# Unified Human and Robot Command for Disaster Recovery Situations

Carlos Ibarra Lopez, James Kuczynski, Holly A. Yanco
University of Massachusetts Lowell
Department of Computer Science
{clopez, jkuczyns, holly}@cs.uml.edu

*Abstract*— **The system described herein was designed to allow sending commands to and obtaining location data from humans in a manner similar to robots, thus simplifying the command of combined human/robot teams. We designed and implemented a system that allows humans to be interfaced with a robot framework (ROS, or Robot Operating System), providing sensor information and waypoint based navigation for a team of people. An experiment is also described in which participants were asked to drive a team of units, first described as all robots, then described as both robots and humans. We discovered that participants have a similar level of success when using such a system to control units described as robots as when they use it to control units described as both humans and robots.**

## I. INTRODUCTION

Robots have been used in disaster recovery situations for over a decade (e.g., the World Trade Center in 2001; in Biloxi, MS, after Hurricane Katrina in 2005). One major limitation of the current generation of these systems is the difficulty of information sharing – between first responders in the field, between the field and command, etc. [1], [2] – particularly as we increase the amount of available digital data from satellites, robots, handheld sensors, and many other sources. Responders on the ground are not as well connected to their command and control centers, to each other, or to the available data as they could be. There is a need to improve disaster response through more effective information sharing, a problem that we propose to solve with the use of Google Glass and Project Tango.

While some existing systems try to unify human and robot command by making robot command similar to human command, using methods such as natural language processing [3], [4], we propose that for disaster recovery situations, the opposite can be more accurate, as we can take advantage of the precision that robot command interfaces provide.

This paper describes a set of instruments that, when carried by a human user, provide him or her with sensors usually associated with robots, specifically data from inertial measurement sensors and 3D imagery in the form of a point cloud. When combined with an interface that can translate movement and other commands to make them understandable for the user, our system allows humans to be agents that can utilize many of the utilities originally developed for use with robots, including but not limited to, multi-agent command software, of which we provide an example in the form of a touch screen interface for simultaneous human and robot command that aims to solve the problem mentioned above, by allowing humans and robots to be commanded using the same interface.

Similar systems have been designed to localize humans using foot mounted sensors [5] and sensors normally used in robots, including laser range finders [6], our system differs in that it uses an integrated, comercially available solution for sensing.

We also describe an experiment that we conducted in which users were asked to command two groups, one consisting of all robots, and another consisting of both robots and humans, in order to better understand how a user might behave differently when commanding humans, which will allow us to build better user interfaces for commanding combined human/robot teams.

## II. SYSTEM

The goal of our system is to simplify combined human/robot team command, by outfitting humans with sensors that are commonly available in robots, providing them with a way to receive commands in a manner analogous to how robots receive them, and providing a unified command system for the base station, allowing for seamless command of human and robot agents. This system is divided in two parts, a field user component, used by the human agents that are part of a joint human robot team, and a base station user interface, used by the commander.

### A. Hardware

It is desirable that the field user has some sensor input of his or her surroundings, and an ability to estimate location. It is also necessary to keep equipment light and easy to carry, since a focus use case for our system is first responder teams, who need to be able to move unobstructed and are already carrying a considerable amount of equipment. To fulfill these requirements, we selected a Google Project Tango tablet, which provides a 3D image sensor and position estimation, and a pair of Google Glass, allowing command outputs to be visualized by the agent. An image of a user wearing the equipment is shown in figure 1. For the base station user interface, a computer with a touch screen is used.

### B. Software

*1) ROS interfacing:* We decided to make our software compatible with ROS [7], an open source framework designed for robot intercommunication, for passing information between our different components. This has the advantage of

Fig. 1. Field User Wearing Google Project Tango Tablet and Google Glass.

making our software directly compatible with a large library of software, including tools for mapping and robot control.

*2) Project Tango Sensor Data:* As mentioned above, the Tango captures 3D point clouds, which are useful for tasks such as mapping or path planning, and estimates its position in space. We wrote software that enables this data to be transmitted in a format that is compatible with other software running ROS, an example being our base station user interface.

*3) Glass User Interface:* Glass runs a program which receives desired linear and angular velocities, and shows movement instructions as arrows on its screen or navigation voice prompts (i.e. spoken messages with instructions such as "walk forward" or "turn right"). This software can also receive text input, displaying the text on screen for any received messages, allowing for non-movement commands to be sent.

*4) Audio Transmission and Reception:* An audio transmission system was developed, through which users wearing Glass headsets can talk with each other, either in private conversations or multiple user groups, optionally including a field commander at an OCU. This allows Glass to replace a regular radio and reduce the amount of equipment a user has to carry.

*5) Robot Point of View Visualization:* An additional capability developed for Glass is a video visualization tool that allows a wearer to receive video from a robot in the Glass screen, and if the robot is equipped with a pan/tilt unit, it allows for pan/tilt control in a natural way using head movements. This would be useful in case a user needs to evaluate whether it is worth visiting a room in which a robot is located (e.g., to look for people to rescue or dangerous situations). It is also possible to broadcast video from a Glass headset back to the base station or to another Glass headset, in order to assist in information sharing.

Fig. 2. Screenshot of command system displaying a human agent alongside a robot on a section of a map.

*6) Navigation Setup:* Path planning is required for a robot using any level of control besides full teleoperation; given some coordinates, a path planning system controls the robot speed and direction to reach them. For a human agent, path planning might not appear to be necessary, as a human has the capability of avoiding obstacles upon seeing them. Nevertheless, path planning is useful as it is desirable to be able to command humans by giving them waypoints to which to go next. For this purpose, the navigation modules on ROS were configured for a human agent.

Using the software described in the previous sections, the human agents are now equipped with sensors, and interfaced into ROS, enabling them to be interfaced with available software designed to be used with robots with minimal changes.

### *C. Mission Control: Touch Interface for Commanding a Human/Robot Team*

The next element of our system consists of an integrated base station command interface for commanding a combined human/robot team.

It was decided to expand upon the software described and evaluated in [8] for multi-touch based multi-robot control, as many of the requirements are already satisfied, if only for robots. The robot position is shown on the map as part of the interface, waypoints for movement may be sent to a group of robots, and manual control is provided via the DREAM controller [8], a virtual joystick displayed on screen. In addition to technical modifications to make this existing software compatible with our system, the following changes were done:

- *Communicating with different kinds of agents:* By design, the original command software communicates with a specific kind of robot, and in our case it was desired to use at least two different kinds of agents (humans and robots), and ideally for it to be able to accommodate as many different types of agents as a situation requires. For this, changes were made in the software in order to accommodate control of different forms of agents,

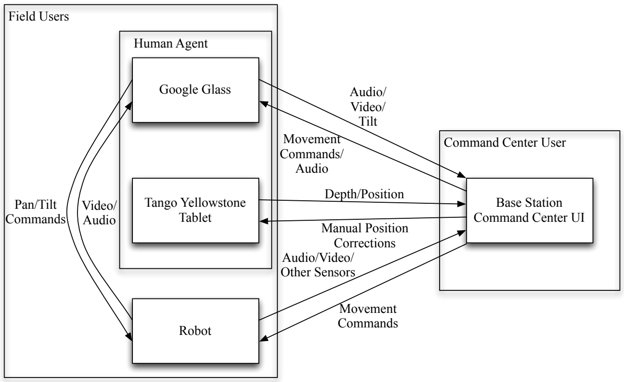

Fig. 3. Software Architecture Diagram. Communications use ROS as a backend, audio additionally uses a ROS controlled SIP server.

which mostly consisted on providing the capability of subscribing and publishing to differently named topics for different types of agents.

- *Appearance:* Given that different kinds of robots, as well as humans, will now be commanded through the interface, changes were made in so that each appears with different icons, allowing the person using the command interface to distinguish if an object is a person or a robot in the field, and, if a robot, which type of robot. Such distinctions could also be made for people in different roles, whether a command structure or by agency.
- *Audio Communications:* In addition to the command capabilities previously mentioned, an audio client was developed to allow voice communication between mobile agents - whether those agents are robots or humans. Base station users are also able to use this as a means of opening a communication channel with a single agent, with a group of agents, or by broadcasting to all agents.

The mission control software provides the capability to command robots and humans simultaneously through touch gestures by selecting either or both, and giving them waypoints to which to navigate, the touch interface can be seen in figure 2. A diagram showing how different parts of the system interact with each other is shown in figure 3

## III. EXPERIMENT

We designed our experiment based on the multi-robot user interface described previously. Sixteen participants were asked to do two runs, completing the tasks described below, on two different maps. On the first run, participants were told they were commanding six robots, and the user interface depicted all icons as robots, participants were shown an image of a ActivMedia Pioneer robot and told that would be the type of robots they are commanding. On the second run participants were told they were commanding three robots and three humans, robots were still described as ActivMedia Pioneers and depicted as such. Humans were described to the participants as wearing odometry sensors in the form of a Google Project Tango Yellowstone tablet, and receiving commands via a wearable display showing directions. Participants were shown an image of a supposed human they were sending commands to, a person wearing a Project Tango in a chest holster and a pair of Google Glass, similar to figure 1. For the purposes of this study, the back end was all run in simulation; both units, the ones described as humans and the ones described as robots were simulated robots, with the same capabilities. Simulated robots and humans were set to have periodic navigation failures (unit forgets all pending goal waypoints, and stops moving), in the interest of recording user attempts at recovering. Additionally, since the unit movement and sensing had simulated noise and blind spots, units at some times experienced failure at navigating a set path, or became stuck. Since sections of the map were hidden prior to exploration, a unit could also fail at navigating when users gave it waypoints to an inaccessible area of the map, or outside of the map.

Participants were told they would be using a multitouch screen interface to either control robots or send commands to human agents. Participants were also told they can select either individual or multiple units, and send either a single goal, or a collection of waypoints that the unit(s) would execute in sequence. They also had the option of overriding this path planning by selecting a single unit and setting it on manual control mode, then using the DREAM controller [8]. The situation was described as a test course representing a search and rescue situation, in which their main goal was to cover as much area as possible, using either the robots or human agents.

Data recorded from the experiment included when a user sent waypoints to robots, humans, or both, when a user used manual control on a robot unit, and when they used manual control on a human unit, separating cases in which new waypoints or manual control were used to recover from a failure. For each time waypoints were sent to units, we recorded the number of waypoints sent.

## IV. EXPERIMENT RESULTS

We recorded data from the experiment using a combination of automatic reporting software from the user interface and video data in the form of screen captures. For the data that was obtained by coding events from the screen captures, we calculated intercoder agreement using Cohen's Kappa. Events being coded were: User selected a unit, User deselected a unit, and User Started manual control. A distinction was made as to whether the unit was a stopped robot, a moving robot, a stopped human or a moving human. Additionally, we coded when a user set a waypoint, when a user started waypoints, when a robot failed at navigation and when a human failed at navigation. Our first kappa was $\kappa = 0.83$ ($\kappa = 0.59$ excluding chance); we discovered that there was significant disagreement in the categories for navigation failure. We decided to code the category of navigation failure again, redefining what constituted a navigation failure, using a more specific definition; coding was performed on a separate video of a different run. After separating the coding for navigation failures, our Kappa value for the coding of all categories except navigation failures was $\kappa = 0.85$ ($\kappa = 0.62$ excluding chance); and our kappa for navigation failures was $\kappa = 0.81$ ($\kappa = 0.64$ excluding chance). According to [9], a Kappa value in the range of 0.61-0.80 indicates substantial agreement among coders. Coding for grouping and reacting to navigation failures was expressed as a combination of the events coded in the Kappa, which is why those events were not coded separately.

### A. *Success*

We measured success as how thoroughly the user completed the assigned task (exploring the map), as a percentage of a completely explored map. We recorded this for every user in both the run using only robots and the run using robots and humans.

We then compared each of the participants' second run against their first one, to see how their performance increased or decreased. We also recorded the time each user took on the task, again comparing the second run against the first one. Results are shown on Table I. A positive number on explored percentage means the participant explored more on second run. Similarly, a positive time means the participant took more time on the second run. Twelve participants had approximately the same amount of coverage in both runs (within 2 percentage points). The mean of the differences between runs was $-1.79$ ($\sigma = 12.41$). Of the 4 participants with significant differences between runs, 2 covered more area on the second run and two covered less area.

One of the participants who covered less area (P9) left several rooms unexplored after visiting them with a human. A hypothesis is that some users might take for granted that a human would be able to see the whole room (even when it was told to users that humans would have the same exploration capabilities as robots for the purpose of this experiment). However since the sample size for users who differed in the runs is small, additional testing would be required to reach a conclusion about this. The other participant who covered less area in the second run (P15), did not seem to exhibit this behavior; instead this participant missed a large area in one edge of the map.

Participants 11 and 16, the cases where there was better performance on the second run, did not exhibit a different exploration pattern on the second run. Even though these participants' coverages were better on the second run, they still left a large area unexplored.

We also compared time results from the first run with the second. Most participants had a shorter run on the second, except for P6 and P14. The shorter runtime was likely caused from having previous experience with the interface from the first run.

## V. GROUPING

We recorded the amount of times participants selected a group of units (i.e. when a participant sent commands to two or more units at the same time). On the second run we divided groups in "Human Only", "Robots Only", and "Mixed". In general, grouping was rare; users seemed to prefer commanding units individually more. However out of the cases where there was grouping, mixed unit groups were more common than groups of the same unit, with human only groups being particularly rare (only P16 did a human only group, and only twice). This would indicate that users do not have a problem with selecting human and robot units together when needed.

TABLE I

COMPARISON OF PARTICIPANTS' RESULTS IN SECOND RUN AGAINST FIRST RUN. SECOND COLUMN SHOWS THE TIME DIFFERENCE (RUN TIME IN SECOND RUN - RUN TIME IN FIRST RUN), THIRD COLUMN SHOWS THE DIFFERENCE OF EXPLORED AREA (PERCENTAGE EXPLORED IN SECOND RUN - PERCENTAGE EXPLORED IN FIRST RUN)

| Participant | Time R2 - R1 | Explored R2-R1 |
|---|---:|---:|
| 1 | -02:33 | 0 |
| 2 | -09:57 | -0.2 |
| 3 | -01:34 | 0.6 |
| 4 | -08:12 | -0.1 |
| 5 | -19:58 | -0.2 |
| 6 | 04:07 | 0.8 |
| 7 | -00:25 | -0.1 |
| 8 | -10:47 | -0.2 |
| 9 | -04:44 | -39 |
| 10 | -00:07 | 0.5 |
| 11 | -07:26 | 16.15 |
| 12 | -03:49 | 2 |
| 13 | -00:30 | 0 |
| 14 | 03:31 | -0.1 |
| 15 | -04:59 | -23 |
| 16 | -08:48 | 14.14 |
| Mean | -04:46 | -1.79 |
| Std. Dev. | 05:52 | 12.41 |

### A. *Manual Control*

We recorded how many times a user entered manual control mode on a unit. On the second run, we separated our records by humans and robots. On the second run, 7 participants used manual control more on humans than robots, 5 used it more times on robots than humans, 2 used it the same amount of times, and one did not use manual control at all. Participants who used manual mode more on the first run seemed more likely to use it on the second run. We initially thought manual control would not be used as much on humans, since we suspected it was not a natural way to send commands to a person. However, all of the participants who used manual control on the second run used it at least once on a human.

### B. *Open Question*

Participants were asked "What additional features would you have preferred for commanding humans?" Five participants did not reply to the question, or replied with a non-suggestion (e.g. None, I'm not sure). Four participants left suggestions for the interface in general, including having the manual control joystick be operable with the right hand, a request for more simple controls and a faster refresh rate, more responsiveness in the joystick and the ability to drag waypoints to change a path without completely cancelling it. Three other participants wrote problems they perceived when operating the UI (e.g. Sometimes robots were not following commands, I had trouble getting robots unstuck with manual mode). The other 4 participants left the following suggestions for features when commanding humans:

- “P2: Humans don’t need to hit every inch of the area because we can see further than the robots sensors. The humans should be knowing where walls are (sic).” This comment seems to reinforce what we hypothesized about users believing humans have better sight than robots. In general this would be true, however if the task requires exploration closer than what a human can see (for example, if using a human mounted sensor to get a reliable map or a human navigating on an area where visibility is limited), the commander must be taught to not rely on what he/she assumes a human can normally see.
- “P3: Humans seemed to get stuck more easily than robots. Maybe some kind of callback system to just have them turn around. Also maybe a pattern for them like back and forth so you don’t have to micromanage them.” Units representing humans were configured exactly the same as robots for this experiment. This participant might have been more aggressive with human commands hence the participant saw more navigation failures. His second comment about having a pattern control for humans could be implemented by using a wandering algorithm.
- “P4: Giving the humans more autonomy would be helpful.” Again, commanders expect humans to be able to use their abilities (e.g. better vision, autonomy), so in situations where this is not possible, the commander must know why.
- “P9: I would say that humans unlike robots have a full range of motion that isn’t limited to X and Y position. For Example a robot cannot turn while moving, but a human can. If there were any ways to give humans a more extensive range of motion, I think that could help a lot.” On a real system, as opposed to simulation, this would have been visible, as the interface shows a unit’s current position, so the human agent on the field would be able to move as he/she finds necessary, and the interface would reflect the new position.

## VI. PARTICIPANT SURVEY

Participants were asked to fill out a survey about their experience using the software once they were done with both tasks. They rated their performance, stress level, mental demand and frustration using the interface on a 7 point Likert scale, where 1 was Very Low and 7 Very High. The results are in Table II. Means were 5.68 for Performance, 2.75 for Stress, 3.62 for Mental Demand and 2.87 for Frustration.

Participants who had a low performance on the exploration task (Percentage explored $< 70\%$ in either of the runs, P9, P11 and P15) rated their performance as 5, 6 and 6. This indicates participants were not aware of their low performance on the task.

TABLE II

Participants’ experience with the user interface (1 is Very Low, 7 is Very High)

| Participant | Performance | Stress | Mental Demand | Frustration |
|---|---|---|---|---|
| 1 | 6 | 2 | 4 | 3 |
| 2 | 7 | 1 | 3 | 1 |
| 3 | 5 | 2 | 5 | 3 |
| 4 | 6 | 2 | 2 | 4 |
| 5 | 5 | 2 | 4 | 2 |
| 6 | 5 | 2 | 2 | 3 |
| 7 | 6 | 3 | 4 | 2 |
| 8 | 7 | 2 | 2 | 3 |
| 9 | 5 | 2 | 2 | 1 |
| 10 | 6 | 5 | 5 | 7 |
| 11 | 6 | 1 | 1 | 2 |
| 12 | 6 | 4 | 5 | 2 |
| 13 | 6 | 6 | 6 | 4 |
| 14 | 6 | 3 | 4 | 1 |
| 15 | 6 | 2 | 4 | 3 |
| 16 | 3 | 5 | 5 | 5 |
| Mean | 5.69 | 2.75 | 3.62 | 2.87 |
| Std. Dev. | 0.92 | 1.44 | 1.41 | 1.54 |

### A. *Effects of previous experience with robots*

On our demographic survey, out of the 16 participants, 7 (P1, P3, P4, P6, P10, P13, P16) self reported on the lower half of the Likert scale (1-3) for “I have experience operating robots”, 2 (P7, P12) reported the middle option (4), and 7 (P2, P5, P8, P9, P11, P14, P15) self reported on the upper half. We analyzed the differences in control behaviors for the upper and lower half. Manual control data grouped by robot experience is shown on tables III and IV. We discovered a difference in the percentage of times a participant used manual control on human agents.

For analysis of manual control differences, we removed P13 from the dataset, as that participant did not use manual control at all on the second run. We analyzed the data by calculating the percentage of times when a user activated manual control on a human agent in the second run, out of the total times the user activated manual control on the second run. Participants who had less experience with robots used manual control more on human agents than on robots, while participants who had more experience with robots used manual control more on robot units. To determine statistical significance of the data, we did a one tailed, two sample equal variance T-test on the results. Our null hypothesis was $h_0$ = “A user’s past experience operating robots has no effect on whether a user activates manual control more on human units or robots”. We found a significant difference in the percentage of times manual control was used on human agents by users with low experience with robots (Mean $= 74.24$, $\sigma = 22.43$) when compared to users that had more experience with robots (Mean $= 41.68$, $\sigma = 15.18$), $p = 0.005$.

## VII. CONCLUSIONS

Most participants had a similar level of success when they were told units were both humans and robots as they did when they were told all units were robots. This leads us

TABLE III

PERCENTAGE OF TIMES MANUAL CONTROL WAS ACTIVATED ON HUMANS BY USERS WITH LOW PREVIOUS EXPERIENCE WITH ROBOTS

| Participant | Percentage Manual Humans |
|---|---|
| 2 | 50.00 |
| 5 | 33.33 |
| 8 | 66.67 |
| 9 | 33.33 |
| 11 | 20.00 |
| 14 | 38.46 |
| 15 | 50.00 |
| Mean | 41.68 |
| Std. Dev. | 15.18 |

TABLE IV

PERCENTAGE OF TIMES MANUAL CONTROL WAS ACTIVATED ON HUMANS BY USERS WITH HIGH PREVIOUS EXPERIENCE WITH ROBOTS

| Participant | Percentage Manual Humans |
|---|---|
| 1 | 80.00 |
| 3 | 100.00 |
| 4 | 80.00 |
| 6 | 83.33 |
| 10 | 68.75 |
| 16 | 33.33 |
| Mean | 74.24 |
| Std. Dev. | 22.43 |

to believe such a system would be an effective method to command combined human/robot teams.

When asked which features they would want added to the system to control humans, some participants suggested adding features that take advantage of abilities humans have and robots don't (such as better depth perception capabilities). While we believe an existing robot command system is a good starting point for a team that contains humans, adaptations need to be made in order to better fit human abilities.

## VIII. FUTURE WORK

### A. Project Tango Application Improvements

We plan to implement several improvements to the application developed for the tablet, including:

- Colorized point clouds: Using the information available about position and optics of the color camera in the Tango to colorize the 3D point cloud provided. This would aid in visualization and point cloud processing.
- 3D Mapping: We plan to connect the Tango with a 3D mapping system, to allow multiple Tangos to collaboratively create a 3D map of an area. We believe this would help agents to share information about the current state of an area, which is particularly useful in disaster recovery, since an area might be altered and no longer match its description in existing maps. Our candidate for implementing 3D mapping is Octomap [10], given its efficiency and the fact that maps' resolutions can be easily altered, allowing maps to be downsized when they need to be shared over a network.

### B. Additional User Testing

The completed system would benefit from user testing on the field user interface. A future experiment could include having participants wearing the Google Glass while being asked to navigate through a staged disaster scenario and while following instructions given on the display. This would allow us to find out which is the best method to display those instructions, whether it is an arrow system, a voice guidance system, or displaying a zoomed in section of the map with the path drawn on it. This would also allow us to test for different behaviors with different interfaces (e.g., a user might be more successful in reaching the final destination with the zoomed in map, but they might be more compliant with the specific instructions from the base when using the arrows). Running an experiment like this would allow us to build a more robust system and incorporate improvements discovered during such a study. Additionally, more focused experiments can be done on the commander user interface to further validate the data obtained in this experiment.

## IX. ACKNOWLEDGMENTS

This work has been supported in part by Google. Carlos Ibarra Lopez was supported in part by a CONACyT Mexico Graduate Scholarship.

## REFERENCES

[1] M. Micire and H. Yanco, "Improving disaster response with multi-touch technologies," in *Intelligent Robots and Systems, 2007. IROS 2007. IEEE/RSJ International Conference on*, Oct 2007, pp. 2567–2568.

[2] B. Manoj and A. H. Baker, "Communication challenges in emergency response," *Commun. ACM*, vol. 50, no. 3, pp. 51–53, Mar. 2007. [Online]. Available: http://doi.acm.org/10.1145/1226736.1226765

[3] D. J. Brooks, C. Lignos, C. Finucane, M. S. Medvedev, I. Perera, V. Raman, H. Kress-Gazit, M. Marcus, and H. A. Yanco, "Make it so: Continuous, flexible natural language interaction with an autonomous robot," in *Grounding Language for Physical Systems Workshop at the AAAI Conference on Artificial Intelligence*, 2012.

[4] R. C. Simpson and S. P. Levine, "Voice control of a powered wheelchair," *IEEE Transactions on Neural Systems and Rehabilitation Engineering*, vol. 10, no. 2, pp. 122–125, June 2002.

[5] R. Stirling, K. Fyfe, and G. Lachapelle, "Evaluation of a new method of heading estimation for pedestrian dead reckoning using shoe mounted sensors," *Journal of Navigation*, vol. 58, no. 1, pp. 31–45, 01 2005.

[6] M. Baglietto, A. Sgorbissa, D. Verda, and R. Zaccaria, "Human navigation and mapping with a 6 dof imu and a laser scanner," *Robotics and Autonomous Systems*, vol. 59, no. 12, pp. 1060 – 1069, 2011. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S0921889011001540

[7] M. Quigley, K. Conley, B. P. Gerkey, J. Faust, T. Foote, J. Leibs, R. Wheeler, and A. Y. Ng, "Ros: an open-source robot operating system," in *ICRA Workshop on Open Source Software*, 2009.

[8] M. J. Micire, "Multi-touch interaction for robot command and control," Ph.D. dissertation, University of Massachusetts Lowell, Lowell, MA, 2010.

[9] R. Landis and G. Koch, "The measurement of observer agreement for categorical data," *Biometrics*, vol. 33, no. 1, pp. 159–174, 1977. [Online]. Available: http://www.jstor.org/stable/2529310

[10] A. Hornung, K. M. Wurm, M. Bennewitz, C. Stachniss, and W. Burgard, "OctoMap: An efficient probabilistic 3D mapping framework based on octrees," *Autonomous Robots*, 2013, software available at http://octomap.github.com. [Online]. Available: http://octomap.github.com