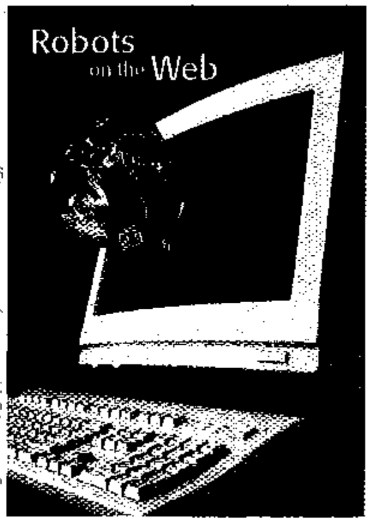

© Digital Stock; gripper courtesy of Barrett Technology

# Web Interfaces for **Mobile Robots** in **Public Places**

*Addressing Issues Such as Shared Control, Safety, and Ease of Use During Web Teleoperation of Museum Tour-Guide Robots*

by DIRK SCHULZ, WOLFRAM BURGARD, DIETER FOX, SEBASTIAN THRUN, and ARMIN B. CREMERS

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 "telepresence" 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. This article 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 article also discusses trade-offs and limitations of web-based robots that interact with people in populated, public places.

## Web-Based vs. Conventional Teleoperation

Web-based telecontrol differs from conventional teleoperation 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 teleoperation 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 that are remotely controlled over the web require a much higher degree of autonomy than conventional teleoperation.
- Web users typically lack the technical education and skill required by most existing teleoperation 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 teleoperation applications. Information gathered by the robot (such as graphical maps of an environment) must 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.


1070-9932/00/$10.00©2000IEEE

- Some applications, such as the one described in this article, 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 disrupt either group of users. The web enables users to interact with remote people, who use the robot to establish a remote "telepresence."

## Project Overview

The collection of web interfaces we developed were recently 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 who could control the robots' operation. Some of the web-based operation took place while the museum was open, which meant that both virtual and physical visitors shared the control of the robots. We also explored the utility of web interfaces after the museum closed, when web users could assume exclusive control over the robots. A series of three interfaces are detailed, all of which were 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 prerecorded 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 last chosen.
- The third web interface uses a voting scheme to arbitrate control among web users. Here visitors can vote for a specific tour. At prescheduled points in time, the robot performs the function selected by the largest number of web users.


*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, imlemented by a Java program and running on a touch-screen.*

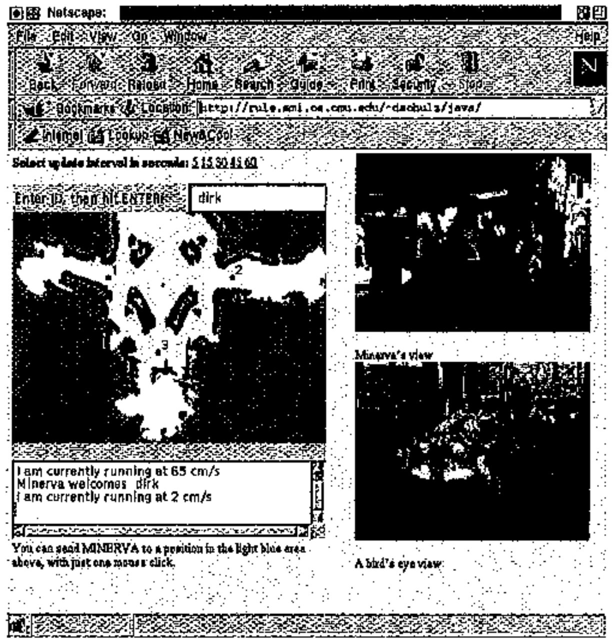


*Figure 3. Minerva's exclusive web control interface. Users give target positions by clicking inside the map. The robot moves to targets in a first-come, first-serve manner.*

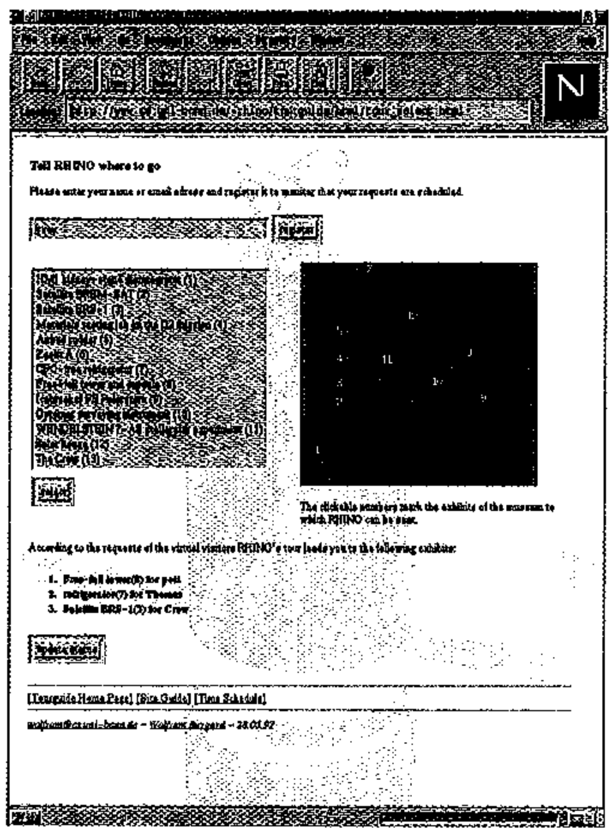


*Figure 4. Control page of Rhino's web interface. Users can choose exhibits they want the robot to go to as well as viewpoints.*

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 article describes all three interfaces in depth as well as their advantages and limitations. During three weeks of on-line operation (one week in Germany and two weeks in the US), approximately 4000 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.

## Related Work

Web-controlled robots have gained serious interest in the last few years. First, there is a variety of manipulators that can be used to telemanipulate items located in their working area. The web interfaces of these systems allow users to issue single commands that are executed immediately. Three early systems are the Mercury Project [3] installed in 1994; Australia's Telerobot on the web [12], which came on-line nearly at the same time; and the Telegarden [4], which replaced the Mercury robot in 1995. While the Mercury robot and the Telegarden enable web users to perform different types of digging tasks, such as excavation of artifacts as well as watering and seeding flowers, the Telerobot on the web gives 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 Telegarden system additionally uses CAD drawings to animate the state of the manipulator. In addition to these manipulators, there is a growing number of web cameras that 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 teleprogram 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 the 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 three-dimensional (3-D) 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 that 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 to operate 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 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 e-mail. Xavier's web interface relies on client-pull and server-push techniques to provide images taken by the robot. Furthermore, it 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 is the case in most of the systems described above. Therefore, the control interfaces of our tour-guide robots, which are described in this article, 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 on a ceiling. Finally, they are able to serve several people at the same time by different types of brokering techniques.

## 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 that are executed in parallel on-board and off-board, using sockets and a tetherless Ethernet bridge for communication [1, 2, 13]. 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.

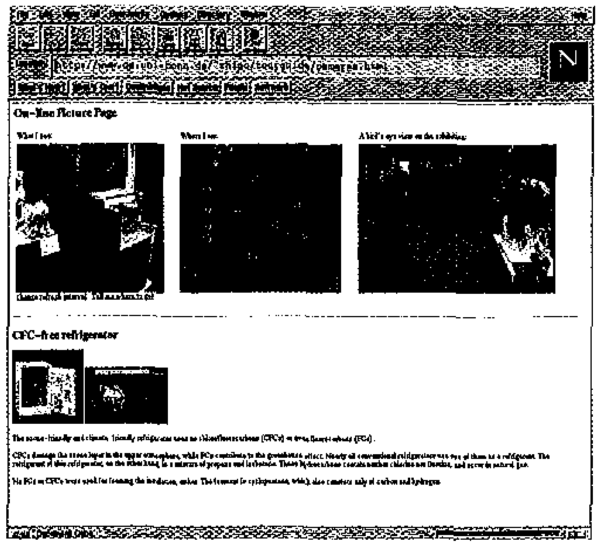


*Figure 5. This web page shows images of both cameras, one mounted at a fixed location and one on board the robot. The lower part of the screen provides hypertext information about the current exhibit.*

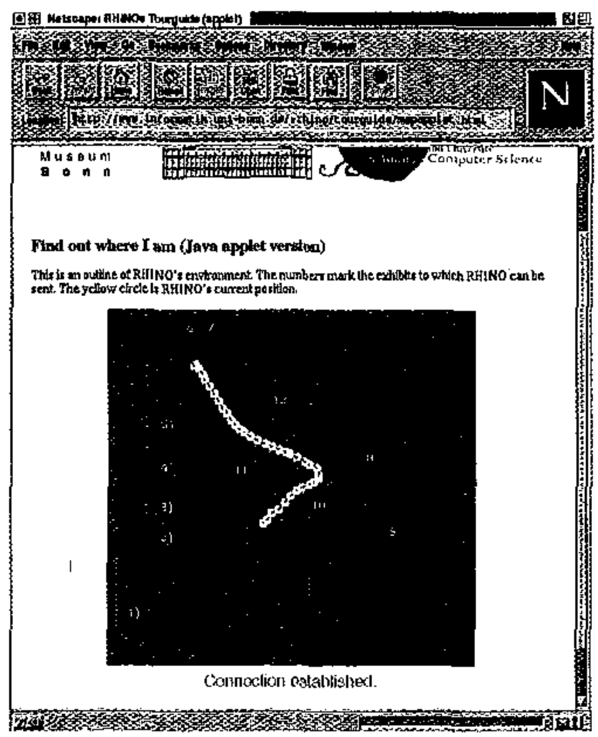


*Figure 6. The monitoring page provides smooth visualization of the robot's current position in a previously learned two-dimensional map of the environment. It also includes a ticker displaying the current actions of the robot.*

For the web interface, this architecture provides autonomous operation, including the ability to navigate safely from point to point even in highly dynamic environments. Thus, the robot's safety and progress 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 nontechnical users can operate the system with ease. It must also be appealing, so that people will be attracted to the robot and will want to 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. In addition to displaying texts and graphics about the exhibits on the screen, the robots played prerecorded 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.

***At peak traffic hours, over 150 people simultaneously teleoperated Rhino through the museum.***

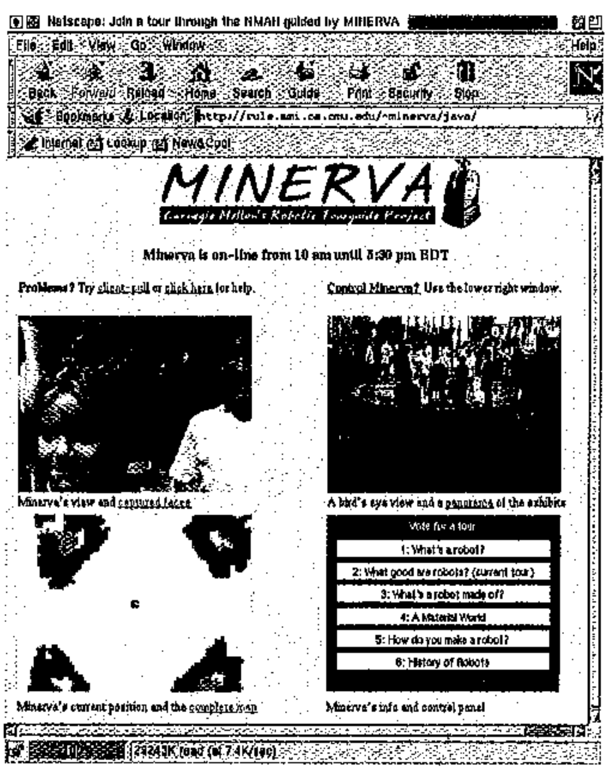


*Figure 7. Minerva's shared user and web interface. The robot's user interface is replicated on the web page (the applet with the yellow buttons).*

## Web Interfaces of the Museum Tour-Guide Robots

### *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 Fig. 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 that 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 nonoccupied 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 four 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.

### *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 with the first one, it only allows web users to take control over the robot during dedicated Internet hours. However, 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 composition (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 Fig. 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 Fig. 5) provides still images from the robot's camera and from two static wall-mounted cameras that 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 infor-

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

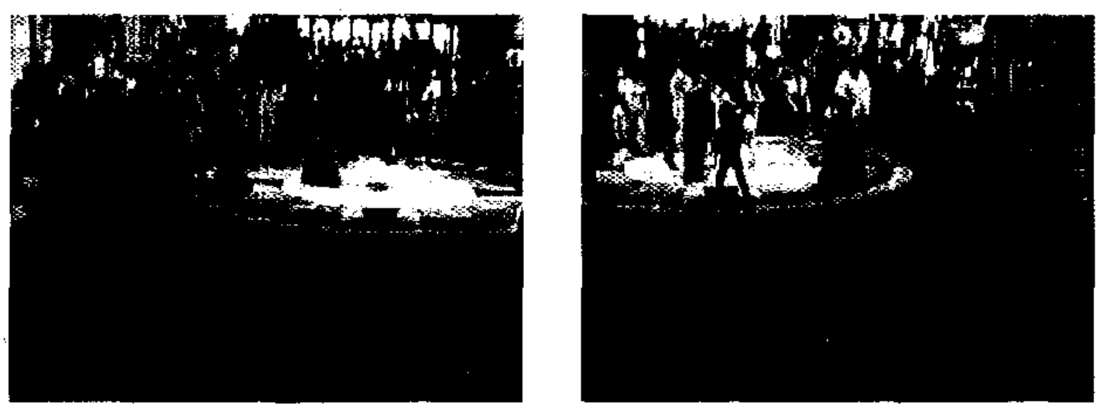

***Figure 9.*** *Typical on-line images recorded with the ceiling-mounted camera tracking the robot.*

mation 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 Fig. 6) is designed to provide 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 that establishes 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.

***Since the delay and throughput of the Internet varies drastically, the communication must be extremely low bandwidth.***

### *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 that 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 Fig. 2). Internet users automatically download the interface as an applet, which then is executed by their web browser (see Fig. 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. In addition to the previous interfaces it also delivers on-line images from an on-board camera as well as from a ceiling-mounted camera that tracks the robot in the museum. All this information is displayed on a single web page (see Fig. 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 vote. 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.

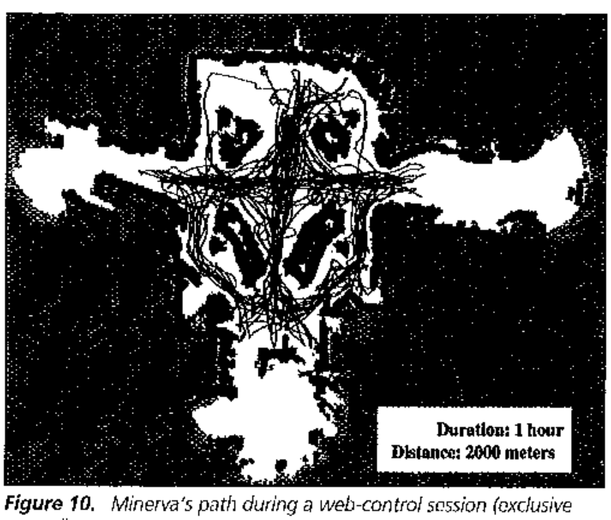

*Figure 10. Minerva's path during a web-control session (exclusive control).*

## Results

During three weeks of on-line mobile robot operation (one week in Germany and two weeks in the US), approximately 4000 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 2144 meters long. The maximum velocity of Minerva was 163 cm/s. 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, 2060 people visited Rhino's web pages. Rhino traveled over 18 km during its 6-day deployment period. Simultaneously with the web visitors, Rhino guided over 2000 physical visitors through the museum. There were 2400 requests to explain exhibits. Rhino managed to successfully arrive at its exhibits and deliver the information to the visitors in 2394 cases, which corresponds to an overall success rate of 99.75%—in the other six cases, Rhino's operation was disrupted by a battery change that led to a restart of Rhino's queue. At peak traffic hours, over 150 people simultaneously teleoperated the robot through the museum, using the web interface described here.

Minerva's shared-control interface (interface #3) was on-line for 91 hours and was accessed by 2885 people. We counted 4561 votes for the six 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 2390 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.

## 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 fixed graphics for displaying the robot's location in its map, the size of which made the display very slow (typical update frequency: once in 10 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 two things in common: they proved relatively easy to use and they were popular—at least according to the feedback we received.

## Conclusions

This article 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 commands and easy-to-understand graphical representations of information acquired by the robot. They use voting, queues, and first-come, first-serve priority 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.

***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.***

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 3-D 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" telepresence, the interface must be bi-directional, relaying back video and audio fed 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.

## Acknowledgments

This project has been supported by the Northrhine-Westphalia Ministry for Schools, Education, Science and Research (MSWWF-NRW, Germany) under contract number IVA3-10703596 and by the IST Programme of the Commission of the European Communities under contract number IST-199-12643.

## Keywords
Autonomous mobile robots, web-based control, telepresence

## 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 *Proc. 15th Nat. Conf. Artificial Intelligence*, 1998, pp. 11-18.

[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*, vol. 114, nos. 1-2, pp. 3-55, 2000.

[3] K. Goldberg, M. Mascha, S. Gentner, N. Rothenberg, C. Sutter, and J. Wiegley, "Desktop tele-operation via the World Wide Web," in *Proc. IEEE Int. Conf. Robotics and Automation*, 1995, pp. 654-659.

[4] K. Goldberg, J. Santarromana, G. Bekey, S. Gentner, R. Morris, J. Wiegley, and E. Berger, "The Telegarden," in *Proc. ACM SIGGRAPH*, 1995, p. 135.

[5] S. Goldberg, G.A. Bekey, and Y. Akatsuka, "DIGIMUSE: An interactive telerobotic system for remote viewing of three-dimensional art objects," in *Proc. IEEE IROS'98 Workshop on Robots on the Web*, 1998, pp. 55-60.

[6] D. Hähnel, W. Burgard, and G. Lakemeyer, "GOLEX - Bridging the gap between logic (GOLOG) and a real robot," in *Proc. 22nd German Conf. Artificial Intelligence (KI'98)*, Bremen, Germany, 1998, pp. 165-176.

[7] H. Hirukawa, T. Matsui, and H. Onda, "Prototypes of teleoperation systems via a standard protocol with a standard human interface," in *Proc. IEEE Int. Conf. Robotics and Automation*, 1997, pp. 1028-1033.

[8] A. Hopp, D. Schulz, W. Burgard, D. Fellner, and A.B. Cremers, "Virtual reality visualization of distributed teleexperiments," in *Proc. 24th Ann. Conf. of the IEEE Industrial Electronics Society (IECON'98)*, 1998, pp. 2210-2215.

[9] P. Saucy and F. Mondada, "KhepOnTheWeb: One year of access to a mobile robot on the Internet," in *Proc. IEEE IROS'98 Workshop on Robots on the Web*, 1998, pp. 22-30.

[10] D. Schulz, W. Burgard, and A. Cremers, "Robust visualization of navigation experiments with mobile robots over the Internet," in *Proc. IEEE/RSJ Int. Conf. Intelligent Robots and Systems (IROS)*, 1999, pp. 942-947.

[11] R. Simmons, R. Goodwin, K. Haigh, S. Koenig, and J. O'Sullivan, "A layered architecture for office delivery robots," in *Proc. First Int. Conf. Autonomous Agents*, Marina del Rey, CA, Feb. 1997, pp. 245-252.

[12] K. Taylor and J. Trevelyan, "Australia's telerobot on the Web," In *Proc. 26th Int. Symp. Industrial Robots*, 1995, pp. 39-44.

[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 tourguide robot," in *Proc. Int. Conf. Robotics and Automation (ICRA'99)*, 1999, pp. 1997-2085.

**Dirk Schulz** is a Ph.D. student at the Department of Computer Science III at the University of Bonn. His main research interests are in the fields of service robotics, Internet-based telepresence, and probabilistic state estimation.

**Wolfram Burgard** is an associate professor at the Department of Computer Science at the University of Freiburg. He received his Ph.D. degree in computer science from the University of Bonn in 1991. His research focuses on mobile robotics and artificial intelligence.

**Dieter Fox** has a postdoctoral position at the Department of Computer science at Carnegie Mellon University, Pittsburgh, PA. His research interests cover robotics, probabilistic reasoning, and artificial intelligence. Fox earned his Ph.D. degree from the University of Bonn in 1998.

**Sebastian Thrun** is currently an assistant professor at the Department of Computer Science at Carnegie Mellon University, Pittsburgh, PA. His research interests include robotics, machine learning, and various other aspects of artificial intelligence. He received his Ph.D. from the University of Bonn in 1995.

**Armin B. Cremers** is a full professor of computer science and chair of the Department of Computer Science III at the University of Bonn. Prior to this position he was a full professor at the University of Dortmund and an assistant professor at the University of Southern California. His scientific research is in the areas of database and information systems, autonomous mobile robots, and artificial intelligence.

***Address for Correspondence*:** Dirk Schulz, Dept. of Computer Science III, University of Bonn, Römerstr. 164, D-53117 Bonn, Germany. Phone: +49 228 73 4267. Fax: +49 228 73 4382. E-mail: schulz@cs.uni-bonn.de