32. Networked Telerobots

Dezhen Song, Ken Goldberg, Nak Young Chong

Telerobots, remotely controlled robots, are widely used to explore undersea terrains and outer space, to defuse bombs, and to clean up hazardous waste. Until 1994, telerobots were accessible only to trained and trusted experts through dedicated communication channels. This chapter describes networked telerobots, a new class of telerobots controllable over networks such as the Internet, that are accessible to the general public. This chapter will describe relevant network technology, the history of networked telerobots within the broader field of teleoperation, properties of networked telerobots, how to build a networked robot, example systems, and topics for future research.

32.1 Overview and Background

As illustrated in Fig. 32.1, the broader field of teleoperation, where the primary concerns are stability and time delay, is covered in Chap. 31. The field of networked *robots*, where autonomous robots and sensors communicate over local networks, is covered in Chap. 41. Networked *telerobots*, the subject of the present chapter, focuses on teleoperated robot systems that are accessible by the public via web browsers.

By 2006, several hundred networked telerobots have been developed and put online for public use. Many papers have been published describing these systems and a book on this subject by *Goldberg* and *Siegwart* is available [32.1]. Updated information about new research and an archive/survey of networked telerobots is available on the website of the IEEE technical committee on networked robots, which fosters research in both networked telerobots and networked robots (IEEE Technical Committee on Networked Robots http://tab.ieee-ras.org/).

Networked telerobots have the following properties

- The physical world is affected by a device that is locally controlled by a network *server*, which communicates with remote human users through web browsers such as Internet Explorer or Firefox, which are generally referred to as *clients*. As of 2006, the standard protocol for network browsers is the hypertext transfer protocol (HTTP), a stateless transmission protocol.
- Most networked telerobots are continuously accessible (online), 24 hours a day, 7 days a week.
- Since hundreds of millions of people now have access to the Internet, mechanisms are needed to handle client authentication and contention.
- Input and output for networked telerobots is usually achieved with the standard computer screen, mouse, and keyboard.
- Clients may be inexperienced or malicious, so online tutorials and safeguards are generally required.

32.2 A Brief History

Like many technologies, remotely controlled devices were first imagined in science fiction. In 1898, *Nicola Tesla* [32.2] demonstrated a radio-controlled boat in New York's Madison Square Garden. The first major experiments in teleoperation were motivated by the need to handle radioactive materials in the 1940s. *Goertz* demonstrated one of the first bilateral simulators in the 1950s at the Argonne National Laboratory [32.3]. Remotely operated mechanisms have been designed for use in inhospitable environments such as undersea [32.4] and space exploration [32.5]. At General Electric, *Mosher* [32.6] developed a two-arm teleoperator with video cameras. Prosthetic hands were also applied to teleoperation [32.7]. More recently, teleoperation has been considered for medical diagnosis [32.8], manufacturing [32.9], and micromanipulation [32.10]. See Chap. 31 and the book from *Sheridan* [32.11] for excellent reviews on teleoperation and telerobotics research.

The concept of hypertext (linked references) was proposed by Vannevar Bush in 1945 and was made possible by subsequent developments in computing and networking. In the early 1990s, Berners-Lee introduced the HTTP. A group of students led by Marc Andreessen developed an open-source version of the first graphical user interface, the *Mosaic* browser, and put it online in 1993. The first networked camera, or *webcam*, went online in November 1993 [32.12].

Approximately nine months later, the first networked telerobot went online. K. Goldberg and M. Mascha's Mercury Project combined an IBM industrial robot arm with a digital camera and used the robot's air nozzle to allow remote users to excavate for buried artifacts in a sandbox [32.13, 14]. Working independently, a team led by *K. Taylor* and *J. Trevelyan* at the University of Western Australia demonstrated a remotely controlled six-axis telerobot in September 1994 [32.15, 16]. These early projects pioneered a new field of networked telerobots. See [32.17–25] for other examples. An online archive of Networked Telerobots is at: http://ford.ieor.berkeley.edu/ir/.

Networked telerobots are a special case of *supervisory control* telerobots, as proposed by *Sheridan* and colleagues [32.11]. Under supervisory control, a local computer plays an active role in closing the feedback loop. Most networked robotics are type c supervisory control systems (see Fig. 32.2).

Although the majority of networked telerobotic systems consist of a single human operator and a single robot [32.26–33], *Chong* et al. [32.34] propose a useful taxonomy: single operator single robot (SOSR), single operator multiple robot (SOMR), multiple operator single robot (MOSR), and multiple operator multiple robot (MOMR).

The decade from 1995–2005 witnessed extensive development in networked telerobots. New systems, new experiments, and new applications now go well beyond the traditional fields such as defense, space, and nuclear material handing [32.11] that motivated teleoperation in early 1950s. As the Internet introduces universal access to every corner of life, the impact of networked robots becomes broader and deeper in modern society. Recent applications range from education, industry, commercial, health care, geology, and environmental monitoring, to entertainment and arts.

Networked telerobots provide a new medium for people to interact with a remote environment. A networked robot can provide more interactivity beyond that provided by a normal videoconferencing system. The physical robot not only represents the remote person but also transmits multimodal feedback to the person, which is often referred as *telepresence* in the literature [32.29]. *Paulos* and *Canny*'s personal roving presence (PRoP) robot [32.35] and *Jouppi* and *Thomas*'s surrogate robot [32.29] are recent representative work.

Networked telerobots have great potential for education and training. In fact, one of the earliest networked telerobot systems [32.36] originated from the idea of a remote laboratory. Networked telerobots provide universal access to the general public, who may have little to no knowledge of robots, with opportunities to understand, learn, and operate robots, which were expensive scientific equipment limited to universities and large corporate laboratories before. Built on networked telerobots, online remote laboratories [32.37, 38] greatly improve distance learning by providing an interactive experience; for example, teleoperated telescopes help students to understand astronomy [32.39]. Teleoperated microscopes [32.40] help student to observe microorganisms. The Tele-Actor project [32.41] allows a group of students to remotely control a human tele-actor to visit environments that are normally not accessible to them such as cleanroom environments for a semiconductor manufactory facility and DNA analysis laboratories.

32.3 Communications and Networking

Below is a short review of relevant terminologies and technologies on networking. For details, see the texts by [32.42].

A communication network includes three elements: *links*, *routers/switchers*, and *hosts*. Links refer to the physical medium that carry bits from one place to another. Examples of links include copper or fiber-optic cables and wireless (radio frequency or infrared) channels. Switches and routers are hubs that direct digital information between links. Hosts are communication end points such as browsers, computers, and robots.

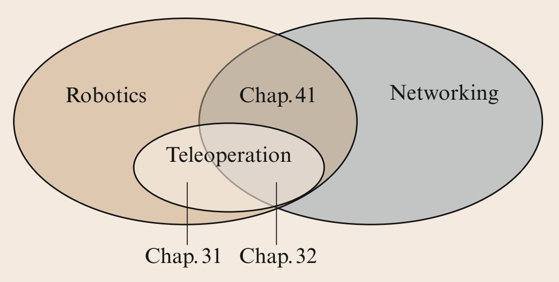


Fig. 32.1 Relationship between the subjects of networked telerobots (Chap. 32, the present chapter), teleoperation (Chap. 31), and networked robots (Chap. 41)

Networks can be based in one physical area (local-area network, or LAN), or distributed over wide distances (wide-area network, or WAN). Access control is a fundamental problem in networking. Among a variety of methods, the *ethernet* protocol is the most popular. Ethernet provides a broadcast-capable multiaccess LAN. It adopts a carrier-sense multiple-access (CSMA) strategy to address the multiple-access problem. Defined in the IEEE 802.x standard, CSMA allows each host to send information over the link at any time. Therefore, collisions may happen between two or more simultaneous transmission requests. Collisions can be detected either by directly sensing the voltage in the case of wired networks, which is referred to as collision detection (CSMA/CD), or by checking the time-out of

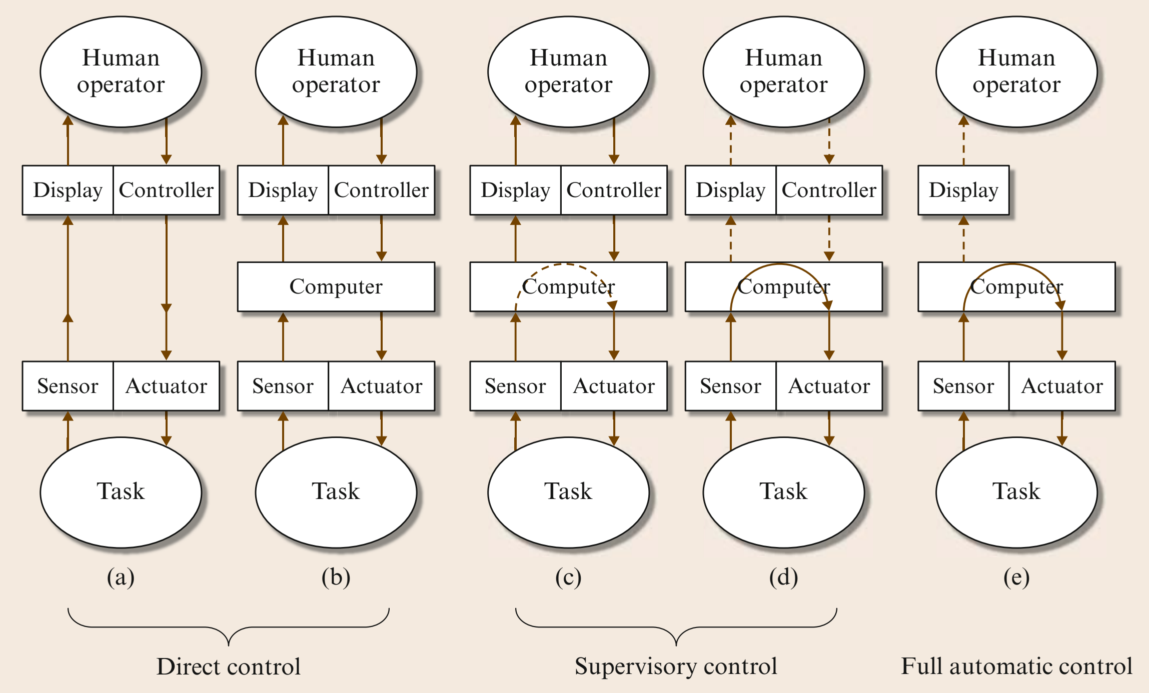


Fig. 32.2 A spectrum of teleoperation control modes adapted from *Sheridan*'s text [32.11]. We label them (a–e), in order of increasing robot autonomy. At the *far left* would be a mechanical linkage where the human directly operates the robot from another room through sliding mechanical bars, and on the *far right* is the system where the human role is limited to observation/monitoring. In (c–e), the *dashed lines* indicated that communication may be intermittent

an anticipated acknowledgement in wireless networks, which is referred to as collision avoidance (CSMA/CA). If a collision is detected, both/all senders randomly back off a short period of time before retransmitting. CSMA has a number of important properties: (1) it is a completely decentralized approach, (2) it does not need clock synchronization over the entire network, and (3) it is very easy to implement. However, the disadvantages of CSMA are: (1) the efficiency of the network is not very high and (2) the transmission delay can change drastically.

As mentioned previously, LANs are interconnected with each other via routers/switchers. The information transmitted is in packet format. A packet is a string of bits and usually contains the source address, the destination address, content bits, and a checksum. Routers/switchers distribute packets according to their routing table. Routers/switchers have no memory of packets, which ensures scalability of the network. Packets are usually routed according to a first-in first-out (FIFO) rule, which is independent of the application. The packet formats and addresses are independent of the host technology, which ensures extensibility. This routing mechanism is referred to as packet switching in the networking literature. It is quite different from a traditional telephone network, which is referred to as circuit switching. A telephone network is designed to guarantee a dedicated circuit between a sender and a receiver once a phone call is established. The dedicated circuitry ensures communication quality. However, it requires a large number of circuits to ensure the quality of service (QoS), which leads to poor utilization of the overall network. A packet-switching network cannot guarantee dedicated bandwidth for each individual pair of transmissions, but it improves overall resource utilization. The Internet, which is the most popular communication media and the infrastructure of networked telerobots, is a packet-switching network.

32.3.1 The Internet

The creation of the Internet can be traced back to US Department of Defense's (DoD) APRA NET network in the 1960s. There are two features of the APRA NET network that enabled the successful evolution of the Internet. One feature is the ability for information (packets) to be rerouted around failures. Originally this was designed to ensure communication in the event of a nuclear war. Interestingly, this dynamic routing capability also allows the topology of the Internet to grow easily. The second important feature is the ability for heterogeneous networks to interconnect with one another. Heterogeneous networks, such as X.25, G.701, ethernet, can all connect to the Internet as long as they can implement the Internet protocol (IP). The IP is media, operating system (OS), and data rate independent. This flexible design allows a variety of applications and hosts to connect to the Internet as long as they can generate and understand IP.

Figure 32.3 illustrates a four-layer model of the protocols used in the Internet. On the top of the IP, we have two primary transport layer protocols: the transmission control protocol (TCP) and the user data protocol (UDP). TCP is an end-to-end transmission control protocol. It manages packet ordering, error control, rate control, and flow control based on packet round-trip time. TCP guarantees the arrival of each packet. However, excessive retransmission of TCP in a congested network may introduce undesirable time delays in a networked telerobotic system. UDP behaves differently; it is a broadcast-capable protocol and does not have a retransmission mechanism. Users must take care of error control and rate control themselves. UDP has a lot less overhead compared to TCP. UDP packets are transmitted at the sender's preset rate and the rate is changed based on the congestion of a network. UDP has great potential, but it is often blocked by firewalls because of a lack of a rate control mechanism. It is also worth mentioning that the widely accepted term TCP/IP refers to the family of protocols that build on IP, TCP, and UDP.

In the application layer of the Internet protocols, the HTTP is one of the most important protocols. HTTP is the protocol for the World Wide Web (WWW). It allows the sharing of multimedia information among

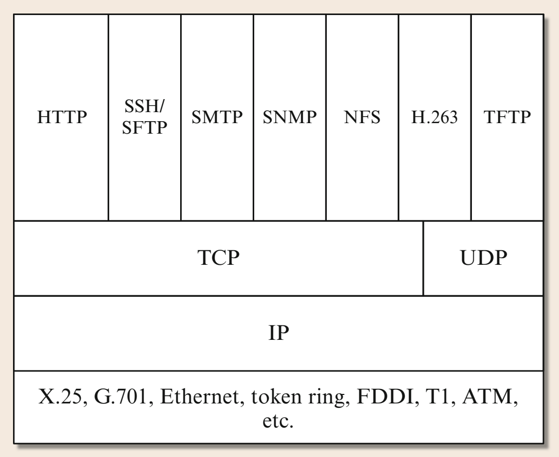


Fig. 32.3 A four-layer model of internet protocols (after [32.42])

Table 32.1 *Last-mile* Internet speed by wired connection type. If not specified, the downstream transmission and the upstream transmission share the same bandwidth

Types	Bits per second
Modem (V.92)	Up to 56 K
Integrated Services Digital Network (ISDN) BRI	64–128 K
High Data Rate Digital Subscriber Line (HDSL)	1.544 M duplex on two twisted-pair lines
Assymetric Digital Subscriber Line (ADSL)	1.544–6.1 M downstream, 16–640 K upstream
Cable modem	2–4 M downstream, 400–600 K upstream
Fiber to the home (FTTH)	5–30 M downstream, 2–5 M upstream
Internet II/III node	$\geq$ 1 G (data are based on the service provided by Verizon)

Table 32.2 Survey of wireless technologies in terms of bit rate and range

Types	Bit rate (bps)	Band (Hz)	Range (m)
Zigbee (802.15.4)	20–250 K	868–915 M/2.4 G	50
3G cellphone	400 K–1.15 M	$\leq$ 3.5 G	15 000
Bluetooth	732 K	2.4 G	100
MWBA (802.20)	1 M	$\leq$3.5 G	15 000
WiFi (802.11a,b,g)	11–54 M	2.4 G or 5.8 G	100
WiMax (802.16)	70 M	2–11, 10–66 G	50 000

heterogeneous hosts and OSs including text, image, audio, and video. The protocol has significantly contributed to the boom of the Internet. It also changes the traditional client/server (C/S) communication architecture to a browser/server (B/S) architecture. A typical configuration of the B/S architecture consists of a web server and clients with web browsers. The web server projects the contents in hypertext markup language (HTML) format or its variants, which is transmitted over the Internet using HTTP. User inputs can be acquired using the common gateway interface (CGI) or other variants. The B/S architecture is the most accessible because no specialized software is needed at the client end.

32.3.2 Wired Communication Links

Even during peak usage, the network backbones of the Internet often run at less than 30% of their overall capacity. The average backbone utilization is around 15–20%. The primary speed limitation for the Internet is the *last mile*, the link between clients and their local Internet service providers (ISP).

Table 32.1 lists typical bit rates for different connection types. It is interesting to note the asymmetric speeds in many cases, where upstream bit rate (from the client to the Internet), are far slower than downstream bit rates (from the Internet to the client). These asymmetries introduce complexity into the network model for teleoperation. Since the speed difference between the slowest modem link and the fastest Internet II node is over 10 000, designers of a networked telerobotic system should anticipate a large variance of communication speeds.

32.3.3 Wireless Links

Table 32.2 compares the speed, band, and range of wireless standards as of 2006. Increasing bit rate and communication range requires increasing power. The amount of radiofrequency (RF) transmission power required over a distance d is proportional to d^k, where $2 \leq k \leq 4$ depending on the antenna type. In Table 32.2, Bluetooth and Zigbee are typical low-power transmission standards that are good for short distances. WiMax and MWBA are currently under development.

By providing high-speed connectivity at low cost, WiFi is the most popular wireless standard in 2006. Its range is approximate 100 m line of sight and the WiFi wireless network usually consists of small-scale interconnected access points. The coverage range usually limits these networks to an office building, home, and other indoor environments. WiFi is a good option for indoor mobile robots and human operators. If the robot needs to navigate in the outdoor environment, the third-generation (3G) cellphone network can provide the best coverage available. Although obvious overlap exists among wireless standards in coverage and bandwidth, there are two import issues that have not been covered by Table 32.2. One is mobility. We know that, if an RF source or receiver is moving, the corresponding Doppler effect causes a frequency shift, which could cause problems in communication. WiFi is not designed for fast-moving hosts. WiMax and 3G cellphone allows the host to move at a vehicle speed under 120 km/h. However, MWBA allows the host to move at a speed of 250 km/h, which is the only protocol that works for high-speed trains. Both WiMax and MWBA are designed with a transmission latency of less that 20 ms. However, 3G cellphone networks have a variable latency of 10–500 ms.

32.3.4 Properties of Networked Telerobotics

As defined by *Mason*, *Peshkin*, and others [32.43, 44], in *quasistatic* robot systems, accelerations and inertial forces are negligible compared to dissipative forces. In quasistatic robot systems, motions are often modeled as transitions between discrete atomic *configurations*.

We adopt a similar terminology for networked telerobots. In quasistatic telerobotics (QT), robot dynamics and stability are handled locally. After each atomic motion, a new state report is presented to the remote user, who sends back an atomic command. The atomic state describes the status of the robot and its corresponding environment. Atomic commands refer to human directives, which are desired robotic actions.

Several issues arise

- *State-command presentation*: How should state and available commands be presented to remote human operators using the two-dimensional (2-D) screen display?
- *Command execution/state generation*: How should commands be executed locally to ensure that the desired state is achieved and maintained by the robot?
- *Command coordination*: How should commands be resolved when there are multiple human operators?

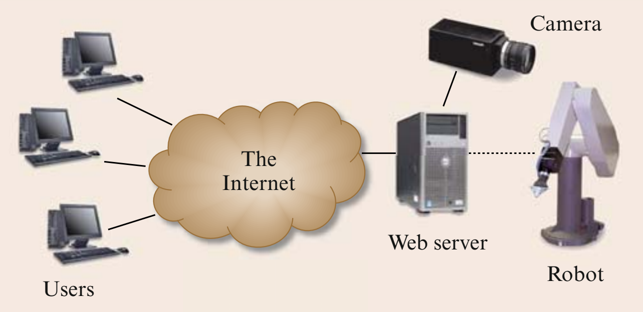


Fig. 32.4 Typical system architecture for a networked telerobot

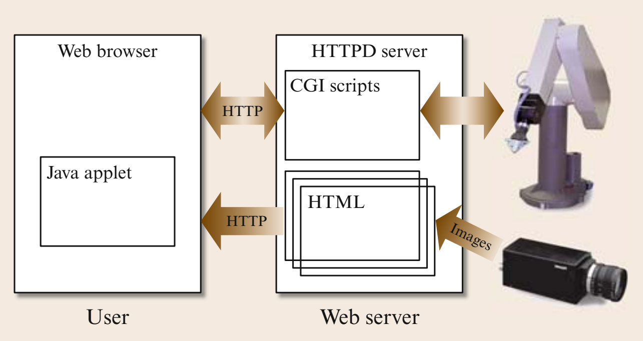


Fig. 32.5 A sample software architecture of a networked telerobot

32.3.5 Building a Networked Telerobotic System

As illustrated in Fig. 32.4, a typical networked telerobotic system typically includes three components:

- users: anyone with an Internet connection and a web browser
- web server: a computer running a web server software
- robot: a robot manipulator, a mobile robot, or any device that can modify or affect its environment

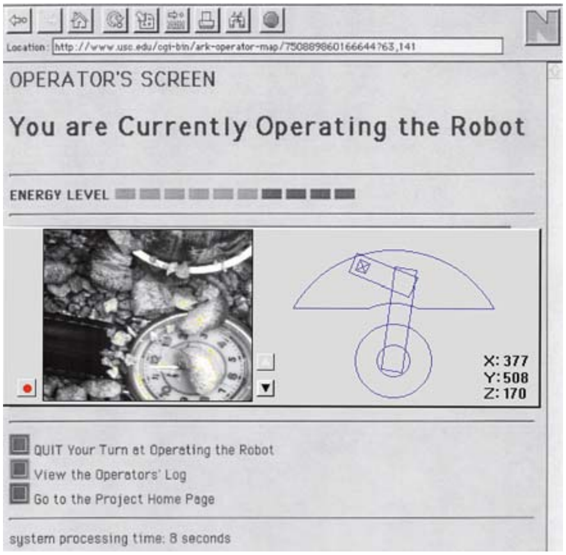
Users access the system via their web browsers. Any web browser that is compatible with W3C's HTML standard can access a web server. In 2006, the most popular web browsers are Microsoft Internet Explorer, Netscape, Mozilla Firefox, Safari, and Opera. New browsers and updated versions with new features are introduced periodically.

A web server is a computer that responds to HTTP requests over the Internet. Depending upon the operating system of the web server, popular server software packages include Apache and Microsoft Internet Information Services (IIS). Most servers can be freely downloaded from the Internet.

To develop a networked telerobot, one needs a basic knowledge of developing, configuring, and maintaining web servers. As illustrated in Fig. 32.5, the development requires knowledge of HTML and at least one local programming languages such as C, CGI, Javascript, Perl, PHP, or Java.

It is important to consider compatibility with the variety of browsers. Although HTML is designed to be compatible with all browsers, there are exceptions. For example, Javascript, which is the embedded scripting language of web browsers, is not completely compatible between Internet Explorer and Netscape. One also needs to master the common HTML components such as forms that are used to accept user inputs, frames that are used to divide the interface into different functional regions, etc. An introduction to HTML can be found in [32.45].

User commands are usually processed by the web server using CGI, the common gateway interface. Most sophisticated methods such as PHP, Java Server Pages (JSP), and socket programming can also be used. CGI is invoked by the HTTP server when the CGI script is referred in the Uniform Resource Locator (URL). The CGI program then interprets the inputs, which is often the next robot motion command, and sends commands to the robot via a local communication channel. CGI scripts can be written in almost any programming language. The most popular ones are Perl and C.


Location: http://www.usc.edu/cgi-bin/ark-operator-map/7508898601666447?63,141

OPERATOR'S SCREEN

You are Currently Operating the Robot

ENERGY LEVEL

X: 377
Y: 508
Z: 170

QUIT Your Turn at Operating the Robot
View the Operators' Log
Go to the Project Home Page

system processing time: 8 seconds

Fig. 32.6 Browser's view of the first networked telerobot interface [32.46]. The schematic at *lower right* gives an overhead view of position of the four-axis robot arm (with the camera at the end marked with X), and the image at the *lower left* indicates the current view of the camera. The *small button marked with a dot at the left* directs a 1 s burst of compressed air into the sand below the camera. The Mercury Project was online from August 1994 to March 1995

A simple networked telerobotic system can be constructed using only HTML and CGI. However, if the robot requires a sophisticated control interface, a Java applet is recommended. Java applets run inside the web browser on the client's computer. Information about Java can be found at Sun Microsystems' official Java home page.

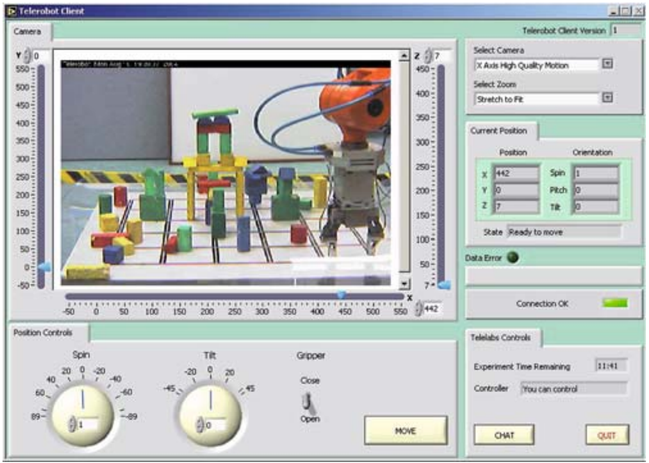
Most telerobotic systems also collect user data and robot data. Therefore, database design and data processing program are also needed. The most common used databases include MySQL and PostgresSQL. Both are open-source databases and support a variety of platforms and operation systems. Since a networked telerobotic system is online 24 hours a day, reliability is also an important consideration in system design. Website security is critical. Other common auxiliary developments include online documentation, online manual, and user feedback collection.

32.3.6 State-Command Presentation

To generate a correct and high-quality command depends on how effectively the human operator understands the state feedback. The state-command presentation contains three subproblems: the 2-D representation of the true robot state (state display), the assistance provided by the interface to generate new commands (spatial reasoning), and the input mechanism.

Browser Displays

Unlike traditional point-to-point teleoperation, where specialized training and equipment are available to ope-



Telerobot Client Version 1

Select Camera: X Axis High Quality Motion

Select Zoom: Stretch to Fit

X 442 Spin 1
Y 0 Pitch 0
Z 7 Tilt 0

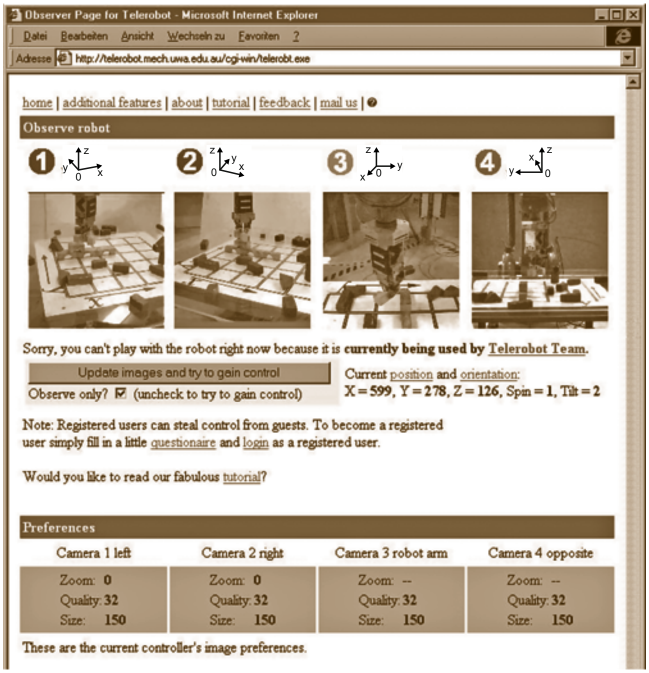
State: Ready to move

Connection OK

Experiment Time Remaining 11:41

Controller: You can control

Fig. 32.7 Browser interface to the Australian networked telerobot which was a six-axis arm that could pick up and move blocks [32.16]


Observer Page for Telerobot - Microsoft Internet Explorer
Datei Bearbeiten Ansicht Wechseln zu Favoriten ?
Adresse http://telerobot.mech.uwa.edu.au/cgi-win/telerobt.exe
home | additional features | about | tutorial | feedback | mail us |
Observe robot
1 z y x 0
2 z y x 0
3 z y x 0
4 x z y 0
Update images and try to gain control
Preferences

Sorry, you can't play with the robot right now because it is **currently being used by** Telerobot Team.

Observe only? ☑ (uncheck to try to gain control)

Current position and orientation:
X = 599, Y = 278, Z = 126, Spin = 1, Tilt = 2

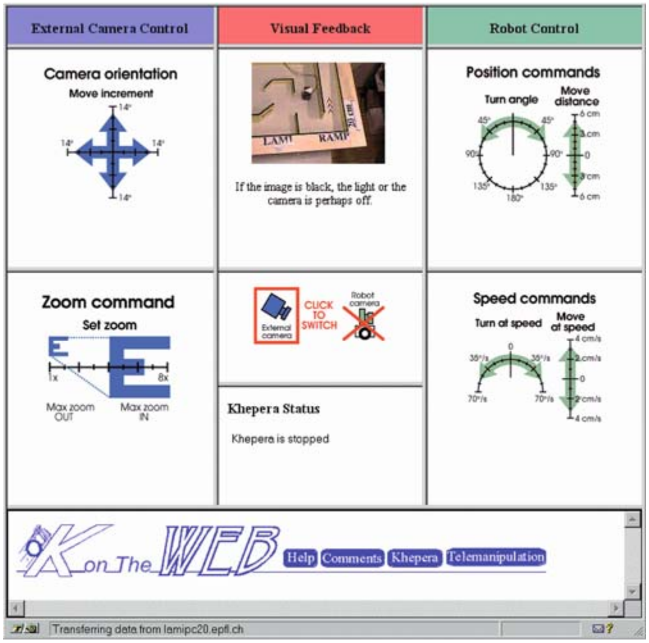
Note: Registered users can steal control from guests. To become a registered user simply fill in a little questionaire and login as a registered user.

Would you like to read our fabulous tutorial?

Camera 1 left	Camera 2 right	Camera 3 robot arm	Camera 4 opposite
Zoom: 0 Quality: 32 Size: 150	Zoom: 0 Quality: 32 Size: 150	Zoom: -- Quality: 32 Size: 150	Zoom: -- Quality: 32 Size: 150

These are the current controller's image preferences.

Fig. 32.8 Use of a multicamera system for multi-viewpoint state feedback [32.47]



If the image is black, the light or the camera is perhaps off.

Khepera Status

Khepera is stopped

Transferring data from lamipc20.epfl.ch

Fig. 32.9 Camera control and mobile robot control in Patrick Saucy and Francesco Mondada's Khep on the web project

rators, networked telerobots offer wide access to the general public. Designers cannot assume that operators have any prior experience with robots. As illustrated in Fig. 32.6, networked telerobotic systems must display the robot state on a 2-D screen display.

The states of the teleoperated robot are often characterized in either world coordinates or robot joint configuration, which are either displayed in numerical format or through a graphical representation. Figure 32.6 lists robot *XYZ* coordinates on the interface and draws a simple 2-D projection to indicate joint configurations. Figure 32.7 illustrates another example of teleoperation interface that was developed by *Taylor* and *Trevelyan* [32.36]. In this interface, *XYZ* coordinates are presented in a sliding bar near the video window.

The state of the robot is usually displayed in a 2-D view as shown in Figs. 32.6 and 32.7. In some systems, multiple cameras can help the human operator to understand the spatial relationship between the robot and the objects in the surrounding environment. Figure 32.8 shows an example with four distinct camera views for a six-degree-of-freedom industrial robot.

Figure 32.9 demonstrate an interface with a pan–tilt–zoom robotic camera. The interface in Fig. 32.9 is designed for a mobile robot.

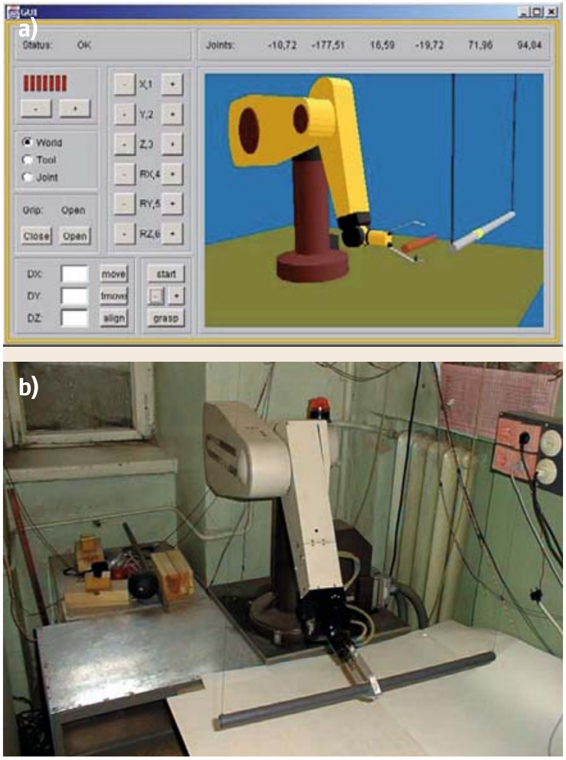
More sophisticated spatial reasoning can eliminate the need for humans to provide low-level control by automatically generating a sequence of commands after it receives task-level commands from the human operator. This is particularly important when the robotic system is highly dynamic and requires a very fast response. In this case, it is impossible to ask the human to generate intermediate steps in the robot control; for example, *Belousov* et al. adopt a shared autonomy model to direct a robot to capture a moving rod [32.27]. *Fong* and *Thorpe* [32.48] summarize vehicle teleoperation systems that utilize these supervisory control techniques. *Su* et al. developed an incremental algorithm for better translation of the intention and motion of operators into remote robot action commands [32.32].

Human Operator Input

Most networked telerobotic systems only rely on mouse and keyboards for input. The design problem is what to click on in the interface. Given the fact that user commands can be quite different, we need to adopt an appropriate interface for inputs; for example, inputs could be Cartesian *XYZ* coordinates in world coordinate system or robot configurations in angular joint configurations.

For angular inputs, it is often suggested to use a round dial as a control interface, as illustrated in bottom left of Fig. 32.7 and the right-hand side of Fig. 32.9. For linear motion in Cartesian coordinate, arrows operated by either mouse clicks or the keyboard are often suggested. Position and speed control are often needed, as illustrated in Fig. 32.9. Speed control is usually controlled by mouse clicks on a linear progress bar for translation and a dial for rotation.

The most common control type is position control. The most straightforward way is to click on the video image directly. To implement the function, the software needs to translate the 2-D click inputs into three-dimensional (3-D) world coordinates. To simplify the problem, the system designer usually assumes that the clicked position is on a fixed plane; for example, a mouse click on the interface of Fig. 32.6 assumes the robot moves on the X–Y plane. The combination of a mouse click on the image can also allow abstract task-level command. The example in Fig. 32.12 uses mouse clicks to place votes on an image to generate a command that directs a robot to pick up a test agent at the task level.


Status: OK

Joints: -10,72 -177,51 16,59 -19,72 71,96 94,04

Grip: Open

Fig. 32.10a,b A web-based teleoperation system that allows robot to capture a fast-moving rod [32.27] **(a)** User interface and **(b)** system setup

32.3.7 Command Execution/State Generation

When a robot receives a command, it executes the command and a new state is generated and transmitted back to the human operator. However, commands may not arrive in time or may get lost in transmission. Also, because users are often inexperienced, their commands may contain errors.

Belousov and colleagues demonstrated a system that allowed a web user to capture a fast rod that is thrown at a robot manipulator [32.27]. Over the limited communication channel, it is impossible to ask the human to control the manipulator directly. Computer vision and augmented-reality-based local intelligence is required

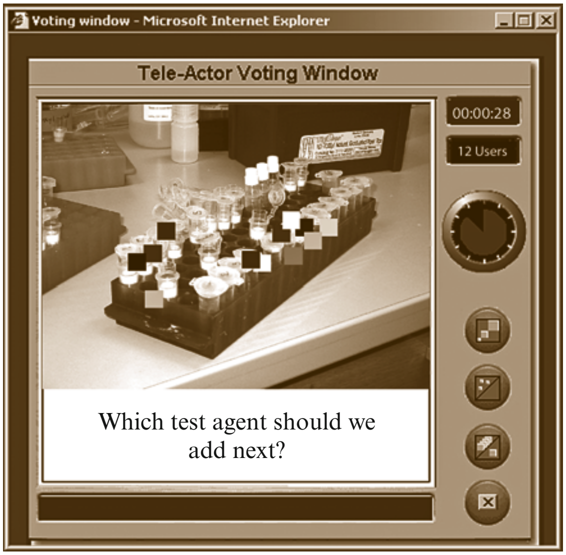

Which test agent should we add next?

Fig. 32.11 Spatial dynamic voting interface for the Tele-Actor system [32.41]: the spatial dynamic voting (SDV) interface as viewed by each user. In the remote environment, the Tele-Actor takes images with a digital camera, which are transmitted over the network and displayed to all participants with a relevant question. With a mouse click, each user places a color-coded marker (a *votel* or voting element) on the image. Users view the position of all votels and can change their votel positions based on the group's response. Votel positions are then processed to identify a *consensus region* in the voting image that is sent back to the Tele-Actor. In this manner, the group collaborates to guide the actions of the Tele-Actor

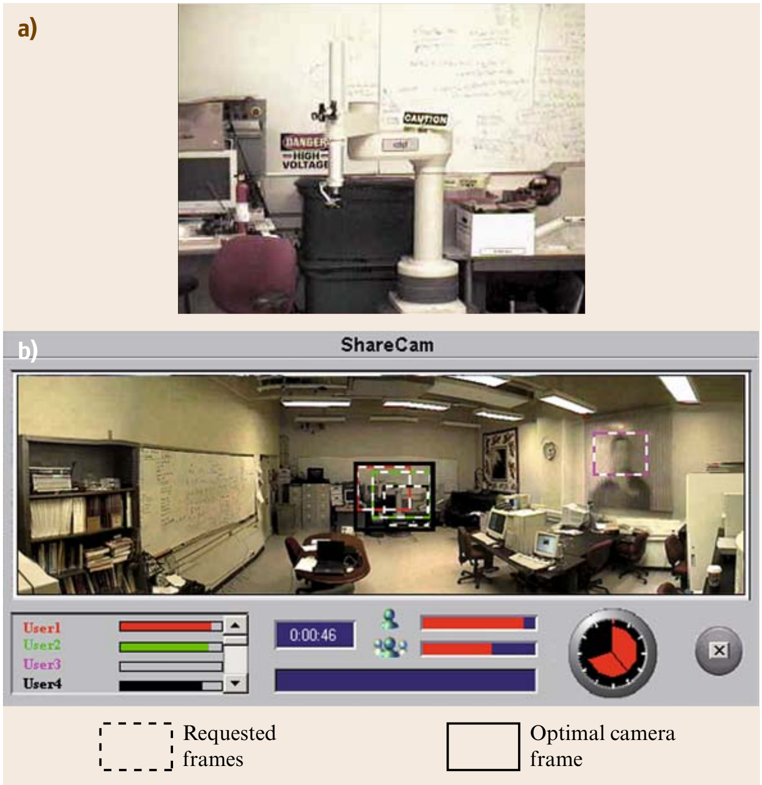


Fig. 32.12a,b Frame selection interface [32.49]. The user interface includes two image windows. The lower window **(b)** displays a fixed panoramic image based on the camera's full workspace (reachable field of view). Each user requests a camera frame by positioning a *dashed rectangle* in **(b)**. Based on these requests, the algorithm computes an optimal camera frame (shown with a *solid rectangle*), moves the camera accordingly, and displays the resulting live streaming video image in the upper window **(a)**

to assist the human operator. The rod is on bifilar suspension, performing complicated oscillations. Belousov et al. designed a shared-autonomy control to implement the capture. First, an operator chooses the desired point for capture on the rod and the capture instant using a 3-D online virtual model of the robot and the rod. Then, the capturing operation is performed automatically using a motion prediction algorithm that is based on the rod's motion model and two orthogonal camera inputs, which perceive the rod's position locally in real time.

32.3.8 Collaborative Control

When more than one human is sharing control of the device, command coordination is needed. According to [32.50], multiple human operators can reduce the chance of errors, cope with malicious inputs, utilize operators' different expertise, and train new operators. In [32.51], a *collaborative telerobot* is defined as a telerobot simultaneously controlled by many participants, where input from each participant is combined to generate a single control stream.

When group inputs are in the form of direction vectors, averaging can be used as an aggregation mechanism [32.52]. When decisions are distinct choices or at the abstract task level, voting is a better choice [32.41]. As illustrated in Fig. 32.12, *Goldberg* and *Song* develop the Tele-Actor system using spatial dynamic voting. The Tele-Actor is a human equipped with an audio/video device and controlled by a group of online users. Users indicate their intensions by positioning their votes on a 320 × 320 pixel voting image during the voting interval. Votes are collected at the server and used to determine the Tele-Actor's next action based on the most requested region on the voting image. (see http://www.tele-actor.net)

Song and *Goldberg* [32.49, 53] developed a controllable camera that allowed many clients to share control of its camera parameters, as illustrated in Fig. 32.12. Users indicate the area they want to view by drawing rectangles on a panoramic image. The algorithm computes an optimal camera frame with respect to the user satisfaction function, which is defined as the frame selection problem.

32.4 Conclusion and Future Directions

As this technology matures, networked telerobots will gradually go beyond university laboratories and find application in the real world. A new project, the Collaborative Observatory for Nature Environments (CONE) project, proposed by *Song* and *Goldberg* [32.54], aims to design a networked robotic camera system to collect data from the wilderness for natural scientists. The fast development of networked telerobot system is not limited to North America. Japan's Advanced Telecommunications Research Institute International (ATR) Intelligent Robotics and Communication Laboratory has also announced its networked robot project led by Norihiro Hagita (ATR). Its mission is to develop network-based intelligent robots for applications such as service, medical, and safety. Hideyuki Tokuda (Keio University) chaired the Networked Robot Forum in Spring 2005, which promotes research and development (R&D) and standardization on network robots through activities to support awareness campaigns and verification experiments in collaboration among wide-ranging parties, which includes over 100 industry and academic members. Korea's Ministry of Information and Communication has also announced the Ubiquitous Robotic Companion (URC) project to develop network-based intelligent robots.

Networked telerobots have allowed tens of thousands of nonspecialists around the world to interact with robots. The design of networked telerobots presents a number of engineering challenges to build reliable systems that can be operated by nonspecialists 24 hours a day, 7 days a week and remain online for years. Many new research challenges remain.

- *New interfaces*: As portable devices such as cellphones and portable digital assistants (PDAs) becomes grow in computation power, networked telerobotics should be able to adopt them as new interfaces. As computers becomes increasingly powerful, they become capable of visualizing more sophisticated sensor inputs. Designers of new interfaces should also keep track of new developments in hardware such as haptic interfaces and voice recognition systems. New software standards such as flash, extensible markup language (XML), extensible hyper text markup language (XHTML), virtual reality modeling language (VRML), and wireless markup language (WML) will also change the way we design interface.
- *New algorithms*: Algorithms determine performance. Scalable algorithms that are capable of handing large amounts of data such as video/sensor network inputs and utilize fast-evolving hardware capability such as distributed and parallel computation will become increasingly important in the networked telerobotics.
- *New protocols*: Although we have listed some pioneering work in changing the network environment to improve teleoperation, there are still a large number of open problems such as new protocols, appropriate bandwidth allocation [32.55], QoS [32.56], security, routing mechanisms [32.28], and many more. Network communication is a very fast-evolving field. The incorporation/modification of network communication ideas into networked telerobotic system design will continue to be an active research area. The common object request broker architecture (CORBA) or real-time CORBA [32.19, 20, 57, 58] have great potential for networked telerobotics.
- *Applications*: Many new applications are emerging in areas such as security, inspection, education, and entertainment. Application requirements such as reliability, security, and modularity will continuous to pose new challenges for system design.

References

32.1 K. Goldberg, R. Siegwart (Eds.): *Beyond Webcams: An Introduction to Online Robots* (MIT Press, Cambridge 2002)

32.2 N. Tesla: Method of an apparatus for controlling mechanism of moving vessels or vehicles http://www.pbs.org/tesla/res/613809.html (1898)

32.3 R. Goertz, R. Thompson: Electronically controlled manipulator, Nucleonics **12**(11), 46–47 (1954)

32.4 R.D. Ballard: A last long look at titanic, National Geographic **170**(6), 698–727 (1986)

32.5 A.K. Bejczy: Sensors, controls, and man–machine interface for advanced teleoperation, Science **208**(4450), 1327–1335 (1980)

32.6 R.S. Mosher: Industrial manipulators, Sci. Am. **211**(4), 88–96 (1964)

32.7 R. Tomovic: On man–machine control, Automatica **5**(4), 40–404 (1969)

32.8 A. Bejczy, G. Bekey, R. Taylor, S. Rovetta: A research methodology for tele-surgery with time delays, First Int. Symp. Med. Robot. Comp. Assist. Surg. (1994)

32.9 M. Gertz, D. Stewart, P. Khosla: A human–machine interface for distributed virtual laboratories, IEEE Robot. Autom. Mag. **1**(4), 5–13 (1994)

32.10 T. Sato, J. Ichikawa, M. Mitsuishi, Y. Hatamura: A new micro-teleoperation system employing a hand-held force feedback pencil, IEEE Int. Conf. Robot. Autom. (1994)

32.11 T.B. Sheridan: *Telerobotics, Automation, and Human Supervisory Control* (MIT Press, Cambridge 1992)

32.12 http://www.cl.cam.ac.uk/coffee/qsf/timeline.html

32.13 K. Goldberg, M. Mascha, S. Gentner, N. Rothenberg, C. Sutter, J. Wiegley: Robot teleoperation via WWW, IEEE Int. Conf. Robot. Autom. (1995)

32.14 K. Goldberg, M. Mascha, S. Gentner, N. Rothenberg, C. Sutter, J. Wiegley: Beyond the web: Manipulating the physical world via the WWW, Comp. Netw. ISDN Syst. J. **28**(1), 209–219 (1995), Archives can be viewed at http://www.usc.edu/dept/raiders/

32.15 B. Dalton, K. Taylor: A framework for internet robotics, IEEE Int. Conf. Intell. Robot. Syst. (IROS): Workshop on Web Robots (Victoria 1998)

32.16 http://telerobot.mech.uwa.edu.au/

32.17 H. Hu, L. Yu, P.W. Tsui, Q. Zhou: Internet-based robotic systems for teleoperation, Assembly Autom. **21**(2), 143–151 (2001)

32.18 R. Safaric, M. Debevc, R. Parkin, S. Uran: Telerobotics experiments via internet, IEEE Trans. Ind. Electron. **48**(2), 424–431 (2001)

32.19 S. Jia, K. Takase: A CORBA-based internet robotic system, Adv. Robot. **15**(6), 663–673 (2001)

32.20 S. Jia, Y. Hada, G. Ye, K. Takase: Distributed telecare robotic systems using CORBA as a communication architecture, IEEE Int. Conf. Robot. Autom. (ICRA) (Washington 2002)

32.21 J. Kim, B. Choi, S. Park, K. Kim, S. Ko: Remote control system using real-time mpeg-4 streaming technology for mobile robot, IEEE Int. Conf. Consum. Electron. (2002)

32.22 T. Mirfakhrai, S. Payandeh: A delay prediction approach for teleoperation over the internet, IEEE Int. Conf. Robot. Autom. (ICRA) (2002)

32.23 K. Han, Y. Kim, J. Kim, S. Hsia: Internet control of personal robot between kaist and uc davis, IEEE Int. Conf. Robot. Autom. (ICRA) (2002)

32.24 L. Ngai, W.S. Newman, V. Liberatore: An experiment in internet-based, human-assisted robotics, IEEE Int. Conf. Robot. Autom. (ICRA) (2002)

32.25 R.C. Luo, T.M. Chen: Development of a multibehavior-based mobile robot for remote supervisory control through the internet, IEEE/ASME Trans. Mechatron. **5**(4), 376–385 (2000)

32.26 D. Aarno, S. Ekvall, D. Kragi: Adaptive virtual fixtures for machine-assisted teleoperation tasks, IEEE Int. Conf. Robot. Autom. (ICRA) (2005) pp. 1151–1156

32.27 I. Belousov, S. Chebukov, V. Sazonov: Web-based teleoperation of the robot interacting with fast moving objects, IEEE Int. Conf. Robot. Autom. (ICRA) (2005) pp. 685–690

32.28 Z. Cen, A. Goradia, M. Mutka, N. Xi, W. Fung, Y. Liu: Improving the operation efficiency of supermedia enhanced internet based teleoperation via an overlay network, IEEE Int. Conf. Robot. Autom. (ICRA) (2005) pp. 691–696

32.29 N.P. Jouppi, S. Thomas: Telepresence systems with automatic preservation of user head height, local rotation, and remote translation, IEEE Int. Conf. Robot. Autom. (ICRA) (2005) pp. 62–68

32.30 B. Ricks, C.W. Nielsen, M.A. Goodrich: Ecological displays for robot interaction: a new perspective, Int. Conf. Intell. Robot. Syst. (IROS), Vol. 3 (2004) pp. 2855–2860

32.31 D. Ryu, S. Kang, M. Kim, J. Song: Multi-modal user interface for teleoperation of robhaz-dt2 field robot system, Int. Conf. Intell. Robot. Syst. (IROS), Vol. 1 (2004) pp. 168–173

32.32 J. Su, Z. Luo: Incremental motion compression for telepresent walking subject to spatial constraints, IEEE Int. Conf. Robot. Autom. (ICRA) (2005) pp. 69–74

32.33 I. Toshima, S. Aoki: Effect of driving delay with an acoustical tele-presence robot, telehead, IEEE Int. Conf. Robot. Autom. (ICRA) (2005) pp. 56–61

32.34 N. Chong, T. Kotoku, K. Ohba, K. Komoriya, N. Matsuhira, K. Tanie: Remote coordinated controls in multiple telerobot cooperation, IEEE Int. Conf. Robot. Autom., Vol. 4 (2000) pp. 3138–3343

32.35 E. Paulos, J. Canny, F. Barrientos: Prop: Personal roving presence, SIGGRAPH Vis. Proc. (1997) p. 99

32.36 K. Taylor, J.P. Trevelyan: Australia's telerobot on the web, 26th Symp. Ind. Robot. (Singapore 1995) pp. 39–44

32.37 A. Khamis, D.M. Rivero, F. Rodriguez, M. Salichs: Pattern-based architecture for building mobile robotics remote laboratories, IEEE Int. Conf. Robot. Autom. (ICRA) (Taipei 2003) pp. 3284–3289

32.38 C. Cosma, M. Confente, D. Botturi, P. Fiorini: Laboratory tools for robotics and automation education, IEEE Int. Conf. Robot. Autom. (ICRA) (Taipei 2003) pp. 3303–3308

32.39 K.W. Dorman, J.L. Pullen, W.O. Keksz, P.H. Eismann, K.A. Kowalski, J.P. Karlen: The servicing aid tool: A teleoperated robotics system for space applications, The Seventh Annual Workshop on Space Operations Applications and Research (SOAR 1993), Vol. 1 (Johnson Space Center, Houston 1994)

32.40 C. Pollak, H. Hutter: A webcam as recording device for light microscopes, J. Comp. Assist. Microsc. **10**(4), 179–183 (1998)

32.41 K. Goldberg, D. Song, A. Levandowski: Collaborative teleoperation using networked spatial dynamic voting, Proc. IEEE **91**(3), 430–439 (2003)

32.42 J. Walrand, P. Varaiya: *High-Performance Communication Networks*, 2nd edn. (Morgan Kaufmann, San Francisco 2000)

32.43 M.A. Peshkin, A.C. Sanderson: Minimization of energy in quasi-static manipulation, IEEE Trans. Robot. Autom. **5**(1), 53–60 (1989)

32.44 M.T. Mason: On the scope of quasi-static pushing, 3rd Int. Symp. Robot. Res., ed. by O. Faugeras, G. Giralt (MIT Press, Cambridge 1986)

32.45 E. Ladd, J. O'Donnell: *Using HTML 4, XML, and Java 1.2* (QUE Press, 1998)

32.46 K. Goldberg, M. Mascha, S. Gentner, N. Rothenberg, C. Sutter, J. Wiegley: Desktop tele-operation via the world wide web, IEEE Int. Conf. Robot. Autom. (Nagoya 1995)

32.47 H. Friz: Design of an Augmented Reality User Interface for an Internet Based Telerobot Using Multiple Monoscopic Views. Ph.D. Thesis (Technical University of Clausthal, Clausthal-Zellerfeld 2000)

32.48 T. Fong, C. Thorpe: Vehicle teleoperation interfaces, Auton. Robot. **11**, 9–18 (2001)

32.49 D. Song, A. Pashkevich, K. Goldberg: Sharecam part II: Approximate and distributed algorithms for a collaboratively controlled robotic webcam, IEEE/RSJ Int. Conf. Intell. Robot. (IROS), Vol. 2 (Las Vegas 2003) pp. 1087–1093

32.50 K. Goldberg, B. Chen, R. Solomon, S. Bui, B. Farzin, J. Heitler, D. Poon, G. Smith: Collaborative teleoperation via the internet, IEEE Int. Conf. Robot. Autom. (ICRA), Vol. 2 (2000) pp. 2019–2024

32.51 D. Song: Systems and Algorithms for Collaborative Teleoperation. Ph.D. Thesis (Department of Industrial Engineering and Operations Research, University of California 2004)

32.52 K. Goldberg, B. Chen: Collaborative teleoperation via the internet, Int. Conf. Intell. Robot. Syst. (IROS) (2001)

32.53 D. Song, K. Goldberg: Sharecam part I: Interface, system architecture, and implementation of a collaboratively controlled robotic webcam, IEEE/RSJ Int. Conf. Intell. Robot. (IROS), Vol. 2 (Las Vegas 2003) pp. 1080–1086

32.54 D. Song, K. Goldberg: CONE Project (www.c-o-n-e.org)

32.55 P.X. Liu, M. Meng, S.X. Yang: Data communications for internet robots, Auton. Robot. **15**, 213–223 (2003)

32.56 W. Fung, N. Xi, W. Lo, B. Song, Y. Sun, Y. Liu, I.H. Elhajj: Task driven dynamic qos based bandwidth allcoation for real-time teleoperation via the internet, IEEE/RSJ Int. Conf. Intell. Robot. Syst. (Las Vegas 2003)

32.57 M. Amoretti, S. Bottazzi, M. Reggiani, S. Caselli: Evaluation of data distribution techniques in a CORBA-based telerobotic system, IEEE/RSJ Int. Conf. Intell. Robot. Syst. (Las Vegas 2003)

32.58 S. Bottazzi, S. Caselli, M. Reggiani, M. Amoretti: A software framework based on real time COBRA for telerobotics systems, IEEE/RSJ Int. Conf. Intell. Robot. Syst., EPFL (Lausanne 2002)