Replacing the Office Intern: An Autonomous Coffee Run with a Mobile Manipulator

Anthony Pratkanis[1], Adam Eric Leeper[2] and Kenneth Salisbury[3]

***Abstract*— We describe our development of an autonomous robotic system that safely navigates through an unmodified campus environment to purchase and deliver a cup of coffee. To accomplish this task, the robot navigates through indoor and outdoor environments, opens heavy spring-loaded doors, calls, enters, and exits an elevator, waits in line with other customers, interacts with coffee shop employees to purchase beverages, and returns to its original location to deliver the beverages. This paper makes four contributions: a robust infrastructure for unifying multiple 2D navigation maps; a process for detecting and opening transparent, heavy spring-loaded doors; algorithms for operating elevators; and software that enables the intuitive passing of objects to and from untrained humans.**

Fig. 1. An autonomous robot waits in line to purchase a cup of coffee.

I. INTRODUCTION

Autonomous robots simultaneously amaze and frustrate us. They can track and catch objects at incredible speed yet have difficulty manipulating a child's toy; they can spot subtle patterns amid noisy data yet their programs fail to adapt to unexpected events. In order for robots to move from the laboratory and factory to unstructured and unpredictable indoor and outdoor environments, researchers must learn how to synthesize various, often unrelated robot capabilities into a reliable system.

The aim of this paper, then, is to describe the methods and lessons learned in programming a commercial mobile manipulator to accomplish a common human task: the purchase (and delivery) of a cup of coffee from the nearby campus coffee shop. The goal was to accomplish this task with minimal "special-case" modifications to the robot and **no** modifications to the environment. While in the future it may become more socially acceptable to engineer environments with markers and sensors for robots to exploit, it remains important to find ways for robots to operate in environments that were not designed with robots in mind.

In this paper, we describe strategies for dealing with four unique challenges in accomplishing the coffee delivery task:

1) *Multi-map navigation*. While research in 2D and 3D navigation is relatively mature, it is non-trivial to implement a navigation planner that will efficiently deal with large, multi-floor environments. To resolve this we developed an infrastructure to allow manual unification of multiple 2D SLAM maps via gateways to enable seamless navigation.
2) *Opening heavy, spring-loaded, glass-panelled doors*. The doors in our building are transparent and flush with the wall, so we propose suitable detection features that are different than those used in previous work. Further, the robot's arm is too weak to pull open the doors, so we describe a new maneuver that uses the base and one arm to pull, prop, and push the doors.
3) *Operating an elevator*. There are numerous obstacles and failure conditions for the elevator, such as humans getting on and off or calling it to the wrong floor. We developed a robust system for navigating the elevator by taking advantage of existing markers typical of elevator environments.
4) *Passing objects to humans*. We developed a system for *untrained* humans to intuitively pass objects (such as coffee cups and bags of money) to and from the robot.

The software developed for this project is built on *ROS* and is freely available under an open-source license.

II. RELATED WORK

We developed a new integrated robotics system based on PR2, a mobile manipulator produced by Willow Garage[1]. Similar demonstrations of robot systems integration have been done by Meeussen et. al. [8] and Okada et. al. [4].

*This work was supported in part by Stanford University's Summer Undergraduate Research Institute (SURI).

[1]Student, Department of Computer Science, Stanford University, Stanford, CA, USA `atp42 at stanford.edu`

[2]Student, Department of Mechanical Engineering, Stanford University, Stanford, CA, USA `aleeper at stanford.edu`

[3]Faculty, Departments of Computer Science and Surgery, Stanford University, Stanford, CA, USA `jks at robotics.stanford.edu`

[1]See www.willowgarage.com for details

However, our work addresses specific challenges that were absent in previous work.

A. Multi-map Navigation

Two dimensional localization techniques have been used to accomplish impressive feats of autonomous navigation [8]. However, simple 2D navigation implementations do little to address scalability. Multi-story buildings break the assumptions of 2D navigation, and extremely large environments present memory and performance problems. Our approach resolves both issues by allowing multiple 2D maps to be connected together by various transitions. Okada et. al. [4] demonstrated similar multiple map navigation with the PR2. We believe our approach is simpler, more user-friendly and more scalable to different types of environments.

B. Door Detection and Opening

Many groups have worked on door opening with mobile manipulation platforms. Meeusen et. al. [8] suggest a method for opening doors with the PR2 based on 3D laser detection of the handle and door frame, supplemented by 2D recognition of the handle. This works well for solid interior doors; one can further exploit the fact that these doors are often recessed from the surrounding wall, resulting in clearly defined edges useful for pose estimation. Klingbeil et. al. [6] also used laser and vision detection to find the door handle on previously unseen doors. In both cases, laser and vision techniques consistently failed on transparent doors. Our approach avoids these problems by utilizing a haptic strategy to find and estimate the pose of the handle.

With regard to actually opening doors, Meeusen et. al. [8] and Chitta et. al. [1] demonstrated opening of interior doors by pushing and pulling. It is significantly more difficult, however, to open spring-loaded doors, that is, doors that close when released by the robot. Gray et. al. [3] used both arms of the PR2 to open a spring-loaded door, switching from one arm to the other while opening. However, this approach is limited to robots with two arms, and fails with doors that are too heavy for the robot's arm. Our approach uses only one arm and the much stronger base to pull and then prop open the door.

C. Elevator Operation

Elevator operation poses a daunting challenge because robots must be able to detect and manage a wide variety of buttons and signs, while handling numerous possible failure modes. Klingbeil et. al. [5] demonstrated detection of buttons in a diverse set of elevators. The more general elevator navigation problem has been explored by a number of groups ([4],[9]), but in each case unique properties of the elevators were exploited, such as back-lit buttons [9] and visible signs [4]. Our work adds robustness to elevator navigation by ensuring that the robot exits on the correct floor, and by exploiting mechanical alignment techniques to minimize the dependence on vision.

D. Object Passing

Passing of objects between robots and humans is a complex field in itself. Two general approaches have been studied: those in which the human approaches the robot [2], and those where the robot approaches the human [7]. We build on the work of Edsinger and Kemp [2], allowing the human to approach the robot while making use of a forearm-mounted camera to improve detection of objects inserted into the gripper.

III. PROBLEM STATEMENT

A. Problem Description

We examine the problem of successful navigation and object transport across an unmodified indoor and outdoor campus setting. The robot must autonomously, without human intervention:

- navigate to any reachable location.
- open doors and operate elevators.
- interact with human employees and purchase a beverage at a coffee shop.
- manipulate and transport a filled beverage cup.

We also require that the robot be capable of handling non-standard environmental conditions such as doors held open by humans, elevators arriving on the wrong floor, and human failure at the beverage passing task. We require that in such failure cases, the robot keep trying until it accomplishes its goal.

An instance of the problem has three main components:

1) A human beverage requester. This is the human to whom the beverage will be given after it is procured. Currently, this is interpreted as a 2D map location to which the robot will attempt to navigate.
2) The coffee shop location. This consists of the physical location of the counter where beverages can be purchased as well as a course of locations where people form a line while waiting to be served.
3) The environment map. This consists of a number of 2D maps and a list of locations for doors and elevators that allow transitions between the maps.

The task proceeds as follows. The beverage requester fills a small plastic bag with money and a piece of paper describing the beverage order, and hands this to the robot. The robot then proceeds to the coffee shop. In our instance of the problem, the robot must at this point navigate five doors and one elevator in order to navigate from the human to the coffee shop. Once at the coffee shop, the robot drives a fixed course of locations to wait in line. When it reaches the front of the line, the robot passes the plastic bag containing money and paper instructions to the coffee shop employee at the counter. The employee is asked to place the beverage cup (and change!) in the plastic bag (to help avoid damage in case of spills), and to give the bag and cup to the robot. The robot places the cup in a custom shoulder-mounted cup holder, and returns to the beverage requester via the same route it took earlier. The robot has completed its task once it has handed the cup to the requester.

B. Supporting Infrastructure

Our platform is the PR2 robot[2]. This task exploits most physical capabilities of the robot: a pseudoholonomic base, two 7-DOF arms with 1-DOF grippers, a pan-tilt sensor head, and a telescoping spine. The sensors include cameras on the head and forearms, and Hokuyo UTM-30 laser scanners.

The robot runs ROS, a set of open source packages that implement a wide variety of hardware drivers, controllers, and robot algorithms. See www.ros.org for details.

IV. APPROACH

Our general approach was to develop simple heuristics that allowed us to accomplish the various tasks. To enhance robustness, we attempted to avoid the use of complex algorithms that lack transparency and pose challenges for generalizability.

A. MULTI-MAP NAVIGATION

Navigating an environment typically involves global route planning using a pre-made map, as well as local obstacle avoidance and re-planning. However, typical implementations do not scale well. In our case, we needed support for large maps (a memory issue) and multiple floors (which is out of scope for standard 2D planning, and for which full 3D planning is overkill). These challenges can be solved by creating robust ways to transition between multiple 2D maps at appropriate locations (see figure 2).

In a building setting there are four types of transition: proximity to a given location, such as a hallway opening (bidirectional); door pushing or door pulling (each classified as separate unidirectional actions); and elevator management to go between floors (multi-directional).

The process of multi-map navigation begins with a known starting location (assuming the robot is localized on a map via AMCL) and a goal (a pose specified on one of the available maps). The system uses these points and the knowledge of the transitions between maps to build a directed graph of potential routes. It then uses Dijkstra's algorithm to plan a high-level path through each of these nodes. Finally, it run a series of programs to traverse each link of the graph, including the standard ROS navigation stack, and the new door and elevator management software described in the sections that follow. The above algorithms are implemented in a new ROS package, `multi_map_navigation`. [3]

B. DOOR OPENING

Manipulating transparent spring-loaded doors can be broken into 4 sub-problems:

1a Detecting and aligning with doors that push open.
1b Pushing open the doors.
2a Detecting and aligning with doors that pull open.
2b Pulling open the doors.

The multi-map navigation software is aware of the directionality of doors, and therefore is able to cue the right kind of behavior as the robot approaches a door.

[2]See www.willowgarage.com for details

[3]Multi-map navigation is designed to be completely re-usable. See http://ros.org/wiki/multi_map_navigation for details.

Fig. 2. Multimap navigation in simulation. The blue cylinders represent transitions between two maps.

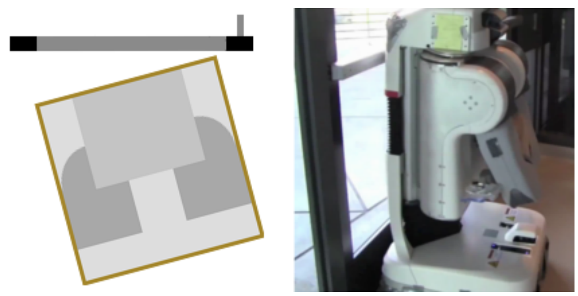

Fig. 3. The robot preparing to push the door open.

1) Door Pushing Alignment: The primary challenge with detecting doors in our environment is that they are flush with the wall, and thus traditional detection features are absent. However, the glass panels in our doors are largely transparent to the laser scanner, giving a new type of feature that can be exploited. Hence, we propose glass doors can be localized by detecting an expected gap in the laser data. It is important to note that directing the laser scanner downward so that it makes an angle of $30°$ with the horizontal helps to avoid signs and other obstructions that humans place on doors, and minimizes refraction artifacts in the data. The robot centers itself on the door and aligns with the plane of the door using the laser data.

2) Door Pushing: The robot uses wheel odometry to ensure that it traverses the proper course while pushing through a door. With the PR2 it is easier to make initial contact with the *back* of the robot, which depresses the push bar on the door. Once the robot touches the door, it continues to use velocity control to apply more force as needed. The robot is successful after driving 1.5 meters past the door.

3) Door Pulling Alignment: Door pulling requires detection of the door and handle. Since alignment is critical and the shiny handle is very difficult to see using sensors, this step is performed using machine haptics.

To align the robot with the door, the robot moves in to make contact with its base. It then performs small rotations of the base while driving forward to settle into alignment with the plane of the door.

To find the door handle, the robot backs away from the door and raises its gripper to an area near, but far to the right of, the estimated location of the handle. The gripper then makes the following movements:

1) forward, until the gripper contacts the door, as sensed by the finger tips. This cancels out depth errors.
2) left, until the gripper touches the handle, as sensed by controller error. This provides a target grasp point.

This machine haptics method is robust because the motion can tolerate up to 1.1 meters of error in the position of the handle, far higher than any observed error in the navigation system. Once the robot has found and grasped the handle, it moves its base in such a way that the gripper and door handle will be at a known location with respect to the base. This causes the robot to be positioned extremely accurately with respect to the door.

4) Door Pulling: As safety and low-power have become important design considerations for mobile manipulators, it is common for robot arms to be too weak to directly pull open heavy, spring-loaded doors.

We propose a spinning maneuver that makes use of the base to pull, prop, and finally push open the door. The sequence is described below and illustrated in figure 4:

1) With the handle grasped (in our case, with the right arm), the robot spins 160° (in our case, clockwise) while making a slight movement toward the door hinge (right). When the robot arm becomes “wrapped” around the robot’s body it forces the door open to an angle of approximately 45°.
2) The robot moves its base to press against the edge of the open door, propping it open.
3) The robot releases the handle and continues to rotate (clockwise) to 270° from the original orientation, with its back against the open door.
4) The robot drives sideways through the open door.

In comparison to the approach used in [3], this sequence has the added benefit of requiring only one arm, making it useful on a wider variety of lower-cost robots.

C. Elevator Management

In the scheme of multi-map navigation, elevators represent a many-to-many kind of transition. The robot must operate the elevator while ensuring it exits on the correct floor. There are four main phases to navigating the elevator:

1) Detection and operation of the elevator call buttons.
2) Entering the elevator.
3) Detection and operation of the control buttons.
4) Exit from the elevator on the correct floor.

We note that, due to concerns for the safety of humans and of the robot, we decided that we would not allow the robot to ride an elevator with humans.

Calling the elevator. To begin the elevator sequence, the robot navigates to the approximate (known) map location

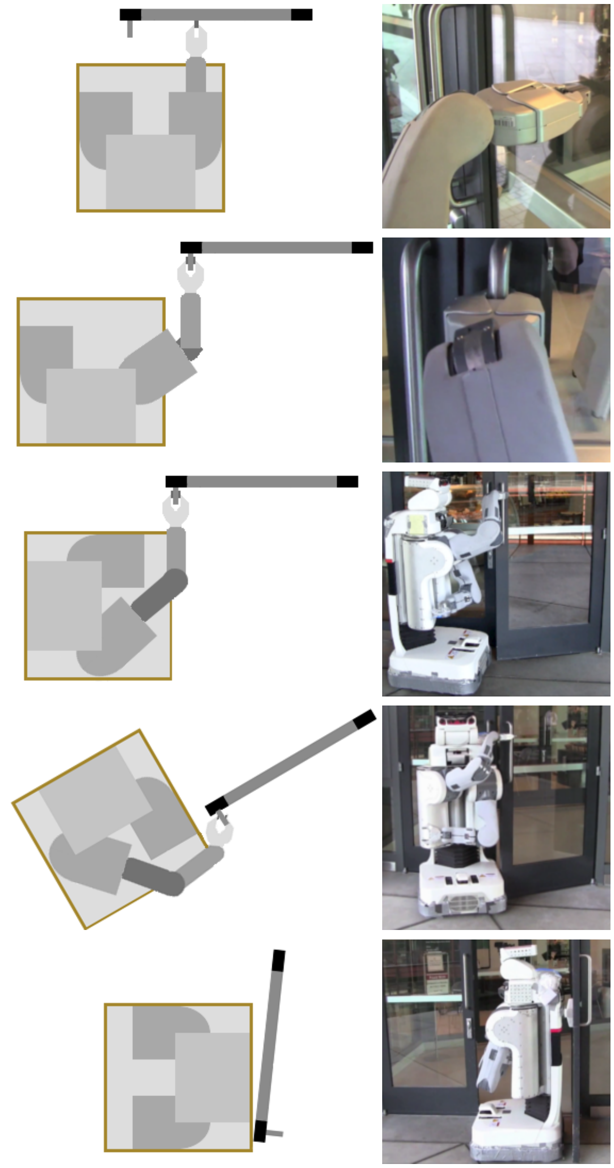

Fig. 4. The robot (a) lining up with the door, (b) in position to open the door, (c) beginning a clock-wise rotation to pull open the door, (d) wedged on the partly-open door and releasing the handle, (e) opening the door the rest of the way by continuing to spin and back into the door. From this pose the robot can drive sideways into the opening.

of the elevator call buttons and uses data from the laser scanner to align with the wall. The robot extends its forearm and performs a sweeping motion with the arm to survey the wall with the forearm camera. The buttons are localized in the camera images using OpenCV’s HoughCircles() function. Finally, the desired button is pressed.

Entering the elevator. When the elevator arrives, the robot performs some checks to ensure that it is appropriate to enter. The lights by the door are checked to see whether the elevator is going up or down; since our instance of the problem involved two elevators side by side, it was important to note which lights belonged to which elevator. Once a suitable open elevator is detected, the robot attempts to navigate into it while scanning for obstacles along the way. The most

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

Fig. 5. (a) Pushing the elevator call button, (b) entering the elevator, (c) selecting a floor to go to, and (d) identifying the current floor by equalizing and thresholding a segment of the camera image.

common obstacle is humans entering or exiting the elevator. If the obstacles do not clear within 5 seconds the robot returns to the starting location and calls the elevator again. If the robot reaches the elevator door frame, it ensures that there are no people or objects inside the elevator using data from its laser scanner. If the elevator is not empty then the robot backs out and starts the process all over again. While it may take several attempts, these safeguards ensure that the robot will only enter a vacant elevator.

Operating the elevator control buttons. Due to the high reflectivity of the interior of the elevator, we once again opted for mechanical alignment over laser and vision techniques to improve reliability. By gently driving against a wall of the elevator and executing small rotations the robot becomes aligned, and localization inside the elevator can be done using a combination of wheel odometry and sensed contact with the walls. Elevator button operation is thus done purely mechanically and with high reliability.

Exit from the elevator on the correct floor. The robot uses its laser scanner to monitor the door. If the door does not open for more than 30 seconds, the robot tries to push the control buttons again. When the door opens, the robot cannot assume that it has arrived on the correct floor because the elevator may have been called by humans at another floor. Therefore, the robot moves into the door frame and uses its camera to inspect the sign indicating the floor number, which is highly visible and easily readable by the robot. OpenCV functionality is used to equalize and threshold the image, and template matching is used to recognize the signs from each floor. Based on the recognition result the robot either retreats into the elevator to wait for the next floor, or continues exiting and considers elevator navigation a success. The entire elevator sequence is illustrated in figure 5

D. *Object Passing*

To enable the actual task of purchasing a beverage, we created a system for allowing the robot to pass objects to and from humans (figure 6(a)). The goal was to make interaction with the robot intuitive enough that untrained personnel could interact with it successfully.

A fairly common approach, and one we tried initially, involves using accelerometer or controller (joint) error to detect when a human is passing or taking an object from the robot. This was sufficiently reliable for the case of taking objects from the robot, since the human would deflect the arm by pulling on the object until the robot released it. However, humans tend to give objects much more gently, often avoiding direct contact with the gripper. Since it is not intuitive for humans to push objects forcefully into the hand of another human, this interaction with the robot was frustrating.

(a) (b)

Fig. 6. (a) The robot takes the beverage from the coffee shop employee, and (b) stashes the cup in its shoulder-mounted cupholder. The plastic bag around the cup has been omitted in (b) for clarity.

To solve this we used the robot's forearm camera to detect when objects were inserted into the gripper. An initial image of the scene was taken when the gripper was first readied to grab an object, and this image was compared against subsequent images. Once a significant difference was reached between the two images, the robot attempted to grasp the object. While some false positives were evident, overall reliability was not impacted because the robot would simply attempt a re-grasp if it detected that the gripper closed with no object in it.

Since the robot is not able to navigate well while grasping a bulky object, a simple cupholder was designed to facilitate the transportation task (figure 6(b)).

V. INTEGRATED EXPERIMENTAL RESULTS

Our primary experimental goal was to verify that the robot could robustly perform its tasks. We demonstrated the

Task	Time (s)	Iterations	Average Time (s)
Driving	1165	N/A	N/A
Taking Objects	75	3	25
Giving Out Objects	85	3	28
Waiting for Humans	222	N/A	N/A
Alignment for Door Pushing	146	5	29
Door Pushing	120	5	24
Alignment for Door Pulling	332	5	66
Door Pulling	243	5	49
Pushing the Elevator Call Buttons	184	4	46
Entering the Elevator	41	2	20
Pushing the Elevator Control Buttons	91	2	45
Exiting the Elevator	40	2	20
Waiting for the Elevator	91	N/A	N/A

Fig. 7. Break down of the time spent on one run which took a total 2835 seconds. Note that the elevator call buttons had to be pressed 3 times because the elevator was busy on other floors and timed out.

system's reliability by achieving three successful integration tests. A breakdown of the time taken for each component on one of the three integration tests is shown in figure 7. The robot achieved its goals despite the large number of unpredicted obstacles and the failures of individual components of the system. This demonstrates, in accordance with [8], how individually unreliable components can be integrated for high overall reliability through retry logic.

The system avoided a variety of obstacles and failure conditions during the testing. A common case was the presence of people and carts in the environment. The robot successfully avoided these obstacles by either driving around them or by waiting for them to clear. A second issue occurred when humans called the elevator to another floor before the robot could successfully press the control button. The robot avoided this problem by detecting and rejecting incorrect floors during the exit phase of the elevator navigation process. A third common issue was failure to push the buttons inside or outside the elevator. Recovery was achieved because the robot re-ran these processes when the elevator did not act as expected within 30 seconds.

In addition to the failures of the robot described above, the robot avoided other cases caused by humans and environmental issues. One failure mode was that the coffee shop employee did not fully push the beverage cup into the robot's hand. This cause the robot's hand to slip off the beverage cup as the robot attempted to pull the cup back from the human. However, the robot detected that no object was present in its gripper and asked the human to try again. Another challenge arose from a case where a human pushed a door open and then released it while the robot was lining up to navigate the door. The robot avoided this problem by waiting for sensor readings to stabilize, given its quasi-static approach to navigating the world.

VI. CONCLUSIONS

We demonstrated a robot capable of reliably delivering beverages to humans in an unaltered, challenging human environment with heavy, spring-loaded doors, multiple floors, and elevators. The robot received no assistance from humans outside of the interactions with the requester and the coffee shop employees. Our system suggests the ability of a personal robotics platform to accomplish a wide variety of tasks without significant physical modification. The results of our tests support previous work by finding that a robotic system can be reliable as a whole despite the failures of individual components and the unstructured nature of the environment.

All of the code developed for this project is freely available at http://code.google.com/p/jks-ros-pkg. In addition, we have released the multi-map navigation software for others to use with the PR2 and other robots running ROS.

ACKNOWLEDGMENTS

We would like to thank Guity Elahi, Victoria Quivoz, Adriana Pedroza and Leticia Cunningham of Peet's Coffee in the Clark Center for allowing the robot to enter their store and for their patience with the robot.

REFERENCES

[1] S. Chitta, B. Cohen, and M. Likhachev. Planning for autonomous door opening with a mobile manipulator. Anchorage, Alaska, 2010. International Conference on Robotics and Automation.

[2] A. Edsinger and C. Kemp. Human-robot interaction for cooperative manipulation: Handing objects to one another. In *Robot and Human interactive Communication, 2007. RO-MAN 2007. The 16th IEEE International Symposium on Robots and Human Interactive Communications*, pages 1167 –1172, aug. 2007.

[3] S. Gray, C. Clingerman, S. Chitta, V. Kumar, and M. Likhachev. Search-based planning for autonomous spring-loaded door opening. Sydney, Australia, 2012. RSS Mobile Manipulation Workshop.

[4] H. C. K. Okada, M. Saito and M. Inaba. Towards real robot service application that we can use everyday. San Francisco, USA, 2011. IEEE/RSJ International Conference on Intelligent Robots and Systems.

[5] E. Klingbeil, B. Carpenter, O. Russakovsky, and A. Ng. Autonomous operation of novel elevators for robot navigation. Anchorage, Alaska, 2010. International Conference on Robotics and Automation.

[6] E. Klingbeil, A. Saxena, and A. Y. Ng. Learning to open new doors. Zurich, Switzerland, 2008. RSS Mobile Manipulation Workshop.

[7] K. L. Koay, E. A. Sisbot, D. S. Syrdal, M. L. Walters, K. Dautenhahn, and R. Alami. Exploratory study of a robot approaching a person in the context of handing over an object. In *In Proceedings of the AAAI Spring Symposium on Multidisciplinary Collaboration for Socially Assistive Robotics*, page 2007.

[8] W. Meeussen, M. Wise, S. Glaser, S. Chitta, C. McGann, P. Mihelich, E. Marder-Eppstein, M. Muja, V. Eruhimov, T. Foote, J. Hsu, R. B. Rusu, B. Marthi, G. Bradski, K. Konolige, B. P. Gerkey, and E. Berger. Autonomous door opening and plugging in with a personal robot. International Conference on Robotics and Automation, 2010.

[9] J. Miura, K. Iwase, and Y. Shirai. Interactive teaching of a mobile robot. Barcelona, Spain, 2005. International Conference on Robotics and Automation.