A simplified system can be made more robust than advanced manipulators that require more engineering (and more points of failure.) The low mechanical overhead also can mean a lower cost and a lower footprint on the robot. Having only a single degree of freedom also allows the system to use a simple grasping strategy.

#### D. 3D Perception

Tabletop objects are detected using a 3d laser scanner mounted on our robot, shown in Figure 1. The sensor consists of a Hokuyo UTM-30-LX laser scanner mounted on a Directed Perception PTU-D46-70 pan tilt unit. Panning the laser scanner allows us to build up a 3D point cloud of the scene in front of the scanner. The resulting point clouds are then processed using the Point Cloud Library (PCL) to detect planar surfaces such as tables, as well as objects resting on these surfaces. Our technique is similar to that presented by Rusu *et. al.* in [7], [8]. For the purposes of grasping, we assume that the robot is positioned roughly in front of a table. This is reasonable for applications where we are navigating to a table location as taught by a human, as is the case in Robocup tasks.

Segmenting objects in front of the robot begins by finding the nearest table. Only points within a relatively small bounding box in front of the sensor are considered. For this work, we discarded points that were not within 1.5m x,y bounding box in the front of the robot, and were more than 30cm above the ground plane, because we don't wish to consider the ground as a surface relevant to manipulation. We then detect the dominant plane in this region using a RANSAC technique. Point inliers are projected down to the plane, and a convex hull is constructed. The hull is used to extract points above the planar surface, which are then clustered into discrete objects. The result is one or more sets of clustered points, corresponding to objects resting on the planar surface. The centroid of each resulting object point cloud is then computed. These clouds can serve as input to our visual object recognition, or be used directly for grasping of the objects.

#### E. Object Recognition

To identify each object we make use of a Prosilica GigE 650C camera mounted on the PTU adjacent to the Hokuyo laser scanner. The camera is servoed by the PTU to point the centroid of the segmented object point cloud. By projecting the 3D points into the image taken from the camera at this viewpoint, a small region of interest is extracted corresponding to the extent of the object in the image. We extract SURF

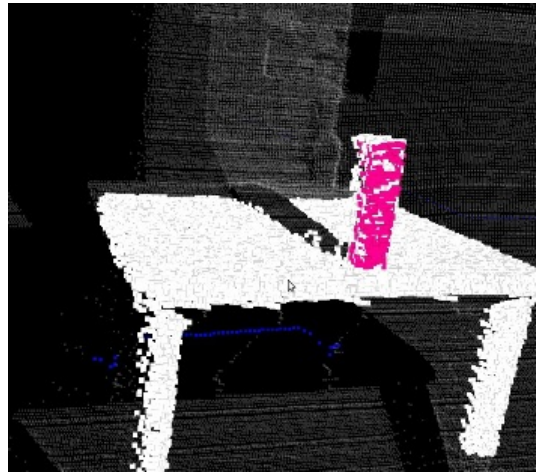


Fig. 3: 3D point cloud and object segmentation.

[1] features from this smaller image. The SURF features are compared against the SURF features from a library of objects at a variety of poses which has been trained *a priori* using the ratio test. As a final step, a homography is computed with RANSAC [5]. The test object's identity is assigned to be the identity of the library object which had the largest number of inlier SURF features from this test.

#### IV. PROBLEM DESCRIPTION

As it has been stated previously, this paper is presented in the context of the RoboCup@Home scenarios and task specifications. The RoboCup@Home league is concerned with challenging robots at tasks in various home settings and situations that would need to be considered for domestic service robots. A number of different tasks have been required in the last few years since the league began, including person following through crowded environments with dynamic obstacles, locating and manipulating various objects in a simulated home environment, and interacting with and manipulating objects in the presence of humans.

As many of the challenges in the contest deal with grasping objects from planar surfaces as part of the task requirements, an algorithm was designed for completing specifically this task. Namely, the robot begins with a candidate surface being specified. The robot then navigates to some predefined destination in the neighborhood of the desired surface, oriented such that the surface is visible to the camera and laser range scanner. This motion is controlled by one of the ROS modules used by the system. A 3D point cloud is obtained from the sensor package, and objects on the surface are segmented from the point cloud. From the point clouds of the individual objects the centroid is calculated. This centroid is used as the control point for the grasping controller.

At this point, control of the pose of the robot is switched to a local controller to drive the robot to the grasp point, or centroid of the object to be manipulated. The states associated with this situation are  $\{dX, dY, dZ, \theta\}$ , where  $dX$  and  $dY$  are the distances between the current robot pose and

the object centroid along the coordinate axes in the local coordinate frame of the robot, and  $\theta$  is the bearing to the object centroid from the current robot heading.  $dZ$  is used to compute the vertical offset between the object centroid and the grasping center of the gripper.

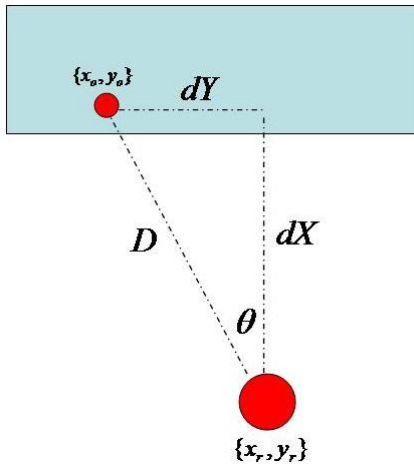


Fig. 4: Overhead view of the table approach, where  $(x_r, y_r)$  is the position of the robot and  $(x_o, y_o)$  is the position of the object centroid.

The control variables for the proposed control scheme are  $\theta$  and  $D$ , where  $D$  is the Euclidean distance between the current robot pose and the object centroid and in general is found by the relationship

$$D = \sqrt{dX^2 + dY^2}.$$

The selection of these control variables was fairly straightforward, as the control input into the Segway mobile base used by the Jeeves platform accepts linear velocity and angular velocity as the control commands. This logically leads to the consideration of  $D$  and  $\theta$  as control variables as units that deal with linear and angular disparities, respectively. Thus the overall control strategy is to minimize the distance  $D$  and the bearing  $\theta$ , which drives the system to the desired setpoint, a robot pose that is able to manipulate the object.  $\theta$  is determined by the relationship

$$\theta = \arctan\left(\frac{dY}{dX}\right).$$

The controller utilized for driving the distance  $D$  to zero obeys the control law,

$$v_{command} = K_{p-v} * D,$$

where  $v_{command}$  is the linear velocity applied to the mobile base and  $K_{p-v}$  is the gain. While a more complex controller was considered, initial testing showed that this simple controller worked reliably within the expectations for object grasping tasks.

The control input for the orientation of the robot is determined using a similar control law,

$$\omega_{command} = K_{p-\omega} * \theta,$$

with  $\omega_{command}$  being the commanded angular velocity and  $K_{p-\omega}$  the proportional gain applied for angular acceleration. One problem with directly applying this simple control law to the problem of angular correction is that the calculation for the bearing (and thus the angular error) depends on a non-linear term,  $\arctan\left(\frac{dY}{dX}\right)$ . While using the direct control will have reasonable results when far away from the object centroid it tries to move to, those results will increasingly degrade as the robot comes close to the object. As the distances become smaller upon approach, so does the effect that measurement and mechanical noise in the system increasingly has a greater influence on the accuracy of the results. This poor, noisy behavior becomes even more pronounced within several centimeters of the robot, often making actual grasping a challenge. That is the reason it was decided to use a gain scheduling approach to solve this nonlinear problem.

In essence, gain scheduling is the use of different gain parameters that induce system properties (such as stability) in only certain regions of the controllable subspace due to the non-linear constraints of the system. In this way, the entire state space of the system can be made controllable by scheduling different gains to properly influence whatever equilibrium the system happens to be in the neighborhood of. The angular bearing controller utilized by the Jeeves system uses three different gain schedules. For distances  $D$  of greater than 20 centimeters, a larger gain is used. Between 20 centimeters and 7 centimeters, a much smaller gain is used to further control to the centroid of the object, and at distances less than 7 centimeters the gain is reduced to 0. The gain scheduling technique for dealing with non-linearities in systems controller design has been successful with the Jeeves platform, and has demonstrated that it can deliver reliable and repeatable performance in a number of lab tests. It is the premise of this paper that this controller can robustly deal with the problem of manipulation in domestic settings using this unique combination of simple and non-linear controllers.

## V. EXPERIMENTS

A single experiment was designed to test the efficacy of the grasping controller used on the Jeeves platform. The test was constructed in the context of the RoboCup@Home scenarios. Specifically, many of the RoboCup@Home tasks that require manipulation involve grasping an object from a planar surface such as a table or a shelf and then taking the object to a secondary location, such as delivering it to a human.

In the spirit of these tasks, the experiment designed to test the system requires the robot to grasp an object from a specified location on a table and then deliver this to a human test operator. A grasp is defined to be the robot making contact with the object, and lifting and holding it for at least ten seconds, following those rules set forth by the RoboCup@Home organizers. A single test in the experiment would begin by the robot initiating a scan of the environment using the 3D scanning sensor package, assuming that it has already achieved a position in front of some planar



surface. When the point cloud is processed and the object is segmented from the table top, a centroid for the object is returned to the controller, which it then uses as the setpoint to servo to the object location. After the object has been grasped, Jeeves returns to the starting location and orients itself away from the table briefly, allowing time for the test operator to retrieve the object from the gripper and return it to the table for the next part of the experiment. After a few seconds, Jeeves turns again to face the table, and the sequence is repeated.



Fig. 5: Objects used in the manipulation experiment.

Ten objects were selected to serve as the representative categories of general objects found in domestic environments, and can be seen in figure 5. As can be seen in the figure, effort was made to select from a wide range of common household products, including beverage and food containers, cleaning supplies, pill bottles, and toys. Among these types of products, the objects chosen share some basic geometric shape properties, such as cylindrical or box shaped. These suggest four categories from the ten selected objects. Wide cylindrical objects are cylindrical objects whose diameter is within a few centimeters of the maximum grasping width of the gripper. Narrow cylindrical objects are cylindrical objects whose width is much less than the maximum grasping width of the gripper. Box shaped objects are simply that - boxes, or in this case, tea boxes. Finally, odd-shaped objects do not easily fall into any one category. The cleaner has an oval-shaped container, while the pill bottle is a box with rounded sides. The stuffed doll object was chosen to provide a stark contrast to the other relatively simply described objects selected. The doll is placed on table for grasping in a sitting position, which is inherently a balancing position. This adds yet another layer of difficulty onto the task of grasping the doll.



Fig. 6: Marked locations on the table used during the manipulation experiment.

Each object was attempted a total of 24 times, cycling

the objects through a set of six points on the table (see figure 6.) The points on the table are measured at depths of five and fifteen centimeters from the front edge of the table, while the side points are also five centimeters from their respective sides. Each table location is also fifteen centimeters away from its neighbors. The decision to rotate through these various depths and lengths across the table was to provide a method to test the robustness of the control algorithm to variations in the object pose with respect to the robot's starting pose. The initial robot pose also varies slightly between tests as the robot moves, adding to the presumed robustness of the controller.

## VI. EXPERIMENTAL RESULTS

It took between ten and eleven hours to complete testing for the experiment described in section V. The results of this experiment can be seen in table I.

Object	Success (%)	Shape	Overall (%)
Pringles	100	narrow	100
Fanta	100		
Coke	100		
Cable Ties	91.7	wide	93.75
Clorox	95.8	wide	
Mango Tea	100	box	91.67
Earl Grey	83.3	box	
Goo Gone	91.7	odd	83.33
Vitamins	100	odd	
Buzz	58.3	odd	

TABLE I: Results of grasping experiment using different classes of objects.

The first evident observation from the data presented in table I is that the controller works well across all categories of objects. In fact, every object in the narrow cylindrical shaped objects category was grasped with a one hundred percent success rate.

Another interesting observation, as seen in figure II, is that by far the most grasp failures happen on locations that are toward the edges of the table.

Location	Wide	Narrow	Box	Odd	Total
1	0	0	0	3	3
2	0	0	2	2	4
3	1	0	0	1	2
4	1	0	0	1	2
5	0	0	0	2	2
6	1	0	2	3	6

TABLE II: Failed grasp attempts based on the location on the table and the object shape.

There are a number of possible causes for the failed grasps in the experiment. These include variability in the perception and segmentation of objects in the scene, the effect of the caster wheel design, and the strong angles required to grasp objects at the edges of the table. One possible source of error is in the processing of the point clouds. From time to time the segmentation will not group together all of the points in the point cloud that actually belong to the object. This

skewed perspective can push the calculated object centroid away from the actual centroid by a few centimeters. When this occurs, it is possible that the disparity could be enough to induce a failed grasp.

Another possible source of process noise in this experiment is the caster wheel design. As seen in figure 1, Jeeves has two casters attached to the front and back of the unit for a total of four casters. Depending on the configuration of the casters, especially the front casters, they can cause the platform to have to exert more effort than usual to move, which can cause significant overshoot, which in turn induces unwanted oscillation in the platform as it is trying to precisely servo to the grasp point.

One downside of the approach proposed in this paper is that it doesn't take it into account the existence of oriented objects, or objects that have a particular orientation. For example, when trying to grasp the box of tea, Jeeves does not know whether the object it is grasping is facing directly out from the table, or if it is rotated by some number of degrees. During the experiment this led to a number of "angled grasps", or successful grasps where the object was rotated in the gripper. It would seem that the combination of these factors, which would all come into play during an edge location grasp, could lead to grasp failure.

Nevertheless, the control scheme proposed and implemented is shown in this experiment to be robust to variations of an object pose in the robot's workspace, along with variations in the starting pose of the robot.



Fig. 7: Jeeves grasping an object.

## VII. SUMMARY

In this paper we have presented the Jeeves mobile manipulation platform as it functioned at the 2010 RoboCup competition in the RoboCup@Home league. Specifically, we have shown that an approach to object manipulation in domestic environments using a low degree of freedom manipulator can be both a viable and a cost-effective solution to the mobile manipulation problem in home settings. Through experimental results we have shown that with a single degree of freedom manipulator and the control scheme proposed and implemented on the Jeeves mobile manipulation platform, we

can achieve reliable and repeatable results for a number of varied household object types. In the future we plan to further improve the system by including a sense of the orientation of an object, to better improve the robustness of the solution.

## VIII. ACKNOWLEDGMENTS

The authors gratefully acknowledge the support of Boeing for funding this research.

## REFERENCES

- [1] H. Bay, A. Ess, T. Tuytelaars, and L. Van Gool. Speeded-up robust features (SURF). *Computer Vision and Image Understanding*, 110(3):346–359, 2008.
- [2] R. Brooks. A robust layered control system for a mobile robot. *IEEE journal of robotics and automation*, 2(1):14–23, 1986.
- [3] R.A. Brooks. Elephants don't play chess. *Robotics and autonomous systems*, 6(1-2):3–15, 1990.
- [4] R.A. Brooks, J.H. Connell, and P. Ning. Herbert: A second generation mobile robot. *Technical Report*, 1988.
- [5] Martin A. Fischler and Robert C. Bolles. Random sample consensus: a paradigm for model fitting with applications to image analysis and automated cartography. *Commun. ACM*, 24(6):381–395, 1981.
- [6] Morgan Quigley, Ken Conley, Brian Gerkey, Josh Faust, Tully B. Foote, Jeremy Leibs, Rob Wheeler, and Andrew Y. Ng. ROS: an open-source robot operating system. In *International Conference on Robotics and Automation*, Open-Source Software workshop, 2009.
- [7] R.B. Rusu, N. Blodow, Z.C. Marton, and M. Beetz. Close-range Scene Segmentation and Reconstruction of 3D Point Cloud Maps for Mobile Manipulation in Human Environments. In *Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, St. Louis, MO, USA, 2009.
- [8] R.B. Rusu, Z.C. Marton, N. Blodow, A. Holzbach, and M. Beetz. Model-based and learned semantic object labeling in 3D point cloud maps of kitchen environments. In *Proceedings of the IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, St. Louis, MO, USA, 2009.