Web Interfaces for Mobile Robots in Public Places

Dirk Schulz† Wolfram Burgard†† Dieter Fox‡ Sebastian Thrun‡ Armin B. Cremers†
†Department of Computer Science III, University of Bonn, Germany
††Department of Computer Science, University of Freiburg, Germany
‡School of Computer Science, Carnegie Mellon University, Pittsburgh, PA, USA

Abstract

The emergence of the World Wide Web provides a unique opportunity to connect robots to the Internet, enabling people all over the World to control them and monitor their operation. This paper describes a series of Web interfaces, designed to remotely operate mobile robots in public places through the Web. The design of these interfaces specifically addresses issues such as low bandwidth of Internet connections, control brokering, and shared control as well as interaction with people in the robot's environment, which arise naturally in applications with Web-based robot control. The interfaces have been tested extensively using two deployed service robots, which were installed as interactive tour-guides in two museums. The Web interfaces proved highly effective to provide a "tele-presence" to people all over the world. The paper also discusses trade-offs and limitations of Web-based robots that interact with people in populated, public places.

1 Introduction

The recent growth of the World Wide Web provides unique opportunities to bring robots closer to people. The vision behind such endeavors ranges from relatively simple Web-based inspections and surveillance applications, to highly versatile applications that use robots connected to the Web to establish a remote "tele-presence" in dynamic and populated environments. In the latter scenario, robots play the role of a physical mediator, enabling remote people to acquire information, explore, manipulate, communicate, and interact physically with people far away.

Web-based tele-control differs from conventional tele-operation in several aspects:

- The delay and the throughput of the Internet are highly unpredictable. Variations by several orders of magnitude are not uncommon. Such conditions differ from those found in traditional tele-operation interfaces, which typically assume fixed and guaranteed delays and throughput rates. With communication as unpredictable as it is on the Web, *autonomy* plays a much greater role for Web-controlled robots. At a minimum, the Web interface must ensure "safe" operation even if communication breaks down. More desirable, however, are interfaces that enable Web users to control a robot sensibly even if the communication is highly unreliable. Thus, robots which are remotely controlled over the Web require a much higher degree of autonomy than conventional tele-operation.

- Web users typically lack the technical education and skill required by most existing tele-operation interfaces. Graphical displays and easy-to-use command interfaces are therefore critical, posing challenges on the design of Web interfaces that are typically not encountered in more traditional tele-operation applications. Information gathered by the robot (such as graphical maps of an environment) may not only be robot-usable, but must also be easy to understand for people on the Web.

- If one wants to make a single physical robot amenable to many people on the Web, the classical paradigm of *exclusive control*, in which a single person controls the robot over extended periods of time, is clearly undesirable. Instead, innovative mechanisms for brokering control among multiple Web users are needed. This is particularly important for the Web, where the number of potential "users" of a robotic system is enormous.

- Some applications, such as the one described in this paper require that people on the Web *share* the control of the robot with other people physically located close to the robot. These other people might not use the Web at all; instead, they control the robot through more conventional means. This raises the issue of mixed-control interfaces where different types of commands must be mediated, in ways that minimally disrupts either group of users. The Web enables users to interact with remote people, who use the robot to establish a remote "tele-presence."

This paper describes a collection of Web interfaces, which recently were tested using robots operating in highly populated public places. The specific robots studied here were deployed in two museums, the Deutsches Museum in Bonn and the Smithsonian's National Museum of American History in Washington, DC. Both robots, Rhino [1, 2] and Minerva [13], performed the function of a robotic tour-guide, leading people in the museums through exhibitions. The Web was used to make the museum accessible to remote museum visitors, who watched images recorded in the museum and could control the robots' operation. Some of the Web-based operation took place during the opening hours of the museum, in which virtual and physical visitors shared the control of the robots. We also explored the utility of Web interfaces after opening hours, when Web users could assume exclusive control over the robots.

This paper describes a series of three interfaces, all of which developed for Web-based monitoring and control of interactive robots deployed in public places. These interfaces differ in their degree of abstraction

- The first interface enables visitors to send a robot to a user-specified target location anywhere in the museum, assuming that the target location is in fact reachable. Control brokering is achieved through a first-come-first-serve basis, using a limited-size queue to schedule requests.

- The second interface enables Web users to select exhibits and viewpoints. This interface explains exhibits graphically with pre-recorded information when the robot has reached an exhibit. Requests are scheduled according to the length of the tour and according to the time expired since the exhibit was chosen before.

- The third Web interface uses a voting scheme to arbitrate control among Web users. Here visitors can vote for a specific tour. At pre-scheduled points in time, the robot performs the function selected by the largest number of Web users.

The last two interfaces, whose commands are extremely high-level with an execution time frame of several minutes, are particularly well suited for sharing control between physical and virtual museum visitors.

This paper describes all three interfaces in depth, and discusses their advantages and limitations. During three weeks of on-line operation (1 week in Germany and 2 weeks in the US), approximately 4,000 Web users controlled the robots. At certain times, the robots were under exclusive control of the World Wide Web, while at other times Web users and conventional museum visitors shared control. Unfortunately, the effectiveness of the different interfaces is difficult to evaluate systematically, as most feedback we received is anecdotal in nature. This article also discusses usability aspects of the interfaces, and people's experience of using them.

2 Related Work

Web controlled robots have gained serious interest in the last few years. First, there is a variety of manipulators which can be used to tele-manipulate items located in their working area. The Web interfaces of these systems allow users to issue single commands which are executed immediately. Three early systems are the Mercury Project [3] installed in 1994, Australia's Tele-robot on the Web [12], which came on-line nearly at the same time, and the Tele-Garden [4], which replaced the Mercury robot in 1995. While the Mercury robot and the Tele-Garden enable Web users to perform different types of digging tasks, such as excavation of artifacts as well as watering and seeding flowers, the Tele-robot on the Web give Web users the opportunity to build complex structures from toy blocks. The interfaces of these robots use different techniques to visualize the actions of the robot and their effects. While most of the systems provide still images of the robot's operation area, the Tele-Garden system additionally uses CAD drawings to animate the state of the manipulator. Additionally to these manipulators, there is a growing number of Web cameras which deliver image streams to the Web. Some of these cameras, such as [5] are installed on a robot arm and can even be controlled over the Web by issuing single commands.

In order to let individual users tele-program a robot or to perform sequences of actions, more complex interfaces are required. A typical example is the PumaPaint Project which allows people to draw a complete painting by controlling a PUMA 760 robot arm. The Web interface provides a virtual canvas implemented in Java on which the user draws his painting. The interface forwards all actions of the user to the robot, so that almost the same image appears on the real canvas. The system also provides visual feedback in form of periodically updated live images from the robot. Hirukawa et al. [7] describe a Web operation interface, where Web users can perform manipulation tasks using a 3D graphics simulation contained in the Web browser. The specified tasks are first tested on a simulator. A verified sequence of actions can be transmitted to the real robot.

Additionally, there is a long list of mobile robots which can be controlled over the Web. Most of these systems provide exclusive remote control to a single person or provide queues to schedule user requests. KhepOnTheWeb [9] is a typical representative of mobile robots with a Web interface. Users can give elementary movement actions to the robots and observe them using several Web-cameras. Xavier [11] was probably the first mobile robot which operates in a populated office building controlled through the Web. Xavier can be advised by Web users to move to an office and to tell a knock-knock joke after arrival. The robot collects requests

Figure 1. On-board interface of the interactive museum tour-guide robot Rhino.

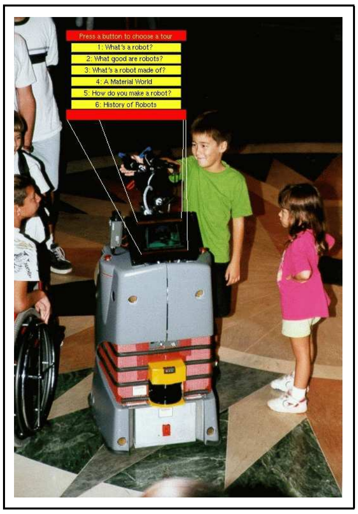

Figure 2. On-board interface of the museum tour-guide robot Minerva, implemented by a Java program and running on a touch-screen.

both off-line and on-line, and processes them during special working hours. After the successful execution of the mission, Xavier informs the Web user via email. Xavier's Web interface relies on client-pull and server-push techniques to provide images taken by the robot. It furthermore provides a map of the environment and indicates the robot's current position in regular intervals.

In dynamic environments, like a populated museum, unforeseen events may occur during task execution. Moreover, since the task of a tour-guide robot is to interact with people in the museum, the actions of the robot cannot always be predicted, as it is the case in most of the systems described above. Therefore, the control interfaces of our tour-guide robots, which are described in this paper, provide various forms of instant feedback to the users. For example, they continuously inform users about other users interacting with the robot, the current goals to be reached, and about the actions currently being carried out. Since they are based on Java applets, they provide frequent updates of the robot's state even over low-bandwidth connections. The interfaces also deliver images taken with the robot's cameras as well as with a camera mounted at a ceiling. Finally they are able to serve several people at the same time by different types of brokering techniques.

3 Software Architecture of the Museum Tour-Guide Robots

The ability to move safely through a dynamic environment plays a fundamental role for Web-based control. Hence, we will begin by briefly outlining our software architecture for autonomous navigation, referring the interested reader to more detailed literature. This software architecture plays an important role, as it guarantees safety and mission compliance under arbitrary latency conditions.

The overall software architecture of the museum tour-guide robots Rhino and Minerva consists of decentralized modules which are executed in parallel on-board and off-board, using sockets and a tetherless Ethernet bridge for communication [1, 13, 2]. At the lowest level, various modules interface directly to the robot's sensors and effectors (lasers, sonars, cameras, motors, pan/tilt unit, face, speech unit, touch-sensitive display, Internet server, etc.). On top of that, several navigation modules provide autonomous navigation capabilities, by carrying out functions like mapping, localization, collision avoidance, and path planning. The interaction level controls the robot's voice, head direction, and other interactive means such as the facial expression. Finally, both robots also employ high-level planners that coordinate entire tours at an abstract, symbolic level.

For the Web interface, this architecture provides autonomous operation, including the ability to navigate safely from point to point even in highly dynamic environments. The robot's safety and progress, thus, does not hinge on the availability of the Internet. In addition, the control software drastically reduces the amount of incoming sensor information, and extracts from it knowledge that is much easier to understand by people, such as graphical maps of the environment (floor plan, ceiling image).

The on-board user interface is an important aspect of tour-guide robots. In application domains such as museums, the user interfaces must be intuitive, so that untrained and non-technical users can operate the system with ease.

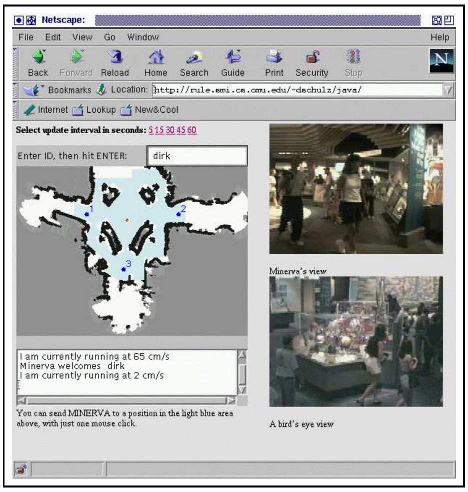

Figure 3. Minerva's exclusive Web control interface. Users give target positions by clicking inside the map. The robot moves to targets in a first-come-first-serve manner.

It must also be appealing, so that people feel attracted by the robot and participate in a tour. Figures 1 and 2 show the on-board user interfaces of the robots Rhino and Minerva, which enabled museum visitors to enter commands simply by clicking buttons or by touching the robot's touch-sensitive monitor. Additionally to displaying texts and graphics about the exhibits on the screen, the robots played pre-recorded sounds, for example, to explain exhibits or to ask for clearance. The design of appropriate Web interfaces, however, is even more challenging, since people on the Web need additional information to understand what is happening at the remote site.

4 Web Interfaces of the Museum Tour-Guide Robots

4.1 Interface #1: Exclusive Control by the Web

The first Web interface described here is designed for mobile robots under *exclusive* control of people on the Web. The purpose of the interface is to enable people all around the world to gather information interactively at a remote site (a museum). The main Web page of the interface is shown in Figure 3. The interactive part of the interface is implemented as one Java applet, which is located in the left part of the page. The right half of the page displays on-line images from the robot and from a ceiling-mounted camera using the client-pull technique.

The layout of the Java applet is dominated by a map of the exhibition area acquired with the robot's map-building component. In addition, the applet contains a text field for user registration and a larger text window for status information. The applet provides different types of information; within the map it displays the robot's current position as well as all the target locations which have been entered by Web users. The text window gives additional status information like the registration of new Web users, the current speed of the robot, and the name of the Web user who has chosen the currently approached target location.

To enter target locations, a user first has to register his name (or whatever name he likes to assume). After that, he can send the robot to arbitrary non-occupied locations in the map, simply by marking this location using his pointing device (e.g., mouse). To schedule the requests of different users, the system uses a small queue containing at most five requests which are served on a first-come-first-serve basis. To avoid long waiting times, the interface only accepts new target locations when the queue is sufficiently empty. When the robot reaches a target location, it takes a picture and informs everyone on the Web that the location has been reached, providing the name of the person who requested the target location. A second camera, mounted on a pan-tilt unit at a stationary location, tracks the robot and continuously takes pictures broadcasted via the Web.

The applet exchanges information with the robot control system over a TCP connection. To provide frequent updates of up to 4 times a second even under extremely low throughput, the applet initially downloads the complete map from the robot control system. Afterwards, the applet only receives the current position of the robot, the selected target locations, and eventually new status information. This small amount of information can be transferred at very high update rates even over low bandwidth connections.

4.2 Interface #2: Perpetual Monitoring With Exclusive Web Control

The second interface, used with the robot Rhino, is designed for monitoring the actions of the robot over the Internet. Just as the first one, it only allows Web users to take the control over the robot during dedicated Internet hours, but it also provides a means for Web users to watch the robot's operation while under control by museum visitors. The main design decision regarding the Web interface of Rhino is to increase the number of simultaneously served Web users instead of giving only a small number of individual persons the right to exclusively control the robot. To achieve a higher degree of concurrency, Rhino's Web interface has a limited number of exhibits that the robot can visit. On the Internet, every user can choose the exhibits and the corresponding viewpoint he wants the robot to provide information about. Thus, this interface is more abstract than the first. It also facilitates Web-based control during the museum's opening hours, as people in the museum, unaware of the Web interface, can still participate in tours whose composi-

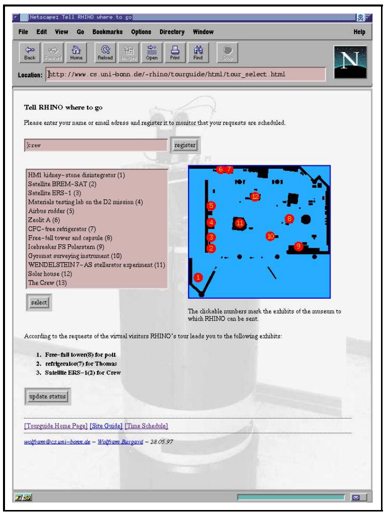

Figure 4. Control page of Rhino's Web interface. Users can choose exhibits they want the robot to go to as well as viewpoints.

tion (sequence of exhibits) is now dictated by Web users.

The interaction between Web visitors and the robot is carried out using three different Web pages. The "control page," shown in Figure 4, allows visitors to register their names and select target exhibits and viewpoints. The communication with the robot control system is carried out via CGI scripts. The high-level planner [6] schedules all requests into a round trip tour along the selected exhibits. Identical requests are merged and newly arriving requests are inserted into a running tour. In return, the control system always lists the current schedule on the Web page, along with the names of the users who chose the corresponding exhibits.

This interface contains two additional pages to give feedback to the Web users while the robot is carrying out its task. The "on-line images" page (see Figure 5) provides still images from the robot's camera and from two static wall-mounted cameras which give an overview of the exhibition hall (only one is active at any point in time). In addition, it displays a map of the environment including the position of the robot at the time when the pictures were taken. All displays are updated in regular but user-controllable time intervals using the client-pull technique. Consequently, Web users can adjust the update rate according to the bandwidth of their Internet connection. At the bottom, this page provides general status information while the robot is in motion, along with more specific information when the robot has reached an exhibit. This is achieved by integrating a Java applet, which receives the URL of a corresponding Web page from the robot control system via a UDP socket connection. A third page (see Figure 6) is designed to pro-

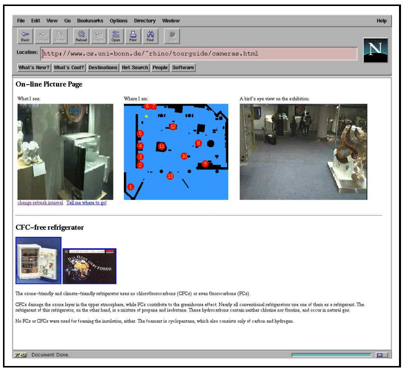

Figure 5. This Web page shows images of both cameras, one mounted at a fixed location and one on board the robot. The lower part of the screen provides hypertext information about the current exhibit.

Figure 6. The monitoring page provides smooth visualization of the robot's current position in a previously learned two-dimensional map of the environment. It also includes a ticker displaying the current actions of the robot.

vide a smooth visualization of the robot's path in the map. This page operates similar to the map display of Minerva's exclusive control interface described above. It contains a ticker displaying the current intentions (plan) of the robot. For example, it might announce the name of the next exhibit. Just as above, this page is also implemented using a Java applet which established a UDP socket connection to the robot's control system. For example, it uses the current position received from the control system to smoothly visualize the movements of the robot. The status information displayed in the ticker is received via the same connection.

Figure 7. Minerva's shared user and Web interface; The robot's user interface is replicated on the Web page (the applet with the yellow buttons).

4.3 Interface #3: Shared Control by Visitors and People on the Web

Both interfaces described above only support exclusive control of the robot. They provide no means for sharing the robot's functionality between users in the museum and virtual (remote) visitors. To enable physical and remote visitors to operate the robot simultaneously, we developed a Web interface which allows visitors to choose complete tours, instead of single target locations. Tours contain between three and six exhibits. This interface, which is implemented in Java, is executed on board the robot and displayed on the robot's touch-screen (see Figure 2). Internet users automatically download the interface as an applet, which then is executed by their Web browser (see Figure 7). All texts and images with information about the exhibits are simultaneously displayed on the touch-screen of the robot and by the Java applet. As a result, the visitors in the museum and the virtual visitors on the Internet have the same control interface. Minerva's Web interface additionally displays the current position of the robot in the Museum. As well as the previous interfaces it also delivers on-line images from an on-board camera as well as from a ceiling-mounted camera which tracks the robot in the museum. All this information is displayed on a single Web page (see Figure 7).

To decide which tour should be chosen next, Minerva follows a simple approach. In contrast to physical visitors in the museum, which select tours on a first-come-first-serve basis, Web users are voting. For this purpose, all applets are connected to the robot control system via a TCP socket connection. The task level control module counts all incoming votes of Web users and executes in regular time intervals a tour for the Web users based on the majority vote. The scores of the different tours are displayed on all applets.

5 Results

During three weeks of on-line mobile robot operation (1 week in Germany and 2 weeks in the US), approximately 4,000 Web users controlled the robots. Our evaluation is strictly anecdotal in nature, since we did not obtain systematic feedback from a sufficiently large fraction of the Web users. Nevertheless, we believe the Web interfaces were very attractive to people on the Web. A number of Web users left comments in our guest book or otherwise sent us feedback. A large number of people, often novices to the field, liked to control the robots but also enjoyed seeing other people interacting with the robot. Figures 8 and 9 show typical images recorded with the on-board and ceiling-mounted cameras.

In one experiment, the exclusive control interface of Minerva (interface #1) was on-line for 3 hours and 40 minutes. During this time, 66 users loaded the exclusive control page. They entered 278 target locations. Figure 10 shows a typical trajectory of Minerva when it was under exclusive control. This particular run was recorded during an Internet session lasting 73 minutes on September 4, 1998. The specific run is 2,144 meters long. The maximum velocity of Minerva was 163 cm/sec. The control system received 103 target locations. Consequently, the average distance between consecutive target locations was 20.8 meters.

Rhino's Web interface (interface #2) could be accessed for a total of 32 hours and 18 minutes. During this time, 2,060 people visited Rhino's Web pages. Rhino traveled over 18 km during its 6 day deployment period. Simultaneously to the Web visitors, Rhino guided over 2,000 physical visitors through the museum. There were 2,400 requests to explain exhibits. Rhino managed to successfully arrive at its exhibits and deliver the information to the visitors in 2,394 cases which corresponds to an overall success rate of 99.75%—in the other 6 cases, Rhino's operation was disrupted by a battery change which let to a restart of Rhino's queue. At peak traffic hours, over 150 people simultaneously tele-operated the robot through the museum, using the

Figure 8. Pictures taken with the robot's on-board camera.

Web interface described here.

Minerva's shared-control interface (interface #3) was on-line for 91 hours, and was accessed by 2,885 people. We counted 4,561 votes for the 6 different tours the robot offered to the visitors. Minerva traveled 38.5 km under shared Web and on-line control, which is 87.5% of the overall distance traveled by the robot during the entire exhibition. During 620 tours, Minerva provided information about 2,390 exhibits. A total number of 87 tours were scheduled for virtual (Web-based) visitors. The maximum number of votes for a tour at any point in time was 259.

6 Lessons Learned

While the original design of the three interfaces was somewhat evolutionary in nature, they are nevertheless distinct in the type of control they provide, and the way they allow for sharing control among Web users and between Web users and people who directly interact with the robot. Among the most important lessons that we learned during the evaluation phases were the following:

1. Since the delay and throughput of the Internet varies drastically, the communication must be extremely low-bandwidth. High initial start-up costs, inferred by the Java-based simulator that Web users had to download, are preferable over high-bandwidth communication during operation (e.g., periodically downloading GIF images to display the latest robot position). Rhino's interface, used fix graphics for displaying the robot's location in its map, the size of which made the display very slow (typical update frequency: once in ten seconds). In Minerva's case, a simulator was downloaded up front, from which point on only the robot's actual position had to be communicated. Here we achieved update frequencies of 1 to 5 Hertz.

2. If one wants more than a single person to use the interface, interfaces that support exclusive control may overly restrict the access to the robot. Nevertheless, an interface that gives Web users momentary control (e.g., by scheduling a target location in the queue) are more effective than voting-based interfaces, where the robot's actions are chosen by the majority. The voting interface didn't give individual users sufficient control to be truly excited; therefore, only considerably few Web users used this interface for robot control.

3. Interleaving Web-based control with control by the people that directly interact with the robot is a tricky issue. The technique developed for Minerva turned out to be effective in providing the same service to visitors in the museum and virtual visitors. Fortunately, the visitors turned out to be very cooperative and never complained when Minerva chose a tour for the Web-visitors.

4. Placing all important on-line information on a *single* Web page is essential. In fact, Rhino's interface, which was historically the first, did not comply with this rule. Minerva's interfaces were more effective in this aspect, measured by (anecdotal) user satisfaction.

Despite these differences, all interfaces had in common that they proved relatively easy to use and that they were popular—at least according to the feedback we received.

7 Conclusions

This paper presented three Web interfaces designed for remote control of mobile robots operating in public places. Our approach specifically addresses issues that arise when a large number of Web users control a single resource, such as control brokering and shared control with people in the robot's environment. It also addresses safety and ease-of-use under unpredictable latency and throughput due to the Internet. All interfaces described here provide high-level command and easy-to-understand graphical representation of information acquired by the robot. They use voting,

Figure 9. Typical on-line images recored with the ceiling-mounted camera tracking the robot.

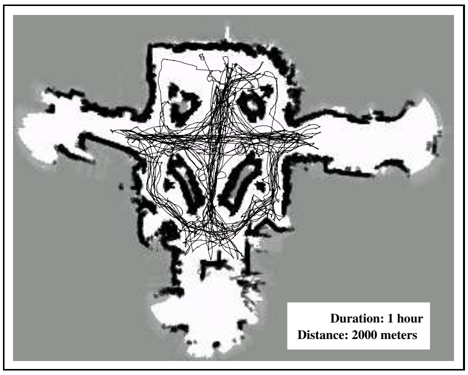

Figure 10. Minerva's path during a Web-control session (exclusive control).

queues, and first-come-first-serve to broker control. In two field experiments, carried out in two public museums, all interfaces were found to be highly suitable for Web-based monitoring and control.

While these interfaces have been evaluated in the context of robotic tour-guide applications, we conjecture that the lessons learned here can equally be applied to other domains involving robots that operate at public places and interact with people.

Future work will concentrate on improved visualizations. So far, we only used two-dimensional graphics to animate the robot's actions. We recently developed a prototype of a three-dimensional visualization, which incorporates predictive simulation techniques for delay compensation [10] and automatic viewpoint selection techniques [8]. We believe that the integration of such improved visualization methods will improve the effectiveness of Web interfaces for autonomous mobile robots in public places. A second issue that warrants future research is the concept of remote telepresence. To establish a "true" tele-presence, the interface must be bi-directional, relaying back video and audio feed from the end user to the people in the museum. However, the current Internet does not (yet) provide sufficiently high bandwidth and low latency for such experiments.

References

[1] W. Burgard, A. B. Cremers, D. Fox, D. Hähnel, G. Lakemeyer, D. Schulz, W. Steiner, and S. Thrun. The interactive museum tour-guide robot. In *Proceedings of the Fifteenth National Conference on Artificial Intelligence*, 1998.

[2] W. Burgard, A. B. Cremers, D. Fox, D. Hähnel, G. Lakemeyer, D. Schulz, W. Steiner, and S. Thrun. Experiences with an interactive museum tour-guide robot. *Artificial Intelligence*, 114(1-2), 2000.

[3] K. Goldberg, M. Mascha, S. Gentner, N. Rothenberg, C. Sutter, and J. Wiegley. Desktop tele-operation via the World Wide Web. In *Proceedings of the IEEE International Conference on Robotics and Automation*, 1995.

[4] K. Goldberg, J. Santarromana, G. Bekey, S. Gentner, R. Morris, J. Wiegley, and E. Berger. The telegarden. In *Proc. of ACM SIGGRAPH*, 1995.

[5] S. Goldberg, G. A. Bekey, and Y. Akatsuka. DIGIMUSE: An interactive telerobotic system for remote viewing of three-dimensional art objects. In *Proceedings of the IEEE IROS'98 Workshop on Robots on the Web*, 1998.

[6] D. Hähnel, W. Burgard, and G. Lakemeyer. GOLEX — bridging the gap between logic (GOLOG) and a real robot. In *Proc. of the 22nd German Conference on Artificial Intelligence (KI'98), Bremen, Germany*, 1998.

[7] H. Hirukawa, T. Matsui, and H. Onda. Prototypes of teleoperation systems via a standard protocol with a standard human interface. In *Proceedings of the IEEE International Conference on Robotics and Automation*, 1997.

[8] A. Hopp, D. Schulz, W. Burgard, D. Fellner, and A. B. Cremers. Virtual reality visualization of distributed tele-experiments. In *Proceedings of the 24th Annual Conference of the IEEE Industrial Electronics Society (IECON'98)*, 1998.

[9] P. Saucy and F. Mondada. KhepOnTheWeb: One year of access to a mobile robot on the Internet. In *Proceedings of the IEEE IROS'98 Workshop on Robots on the Web*, 1998.

[10] D. Schulz, W. Burgard, and A. Cremers. Robust visualization of navigation experiments with mobile robots over the internet. In *Proc. of the IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 1999.

[11] R. Simmons, R. Goodwin, K. Haigh, S. Koenig, and J. O'Sullivan. A layered architecture for office delivery robots. In *Proceedings of the First International Conference on Autonomous Agents*, Marina del Rey, CA, February 1997.

[12] K. Taylor and J. Trevelyan. Australia's telerobot on the Web. In *Proceedings of the 26th International Symposium On Industrial Robots*, 1995.

[13] S. Thrun, M. Bennewitz, W. Burgard, A. B. Cremers, F. Dellaert, D. Fox, D. Hähnel, C. Rosenberg, J. Schulte, and D. Schulz. MINERVA: A second-generation museum tour-guide robot. In *Proceedings of the International Conference on Robotics and Automation (ICRA '99)*, 1999.