SIMULTANEOUS HUMAN-ROBOT COMMAND AND LOCALIZATION

BY

CARLOS J.R. IBARRA LOPEZ
B.S. UNIVERSIDAD DE SONORA (2014)

SUBMITTED IN PARTIAL FULFILLMENT OF THE REQUIREMENTS
FOR THE DEGREE OF MASTER OF SCIENCE
DEPARTMENT OF COMPUTER SCIENCE
UNIVERSITY OF MASSACHUSETTS LOWELL

Signature of Author: ______________________ Date: 12/15/2016

Signature of Thesis Supervisor: ______________________
Holly Yanco, Ph.D.

Signatures of Other Thesis Committee Members: ______________________
Jill Drury, Sc.D.

SIMULTANEOUS HUMAN-ROBOT COMMAND AND LOCALIZATION

BY

CARLOS J.R. IBARRA LOPEZ

ABSTRACT OF A THESIS SUBMITTED TO THE FACULTY OF THE
DEPARTMENT OF COMPUTER SCIENCE
IN PARTIAL FULFILLMENT OF THE REQUIREMENTS
FOR THE DEGREE OF
MASTER OF SCIENCE
UNIVERSITY OF MASSACHUSETTS LOWELL
2016

Thesis Supervisor: Holly Yanco, Ph.D.

Professor, Department of Computer Science

Humans interact with robots in many scenarios including disaster recovery situations such as search and rescue; however people are not usually included in the command interfaces used for robots. A system was developed that augments a person with sensors commonly used in robots and provides a display to receive commands; this system was used to integrate humans into robot command interfaces, allowing for simplified control of combined human-robot teams. The design choices for this system, along with its implementation, are detailed and discussed. An experiment is then described in which participants are asked to complete exploration tasks, first using robots, then with a team consisting of humans and robots. We discovered participants had similar success levels on the task when commanding both people and robots, as they had when using only robots. Participants exhibited some other differences in their control behaviors depending on their previous experience level with robots.

Acknowledgments

I would like to thank Dr. Holly Yanco for being my advisor, for her patience and help in the writing of this thesis, and for providing an amazing place to work at the Robotics Lab. I'd also like to thank Dr. Jill Drury for being in my commitee, and for all her suggestions for this work.

Thanks to everyone in the Robotics Lab for all their reccomendations and help. In particular, I'd like to thank Jordan Allspaw and Eric Marcoux for their help on software development, especially in the first version of the Project Tango drivers and the audio integration, James Kuczynski, for the help during experiment runs and all the hours coding video, and Dan Brooks for taking the time to suggest last minute improvements for this work.

Thanks to my family, in particular to my mother for all her support along the way (and prior to the start of the journey).

Finally, thanks to my wife, Flor, for all her support during the years, and for not freaking out when I asked her to move with me to another country, 4500 km away from home, so I could go to grad school and play with robots.

This work was supported in part by Google and a CONACyT Mexico Graduate Scholarship (Scholarship No. 382799).

Contents

List of Figures

List of Tables

Chapter 1

Introduction

1.1 Problem Statement

Robots have been used in disaster recovery situations for over 15 years (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. (Micire and Yanco, 2007; Manoj and Baker, 2007) – 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.

When teams include both humans and robots, these problems are increased by the fact that communications differ between them. First responder teams usually communicate by radio (Manoj and Baker, 2007), while teleoperated robots are controlled using some form of Operator Control Unit (OCU). This adds a complication to commanding people and robots at the same time; even if both need to be directed to the same place, the commander needs to make two separate orders, one via the radio for the human and one via the OCU for the robot, which adds unnecessary delays and introduces the possibility of human error. While in some cases this could be mitigated by having separate commanders for humans and robots, this solution increases the number of responders required at the command center, and an additional trained person might not be available in all cases. This also requires collaboration between commanders, introducing another point where human error can occur.

There are a multitude of tools available for interacting with robots, particularly as part of ROS (Robot Operating System; *ros.org*), including, but not limited to, those for pose estimation, mapping, command, video transmission, and many others. While some of that software is meant to compensate for abilities that humans have innately and robots don't, such as obstacle avoidance and object identification, there is potential for humans to utilize some of it, particularly in the area of communication, team command, mapping, and navigation.

Yet, while many of those capabilities are as desirable for human teams as they are for robots, they are either not available or are only available as completely separate tools that do not interact with one another. We designed a system aided by 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 controlled using the same type of commands.

In cases where our system adds capabilities a human did not have before, such as localization, which allows the commander to see a representation of the agent on a map, there is clearly a benefit to the commander as compared to not having the ability to localize. However, in other cases, such as unit command, software designed to command robots might not translate directly to humans, as methods that work with robots might seem unnatural for humans. To improve our system by accounting for these differences in user behavior, we designed and ran an experiment that aims to identify which command methods are used differently when applied to humans as opposed to robots.

There are a wide variety of user interface guidelines that have been tested for commanding robots, including methods for providing better situation awareness to the operator (e.g., Scholtz et al., 2004; Yanco and Drury, 2004; Drury et al., 2007) and for allowing a single operator to control large amounts of robots (Micire et al., 2009), among others. Those interface design guidelines usually focus specifically on robot control; however in situations such as robot aided search and rescue, teams usually include humans as well as robots (e.g., Casper and Murphy, 2003). Our interest is to discover how a human operator's behaviors differ when using a robot control interface to send commands to humans, in order to adapt those guidelines and allow us to build a better interface that provides simultaneous human and robot command and localization capabilities.

1.2 Thesis Contributions

This work describes a system that was designed and implemented to allow us to integrate humans into a robot communication system, augmenting their capabilities with wearable sensors and a wearable display, permitting them to receive commands from an interface traditionally used to control robots, and to send sensor information back to the control station.

Implementing the system included developing software for a Google Project Tango Yellowstone tablet, interfacing it with ROS, allowing it to send its 3D depth sensor information (in form of a pointcloud) and its position information. This software was released as open source, and instructions for its use are provided with it, enabling anyone who has access to a Yellowstone tablet to integrate it with ROS, whether for the same purpose as ours, or for an alternate use (e.g., using it to easily outfit a robot with a self contained sensor suite).

We also implemented a Google Glass application that translates ROS movement commands to on screen directions or audio prompts. This application has also been released and is available for anyone who has access to a Glass device.

An existing multi-touch interface for multiple robot control (Micire, 2010) was modified to make it compatible with different types of units and fully linked with ROS. This interface was connected to our Yellowstone/Glass system. These modifications were necessary for our system to allow the interface to command both humans and robots; however, the updated interface is also useful for someone who intends to use it with multiple types of robots.

We ran an experiment with human subjects to determine how their behaviors would differ when they are aware they are controlling humans, as compared to when they are controlling multiple units of robots. Since the goal was to determine differences in commands as influenced only by the fact that some units were humans, the experiment was run using simulated units in which the only difference was the label. What we learned from such an experiment will be useful for us to improve the next iteration of our software, and is useful as a reference for designing a human command system, particularly if reusing an interface that was designed to command robots.

We used the information gathered from the experiment to create a set of design guidelines for interfaces that send commands to both robots and humans, which can be used to create better human command interfaces in the future.

Chapter 2

Related Work

This chapter reviews existing systems that perform tasks similar to our goals, including indoor localization systems, wearable sensors and display devices, and research involving people and robots colaborating towards a goal. Further, we describe how our system relates and builds upon existing technology and research.

2.1 Localization

Some research on localization in robotics can also be applied to people who are carrying or wearing sensors. In the case of outdoor localization, usually GPS can give a precise enough location that it can be reliably used for navigation; however for indoor navigation one cannot rely on GPS, as coverage is lost under cover. Several alternatives have been devised for indoor navigation, such as systems with markers in known positions which function as beacons for localization, including infrared beacons detected by a camera (e.g. Brassart et al., 2000) and radio frequency identification (RFID) tags detected by a sensor (e.g. Ni et al., 2004; Hightower et al., 2000; Yang et al., 2013). Alternatives to dedicated beacons have been developed, including systems that localize by triangulating among existing signals, including wireless network access points (e.g. Biswas and Veloso, 2010; Lim et al., 2007; Ferris et al., 2007), and cellular phone antennas (Roxin et al., 2007). In disaster recovery situations, systems that require beacons to be placed beforehand are not desirable, because the spaces being explored are usually unknown, and even if a building had beacons placed beforehand, they might have been moved or rendered non-operational. Localization using wireless network or cellular phone signals can also present a problem, as even in areas that are traditionally covered with those signals, they might not be operational after a disaster (e.g. due to loss of electric power or physical damage to transmitters).

In the case of mobile wheeled robots, localization as an estimation of movement relative to the starting location, known as odometry, can be obtained by using encoders in the wheels or derived from the speed that motors were set to run. These methods, commonly referred as dead reckoning when used by themselves, are prone to drift, or accumulation of error, and for that reason require corrections in order to be useful over long distances. An example of said corrections is Monte Carlo localization (Dellaert et al., 1999), a probabilistic method that uses sensor information to localize on a known map and compensate for odometry drift. Since they rely on hardware specific to robots, methods that use wheel or motor information to estimate odometry are not appropriate to localize people.

Inertial Measurement Units (IMUs) are sensors that measure acceleration and rotation of a body by using a combination of gyroscopes and accelerometers; they have also been used to aid navigation, both in robots (e.g. Yi et al., 2007; Lee et al., 2009) and people (e.g., Stirling et al., 2005; Ruiz et al., 2012). Cameras can also be used for movement estimation in the form of visual odometry. This uses either a single camera, or a stereo pair to estimate movement relative from the starting location. In the system described here, we use a device, Google Project Tango, that implements a combination of IMU sensors and visual odometry to estimate movement relative to a starting location.

2.2 Wearable Sensors

The idea of giving people increased abilities by attaching sensors to them has been developed in different areas. Examples in the medical field include using sensors to diagnose rehabilitation progress (e.g. Jovanov et al., 2005; Winters et al., 2003), for constant monitoring of vital signs (e.g. Gao et al., 2005; Yilmaz et al., 2010), for monitoring patients for falls (Chen et al., 2005) and other situations where people might need assistance. Wearable sensors have also been used for recognizing user context (Lee and Mase, 2002; Kern et al., 2003), that is, a rough estimate of the type of location the wearer might be in, and the activity he or she is doing.

Additionally, Stirling et al. (2005) presents a system that uses a foot mounted IMU that detects steps, then uses that data to estimate human odometry information, and attempts to localize. A similar system is presented in Tian et al. (2014) that uses a combination of an IMU and a camera to localize a human wearing them.

Light Detection and Ranging (LIDAR) devices, which emit light, typically infrared, and measure the time it takes to rebound to detect multiple distances from a point, have also been mounted on people with the purpose of applying algorithms such as mapping and localization that are traditionally used in robotics. The system described in Baglietto et al. (2011) combines a LIDAR and an IMU mounted on a helmet to perform Simultaneous Localization and Mapping (SLAM) with people wearing it, adapting traditional SLAM algorithms to compensate for head movement.

Our system uses a single device for sensing, a Google Tango tablet with built in IMUs and a 3D camera system, enabling localization and mapping while maintaining a smaller profile. It includes built in wireless networking, which makes system integration easier, and has built in computation capabilities, allowing for on board data processing and future expandability. We used a chest mount to convert the tablet into a wearable device.

2.3 Wearable Technology for Command

In Wilson and Wright (2009), the authors tested a custom head mounted display system for giving navigation directions to first responders. They found that responders had positive results when using the system, including faster completion times and lower rates of navigation errors. First responders that tested the system were positive about the use of head mounted technologies to aid in navigation.

Malek Newaz et al. (2015) tested the usage of a head mounted display by first responders. They describe a survey conducted with police officers, in which they ranked the order of importance of features in a head mounted display. Officers ranked how useful they found the categories of Remote Guidance, Voice, Map, Image and Text. Of those categories, every single one received more than 80 percent of responses as "Useful" or better. In the case of Map, all participants responded with "Useful" or better. They designed a prototype of a Google Glass application that had text mission debriefing, navigation and video streaming. They then tested a navigation task using the Google Glass, and compared it against smart phone guidance. In that scenario, they found no meaningful differences between Google Glass and a traditional smart phone.

We also use Google Glass in our system; we currently have implemented four out of the five categories that responders found useful in the survey, except for map visualization which is a planned future improvement. Additionally we integrated our Google Glass system into a robotics communication framework, and added other features that we postulate will aid working alongside with robots, including bidirectional audio/video communications with robots, and control of a pan tilt camera using head movement.

2.4 Human-Robot Team Cooperation

Research exists regarding making robots cooperate better with humans, including cooperative path planning (Kruse et al., 2010) and joint action planning (Schrempf et al., 2005) systems developed specifically for robots that interact with humans.

Regarding robots working alongside first responders, research includes methods to enhance responders' capabilities or warn them of dangers. In Kumar et al. (2004) a system is designed for robots that work alongside firefighters setting sensors in a building, enabling, among other things, detection of possible victims or areas with dangerously high temperatures. This information is then transmitted to the base station, so it can be relayed to the responders. Our implementation would allow, if working in combination with such a system, for information to be relayed directly from robots to first responders, bypassing the base station and allowing responders to react more promptly, or use video streaming to survey an area by themselves.

2.5 Multiple Unit Control Interfaces

One of our goals is to integrate the ability to integrate robot and people on the same interface. The experience from the point of view of the commander in this situation will be similar to that of operating multiple robots at the same time. Multiple robot control presents several obstacles, such as the need to maintain situational awareness from multiple robots, and being able to divide attention among them. There are a number of different approaches for this, including systems such as the one described in Humphrey et al. (2007), where the interface focuses on data from a single robot, providing only general information about the other ones, and where the robot that can be controlled at a determined time is the one with focus.

In Micire (2010) a touch screen interface that enables simultaneous multiple robot control is described. This system allows control of multiple units, while providing a top down view of the map in which they are located. We expand on that work by integrating our system with it, allowing it to be used to command people and display information gathered from them using the same gestures and features that are currently used for robot command. We opted to enhance this touch screen interface because it provides simple ways for commanding that would not require new users to undergo significant training to operate it.

2.6 Natural Language Command for Robots

While faced with the problem of unifying command methods for people and robots, an approach is to implement systems that command robots using natural language, for example, in Brooks et al. (2012) a system is described that allows control of a mobile robot using natural language in the form of text commands. There are also systems that use voice control to command robots, such as the one described in Simpson and Levine (2002) for controlling a wheelchair. In our case, we took the opposite approach to unifying command, instead of adopting the method used traditionally for giving commands to people (natural language) and applying it to robots, we took an interface designed for commanding robots, and adapted it to send commands to people. For applications in search and rescue, an advantage of robot control interfaces is precision in position commands and feedback, which we believe will be as useful when commanding people as it is when commanding robots.

Chapter 3

System Design

3.1 Objective

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

3.2 Requirements

3.2.1 Field User Interface

Displaying information that is not needed immediately in a user interface can distract the user from the current activity. Information overload, even when said information is related to an important task, is often counter-productive, even resulting in the user not paying attention to future information, dismissing it as unimportant. Interruptions, such as notifications, are known to be especially detrimental (Bailey et al., 2001). Since our main target user group consists of first responders, who need to be completely focused on the task they are performing, any unnecessary distraction needs to be minimized. As such, the field user interface should be as unobtrusive as possible, yet easy to look at when information is required, and it should provide the minimum amount of information necessary to accomplish its goal. Interaction of the field user with the interface should also be kept to a minimum, acting mostly as an output device for information. The information that needs to be available to the user is a form of conveying movement directions received from the base station.

3.2.2 Sensor Information

Besides a method for the field user to receive directions, it is desirable to automatically capture sensor information of the area the field user is in, as this opens possibilities for localization and mapping, along with easier information capture (e.g., temperatures, oxygen levels, thermal images) and sharing. The minimum sensor suite expected by most robot-oriented software is a form of odometry to estimate the position of the agent and sensors that provide information about the surroundings (e.g., cameras, distance sensors, temperature sensors). Information captured by the devices carried by the field user should be transmitted constantly back to the base, without the field user having to interact with the device.

3.2.3 Base Station User Interface

Minimum functionality requirements for the base station software are:

- Provide robot and human position on a map.
- Allow for sending movement commands in the form of waypoints to a group of agents (people and robots) simultaneously.
- Allow for manual control of an individual agent (e.g. to manually clear navigation problems, particularly for robots).

In addition, the base station software should not require any major new training for its intended users, i.e. it should be as similar as possible to existing user interfaces (e.g. typical computer graphical user interfaces and smartphone touch gestures).

3.3 Considerations

3.3.1 Software

ROS (Quigley et al., 2009) is an open source framework designed for robot intercommunication. It has a large community of developers and users, and a large library of software, including device specific drivers and robot control tools. Given that our system involves integration with robot applications, we decided to integrate it into the ROS ecosystem, as this would give it the advantage of being immediately compatible with the existing ROS software. At the time of writing, no other robotics framework existed that could provide the interoperability capabilities that ROS has. More details on the ways our software was integrated with ROS are given in Chapter 4.

3.3.2 Display Device

Possible choices for the output device to present commands to users on the field included smartphones, tablets, and wearables such as smart watches, Google Glass and other forms of smart eyewear. Devices that are not worn at eye level would require an alert, either sound or vibration, to get the user to look at the device when new information is available. As mentioned above, we are trying to eliminate those types of alerts, so we decided to use smart eyewear as the output device. Google Glass (fig. 3-1) was selected among the available smart eyewear, since we determined that out of the available devices, Glass provides the best balance between being visible when needed, yet remaining out of the field of view when not. More details on the Google Glass Hardware are given in Section 4.1.2.

3.3.3 Sensors

Options for augmenting humans with sensors include building a custom sensor system, with components such as an IMU (Inertial Measurement Unit) to sense movement

Figure 3-1: Google Glass Headset

and stereo cameras to gather environment information and allow mapping. Using such a system would require a mounting solution for attaching each of the sensors to a human agent, along with hardware drivers to send this information back to the base station. To simplify system integration, we decided to use the commercially available Google Project Tango Yellowstone tablet: an Android tablet that includes 3D sensing capabilities via an RGB-D (Red, Green, Blue - Depth) camera, similar to the ones available in gaming devices such as the Microsoft Kinect. Traditionally such sensors have been used for video games that are motion controlled, and in the case of the Tango, for applications that allow taking measurements with the device and visualizing objects superimposed at real size. More recently, RGB-D cameras have been repurposed in robotics (Oliver et al., 2012; Suarez and Murphy, 2012) and used for localization, obstacle avoidance and 3D mapping.

We elected to use the tablet because it includes all of our desired sensors, does processing for localization on board, and offers the capability of running other software on it, leaving room for future expansion of the system. Using the tablet also simplifies mounting, as it is a single integrated piece of hardware. We describe the Project

Tango Hardware and capabilities more in depth in Section 4.1.1.

Figure 3-2: Project Tango tablet

Figure 3-3: Project Tango tablet sensor detail
Left to right in the top black bar: Infrared Camera, LED Flash, Fisheye Camera, Infrared Emitter

Chapter 4

System Implementation

The following is a description of the hardware used, the new software developed as part of the system and existing software that was reutilized, detailing, if applicable, the modifications and configurations done to it.

4.1 Hardware

The minimum sensor/actuator suite expected by most robot-oriented software includes sensors to estimate the position of the agent, some sensor input of the surroundings, and a way to execute movement commands. For our application, it is also necessary that the equipment to be used provides minimal obstruction to a person's ability to move, along with minimal added weight. To fulfill these requirements, we selected a Google Project Tango tablet and a pair of Google Glass for each human agent.

4.1.1 Project Tango

The Project Tango Yellowstone is an Android tablet developed by Google that incorporates a 3D depth camera, along with advanced motion tracking, making it an excellent candidate for this project. Motion tracking is realized through its incorporated accelerometer and gyroscope, enhancing it through Visual Odometry; that is, using its color camera to estimate position (Nistér et al., 2004) and correct sensor drift as with

SLAM algorithms for robots. Tango also incorporates *Area Learning*, meaning that it remembers spaces where it has been before, to detect when it is located in a place it has already been, allowing it to further correct drift. Tango exposes this information through an application program interface (API) that gives its estimated position in X, Y, Z coordinates and its estimated orientation as a quaternion.

Another feature provided by Tango is depth capture. It has an infrared (IR) depth camera, which estimates the position in space of a collection of points by emitting an array of points with IR light, then using an IR camera to capture the reflection of said points. Since the optics of the emitter and the camera are known, the system can estimate the locations in space of those points (Zhang, 2012). The tablet exposes depth values as a 3D point cloud, which is a collection of X, Y, Z points. This collection of points is similar to what a robot with a depth sensor would be capturing, making it ideal for integration with software that expects typical robot sensor data.

4.1.2 Google Glass

While Tango provides the sensor inputs needed for the system, it is undesirable for the user to have to interact with the tablet every time he or she receives a new command or information, as looking at a tablet can distract from the primary task being executed. The solution we devised is to use the Tango only for sensing, while for interaction, we use Google Glass, a heads up display in a similar shape to a pair of eyeglasses, running a special version of Android. The Google Glass headset provides an interface that can remain unobtrusive when not being used, yet easy to see as soon at it is required. It provides audio and video transmission capabilities, along with a head mounted accelerometer that allows for sensing the orientation of the wearer's head.

4.1.3 Hardware Mounting

Since the user requires no interaction whatsoever with the Tango tablet after initial setup, the most unobtrusive way of affixing it to the user was via a chest mount, either using a purpose made carrying strap, or incorporating it as part of other equipment.

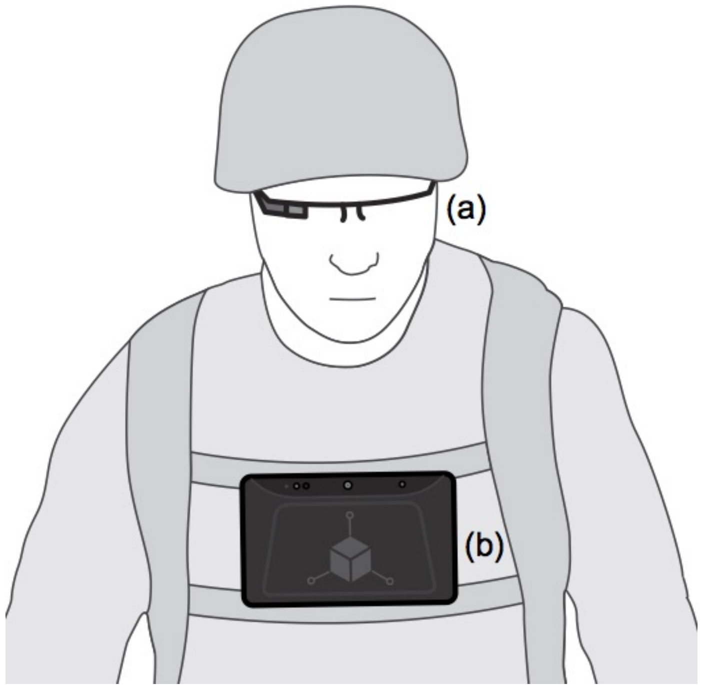

Figure 4-1: Detail of the hardware worn by a user: (a) Google Glass and (b) Project Tango

Meanwhile the Google Glass headset, by design, is worn as a pair of eyewear. (See Figure 4-1.)

4.2 Software System Development

The following describes the new software developed as part of the system, along with description of existing software that was reutilized. Then, a unified control interface for robots and humans is described. An architecture diagram of the full system is shown in Figure 4-4.

4.2.1 ROS interfacing

Since both the Yellowstone tablet and Google Glass run a version of Android, *ROSJava*, a Java ROS client implementation (Kohler, 2012), was the version of choice for the software that runs on Glass, while a native Android port of the ROS C++ libraries was used for the software that runs on the Yellowstone tablet. ROS uses a structure of *nodes*, which are executable programs, that pass *messages* containing data of different types, through *topics*, which are established communication channels between nodes. The following are the ROS nodes developed by us to interface the hardware worn by the human agents with the rest of the ROS ecosystem.

Point Cloud Publishing

As mentioned above, the Tango captures 3D point clouds, which are useful for tasks such as mapping or path planning. The C API provided for Tango by Google allows access to a point cloud every time a new one is available, which is approximately 5 times a second; however, the function that provides this information does not allow blocking for more than a very short amount of time, which is less than the time actually needed to publish each cloud over the network. As a solution, our software relies on Google's Tango Support library, which provides capabilities to do a memory copy of the point cloud and then process it on a separate thread, from which we publish to a ROS topic, to which any other ROS node may subscribe.

Pose Publishing

The other information required from the Yellowstone tablet is the estimated pose. Similar to the point cloud, Google's API allows for retrieval of poses as often as they are available. Publishing this data in real time would overload the device and slow down point cloud publishing. For this reason, pose publication has been manually limited at 15 Hz in our software. As mentioned before, Tango provides the estimated pose as X, Y, Z coordinates along with a rotation quaternion; as this is also the pose format that ROS needs, there is no conversion required. This node, upon receiving a new pose, publishes the distance traveled from when it started as an odometry message, and publishes a set of mappings, known in ROS as transforms. A dynamic transform consists of a mapping from the point where the trajectory started to the current estimated base (in this case, an imaginary circular base around the user's feet) position; a static transform maps from the base position to the depth camera's position.

This node also subscribes to a topic for pose input, and corrects the estimated pose to the received pose every time it receives one. This pose data is useful to manually reset the position of an agent when he or she is at a known location.

Glass User Interface

Glass runs a single ROS node, which subscribes to twist messages, which consist of the desired linear and angular velocities. The ROS node 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 node also subscribes to a topic that takes text input, displaying the text on screen for any received messages.

Glass Head Tilt Sensing

Glass can also detect head tilting and transmit this information back to the base station. This data combined with transmitting the video captured with Glass, allows to receive head tilting commands as audio messages (e.g., "look up", "look down"), and allows for the agent to mimic a camera on a pan/tilt unit.

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, to assist in information sharing.

Audio Transmission and Reception

An audio transmission system was developed so that users wearing Glass headsets can talk with each other, either in private conversations or multiple user groups, including or not 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. The audio transmission and reception system was implemented in the form of a Session Initiation Protocol (SIP) voice over IP server. Software was written to allow SIP communications to be controlled using ROS, allowing any of the devices to have a voice conversation either with a single other device, with multiple of them, or with all the other devices.

Navigation Setup

In addition to the newly developed nodes, it was necessary to configure the path planning software included in ROS to work with our system. 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 person, path planning might not appear to be necessary, as a person has the capability of avoiding obstacles upon seeing them. Nevertheless, path planning is useful as it is desirable to be able to command people by giving them waypoints. For this purpose, the navigation modules on ROS were configured for a person.

Move Base is a software package provided in ROS to provide robot movement to specified waypoints. Once the previously mentioned pose publishing and point cloud publishing nodes were running on the Yellowstone tablet, no software changes were required in the navigation stack; it was simply configured as if the person was a robot. Specifically, it was configured for a 40 centimeter radius holonomic robot with a point cloud sensor. Goal tolerances were set higher than they would usually be for a robot, since when commanding a person to an area of interest, high precision within the specified goal is not usually required.

Using the software described in the previous sections, the people 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 (an example being the just described configuration for the Move Base package).

Figure 4-2: Screenshot of command system displaying a person alongside a robot on a section of a pre-generated map. The map was built from the combination of a floor plan and robot laser sensor input. Varying shades of gray on the map represent probabilities of an obstacle being in a determined location.

4.3 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 controlling a combined human/robot team.

It was decided to expand upon the software described and evaluated by Micire (2010) 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 (Micire, 2010).

Original Software: The original software we are referring to, described in Micire (2010), consists of a touch interface for multiple robot command built in C#, using Microsoft RDS (Robotics Developer Studio). It displays robots as icons on a map, allows users to select either one or a group of them by touching their respective icons, then provide waypoints where the user wants the robots to go, and send the commands to each robot. It also provides manual robot control via the DREAM controller, which is a virtual joystick displayed on the screen when a user places all 5 fingers on it. It allows to drive the robot using the thumb and also provides available information from the robot back to the user (usually video and range finder information).

The following modifications were required for the purposes of linking the original control software with our system:

- *Interfacing with ROS:* The first modification required was to interface the control software with ROS. Since the original uses Microsoft RDS in a Windows environment, the decision was made to keep the user interface as unmodified as possible, while switching the back end from RDS to ROS, replacing the original network communications with ROS.NET (a ROS Client Library for Windows Development in C#, see `https://github.com/uml-robotics/ROS.NET`), enabling the interface to work with ROS powered robots, along with our human agent system.

- *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 purpose, changes were made in the software to accommodate control of different forms of agents, which mostly consisted of 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 controlled through the interface, changes were made 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.

- *Position Display:* The original software derives the position of each robot from data provided by RDS. This approach was changed to a ROS pose topic for each robot/human, to which the program subscribes and from where it gets the information to display each of them on their position in the map. Most ROS based mobile robots (and our Yellowstone pose publishing node described previously) use some form of localization and provide their estimated position through a pose topic, making it easy to integrate them with the interface.

- *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. This uses the SIP server described in Section 4.2.1.

The mission control software provides the capability to control robots and humans simultaneously through touch gestures by selecting both and giving them waypoints to which to navigate. After such a selection, goals are published to the ROS move base goal topic for each selected agent, which then sends to both kinds of agents the appropriate velocity commands to take them to the destination. The humans receive the commands translated to audio navigation prompts or arrows pointing in the direction they should walk, displayed on the Google Glass headset, as shown on Figure 4-3. In Figure 4-2, a screenshot is provided showing the final system, displaying the position of both a robot and a human agent.

Figure 4-3: Image of field user following navigation instructions. Google Glass arrows view shown on top left corner

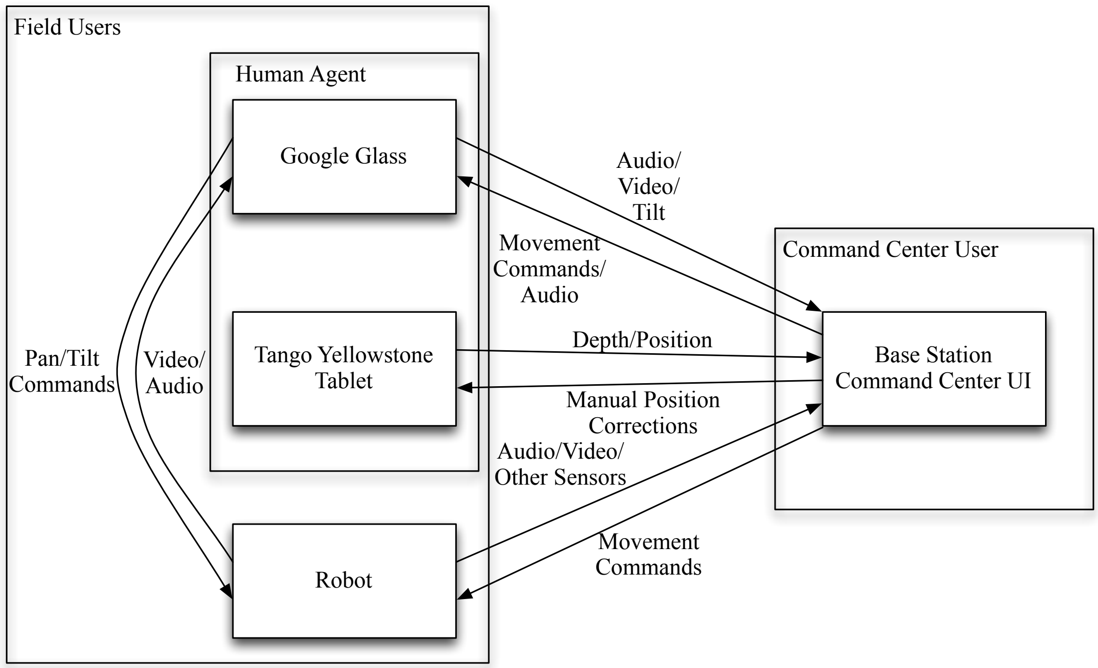

Figure 4-4: Software architecture diagram
Communications use ROS as a back end, audio additionally uses a ROS controlled SIP server.

Chapter 5

Experiment Design

We designed an experiment, approved by UMass Lowell's Institutional Review Board, to discover how users might behave differently when sending commands to humans alongside robots using the same user interface, as opposed as sending commands only to robots. We plan to use what we learn from the experiment to improve both the command system we described in the previous chapters and future interfaces that are designed to send commands to humans.

5.1 Research Questions

In this experiment we were trying to discover specific behaviors users might have when commanding human agents instead of robots, thus helping us discover which changes would be appropriate when adapting a robot user interface to command humans.

RQ1: Would participants success be different when using human units? We are interested in knowing if participants are as successful in the assigned tasks when commanding both people and robots as they are when commanding only robots

RQ2: Would participants group similar units together? We believe that users are likely to group similar units together when sending commands; hence we believe that users will be more likely to group robots separately from humans, even when sending the same navigation command to both.

RQ3: Would participants trust path planning more on human units? We are interested in knowing if a user feels the need to send a more or less specified path when sending commands to humans, that is, whether a single goal is more likely to be sent, rather than a large set of waypoints.

RQ4: Would participants solve navigation problems in a different way for human units? Erratic behavior is common both when commanding robots (e.g., path planning failures, crashes into unexpected objects, etc.) and humans (e.g., human does not follow instructions, does not understand them, is distracted and misses command, etc.), so the reaction the commander has to those needs to be accounted for. We want to know whether the reaction the commander has, whether it is using a joystick to control the stopped unit, sending new navigation waypoints or ignoring the unit, is different between a human and a robot.

RQ5: Would participants use manual controls on humans? We believe participants might not use manual control in humans as mcuh as in robots, as they might not

5.2 Hypotheses

We formulated the following hypotheses, based on what we expect will be the participants' behaviors when using the interface.

H1: Users will be as successful in the assigned task using human units as they are with robots

H2: Users will group units as required, with no difference if the units are humans or robots

H3: Users will be less likely to use manual control on human units than on robot units

5.3 Experiment Format

We designed our experiment based on the multi-robot user interface described previously. Sixteen participants were asked to do two runs. On the first run, participants were told they were commanding six robots, the user interface depicted all icons as robots, and participants were shown an image of a ActivMedia Pioneer robot (Figure 5-1). On the second run, participants were told they were commanding three robots and three humans, and the 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 5-2. The situation was described to participants as a disaster response scenario, in which their goal was to explore as much of the map as possible, using the 6 robots units they have available. For the second run participants were told they had the same task, and same controls, but now 3 of the units are humans equipped with our system. Humans were described to participants as having the same exploration capabilities as robots, with the Tango tablet mimicking a range finder, and as accepting the same commands as robots. Initially, the map in the user interface was shown completely covered, except for the sections the units could see from their initial positions, as seen in Figure 5-3, and as the units moved the map was gradually uncovered, as seen in Figure 5-4. Users were asked to return the units back to the starting position when they were done with the exploration, to provide them a reason to group units at the end of the run.

5.4 Training

Participants were told they would be using a multi-touch screen interface to either control robots or send commands to human agents. Participants were told they could select either individual units or multiple, and send either a single goal, or a collection

Figure 5-1: ActivMedia Pioneer robot

Figure 5-2: Field user wearing a Project Tango Tablet and Google Glass

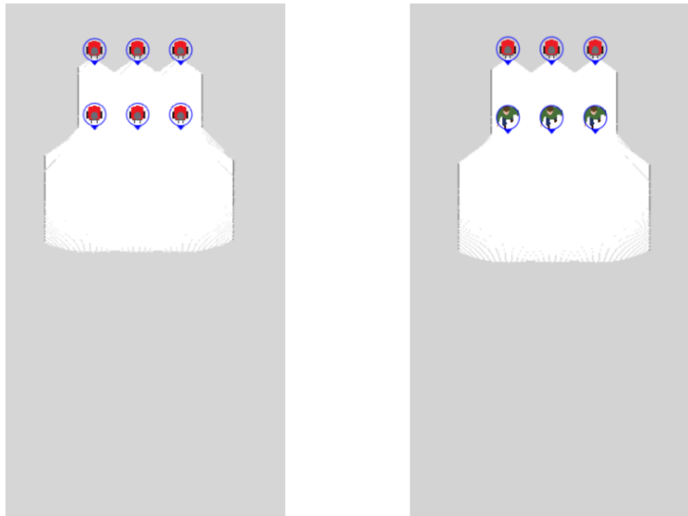

Figure 5-3: Map as initially shown, with unexplored areas hidden
Left image is from first run, with robots only. Right image is from second run with combined robot and human units.

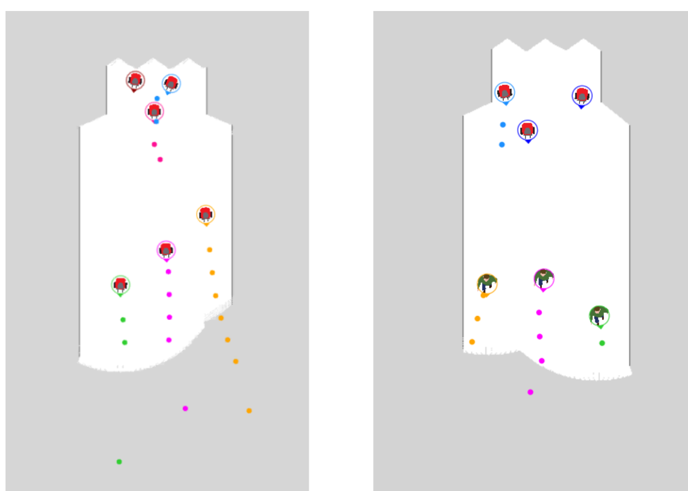

Figure 5-4: Map uncovered as units move
Left image is from first run, with robots only. Right image is from second run with combined robot and human units.

of waypoints that the unit will 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 (Micire, 2010), a virtual joystick displayed on screen. The situation was described as a test course representing a search and rescue situation, in which the main goal was to explore as much area as possible, using either the robots or human agents. The available gestures and control methods of the interface were explained to users, and demonstrated on the touch screen, then users were given unlimited time to practice controls. For this practice round, a separate map was used, consisting of a completely empty square room, with two robot units available for use. No data was recorded from the training.

5.5 Configuration

For our results to be dependent only on whether the user was seeing humans or robots in the command interface, we needed both robots and humans to have the exact same capabilities, and behave in the same way. To reduce variability, we decided to run the units in simulation. We used the Stage simulator (Vaughan, 2008), connected through ROS; all units, both humans and robots, were simulated Pioneer Robots. For navigation purposes, the simulated robots were set to have a front facing laser range finder, with a field of view of 180 degrees and a range of 8 meters, along with a rear facing one with a field of view of 240 degrees and a range of 4 meters. For the purpose of exploration as described to the user and as visible from the command UI, the units had a front facing field of view of 110 degrees and a range of 3 meters. Simulated units were set to have navigation failures every 5 minutes, where a random unit would cancel all their pending waypoints and stop, to record how users respond to those failures. For the first run, we used a map representing a hallway with several rooms (Figure 5-5). Rooms included different features such as separate closed areas in the room, inside doors connecting rooms, and small spaces the units had to navigate through. For the second run, a variation of the map was used (Figure 5-6), in which rooms and the starting position were shifted around, to avoid users memorizing the

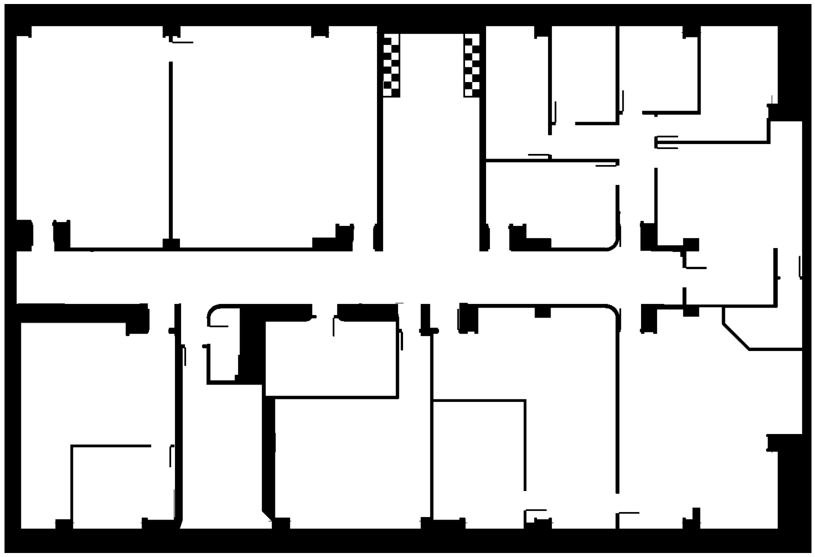

Figure 5-5: Map used for robot only run

map from the first run and thus affecting their behaviors.

5.6 Survey

Once participants were done with the exploration tasks, we asked them to fill out a survey regarding their experience with the system. Participants were asked to rate the following on a Likert scale from 1 (Very Low) to 7 (Very High):

- Performance on the assigned task.
- Stress level while performing the task.
- Mental demand for performing the task.
- Frustration during the task.

Participants were asked to state their level of agreement with the following statements, rating from 1 (Strongly Disagree) to 7 (Strongly Agree).

- I felt the user interface helped me perform the assigned task.
- I felt the user interface was a hindrance in performing the assigned task.

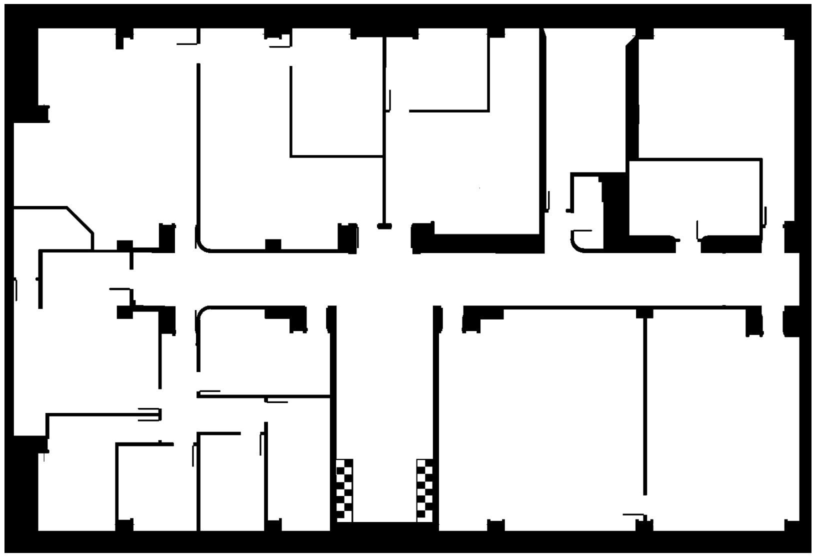

Figure 5-6: Map used for human/robot run

- I felt the user interface was adequate for sending commands to robot units.
- I felt the user interface was adequate for sending commands to human units.

Finally, participants were asked as open question,“Are there any features you feel would make the user interface more adequate for sending commands to human units?”

Chapter 6

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 $\kappa = 0.8307$ ($\kappa = 0.5887$ excluding chance); we discovered that there was significant disagreement in the categories for navigation failure. After analyzing the coding data, we discovered the coders were coding different moments in the video to refer to the same failures. We decided to code the category of navigation failure again, which 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 $\kappa = 0.855$ ($\kappa = 0.6166$ excluding chance); and our kappa for navigation failures was $\kappa = 0.8077$ ($\kappa = 0.6448$ excluding chance). According to Landis and Koch (1977), 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.

Table 6.1: Age and gender distribution of participants

Participant	Age	Gender	Handedness
1	19	M	R
2	18	M	R
3	27	F	R
4	20	M	R
5	18	M	R
6	18	F	R
7	19	M	R
8	18	M	R
9	18	M	R
10	27	M	R
11	23	M	L
12	18	M	R
13	26	M	R
14	54	M	L
15	25	M	R
16	23	F	R

6.1 Demographics

Our participant sample consisted of 16 persons, aged 18 to 54, 18.75% female (more details are in Table 6.1). Participants were recruited from the University of Massachusetts Lowell Computer Science department, by sending a recruitment email to the department's mailing list. Participants were offered a $15 USD Amazon.com gift card in compensation for their time. Two of the 16 participants (12.5%) were left handed.

Participants were asked questions about their familiarity with robots, remote controlled toys, video games, and real time strategy video games. The results of those questions are in Table 6.2, evaluated from 1 to 7 on a Likert scale, where 1 is Strongly Disagree and 7 is Strongly Agree.

Participants were also asked whether they considered themselves good at multi-tasking, and if they believed they had good hand/eye coordination. The results of those questions are in Table 6.3, evaluated from 1 to 7 on a Likert scale, where 1 is Strongly Disagree and 7 is Strongly Agree.

Table 6.2: Demographics results 1

Participant	Robots	Remote Controlled	Video Games	Real Time Strategy
1	3	3	7	7
2	5	4	6	6
3	3	3	7	5
4	3	5	7	6
5	7	7	7	7
6	2	4	7	5
7	4	5	7	5
8	5	4	7	7
9	5	5	5	4
10	1	1	6	4
11	5	6	7	7
12	4	5	7	3
13	2	6	4	5
14	6	3	6	5
15	6	6	7	7
16	2	1	6	1
Mean	3.94	4.25	6.44	5.25
Std. Dev.	1.67	1.68	0.86	1.64

Experience with Robots, Experience with Radio Control vehicles, Experience with Video Games, Experience with Real Time Strategy Video Games

Table 6.3: Demographics results 2

Participant	Multitasking	Hand/Eye Coord.
1	5	5
2	6	6
3	5	4
4	5	5
5	5	5
6	6	6
7	5	6
8	7	7
9	6	5
10	4	5
11	6	7
12	7	5
13	3	6
14	6	6
15	7	7
16	6	7
Mean	5.56	5.75
Std. Dev.	1.06	0.90

Table 6.4: Participants' task success

Participant	Time Run 1	Explored % Run 1	Time Run 2	Explored % Run 2
1	21:20	99.6	18:47	99.6
2	30:04	99.6	20:07	99.4
3	17:47	98.7	16:13	99.3
4	27:15	99.6	19:03	99.5
5	33:34	99.6	13:36	99.4
6	13:00	98.3	17:07	99.1
7	13:21	99.3	12:56	99.2
8	28:25	99.5	17:38	99.3
9	16:11	79	11:27	40
10	19:46	99	19:39	99.5
11	24:50	42.55	17:24	58.7
12	20:50	96.9	17:01	98.9
13	11:42	97.6	11:12	97.6
14	17:19	99.3	20:50	99.2
15	16:53	81.2	11:54	58.2
16	21:50	85.5	13:02	99.64
Mean	20:53	92.20	16:07	90.41
Std. Dev.	06:15	14.48	03:10	18.69

6.2 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. Results are shown on Table 6.4.

We then compared each of the participants' second run against the 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 6.5. 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 -1.794375 ($\sigma = 12.4113$). Of the 4 participants with significant differences between runs, 2 covered more area on the second run and two covered less

Table 6.5: Comparison of participants' results in second run against 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

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 these 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.

Table 6.6: Unit grouping data

Participant	Groups R1	Human Only R2	Robot Only R2	Mixed R2
1	1	0	1	0
2	6	0	0	14
3	1	0	0	0
4	0	0	0	4
5	0	0	2	0
6	0	0	0	0
7	0	0	0	4
8	0	0	0	0
9	4	0	2	0
10	1	0	1	0
11	3	0	1	0
12	1	0	0	0
13	1	0	2	0
14	0	0	0	0
15	4	0	0	2
16	4	2	2	2
Mean	1.62	0.12	0.69	1.62
Std. Dev.	1.87	0.48	0.84	3.48

6.3 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”. Grouping data is shown on Table 6.6. 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.

6.4 Number of Waypoints

When sending waypoints to a unit, participants have the option of sending any number of waypoints they prefer. We recorded how many waypoints they sent each time. The

Table 6.7: Average number of waypoints by unit type and participant

Participant	Run 1	Robots Run 2	Humans R2	All R2
1	6.48	5.28	4.98	5.11
2	16.37	12.52	9.53	11.00
3	12.80	13.33	9.54	11.15
4	10.00	12.02	9.61	10.76
5	10.56	15.44	11.88	13.37
6	18.38	16.87	12.84	14.83
7	12.49	9.37	10.26	9.78
8	9.76	12.04	10.75	11.37
9	6.81	4.95	4.48	4.69
10	10.12	7.55	9.29	8.47
11	8.43	11.77	7.77	9.26
12	12.12	12.72	9.38	10.89
13	19.50	12.29	12.87	12.62
14	10.71	9.94	8.40	8.96
15	19.15	18.07	24.11	20.43
16	17.98	14.31	10.19	12.25
Mean	12.60	11.78	10.37	10.93
Std. Dev	4.22	3.58	4.20	3.56

average of waypoints sent to each unit type is on Table 6.7. On the second run we separated average waypoints sent to groups containing at least a human, and average waypoints sent to a group containing at least a robot. We also recorded average number of waypoints sent to all groups. In eleven out of the sixteen participants, the amount of waypoints sent to humans was smaller than that sent to robots. The mean of the differences is -1.4124 ($\sigma = 2.6832$), which would indicate participants send fewer waypoints to groups containing humans.

6.5 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. Results are shown on Table 6.8. 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

Table 6.8: Manual control results

Participant	Run 1	Run 2 Robot	Run 2 Human	% Humans R2
1	14.00	2.00	8.00	80.00
2	9.00	3.00	3.00	50.00
3	4.00	0.00	5.00	100.00
4	8.00	1.00	4.00	80.00
5	12.00	2.00	1.00	33.33
6	0.00	1.00	5.00	83.33
7	1.00	1.00	2.00	66.67
8	9.00	1.00	2.00	66.67
9	9.00	2.00	1.00	33.33
10	5.00	5.00	11.00	68.75
11	22.00	16.00	4.00	20.00
12	4.00	0.00	2.00	100.00
13	2.00	0.00	0.00	N/A
14	9.00	8.00	5.00	38.46
15	5.00	1.00	1.00	50.00
16	8.00	2.00	1.00	33.33
Mean	7.56	2.81	3.44	60.26
Std. Dev.	5.29	3.94	2.83	24.64

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 control a human. However, all of the participants who used manual control on the second run used it at least once on a human.

6.6 Reaction to Erratic Behavior

We recorded what action a participant took when a unit failed at reaching its navigation goal, whether the participant gave the unit new waypoints or corrected using manual control. On the second run we separated our records by humans and robots. Results are shown on Table 6.9.

For the second run, we calculated the percentages of times robots were corrected using new waypoints and the percentages of times humans were corrected using manual control. Percentages are shown on Table 6.10. We compared the percentages of times manual control was used to solve problems on robots with the percentage of times it

Table 6.9: Participants' actions to correct navigation failures

	Robot Run 1		Robot Run 2		Human Run 2	
Participant	Manual	New WPs	Manual	New WPs	Manual	New WPs
1	14	4	2	1	4	2
2	8	28	1	2	3	6
3	4	11	0	2	5	8
4	4	4	0	4	4	2
5	1	2	0	1	1	2
6	0	8	1	4	4	7
7	1	7	0	2	1	1
8	5	10	0	0	2	1
9	5	11	2	5	1	5
10	4	9	4	6	4	2
11	8	19	1	1	1	2
12	2	3	0	0	2	4
13	2	7	0	0	0	3
14	4	7	5	3	4	4
15	4	9	2	5	1	1
16	3	8	1	2	1	3
Mean	4.31	9.19	1.19	2.38	2.38	3.31
Std. Dev.	3.31	6.22	1.47	1.87	1.54	2.11

was used to solve problems on humans. In 7 cases participants were more likely to use manual control to solve problems with humans than to use it to solve problems on robots, in 4 cases they were less likely, and in 2 cases they were equally likely. The other 3 cases concerned participants that did not have robots that got stuck.

6.7 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 6.11. 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,

Table 6.10: Percentage of times participants used manual control to solve navigation failures on second run.

Participant	Manual % Robots	Manual % Humans	Humans - Robots
1	66.67	66.67	0.00
2	33.33	33.33	0.00
3	0.00	38.46	38.46
4	0.00	66.67	66.67
5	0.00	33.33	33.33
6	20.00	36.36	16.36
7	0.00	50.00	50.00
8	N/A	66.67	N/A
9	28.57	16.67	-11.90
10	40.00	66.67	26.67
11	50.00	33.33	-16.67
12	N/A	33.33	N/A
13	N/A	0.00	N/A
14	62.50	50.00	-12.50
15	28.50	50.00	21.43
16	33.30	25.00	-8.33
Mean	27.91	41.66	15.66
Std. Dev.	22.48	18.82	25.47

6 and 6. This indicates participants were not aware of their low performance on the task.

Participants were also asked their opinions on the User Interface. The questions were whether the user interface helped them on the task, whether the user interface hindered them on the task, whether the user interface was adequate for commanding robots, and whether it was adequate for commanding humans. Responses were on a 7 point Likert scale, where 1 is Strongly Disagree and 7 is Strongly Agree. Results are shown in Table 6.12. Participants were mostly positive about the user interface (mean of 5.75 for Helped, and 2.56 for Hindered), with only one participant (P10) giving a score below 5 for "Helped" and above 3 for "Hindered". Additionally we found that most users found the UI adequate for both robots (Mean=5.56) and humans (Mean=5.44), moreover, we only had two participants (P4 and P10) who found the interface less adequate for humans than for robots.

Table 6.11: Participants' experience with the user interface

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

Table 6.12: Participants' opinions about the user interface

Participant	Helped	Hindered	Adequate for Robots	Adequate for Humans
1	6	2	6	6
2	7	3	6	6
3	6	3	6	6
4	6	2	6	4
5	5	4	5	5
6	7	2	6	6
7	5	3	5	5
8	5	3	6	6
9	6	2	6	6
10	3	6	4	3
11	6	1	6	6
12	6	1	7	7
13	7	2	7	7
14	7	2	6	6
15	5	3	3	4
16	5	2	4	4
Mean	5.75	2.56	5.56	5.44
Std. Dev.	1.03	1.17	1.06	1.12

6.7.1 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 only shows 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.

6.8 Effects of previous experience with robots

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. We discovered a difference in the percentage of times a user 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 (Table 6.13), while participants who had more experience with robots used manual control more on robot units (Table 6.14).

Table 6.13: 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 6.14: 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

6.8.1 Statistical Significance

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, $\sigma = 15.18$), $p = 0.005$.

Chapter 7

Conclusions and Future Work

In this thesis, we described the design and implementation of a system that allows human agents to be integrated into a robot command system. We described the design choices and the implementation details, along with alternatives that we could have chosen and the reasons for using the hardware and software we decided to use. Afterwards we described how our system was integrated in an existing robot command interface. We then detailed the design of an experiment to discover the different ways a human commander behaves when sending commands to humans alongside robots. We discovered that while some changes will improve the existing interface, commanders are successful at using the interface to command human agents. We also discovered that users who have previous experience with robots are less likely to use manual control on human units than on robots. We now present our experiment conclusions, and propose a set of guidelines for human/robot command software, based on what we learned from the experiment. We also developed guidelines for human mounted sensor systems and wearable command interfaces based on our experience on the system's design.

7.1 Experiment Conclusions

H1: *Users will be as successful in the assigned task using human units as they are with robots.* We confirmed H1, as we discovered 12 out of the 16 participants had similar performance on the exploratory task, as defined by being within 2 percentage points of area explored.

H2: *Users will group units as required, with no difference if the units are humans or robots.* Use of the grouping feature by participants was in general very rare, so more research is required in order to confirm or deny H2. We plan to do a future study in which the task will be structured to force users to group units, allowing us to gather more data related to this

H3: *Users will be less likely to use manual control on human units than on robot units.* We rejected H3, we discovered it is true for some participants, particularly those with previous experience with robots, but it is not true for those with less experience with robots, who were in general more likely to use manual control in units marked as humans than they were on units marked as robots.

7.2 Design Guidelines

7.2.1 Human-Robot Interface Guidelines

Using what we learned from the experiment results and observations during the runs, we developed the following set of design guidelines for systems that send commands to robots and human agents.

- *All features accessible to robot command should be available for human command.* As we saw in the case of manual control, commanders expect features to be available for human units, and make use of them, even those whose use is not obvious. If a feature that is available on a robot is not available on a human (e.g. does not apply for technical reasons), it should be marked in the interface so users don't expect it to be available, for example with a grayed out button.

- *Additional features should be added for humans, in cases where their normal human abilities give them more skills than the robots.* Commanders expect humans to use their own abilities to their full extent, such as being able to check a room further, or take more autonomy in path planning, as evidenced by the comments from P2 and P4. The interface should reflect this, for example by showing a video stream of what the field user sees, or by showing the original path that was sent and the one the human decided to take.

- *If a human has restricted abilities (e.g., due to the environment or technical problems), the commander should be made aware.* Participant 9 left rooms unexplored after visiting them with a human. We believe this was because the participant assumed a human was able to see the whole room, even when told humans had the same capabilities as robots for this experiment. Participant 2 also made a comment about humans being able to see further than robots. Due to users' expectations with human units, commanders should be made aware of any limitation affecting units.

- *Command methods for humans should be similar, and compatible with command methods for robots.* We saw with our interface that users would group humans and robots when necessary, that was possible since their control methods were similar and compatible. That is, the interface allowed for selection of both at once. Keeping the controls similar also makes training shorter.

7.2.2 Human Mounted Sensors Guidelines

We provide a set of guidelines for systems that use human mounted sensors to perform tasks such as mapping and navigation, we are designing a future experiment to validate these, they are currently based on experience while designing the system.

- *Sensors should be placed on the torso, not on the head or limbs.* Sensors designed for robots, such as IMUs or cameras used for mapping perform better if they are located near the robot's sensor of gravity. In the case of a human, the torso is an adequate area for sensor placement near a person's center of gravity. Sensors should only be mounted in a limb if they are meant to track the movement of that specific limb in relation to the torso.

- *Sensors should be mounted in a way that does not interfere with a user's movement, nor requires the user to avoid certain motions.* If a sensor is mounted in a way that it restricts movement, users will be less likely to wear them appropriately. Additionally, sensors should not be mounted in a way that a user has to take care as to not interfere with them (e.g., head mounted sensors that require users to minimize head turning, sensors mounted in a way that a user's normal walking movement might cover them).

7.2.3 Wearable Command Interface Guidelines

In the case of Google Glass, Google provides a set of guidelines for user experiences on the headset (Google, 2015)"Design for Glass", "Don't get in the way", "Keep it relevant", "Avoid the unexpected", and "Build for people." Based on those, we provide a set of guidelines when using head mounted displays to send navigation commands.

- *Show only immediately relevant navigation information.* While on larger devices it is common to show a full map or a complete route when displaying navigation information, in smaller head mounted displays attempting to show a full route can cause user confusion. Instead opt for showing either the next action required (e.g., walk forward, turn right), a symbol for it or a close zoom of the map showing only the immediate area around the user.

- *Target messages to specific users.* If using Google Glass to communicate with field users either in text or audio form, send messages targeted to specific users, either by sending them to a specific group, or using other data to categorize them, such as sending it only to users in a particular area. This approach prevents users who are not related to the message, or who would not be able to respond from being distracted unnecessarily from their current task.

- *Limit notifications.* Notifications such as vibrations or sound alarms should only be used for urgent information, such as a mission change or a notice to evacuate. As shown in Bailey et al. (2001), this types of notifications distract users from their ongoing activities, so information such as a non urgent message should just be displayed on screen, allowing the user to read it when appropriate.

7.3 Future Work

7.3.1 Project Tango Application Improvements

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

- Colorized pointclouds: 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 pointcloud 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 (Hornung et al., 2013), 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.

7.3.2 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 our experiments. For example, experiments could test specific behaviors (i.e. one experiment for grouping, a separate one for manual control, and a separate one for number of waypoints), by structuring the tasks so they force the user to do those different types of behaviors.

References

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*, 59(12):1060 – 1069, 2011. ISSN 0921-8890. doi: http://dx.doi.org/10.1016/j.robot.2011.08.005.

B. P. Bailey, J. A. Konstan, and J. V. Carlis. "The Effects of Interruptions on Task Performance, Annoyance, and Anxiety in the User Interface." In *Proceedings INTERACT 01*, pages 593–601. IOS Press, 2001.

J. Biswas and M. Veloso. "WiFi localization and navigation for autonomous indoor mobile robots." In *2010 IEEE International Conference on Robotics and Automation (ICRA)*, pages 4379–4384, May 2010. doi: 10.1109/ROBOT.2010.5509842.

E. Brassart, C. Pegard, and M. Mouaddib. "Localization Using Infrared Beacons." *Robotica*, 18(2):153–161, Mar. 2000. ISSN 0263-5747. doi: 10.1017/S0263574799001927.

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.

J. Casper and R. R. Murphy. "Human-robot interactions during the robot-assisted urban search and rescue response at the World Trade Center." *IEEE Transactions on Systems, Man, and Cybernetics, Part B (Cybernetics)*, 33(3):367–385, June 2003. ISSN 1083-4419. doi: 10.1109/TSMCB.2003.811794.

J. Chen, K. Kwong, D. Chang, J. Luk, and R. Bajcsy. "Wearable sensors for reliable fall detection." In *Engineering in Medicine and Biology Society*, pages 3551–3554, 2005.

F. Dellaert, D. Fox, W. Burgard, and S. Thrun. "Monte Carlo localization for mobile robots." In *Proceedings, 1999 IEEE International Conference on Robotics and Automation*, volume 2, pages 1322–1328 vol.2, 1999. doi: 10.1109/ROBOT.1999.772544.

J. L. Drury, B. Keyes, and H. A. Yanco. "LASSOing HRI: analyzing situation awareness in map-centric and video-centric interfaces." In *Proceedings of the ACM/IEEE International Conference on Human-robot Interaction*, pages 279–286, 2007.

B. Ferris, D. Fox, and N. D. Lawrence. "WiFi-SLAM Using Gaussian Process Latent Variable Models.." In *IJCAI*, volume 7, pages 2480–2485, 2007.

T. Gao, D. Greenspan, M. Welsh, R. R. Juang, and A. Alm. "Vital Signs Monitoring and Patient Tracking Over a Wireless Network." In *2005 IEEE Engineering in Medicine and Biology 27th Annual Conference*, pages 102–105, Jan 2005. doi: 10.1109/IEMBS.2005.1616352.

Google. "Google Glass Design Principles." 2015. Retrieved November 21, 2016 from `https://developers.google.com/glass/design/principles`.

J. Hightower, G. Borriello, and R. Want. "SpotON: An Indoor 3D Location Sensing Technology Based on RF Signal Strength." Technical Report 2000-02-02, University of Washington, Computer Science and Engineering, Seattle, Washington, February 2000.

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. doi: 10.1007/s10514-012-9321-0. Software available at `http://octomap.github.com`.

C. M. Humphrey, C. Henk, G. Sewell, B. W. Williams, and J. A. Adams. "Assessing the scalability of a multiple robot interface." In *Human-Robot Interaction (HRI), 2007 2nd ACM/IEEE International Conference on*, pages 239–246, March 2007. doi: 10.1145/1228716.1228749.

E. Jovanov, A. Milenkovic, C. Otto, and P. C. de Groen. "A wireless body area network of intelligent motion sensors for computer assisted physical rehabilitation." *Journal of NeuroEngineering and Rehabilitation*, 2(1):6, 2005. ISSN 1743-0003. doi: 10.1186/1743-0003-2-6.

N. Kern, B. Schiele, and A. Schmidt. *Multi-sensor Activity Context Detection for Wearable Computing*, pages 220–232. Springer Berlin Heidelberg, Berlin, Heidelberg, 2003. ISBN 978-3-540-39863-9. doi: 10.1007/978-3-540-39863-9_17.

D. Kohler. "Introduction to rosjava." Conference Slides, 2012. Retrieved September 24, 2015 from `http://roscon.ros.org/wp-uploads/2012/06/Introduction\%20to\%20rosjava\%20\%28ROSCon\%202012\%29.pdf`.

T. Kruse, A. Kirsch, E. A. Sisbot, and R. Alami. "Exploiting human cooperation in human-centered robot navigation." In *19th International Symposium in Robot and Human Interactive Communication*, pages 192–197, Sept 2010. doi: 10.1109/ROMAN.2010.5598645.

V. Kumar, D. Rus, and S. Singh. "Robot and Sensor Networks for First Responders." *IEEE Pervasive Computing*, 3(4):24–33, 2004. ISSN 1536-1268. doi: http://doi.ieeecomputersociety.org/10.1109/MPRV.2004.17.

R. Landis and G. Koch. "The Measurement of Observer Agreement for Categorical Data." *Biometrics*, 33(1):159–174, 1977. ISSN 0006341X, 15410420.

S. Lee and K. Mase. "Activity and location recognition using wearable sensors." *IEEE Pervasive Computing*, 1:24–32, 2002.

T. Lee, J. Shin, and D. Cho. "Position estimation for mobile robot using in-plane 3-axis

IMU and active beacon." In *2009 IEEE International Symposium on Industrial Electronics*, pages 1956–1961, July 2009. doi: 10.1109/ISIE.2009.5214363.

C. H. Lim, Y. Wan, B. P. Ng, and C. M. S. See. "A Real-Time Indoor WiFi Localization System Utilizing Smart Antennas." *IEEE Transactions on Consumer Electronics*, 53(2):618–622, May 2007. ISSN 0098-3063. doi: 10.1109/TCE.2007.381737.

F. B. Malek Newaz, A. W. Eide, and A. Pultier. "Supporting first responder in-field communication and navigation using head-mounted displays." In *Proceedings of the ISCRAM 2015 Conference*, Kristiansand, Norway, May 2015. ISBN 978-82-7117-788-1.

B. Manoj and A. H. Baker. "Communication Challenges in Emergency Response." *Commun. ACM*, 50(3):51–53, Mar. 2007. ISSN 0001-0782. doi: 10.1145/1226736.1226765.

M. Micire, M. Desai, A. Courtemanche, K. M. Tsui, and H. A. Yanco. "Analysis of Natural Gestures for Controlling Robot Teams on Multi-touch Tabletop Surfaces." In *Proceedings of the ACM International Conference on Interactive Tabletops and Surfaces*, ITS '09, pages 41–48, New York, NY, USA, 2009. ACM. ISBN 978-1-60558-733-2. doi: 10.1145/1731903.1731912.

M. J. Micire. *Multi-touch Interaction for Robot Command and Control*. PhD thesis, University of Massachusetts Lowell, Lowell, MA, 2010.

M. J. Micire and H. A. Yanco. "Improving disaster response with multi-touch technologies." In *2007 IEEE/RSJ International Conference on Intelligent Robots and Systems*, pages 2567–2568, Oct 2007. doi: 10.1109/IROS.2007.4399558.

L. M. Ni, Y. Liu, Y. C. Lau, and A. P. Patil. "LANDMARC: Indoor Location Sensing Using Active RFID." *Wireless Networks*, 10(6):701–710, 2004. ISSN 1572-8196. doi: 10.1023/B:WINE.0000044029.06344.dd.

D. Nistér, O. Naroditsky, and J. Bergen. "Visual odometry." In *Proceedings of the 2004*

IEEE Computer Society Conference on Computer Vision and Pattern Recognition, 2004. (CVPR 2004), volume 1, pages I–652. IEEE, 2004.

A. Oliver, S. Kang, B. C. Wünsche, and B. MacDonald. "Using the Kinect As a Navigation Sensor for Mobile Robotics." In *Proceedings of the 27th Conference on Image and Vision Computing New Zealand*, IVCNZ '12, pages 509–514, New York, NY, USA, 2012. ACM. ISBN 978-1-4503-1473-2. doi: 10.1145/2425836.2425932.

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.

A.-M. Roxin, J. Gaber, M. Wack, and A. Nait Sidi Moh. "Survey of Wireless Geolocation Techniques." In *IEEE Globecom Workshops*, page 9 Pages, Washington, DC, United States, Nov. 2007. doi: 10.1109/GLOCOMW.2007.4437809.

A. R. J. Ruiz, F. S. Granja, J. C. P. Honorato, and J. I. G. Rosas. "Accurate Pedestrian Indoor Navigation by Tightly Coupling Foot-Mounted IMU and RFID Measurements." *IEEE Transactions on Instrumentation and Measurement*, 61(1): 178–189, Jan 2012. ISSN 0018-9456. doi: 10.1109/TIM.2011.2159317.

J. Scholtz, B. Antonishek, and J. Young. "Evaluation of a human-robot interface: development of a situational awareness methodology." In *Proceedings of the 37th Annual Hawaii International Conference on System Sciences, 2004*, pages 9 pp.–, Jan 2004. doi: 10.1109/HICSS.2004.1265327.

O. C. Schrempf, U. D. Hanebeck, A. J. Schmid, and H. Worn. "A novel approach to proactive human-robot cooperation." In *ROMAN 2005. IEEE International Workshop on Robot and Human Interactive Communication, 2005.*, pages 555–560, Aug 2005. doi: 10.1109/ROMAN.2005.1513838.

R. C. Simpson and S. P. Levine. "Voice control of a powered wheelchair." *IEEE Transactions on Neural Systems and Rehabilitation Engineering*, 10(2):122–125, June 2002. ISSN 1534-4320. doi: 10.1109/TNSRE.2002.1031981.

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*, 58(1):31–45, 01 2005. doi: 10.1017/S0373463304003066.

J. Suarez and R. R. Murphy. “Using the Kinect for search and rescue robotics.” In *Safety, Security, and Rescue Robotics (SSRR), 2012 IEEE International Symposium on*, pages 1–2, Nov 2012. doi: 10.1109/SSRR.2012.6523918.

Y. Tian, W. R. Hamel, and J. Tan. “Accurate Human Navigation Using Wearable Monocular Visual and Inertial Sensors.” *IEEE Transactions on Instrumentation and Measurement*, 63(1):203–213, Jan 2014. ISSN 0018-9456. doi: 10.1109/TIM.2013.2277514.

R. Vaughan. “Massively multi-robot simulation in stage.” *Swarm Intelligence*, 2(2-4):189–208, 2008.

J. Wilson and P. Wright. “Head-mounted display efficacy study to aid first responder indoor navigation.” *Proceedings of the Institution of Mechanical Engineers, Part C: Journal of Mechanical Engineering Science*, 223(3):675–688, 2009.

J. M. Winters, Y. Wang, and J. M. Winters. “Wearable sensors and telerehabilitation.” *IEEE Engineering in Medicine and Biology Magazine*, 22(3):56–65, May 2003. ISSN 0739-5175. doi: 10.1109/MEMB.2003.1213627.

H. A. Yanco and J. Drury. ““Where am I?” Acquiring situation awareness using a remote robot platform.” In *2004 IEEE International Conference on Systems, Man and Cybernetics*, volume 3, pages 2835–2840 vol.3, Oct 2004. doi: 10.1109/ICSMC.2004.1400762.

P. Yang, W. Wu, M. Moniri, and C. C. Chibelushi. “Efficient Object Localization Using Sparsely Distributed Passive RFID Tags.” *IEEE Transactions on Industrial Electronics*, 60(12):5914–5924, Dec 2013. ISSN 0278-0046. doi: 10.1109/TIE.2012.2230596.

J. Yi, J. Zhang, D. Song, and S. Jayasuriya. "IMU-based localization and slip estimation for skid-steered mobile robots." In *2007 IEEE/RSJ International Conference on Intelligent Robots and Systems*, pages 2845–2850, Oct 2007. doi: 10.1109/IROS.2007.4399477.

T. Yilmaz, R. Foster, and Y. Hao. "Detecting Vital Signs with Wearable Wireless Sensors." *Sensors*, 10(12):10837, 2010. ISSN 1424-8220. doi: 10.3390/s101210837.

Z. Zhang. "Microsoft Kinect Sensor and Its Effect." *IEEE MultiMedia*, 19(2):4–10, Feb 2012. ISSN 1070-986X. doi: 10.1109/MMUL.2012.24.