Dynamics of Continuous, Discrete and Impulsive Systems
Series **B:** *Algorithms and Applications* **11 (2004)** 651-664

Remote Control of On-line Mobile Robots over IP Networks

Peter Xiaoping Liu[1], Max Q-H Meng[2] and Simon X. Yang[3]

[1]Department of Systems and Computer Engineering
Carleton University, Ottawa, ON, CANADA
[2]Department of Electronic Engineering
Chinese University of Hong Kong, Hong Kong
[3]Advanced Robotics and Intelligent Systems (ARIS) Lab
School of Engineering,University of Guelph,Guelph, ON, Canada

Abstract. In this paper, we introduce a new scheme for the remote control of mobile robots over IP networks such as the Internet. The hardware configuration of the platform consists mainly of a *Pioneer 2 PeopleBot* mobile robot. On the head of the robot, there is a Sony PTZ video camera which is used to provide live visual feedback on robot's immediate environment. There are two sonar arrays for range measurement. The P2PB mobile robot is connected to the Internet through a pair of wireless LAN adaptors. The system is integrated by a client-server software architecture for robot control and feedback display. In this software architecture, there are two servers, a video server and a control server, and two corresponding clients, a video applet and a control applet. The web server is a Linux Apache web server. Human operators control the mobile robot by using a Java-enabled web browser on an ordinary PC. With this platform, advanced remote control algorithms, interface designs and transmission protocols can be easily tested without large programming efforts.

Keywords: IP networks, mobile robots, remote control, teleoperation

AMS (MOS) subject classification:

1 Introduction

Current applications of computer and communications networks are primarily dominated by information and data transmissions, exchanges and publications. Continuing exponentially increasing advances in microelectronics, photonics and wireless technology are making wired and wireless networks more powerful, accessible and affordable. Consequently, network based applications are rapidly expanding into new areas.

The Internet robot is one of these pioneering areas with great potentials. By means of remote operation of a robotic device or a tool via the Internet, human perceptional and motor capabilities can be extended beyond physical distance limit. The first Internet telerobot appeared in 1994 [1]. By 2001, more than forty such systems had been put online around the world, allowing users to visit museums, tend gardens, navigate undersea, float in blimps or handle protein crystals via the Internet [2-6]. These systems, however, are largely experimental and not ready to provide real-world services. Physical interaction with remote environments over the Internet poses many technical challenges which are still outstanding. For example, new technologies are needed for coping with problems associated with variable time delays and variable bandwidth, for coordinating simultaneous users, for handling unexpected errors, for ensuring trustworthiness (reliability, security and privacy), etc. For these various research purposes, it would be too time-consuming and too involving if it were needed to build an experimental platform of on-line robots from scratch.

In this paper, we present a new modular platform for online mobile robotic systems. On this platform, advanced remote compensation control algorithms, interface designs, transmission protocols and applications can be easily tested without large programming work.

2 The Hardware Architecture

The system hardware consists mainly of a *Pioneer 2 PeopleBot* (P2PB) mobile robot, as shown in Fig. 1 [7]. It contains the basic components for motor control, sensing and navigation, including batteries, drive motors and wheels, position/speed encoders, bumper switches, integrated sonar ranging sensors and a visual system. All these components are managed by an on-board micro-controller and the corresponding server software. The P2PB mobile robot uses a 20MHz Siemens 88C166-based microcontroller with independent motor/power and sonar controller boards for a versatile operating environment. The controller has two RS232-standard communication ports and an expansion bus to support various accessories.

The drive system (as shown in Fig. 2) of the P2PB mobile robot uses high-speed, high-torque, reversible-DC motors. It has a caster wheel and two drive wheels. It can move with an approximate maximum speed of 60 cm per second. Each front drive motor includes a high-resolution optical quadrature shaft encoder that provides 9,850 ticks per wheel revolution (19 ticks per millimetre) for precise position and speed sensing and advanced dead reckoning. The error in distance is about 1 cm per meter and the error in rotation is up to 8 degrees per revolution.

The P2PB mobile robot provides two range-finding sonar arrays. One array, affixed under the front of the Deck and atop the Nose, provides forward and side-range sensing. The other, an optional sonar array is attached just beneath the rear Deck and provides rearward, as well as side sensing. Each array contains eight sonars. Totally up to sixteen sonars surround the robot. The sonar positions are fixed in both arrays: one on each side, and six facing outward at 20-degree intervals, together providing 360 degrees of nearly seamless sensing. The sonar sensitivity ranges from 10 centimetres (6

Figure 1: The P2PB mobile robot

inches) to more than 5 meters (16 feet) (Objects closer than 10 centimetres are not detected). The sonar's firing pattern may also be controlled through software; the default is left-to-right in sequence for the forward array and right-to-left on the rear. One sonar from each array *pings* simultaneously. The sonar array is shown in Fig. 3.

The visual system is detachable and mounted on the head of the P2PB mobile robot. It mainly consists of a Sony D30/31 pan-tilt-zoom (PTZ) colour camera (shown in Fig. 4) and PTZ system software, which is for camera pan-tile-zoom angle control and image grabbing.

The P2PB mobile robot is connected to the Internet through a pair of BreezeCom's BreezeNet PRO 11 indoor wireless Ethernet adaptors (as shown in Fig. 5). The BreezeNet PRO.11 indoor adaptors adhere to the IEEE 802.11 standard, working seamlessly with other 802.11 Frequency Hopping wireless LAN products. By operating in the 2.4 GHz unlicensed ISM band, it offers data communication rate of up to 3 Mbps, at distances of up to 150m (500ft) indoors, and roaming speeds up to 100km/h (60mph). In the platform, an AP-10 adaptor (access point to hub slot) and a SA-10 adaptor (station adaptor installed inside the P2PB mobile robot) are used.

Users can control the mobile robot remotely on any ordinary PC with a Java-enabled web browser and Internet access. An overview of system hardware configuration is shown in Fig. 6.

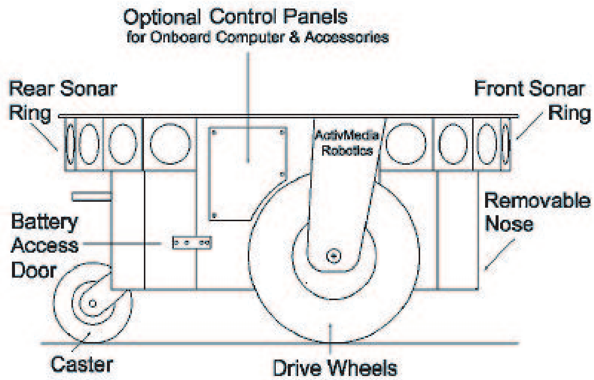

Figure 2: Drive system

3 The Software Architecture

The system software employs a client-server architecture for robot control and feedback information display. In the model, there are two servers, a *Video Server* and a *Control Server*, and two corresponding clients, a *Video Applet* and a *Control Applet*. The web browser is any *Java-enabled* web browser and the web server is a *Linux Apache web server*.

On the client side – the web browser, the Video Applet is responsible for live image decoding and display etc., and the Control Applet is for intercepting and interpreting human control commands (mouse-click events in the control panel) and display of other information feedback, such as robot position, speed and ranging etc.

On the server side – the P2PB mobile robot, the Video Server is in charge of video grabbing, compression, encoding and transmission, while the Control Server takes care of motor control, sonar firing, camera pan-tile-zoom control, and other sensor sampling and encoding. The corresponding server and client pairs exchange information through the web server, the Linux Apache web server.

Various messages need to be exchanged between the robot and the human operator: non real-time administration and configuration data, control commands, live scene images, robot other states such as position and speed, and sonar ranging data. It is clear that data communications require *real-time* delivery except for the once-for-all administrative data. Consequently, in the implementation, we adopt the TCP protocol for the communications of administrative data. A TCP connection is opened when a teleoperation session is started and closed once the teleoperation is geared up. The trinomial protocol [7, 8] is employed for the transmission of information on

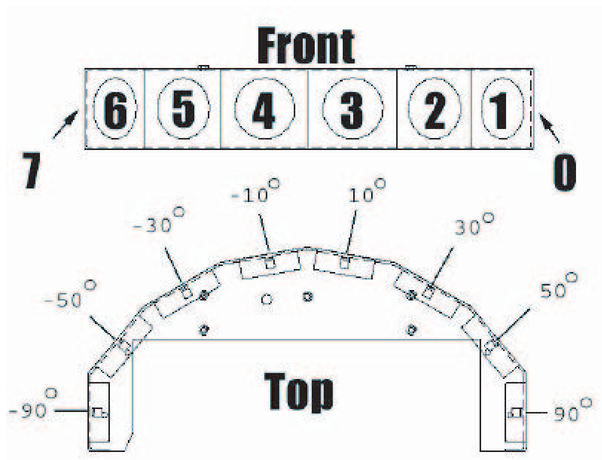

Figure 3: Sonar array

Figure 4: Vision System

live scene images, sonar ranging, and robot positions and speeds. Since the human operator issues control commands based on his/her personal judgements and decisions, the timing of these control signals is random in nature and controlled solely by the human operator. The transmission frequency depends on how often the specific user controls the mobile robot. Thus, the data transmission rate is largely controlled by the human operator instead of the transport protocol. As a result, we still utilize UDP for control command delivery.

A brief functional software structure of the platform is shown in Fig. 7.

Figure 5: Wireless Adaptor

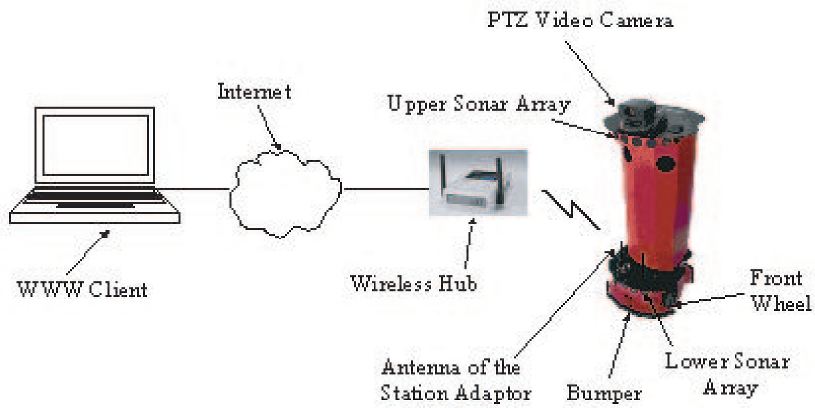

Figure 6: Overview of hardware configuration

3.1 3.1 Server-Client Control Structure

In the server-client control model, the robot's Control Server - *the Pioneer 2 Operating System* (P2OS) - works to manage all the low-level details of the mobile robot system. These include operating the motors, firing the sonars, collecting sonar and wheel encoder data, managing the battery power, controlling the pan-tilt-zoom of the camera, and so on – all on command from and reporting to the corresponding client application, the Control Applet, through the Linux Apache web server.

With this client-server architecture, the client application - the Control Applet, is insulated from the lowest level of controls. The Control Applet communicates with the Control Server, P2OS, via the P2OS client-server interface – the communication packet protocol. The server-client control structure is shown in Fig. 8.

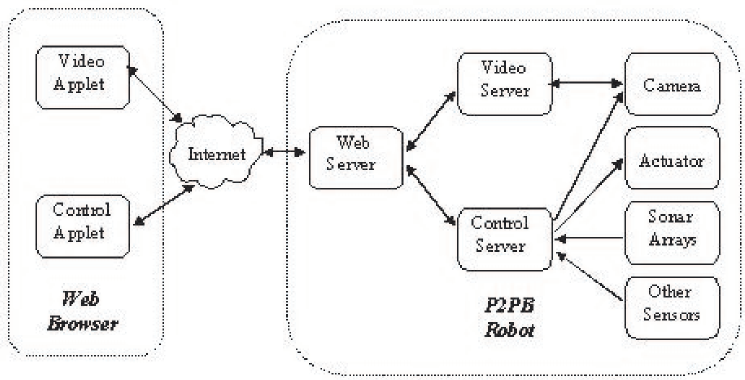

Figure 7: Software architecture overview

3.2 3.2 Server-Client Video Structure

The live video stream of the mobile robot's environment is sent back to the human operator by the Video Server in the P2PB mobile robot. The sending rate is controlled by the trinomial protocol based on network states. The images are captured by a bt848 chipset-based frame grabber and then compressed before transferring to the client application via the Internet. On the client side, the Video Applet buffers, decodes and creates the image when it receives the entire frame and displays it. The server-client video structure is shown in Fig. 9.

For the data communications of Internet robots, video transmission consumes most of the bandwidth and is very costly. Video data are usually compressed before transmission. Rather than employing conventional uniform-resolution compression methods, we adopt the wavelet-based image foveation technology to compress the video data further. The compression rate in the implementation is achieved as high as 30:1 with an acceptable resolution.

The idea of image foveation is to mimic human nonuniform-resolution vision system. Ordinary images are sampled, stored and transmitted with a uniform resolution, but human vision system does not require uniform detail in the field of vision. Less detail is needed on the periphery of the vision field than at the focal point. The concept of image foveation is to take advantage of this feature of human natural vision. A foveated image has non-uniform resolution. A spatially variant filter is applied to the image. This filter maintains high fidelity around the point of interest (e.g., the ground floor in front of the robot for mobile robot tele-navigation and the edges of the object for fine tele-manipulation etc) while reduces spatial resolution away from the region of interest according to the sensitivity function of the human visual

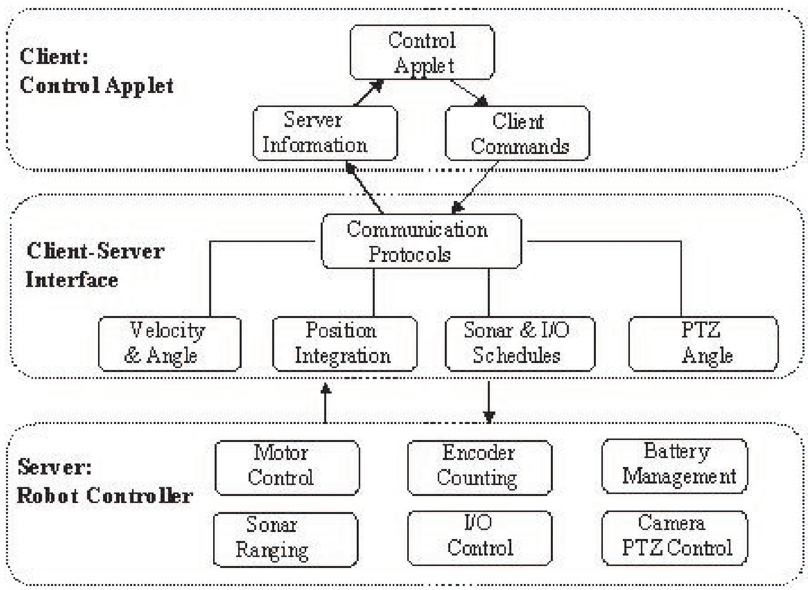

Figure 8: Server-client control structure

system. Thus, substantial savings in bandwidth are achieved compared to traditional uniform-resolution image compression algorithms.

Different from uniform-resolution images, foveated ones characterize themselves by implementing multi-resolution sub-regions in the same image. No matter what coding and transmission algorithm that may be employed, a function to describe the spatial sensitivity of the human visual system must be determined first. In [9], Schwartz presents a logarithmic resolution function, which would be the most appropriate from the biology point of view. However as argued in [10], this sensitivity function is difficult to implement due to singularities along one axis. Therefore, various simplifications are proposed in the literature, among which two are widely adopted. One is a pyramid image representation presented by Geisler and Perry in [11]. The other one is a wavelet-based approach introduced by Chang and Yap [12]. The advantage of the later approach over the former one is that no redundant information is added in order to build the pyramid. In the implementation of the platform, we employ the wavelet-based approach.

As an illustration example, Fig. 10 shows the idea of image foveation. Fig. 10A and 10B are the original uniform-resolution images, and Fig. 10C and 10D are the foveated ones respectively. For the foveated image shown in Fig. 10C, the bottom part of the garbage bin is the point of interest and the

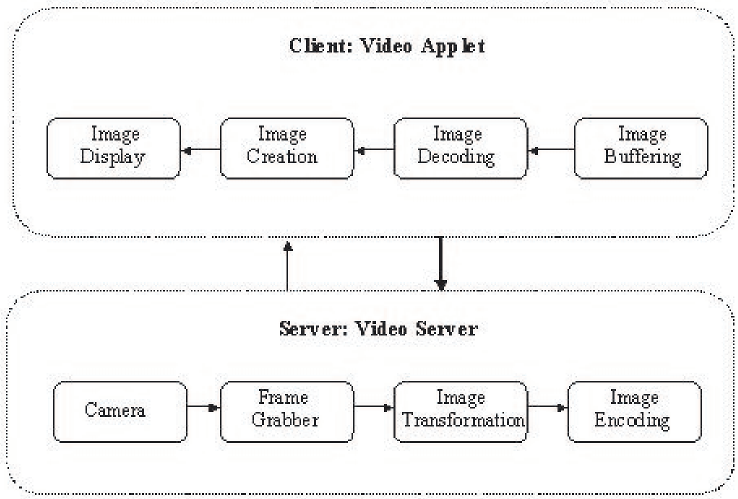

Figure 9: Server-client video structure

resolution around this region is the highest. The resolution fades away from the fovea, the bottom part of the garbage bin. Similarly, for Fig. 10D, the left part of the image, the floor, is the fovea.

4 User Interface

The user interface (UI) is designed in orientation to human operators – making it easy to interact with the remote mobile robot. The outlook of feedback data display, user manipulating platform, operation convenience and feasibility are taken into account when the UI is designed. A snapshot of the teleoperation UI is shown in Fig. 11.

The UI mainly consists of two Java applets: a Control Applet and a Video Applet. It works on any Java-enabled web browser. Brief instructions on how to use the interface to control the remote mobile robot are also provided at the bottom part of the UI.

4.1 Live Video Display

Live visual images are the principal source of information feedback on which the human operator relies to have knowledge of the environment of the remote robot. For a reliable teleoperation, real-time image feedback is usually mandatory although image data transmission on the Internet is very costly. A live video screen is placed on the left-centre of the interface, as shown

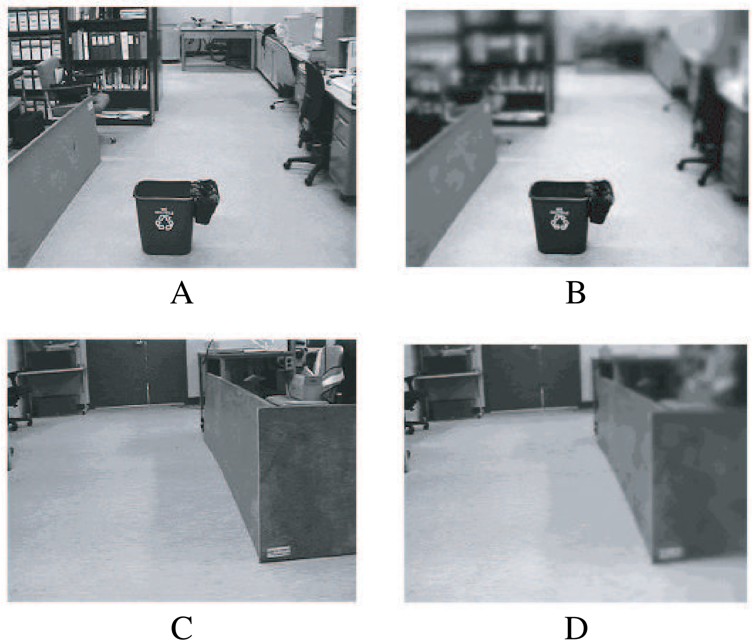

Figure 10: Illustration of image foveation

in Fig. 11. The human operator can expect a higher resolution display of his/her interested area by mouse clicking and dragging.

4.2 Control Panel

The control panel, as placed on the centre of the UI, is an immediate environmental map of the robot. Because of the "skewed right" characteristic and variation of Internet delays, continuous rate control is difficult. Consequently, we use the discrete-command control scheme and a computer mouse rather than a joystick is used as the command-signalling device. The user navigates the robot by clicking the mouse on the control panel to indicate the orientation and velocity of next robot movement. Because of low confidence and low reliability of sonar readings, the sonar readings are displayed in the robot's immediate environmental map only for reference. The user acquires depth information mainly by manipulating the pan-tilt angle of the robot's on-board camera.

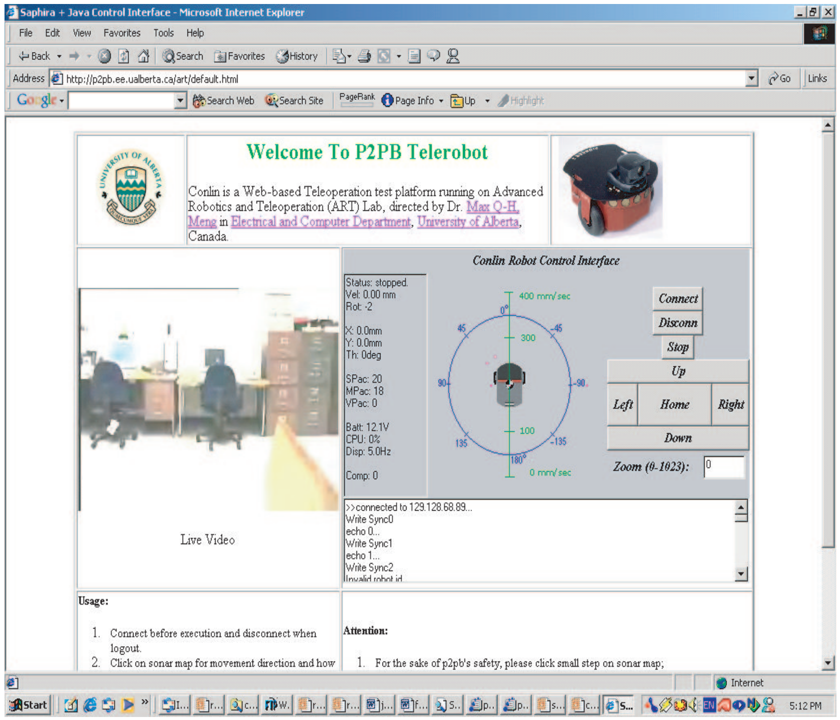

Figure 11: Teleoperation user interface

4.3 Camera Control and Other Information Display

To the immediate left of the control panel is a text area for the display of the robot's other state information feedback, such as speed, position and battery voltage etc. At the right side of the control panel, there are some control buttons for camera control. By clicking the mouse on these buttons, the human operator can adjust the pan-tile angle of the video camera to have a best view of the robot's immediate environment.

5 Experimental Implementation

To demonstrate the validity and feasibility of the developed platform, we have carried out many experiments. In the experiments, by using any Java-enabled web browser, such as Microsoft Internet Explorer or Netscape Communicator, on a regular PC, the users successfully navigated the P2PB mobile robot through a laboratory environment. Fig. 12 shows a sequence of the snapshots of the P2PB mobile robot when it was remotely guided via the Internet to move through a laboratory environment.

6 Conclusion

As the first step towards a real-world network-enabled on-line robotic system, a new modular platform for the remote control of on-line mobile robotic systems is developed in this paper. The system enables Internet users to guide a P2PB mobile robot remotely via the Internet by using any Java-enabled web browser. Experiments have been carried out to test the feasibility of the platform. In the experiments, by using a Java-enabled web browser such as Microsoft Internet Explorer or Netscape Communicator, on a ordinary PC, the users successfully guided the P2PB mobile robot through a laboratory environment remotely over the Internet. The great advantage of this platform is that the client software is insulated from the lowest level details of the mobile robot. Thus, it is very easy to implement and test new advanced teleoperation control algorithms, interface designs and applications on this platform without large programming work.

Acknowledgements

This work was supported by Natural Sciences and Engineering Research Council (NSERC) of Canada and the Canada Research Chairs Program.

References

[1] K. Goldberg, M. Mascha, S. Gentner, N. Rothenberg, C. Sutter and J. Wiegley, "Desktop teleoperation via the World Wide Web," in: *Proc. IEEE Intl. Conf. Robotics and Automation*, Nagoya, Japan, 1995, pp. 654-659.

[2] C. Sayers, *Remote Control Robotics*, Springer Verlag, New York, USA, 1998.

[3] K. Goldberg and R. Siegwart (editors), *Beyond Webcams: An Introduction to Online Robots*, The MIT Press, Cambridge, Massachusetts, USA, 2002.

[4] W. Burgard, A Cremers, D. Fox, D. Haehnel, G. Lakemeyer, D. Schulz, W. Steiner and S. Thrun, "Experiences with an interactive museum tour-guide robot," *Artificial Intelligence*, vol. 114, no. 1, pp. 3-55, 1999.

[5] E. Paulos and J. Canny, "Ubiquitous tele-embodiment: applications and implications," *International Journal of Human Computer Studies*, vol. 46, pp. 861-877, 1997.

[6] O. Michel, P. Saucy and F. Mondada, " 'KhepOnTheWeb': An experimental demonstrator in telerobotics and virtual reality," in: *Proc. IEEE Annual Intl. Conf. Virtual Systems and Multimedia*, Los Alamitos, CA, USA, 1997, pp. 90-98.

[7] P. X. Liu, M. Meng and S. X. Yang, "Data Communications for Internet Robotics," *Autonomous Robots*, vol. 14, no. 3, 2003. To appear in Nov. 2003.

[8] P. X. Liu, M. Meng, X. Ye and J. Gu, "A novel data transport protocol for Internet robots," in: *Proc. 4th IEEE World Congress on Intelligent Control and Automation*, Shanghai, China, 2002, pp. 59-65.

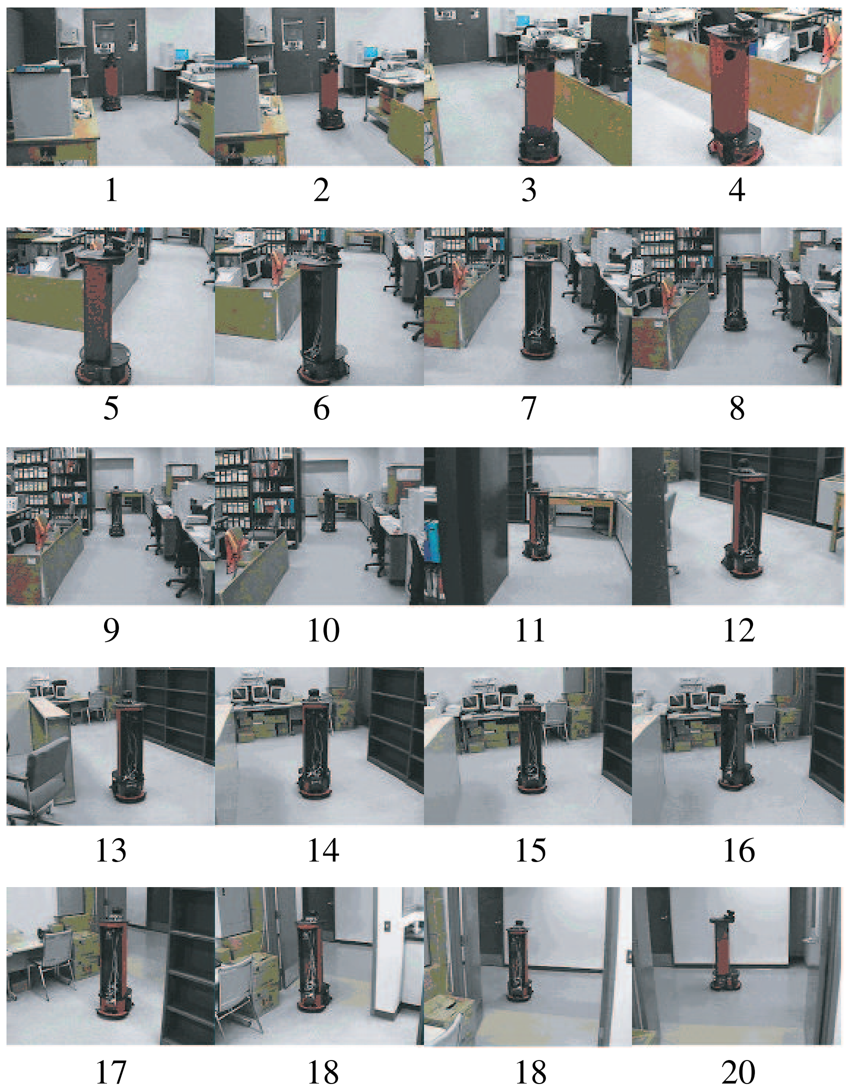

Figure 12: A sequence of robot images showing how the P2PB mobile robot is teleoperated over the Internet by the user to move through a laboratory environment

[9] E. L. Schwartz, "Computational anatomy and functional architecture of striate cortex: A spatial mapping approach to perceptual coding," *Vision Research*, vol. 20, pp. 645-669, 1980.

[10] B. Overall, *Foveated Imaging*, http://www-ise.stanford.edu/class/psych221/99/, Mar 1999.

[11] W.S. Geisler and J. S. Perry, "A real-time foveated multi-resolution system for low-bandwidth video communication," in: B. Rogowitz and T. Pappas (Eds.), *Human Vision and Electronic Imaging, SPIE Proceedings*, vol. 3299, 1998, pp. 294-305.

[12] E. Chang and C. K. Yap, "A wavelet approach to foveating images," in: *13th ACM Symposium on Computational Geometry*, Nice, France, vol. 13, pp. 397-399, 1997.

Received May 2003; revised March 2004.
email:journal@monotone.uwaterloo.ca
http://monotone.uwaterloo.ca/∼journal/