

# Effective Semi-autonomous Telepresence

Brian Coltin<sup>1</sup>, Joydeep Biswas<sup>1</sup>, Dean Pomerleau<sup>2</sup>, and Manuela Veloso<sup>1</sup>

<sup>1</sup> School of Computer Science, Carnegie Mellon University, Pittsburgh, PA, USA

{[bcoltin](mailto:bcoltin@cs.cmu.edu), [joydeepb](mailto:joydeepb@cs.cmu.edu), [mmv](mailto:mmv@cs.cmu.edu)}@cs.cmu.edu

<sup>2</sup> Intel Research, Pittsburgh, PA, USA

[dean.a.pomerleau@intel.com](mailto:dean.a.pomerleau@intel.com)

**Abstract.** We investigate mobile telepresence robots to address the lack of mobility in traditional videoconferencing. To operate these robots, intuitive and powerful interfaces are needed. We present CoBot-2, an indoor mobile telepresence robot with autonomous capabilities, and a browser-based interface to control it. CoBot-2 and its web interface have been used extensively to remotely attend meetings and to guide local visitors to destinations in the building. From the web interface, users can control CoBot-2's camera, and drive with either directional commands, by clicking on a point on the floor of the camera image, or by clicking on a point in a map. We conduct a user study in which we examine preferences among the three control interfaces for novice users. The results suggest that the three control interfaces together cover well the control preferences of different users, and that users often prefer to use a combination of control interfaces. CoBot-2 also serves as a tour guide robot, and has been demonstrated to safely navigate through dense crowds in a long-term trial.

**Keywords:** telepresence, mobile robots.

## 1 Introduction

With the advent of the Internet and videoconferencing software, people have become more and more connected. This is particularly evident in the office, where work groups span countries and continents, but still keep in touch. However, these teleconferencing solutions leave much to be desired due to a lack of mobility—the user can only see what the camera is pointing at.

A mobile telepresence platform is much more powerful than these static videoconferencing solutions. Employees can move down the hall to visit their coworker's office from their own home on another continent, engineers can inspect overseas factories remotely, and collaboration between distant coworkers is enhanced. To this end, mobile telepresence robots have been developed which allow users to physically interact with and move through their environment, including the Willow Garage Texai and the Anybots QB [7]. We have developed CoBot-2, a mobile telepresence robot which is controllable through a web browser. CoBot-2 is unique in the extent of its autonomy: although given high-level instructions by a human, it localizes and navigates through the building entirely autonomously.

In spite of the ever increasing capabilities and robustness of autonomous robots, they are still incapable of performing some tasks on their own. We postulate that even as autonomous robots become increasingly capable, unexpected situations will remain where the assistance of a human is required. We envision this relationship between humans and robots as being *symbiotic* rather than exploitative, such that both the human and robot benefit from the relationship [13]. In the case of CoBot-2, the robot enables the human to visit remote locations, localizing and navigating autonomously to the greatest extent possible. On the other side, the human observes CoBot-2's motion through video to ensure that CoBot-2 doesn't collide with any unseen obstacles, and may reset CoBot-2's localization if the robot becomes lost.

For both telepresence and other forms of assisted autonomy, the effectiveness of the robot directly depends on the ability of humans to interact with the robot via its user interface. This user interface should empower people with little to no training to fully grasp the state of the robot, and to control the robot to perform complex tasks intuitively and quickly. CoBot-2's user interface is designed to be used from a web browser, and the robot's state is displayed in real-time. The interface allows the user to drive CoBot-2 in three ways: a joystick interface, with directional keys to turn and move forward (with buttons or the keyboard), by clicking on a destination point on the floor in the image from CoBot-2's camera, or by clicking on a destination point on the map of the building. A brief user study was conducted to determine the preferences of users between the three control interfaces. Each of the three interfaces were the preferred control input for some subset of the users, so we conjecture that all of the control modalities should be included in the user interface.

First, we will discuss related work in teleoperation interfaces, and then the hardware and autonomous aspects of CoBot-2's software. Next, we will describe the web interface in detail, followed by the user study procedure and results. Finally, we will discuss CoBot-2's debut as a tour guide in a day-long trial.

## 2 Related Work

Vehicles have been teleoperated since at least 1941, when pilotless drone aircraft were used to train pilots [5]. Remote operation has since become even more feasible and practical with the invention of the Internet. The first robot capable of manipulating the environment from the web was most likely the Mercury Project, a robotic arm which could explore a sandbox, deployed by Goldberg et. al. in 1994 [6]. This was followed by a mobile robot, Xavier, in late 1995, which could be sent to points throughout a building via a web browser [9]. In 1998, RHINO was developed to guide visitors at a museum. It combined local interactions with museum visitors with a web interface to observe the robot remotely [3]. These early web interfaces required the user to click on a link or refresh the page to update the robot status due to the limitations of HTML at the time. Recently robots teleoperated over the web have become more common, such as [14]. See [12] for a discussion of several other robots teleoperated

over the web. CoBot-2's web interface uses AJAX and HTML5 to display live updates of the robot's state in a web browser without the installation of any browser plugins. Currently, at least eight companies have developed commercial telepresence robots; some of them with web browser interfaces [7].

Teleoperation interfaces are used in space missions. On the NASA Mars Exploration rovers, the robots autonomous behaviors are complemented by controllers on Earth. Due to the long communication delay between the Earth and Mars, teleoperation in this environment is particularly challenging [11]. Teleoperation interfaces have also been tested in a competitive environment with the RoboCup rescue competitions. Teams have focused on fusing information with the video display [1], and have also used principles learned from video games [8] to make their interfaces easy to learn and use. These robots operate in difficult to traverse rescue environments, and the operators still require extensive training to perform most effectively. Similarly to telepresence, tele-immersion has enabled people to interact with each other in virtual environments [10].

The human-robot interaction community has extensively studied robot teleoperation. Some of the challenges include a limited field of view, determining the orientation of the robot and camera, depth perception, and time delays in communication with the robot [4]. CoBot-2's teleoperation interface attempts to address each of these problems.

### 3 CoBot-2's Autonomous Capabilities

CoBot-2 is a new model of the original CoBot presented in [2] and [13], with new capabilities relating to telepresence. In terms of hardware, CoBot-2 uses the same base as the original CoBot (with omnidirectional motion and LIDAR) with the addition of a touch screen tablet, a StarGazer sensor, and a pan/tilt/zoom camera (see Fig. 1)<sup>1</sup>. CoBot-2 is able to localize itself and navigate through the building autonomously.

From CoBot-2's web interface, the user may command CoBot-2 to move to a global position on a map or a position relative to the robot. Although initiated by the user, these commands rely on the autonomous functionality of CoBot-2, particularly localization, navigation and obstacle avoidance.

#### 3.1 Localization

CoBot-2 uses a particle filter on a graph-based map for localization. For sensor feedback, it uses odometry and StarGazer estimates. StarGazer is an off-the-shelf localization system, which uses regularly placed fiducials on the ceiling. Each StarGazer fiducial consists of a pattern of retroreflective dots arranged on a grid, and is detected by a ceiling facing infrared camera mounted on the robot.

The location estimate of the robot is chosen as the mean of the cluster of particles with the largest weight. Localization is used for autonomous navigation

---

<sup>1</sup> Special thanks to Mike Licitra for designing and constructing CoBot-2, and to John Kozar for building the tablet and camera mount.



**Fig. 1.** CoBot-2 is equipped with 1) an omnidirectional base; 2) a 4 hour onboard battery supply; 3) a Hokuyo LIDAR sensor to detect obstacles; 4) a touch-screen tablet with wireless for onboard processing; 5) a StarGazer sensor; and 6) a pan/tilt/zoom camera

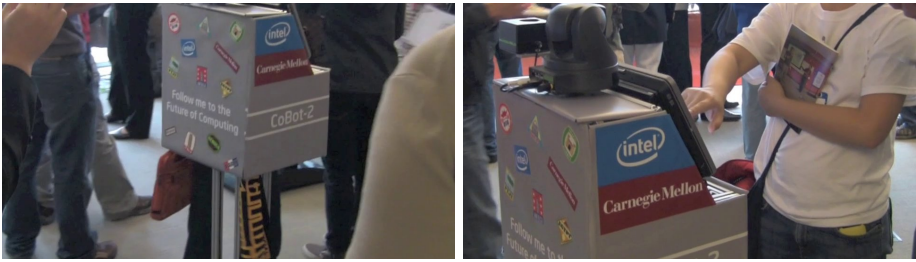
when moving to a global position. In the rare event that localization fails and CoBot-2 is unaware of its position, it may ask a human for help, who will click a map to set the localization position [13].

### 3.2 Navigation

CoBot-2 navigates autonomously to move to a global position on the map. The map of the building is represented as a graph of edges connecting vertices. Each edge is associated with a width indicating the distance the robot is permitted deviate from the edge (for obstacle avoidance). This distance is generally the width of the space the robot is navigating. To navigate between two locations, CoBot-2 finds the nearest points on the graph to these two locations and travels on the shortest path between the two points on the graph. CoBot-2 travels in a straight line while entering or exiting the edges of the graph.

### 3.3 Obstacle Avoidance

Every motion primitive is executed while avoiding obstacles. CoBot-2 avoids obstacles (as detected by its LIDAR sensor) by side-stepping around them. This is possible with the omnidirectional drive, which permits decoupled translation and rotation of the robot. While side-stepping around obstacles, the obstacle avoidance algorithm limits the extent to which CoBot-2 deviates from the center of the edge on the map. If no free path is found to avoid the obstacles, the robot stops and waits until its path is cleared. Once clear of obstacles, the robot moves back to the center of the edge.



**Fig. 2.** CoBot-2 successfully navigated through dense crowds autonomously at the open house for an entire day, guiding visitors to exhibits

### 3.4 Open House Guide

In addition to its telepresence capabilities, CoBot-2’s autonomous navigation has been used for a tour guide robot. To demonstrate and test these capabilities, CoBot-2 served as a tour guide at an all day open-house for a total of six hours. CoBot-2 successfully guided visitors through dense crowds at the open house for the entire day (see Figure 2).

CoBot-2 was supplied beforehand with the locations of the exhibits, which were marked on the web interface’s map and displayed on the touch screen. When the user selected an exhibit, CoBot-2 guided visitors to the exhibit, explaining along the way and avoiding obstacles, including people. If CoBot-2’s path was blocked, it said “Please excuse me” until it could move again.

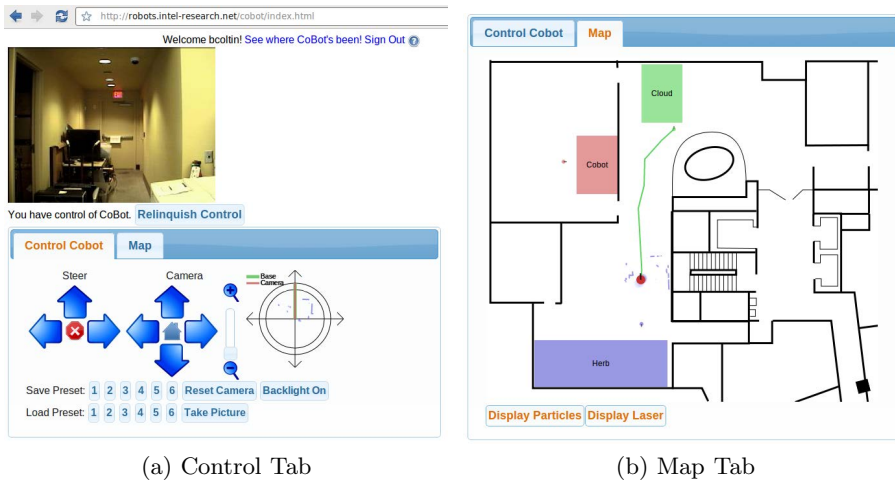
In spite of the dense crowds, which included people of all ages and sizes, and dangerous obstacles such as chairs, table overhangs and posters, which CoBot-2’s lasers could not detect, CoBot-2 did not have a single collision or accident for the entire day, while still moving at a reasonable human walking speed. During the open house, CoBot-2 was not closely monitored by humans— at some points, we did not even know where the robot was and had to go search for it.

One interesting aspect of the open house was seeing people’s responses to the robot. They quickly became accustomed to its presence, and went from seeing it as a surprising novelty to moving comfortably and naturally without being bothered by the robot’s presence. CoBot-2’s interactions at the open house showcased the exciting potential of robots which can interact naturally and safely with humans in unconstrained indoor environments, as telepresence robots must do.<sup>2</sup>

## 4 Web Interface

Users interact with CoBot-2 through a web-based browser interface. With the web-based interface, no special software needs to be installed to interact with

<sup>2</sup> A video of CoBot-2’s interactions with visitors at the open house is available at <http://www.youtube.com/watch?v=ONawEFNXZkE>.



**Fig. 3.** CoBot-2’s web interface, with the Control and Map tabs visible. In the map tab, CoBot-2’s current position, path and LIDAR readings are shown. Users may click on the map to set CoBot-2’s localization position or travel to a point.

CoBot-2. Furthermore, the software runs on multiple devices— desktop computers, mobile devices such as smartphones, and CoBot-2’s own touch screen. The web client repeatedly requests robot state information ten times per second, and requests new images from the robot’s camera five times per second.

A username and password are required to control the robot. If a user is not logged in, they may still view images from the robot and its status but are unable to issue commands to the robot. If multiple users are logged in to the server, only one may control the robot at a time to prevent confusion and contention.

Next, we will examine each component of the web interface. The user may switch between two tabs in the web interface: the Control tab, which contains buttons to control the camera and drive the robot, and the Map tab, which shows the map of the environment and the robot’s position. The image from the robot’s camera is always displayed above the two tabs in the remote interface.

#### 4.1 Camera Controls

From the control tab, users can control the robot’s camera. There are arrow keys to move the camera, surrounding a “home” button which returns the camera to a forward facing position suitable for driving. Next to these arrows is a slider with which the user can set the zoom level of the camera. The rate of the camera’s motion when using the arrow keys depends on the level of the camera’s zoom— at higher zoom levels the camera moves more slowly to allow more precise control. All of these commands have visual icons representing their function (such as magnifying glasses with “+” and “-” for zooming, a picture of a house for the home button) and are associated with keyboard shortcuts (see Fig. 3a). When

the user clicks on an arrow or presses the associated keyboard shortcut, the arrow button on the screen becomes pressed, providing visual feedback.

The Control tab also allows the user to configure six preset camera states with a “save” and “load” button for each preset state, which is convenient for attending meetings with CoBot-2 where there are fixed camera positions for looking at slides, the speaker, or other meeting attendees. Adjacent to the preset buttons is a button to turn the backlight on or off. Toggling the backlight enables the user to optimize image quality for different lighting conditions.

## 4.2 Steering Arrows

Three arrows for steering CoBot-2 are displayed directly to the left of the camera arrows (See Fig. 3a). Although CoBot-2 is capable of omnidirectional motion, these arrows only allow turning in place and moving directly forwards. This is the type of interface we believed would be most familiar to users, and which would also cause the least confusion in conjunction with the movable camera. In the center of the arrows is an emergency stop button. The robot can also be moved with the arrow keys on the keyboard, and can be stopped by pressing space. CoBot-2 autonomously performs obstacle avoidance while controlled with the arrow keys, adjusting its velocity to avoid obstacles.

## 4.3 Compass

On the right side of the Control tab is a compass-like object which displays the relative orientation of CoBot-2 and its camera, each with a colored “compass needle”. The needle representing CoBot-2’s camera always points “north” on the compass, and the needle representing the orientation of the robot’s base moves relative to this based on the current value of the camera’s pan angle. The LIDAR readings are displayed on the compass so that the user can visualize objects in the immediate vicinity of CoBot-2 (see Figure 3a).

The compass is intended to provide situational awareness to a user who has moved the camera, so that they can tell which way the camera is facing, which is essential when driving with the arrow keys. It also allows an experience user to predict which obstacles CoBot-2 will avoid automatically via the LIDAR readings.

CoBot-2 can also be turned precisely using the compass. With the arrow keys, commands are sent to turn a set, small distance when pressed, and to stop turning when the keys are released. However, there is network latency sending the command to stop turning, which will often cause CoBot-2 to overturn. With the compass, CoBot-2 is told to rotate a set angle, so network latency does not cause overturning.

## 4.4 Camera Image: Visualization and Control

A 320x240 image from CoBot-2’s camera is displayed at the top of the display, and refreshed every fifth of a second. Network latency is manageable enough that overseas users can control CoBot-2 without difficulty.



**Fig. 4.** At left, the robot views a scene at the default zoom level. At right, the user zooms in to the boxed area at the maximum level of 18X. The powerful zoom functionality of the robot allows users to inspect small or distant objects, and even read text remotely.

In addition to displaying the environment, the image is used for control. When a user clicks on the image, the camera moves to center on the selected point, providing more precise control than the arrow keys. Additionally, by scrolling the mouse wheel with the cursor over the image, the camera zooms in or out (see Fig. 4).

By clicking on the ground plane in the image while holding the shift key, the user can move CoBot-2 to a specific spot on the floor. The robot computes the clicked spot by considering the height of the robot and the pan, tilt and zoom levels of the camera. We discard clicks above the horizon, and limit the maximum distance travelled by a single click to reduce the consequences of mistaken clicks. With this control scheme, it is irrelevant which direction the camera is facing, and lack of awareness of the robot's camera and orientation is not an issue. Furthermore, moving the robot by clicking on the image does not suffer from the latency problems of the arrow keys.

## 4.5 Map

The map is shown as a separate tab so that the user need not scroll the browser window to view the entire interface. It displays a visualization of CoBot-2's environment, position and orientation. While CoBot-2 moves to a point chosen by the user, the path to its' destination is displayed.

Users set a destination for CoBot-2 to travel to by clicking on the map, then dragging and releasing the mouse button to set the orientation. Additionally, by holding the shift key while choosing a location, the user can set CoBot-2's localization position from the Map tab. This feature is useful in the rare event that CoBot-2's localization becomes lost.

In the Map tab, users may still access much of the functionality from the Control tab, such as driving, emergency stop, and moving the camera, through keyboard shortcuts. The image from the robot's camera remains visible.



## 4.6 Touch Screen and Mobile Interface

One of the goals behind using a web-based interface is to make CoBot-2 accessible from its own touch screen and from mobile devices. However, there are several shortcomings preventing the interface from working as is on touch screens and mobile devices. First, the shift key must be held down to move CoBot-2 by clicking on either the image or the map, but the shift key is not available on touch screens. To resolve this, on touch screens we provide a radio toggle button enabling the user to change modes for both the image and the map. The user then performs a normal click to activate the selected function. On the image, the radio buttons toggle between looking at a point with the camera or moving to a point on the floor, and for the map, the radio buttons switch between setting CoBot-2's localization position and moving to a point on the map.

Second, to set the orientation of CoBot-2 in the map, the user must press the mouse button, drag the mouse and release to set the orientation. This is not possible on certain mobile devices, so for the touch interface, one click sets the position, and a second click sets the orientation.

A third shortcoming is that touch screens are smaller, so the entire interface does not fit on CoBot-2's tablet. To avoid the need to scroll the page, CoBot-2's camera image is only displayed in the Control tab for cell phones and tablets.

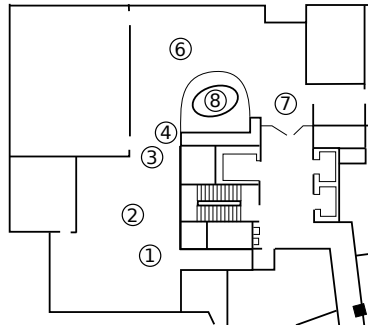
## 5 User Study

To determine the effectiveness of the various user interfaces for controlling CoBot-2, we conducted a small user study. The participants were seven students or recent students in robotics and/or computer science, none of whom had interacted with CoBot-2 before. Their experience with robots varied widely, from people who had never used robots before to people who had worked with them extensively for years. The participants were in different buildings than the robot. The purpose of the study was to demonstrate the telepresence interface's effectiveness and to determine users' preferences for control interfaces among the directional controls (arrow keys), compass, clicking on the image, and clicking on the map. Our hypothesis was that users would prefer to use higher level interfaces: first the map, then clicking on the image, and using the arrow keys least of all.

### 5.1 Setup

The participants in the study were sent instructions by email, and asked to visit CoBot-2's website and login remotely. They then had to complete a set of tasks via the web interface, each using only one of CoBot-2's control interfaces. For the final task, the participants were told to use whichever control scheme they preferred. See Figure 5 for a detailed explanation of the tasks.

After completing the tasks in the instructions, participants were asked to fill out a survey. The survey asked questions to confirm that the participants had completed the assigned tasks, and to rank from 1 to 10 how much they used



**Fig. 5.** In the user study, the participants: 1) Begin with CoBot-2 at this position; 2) Read a nametag at position 2 by moving the camera with the arrows; 3) Drive CoBot-2 to point 3 with the arrow keys; 4) Read a secret message on a post-it note at 4; 5) Rotate CoBot-2 with the compass at 3; 6) Navigate to read a poster at 6 by clicking on the image; 7) Navigate to position 7 with the map; 8) Identify a food item in the kitchen area at 8, using any control scheme; and 9) Return to the original position at 1. Not shown on the map are numerous cubicles, chairs and desks arranged in the lab.

each interface in the final task. The participants were also asked open-ended questions to describe the most frustrating and most enjoyable aspects of CoBot-2's interface, and for any other comments they may have.

## 5.2 Results

All of the participants completed the tasks in approximately fifteen to thirty minutes. The results of the survey did not show a preference for higher level interfaces. We believed that users would greatly prefer driving by clicking on the image or map. However, in the final task, all but one of the subjects used the arrow keys to a significant extent. Furthermore, each of the interfaces was used to the greatest extent by some subset of the participants— two used the map the most, one clicked on the image the most, two used the arrow keys the most, and two used multiple interfaces an equal amount. This indicates that each of the three control schemes is valuable, and the inclusion of multiple control interfaces increases the usability of the web interface. See Table 1 for the complete results.

The impetus for providing multiple interfaces is strengthened by the fact that only two participants used a single interface to complete the final task— subject 4, who used only the arrow keys, and subject 5, who greatly preferred clicking on the image. The remaining participants used a combination of all three control interfaces. Different control methods are appropriate for different situations. We have found the arrow keys to be appropriate for driving small distances and maneuvering in the vicinity of obstacles such as tables, chairs, and posters on easels which cannot be detected by the ground level LIDAR sensor. Clicking on the camera image works very well for line of sight movement, but requires multiple clicks for large distances, and is not feasible for small distances where

**Table 1.** The responses of the seven participants to the question, “To what extent (from 1 to 10, 10 is the greatest extent) did you use each of the following interfaces when driving CoBot-2 home?”

Subject	Arrows	Click on Image	Map
1	7	5	10
2	8	4	7
3	8	7	8
4	9	1	1
5	1	10	1
6	5	5	5
7	5	8	9

the robot cannot see the floor immediately around its base. Similarly, the map is excellent for traversing long distances, but is not very precise. We suggest that user interface designers provide multiple control interfaces to accommodate both user preferences and the requirements of diverse use scenarios.

Another interesting factor observed in the user study was the amount of trust users placed in the robot. The majority of users were willing to drive by clicking on the map or image and expected CoBot to avoid obstacles. However, subjects 4 and 7 commented in the survey that they did not trust the robot to avoid obstacles while moving autonomously. For subject 4, this lack of trust excluded clicking on the map or image at all in the final task. When asked to use the map interface to go to a point, Subject 7 accomplished this by setting individual waypoints for the robot a small distance apart to prevent collisions, also due to a lack of trust. With more familiarity with the robot and a better understanding of the robot’s capabilities, these users would feel more comfortable allowing the robot to move autonomously with the map interface.

## 6 Conclusion

CoBot-2, a mobile telepresence robot, enables visitors to interact with an environment remotely through its browser-based teleoperation interface. CoBot-2 has been shown to perform effectively and safely in indoor environments with humans over a long term. Control of CoBot-2 is symbiotic with a variable level of autonomy, in which the user controls CoBot-2 through one of three control interfaces. Control interface preference varied between users, with many users taking advantage of multiple interfaces. CoBot-2’s semi-autonomous control schemes enabled easy and safe navigation with little user effort.

**Acknowledgements.** This research was funded in part by the NSF and by Intel. The views expressed in this paper are those of the authors only.

The authors would like to thank Mike Licitra for designing and building CoBot-2, and John Kozar for designing and building the mount for the tablet and camera. Special thanks also goes to the participants in the user study and the guests who interacted with CoBot-2 at the open house.

## References

1. Baker, M., Keyes, B., Yanco, H.A.: Improved interfaces for human-robot interaction in urban search and rescue. In: Proc. of the IEEE Conf. on Systems, Man and Cybernetics (2004)
2. Biswas, J., Veloso, M.: Wifi localization and navigation for autonomous indoor mobile robots. In: IEEE International Conference on Robotics and Automation, pp. 4379–4384 (May 2010)
3. Burgard, W., Cremers, A., Fox, D., Hähnel, D., Lakemeyer, G., Schulz, D., Steiner, W., Thrun, S.: The interactive museum tou-guide robot. In: Proc. of the National Conference on Artificial Intelligence, AAAI (1998)
4. Chen, J.Y.C., Haas, E.C., Barnes, M.J.: Human performance issues and user interface design for teleoperated robots. IEEE Transactions on Systems, Man and Cybernetics 37(6), 1231–1245 (2007)
5. Fong, T., Thorpe, C.: Vehicle teleoperation interfaces. Autonomous Robots 11 (2001)
6. Goldberg, K., Mascha, M., Genter, S., Rothenberg, N., Sutter, C., Wiegley, J.: Desktop teleoperation via the world wide web. In: Proc. of IEEE Int. Conf. on Robotics and Automation (1995)
7. Guizzo, E.: When my avatar went to work. IEEE Spectrum, 26–31, 48–50 (September 2010)
8. Kadous, M.W., Sheh, R.K.M., Sammut, C.: Effective user interface design for rescue robotics. In: Proc. of ACM/IEEE Int. Conf. on Human-Robot Interaction (2006)
9. Koenig, S., Simmons, R.: Xavier: A Robot Navigation Architecture Based on Partially Observable Markov Decision Processes, pp. 91–122 (1998)
10. Lien, J.-M., Kurillo, G., Bajcsy, R.: Skeleton-Based Data Compression for Multi-Camera Tele-Immersion System. In: Bebis, G., Boyle, R., Parvin, B., Koracin, D., Paragios, N., Tanveer, S.-M., Ju, T., Liu, Z., Coquillart, S., Cruz-Neira, C., Müller, T., Malzbender, T. (eds.) ISVC 2007, Part I. LNCS, vol. 4841, pp. 714–723. Springer, Heidelberg (2007)
11. Maimone, M., Biesiadecki, J., Tunstel, E., Chen, Y., Leger, C.: Surface Navigation and Mobility Intelligence on the Mars Exploration Rovers, pp. 45–69 (March 2006)
12. Martin, R., Sanz, P., Nebot, P., Wirz, R.: A multimodal interface to control a robot arm via the web: A case study on remote programming. IEEE Transactions on Industrial Electronics 52 (2005)
13. Rosenthal, S., Biswas, J., Veloso, M.: An effective personal mobile robot agent through symbiotic human-robot interaction. In: Proc. of Int. Conference on Autonomous Agents and Multi-Agent Systems (May 2010)
14. Sim, K., Byun, K., Harashima, F.: Internet-based teleoperation of an intelligent robot with optimal two layer fuzzy controller. IEEE Transactions on Industrial Electronics 53 (2006)