

The Mercury Project: A Feasibility Study for Internet Robots *

Ken Goldberg, Steve Gentner, Carl Sutter, and Jeff Wiegley
UC Berkeley and University of Southern California †

As a feasibility study in 1994, we built a system that allows a robot manipulator to be teleoperated via the World Wide Web. Although the field of teleoperation dates back over 50 years, the WWW provides a low-cost and widely-available interface that can make teleoperated resources accessible to a broad range of users. The Mercury Project consisted of an industrial robot arm fitted with a CCD camera and a pneumatic system. We placed a sandbox filled with buried artifacts in the robot workspace. Using the ISMAP feature of HTTP, users could remotely move the camera to view desired locations or direct a short burst of compressed air into the sand to view the newly cleared region.

To our knowledge, the Mercury Project was the first system to permit Internet users to remotely view and manipulate the real world. It was available almost continuously from September 1994 through March 1995 and was accessed by over 50,000 unique hosts. This paper focuses on the interface design, robot hardware, and system architecture of the system, which is archived at

<http://www.usc.edu/dept/raiders/>

The Telegarden,¹ was our second Internet Robot project. It allows users to view and interact with a remote garden filled with living plants. The Telegarden was online for 4 years starting in August 1995:

<http://telegarden.aec.at/>

1 Goals of the Mercury Project

Initiated at CERN in 1992, the World Wide Web provides a standard graphical interface to the Internet [1]; the number

This work was supported in part by NSF Awards IRI-9457523 and IRI-9553197. Parts of this paper appeared in the IEEE International Conference on Robotics and Automation, May 19-26, 1995, Nagoya, JAPAN and at the Second International WWW Conference, Chicago, IL, Oct 17, 1994.

Goldberg, Gentner, and Wiegley were with the Computer Science Department at the University of Southern California when this work was performed. Goldberg is now at UC Berkeley and can be reached at goldberg@ieor.berkeley.edu.

¹Telegarden Project Co-Directors: Ken Goldberg (UC Berkeley) and Joseph Santarromana (UC Irvine) USC Project Team: George Bekey, Steven Gentner, Rosemary Morris, Carl Sutter, Jeff Wiegley, Erich Berger (AEC). The Telegarden was funded in part by USC, Adept Technology, and the Ars Electronica Center.

of users worldwide has grown exponentially. In the Spring of 1994, we conjectured that it might be possible to offer public access to a teleoperated robot via the WWW.

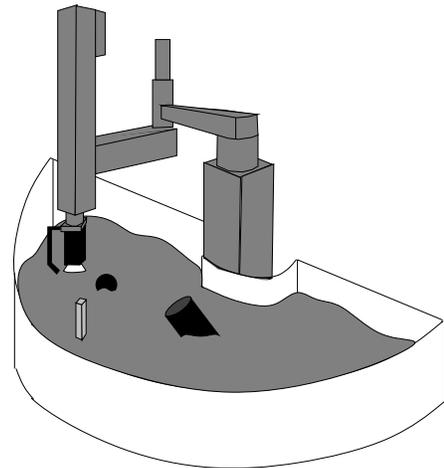


Figure 1: Robot, camera and air nozzle above workspace.

As illustrated in Figure 1, we set up a SCARA-type robot arm over a semi-annular workspace containing sand and buried artifacts. We attached a CCD camera to the end of the arm along with a nozzle to direct air bursts into the sand. We then developed an interface so this hardware could be controlled via the Internet.

Our primary criterion was that the system be reliable enough to operate 24 hours a day and survive user attempts at sabotage. A practical criterion was that the system be low in cost as we had a limited budget. It is worth noting that the manufacturing industry uses similar criteria, reliability and cost, to evaluate robots for production. Thus our experience with RISC robotics [2] proved helpful.

Our secondary goal was to create an evolving WWW site that would encourage repeat visits by users. Toward this end, all of the buried artifacts were derived from an unnamed 19th Century text. Users were asked to identify this text and thereby collectively solve the “puzzle”. After each 5-minute operating session, users are prompted to describe their findings and hypotheses in an ongoing Operator’s Log. When the system was decommissioned the Operator’s Log

contained over 1000 pages of entries. The answer to the puzzle was announced in March 1995: Jules Verne's *Journey to the Center of the Earth*.

2 Related Work

Goertz demonstrated one of the first "master-slave" teleoperators 50 years at the Argonne National Laboratory [3]. Remotely operated mechanisms have long been desired for use in inhospitable environments such as radiation sites, undersea [4] and space exploration [5]. At General Electric, Mosher [6] developed a complex two-arm teleoperator with video cameras. Prosthetic hands were also applied to teleoperation [7]. More recently, teleoperation is being considered for medical diagnosis [8], manufacturing [9] and micromanipulation [10]. See Sheridan [11] for an excellent review of the extensive literature on teleoperation and telerobotics.

Most of these systems require fairly complex hardware at the human interface: exoskeleton master linkages are attached to the human arm to minimize the kinesthetic effects of distance to create a sense of "telepresence". Our objective was to provide widespread access by using only the interface available in standard HTML.

Internet interfaces to coke machines were demonstrated in the early 1980s, well before the introduction of the WWW in 1992. One of the first web cameras was the Trojan coffee pot at Cambridge [12]. We believe the Mercury Project was the first to link a robot to the WWW, but it is important to acknowledge the independent work of Ken Taylor at the University of Western Australia. Taylor demonstrated a remotely controlled six-axis telerobot with a fixed observing camera in October 1994 [13]. Although Taylor's initial system required users to type in spatial coordinates to specify relative arm movements, he and his colleagues have subsequently explored a variety of user interfaces [14].

Also in October 1994, Wallace demonstrated a robotic camera [15] and Cox put up a system that allows WWW users to remotely schedule photos from a robotic telescope [16]. Paulos and Canny have implemented several Internet robots with elegant user interfaces [17, 18]. Bekey and Steve Goldberg used a six-axis robot, rotating platform, and stereo cameras to allow remote scholars to closely inspect a rare sculpture [19]. Since then there have been a number of Internet-controllable mobile robots [20, 21, 22] and NASA has used the Internet for control of vehicles in space [23]. Recently, Stein [24] demonstrated a Java interface to control a painting robot over the Internet. An impressive variety of other Internet robot projects were reported in a recent IROS workshop [25].

Schulz [26], Slotine [27] and Tarn [28] show how control theory used for teleoperation over dedicated lines can be adapted to the Internet, where delays vary unpredictably.

3 System Design and User Interface

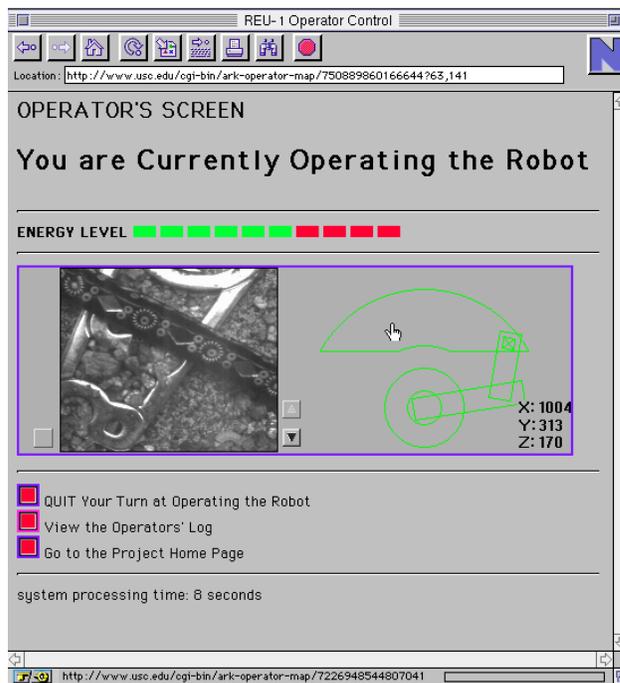


Figure 2: The interface as viewed by a WWW browser.

To facilitate use by a wide audience of non-specialists, we sought to make all robot controls available via the standard point-and-click mouse commands as shown in Figure 2. This forced us to consider a 2D workspace with only a few buttons for out-of-plane effects. Users are trained with an on-line tutorial prior to operating the robot.

The user interface centers around the bitmap that we call the "status image" as shown in Figure 3. Any number of "observers" can simultaneously view the status image, but only the current "operator" can send commands by clicking on the image. To limit access to one operator at a time, we implemented password authentication and a queue that gives each operator 5 minutes at the helm.

When the operator clicks on the status image using the mouse, the XY coordinates are transferred back to our server, which interprets them to decode the desired robot action. This action can be: (1) a global move to center the camera at XY in the schematic workspace, (2) a local move to center the camera at XY in the camera image, (3) moving the camera to one of two fixed Z heights, or (4) blowing a burst of compressed air into the sand directly below the camera.

We worked to reduce the size of the status image to minimize turnaround time when a command is issued. The average image size for the status image, encoded as a .gif file, is 17.3 Kbytes. Although we were able to achieve response times

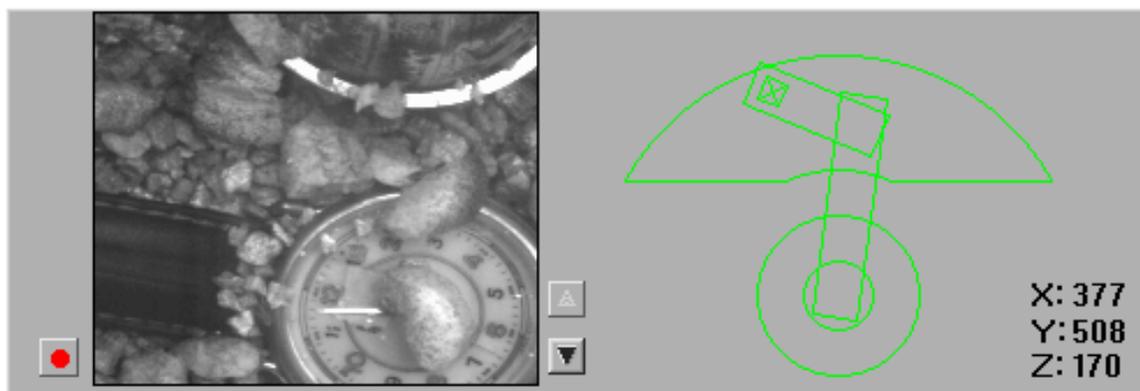


Figure 3: The “status image”. At the right is a schematic top view of the semi-annular workspace and robot linkage. At left is a CCD camera image of the view directly beneath the robot end-effector. Up/Down buttons are included for Z motion of the camera, and the round button is used to blow a burst of compressed air into the sand.

of 10 seconds for on-campus users, cycle times of up to 60 seconds were reported from users in Europe operating via 14.4K telephone lines.

We also created a fictional context for the system, inventing the history of a deceased paleohydrologist who had discovered unexplained artifacts in a radioactive region of southwest Nevada. We explained that the Mercury robot was originally developed to explore that region and that one mandate of our grant was to make our system “available to the broader scientific community”. A hypertext document describing this background provides an online introduction.

4 Robot and Camera

The SCARA robot is an IBM SR5427 built by Sankyo in early 1980. SCARA stands for “Selective Compliance Assembly Robot Arm”; common in industrial assembly for “pick-and-place” operations because it is fast, accurate and has a large 2.5D workspace. We selected this robot over other robots in our lab due to its excellent durability, large workspace, and because it was gathering dust in our lab.

Unfortunately IBM no longer supports this robot and we were forced to read two antiquated BASIC programs and monitor their serial line transmissions to decipher the protocols needed for serial control of the robot. The robot accepts joint motion commands using IEEE format and checksums.

To allow users to *manipulate* the remote environment we initially planned to place a simple gripper at the end effector. Anticipating user attempts at sabotage (which is, after all, the time-honored hacker tradition), we opted to use compressed air as the medium for manipulation.

The CCD camera is an EDC 1000 from Electrim Inc. This camera was chosen based on size and cost. Image data is sent from the camera back through a custom serial line

to a video capture card. The camera image has a resolution of 192 by 165 pixels with 256 shades of gray, which we truncate to 64 shades to reduce transfer time. Exposure time can be changed by software to range between 64ms to 200ms. Although we slowed the robot to minimize dynamic effects, mechanical settling times are long enough to cause image blur at the camera. To avoid this, we implemented a stability check by taking two images separated by 64ms and differencing them. Subsequent images are taken until the two successive images are sufficiently similar.

To avoid the complexity of another servo motor, we use a fixed focus camera and choose a focal point that compromises between the two fixed camera heights. The workspace is primarily illuminated by standard florescent fixtures. We tested a contrast enhancement routine to normalize the lighting of each image captured from the camera. This increased image quality in most cases but exaggerated intensity variations across the workspace.

5 System Architecture

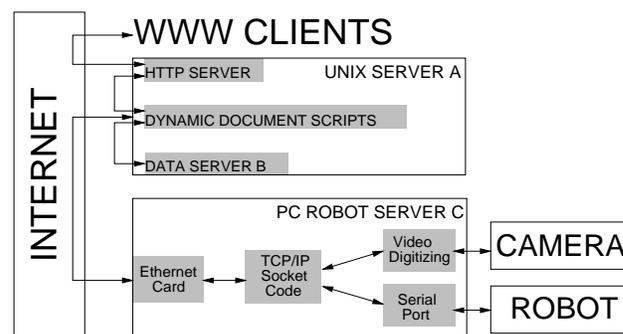


Figure 4: System Architecture

As shown in Figure 4, WWW clients from around the world enter our system through the Internet. The system includes three communicating subsystems. Server A responds to Universal Resource Locator (URL) requests for any file on the raiders/ directory. Server A runs the vanilla NCSA HTTP Demon v.1.3 on a Sun SPARCserver 1000, with SunOS Release 5.3. Server A caches the most recent status image and sends it whenever an observer request comes in.

When a user registers as an operator by entering a password, we use a database server to verify. This server, B, runs on the same machine as Server A. The database server is custom programmed for this project, but performs fairly standard database functions.



Figure 5: Sample camera images: Top row shows scene before burst of compressed air, bottom row after. Left column taken by camera in the up position, Right column by camera in the down position.

When an operator is verified, Server A either adds the operator to the queue or communicates with Server C which controls the robot. Server A decodes the ISMAP X and Y mouse coordinates, and sends them across campus to Server C via Ethernet.

On Server C, a custom program decodes the XY coordinates into a robot command and verifies that the command is legal, e.g., within the robot workspace. If it is, the command is then executed via a command sent to the robot over a 4800 baud serial line. Once the command is completed, server C uses a local frame buffer to capture the image.

Server C is a PC running MS-DOS (there wasn't a camera driver for Linux at that time). Server C generates a new schematic view of the robot in the resulting configuration, combines it with the camera image and appropriately highlighted control buttons to form a new status image. Server

C then compresses this image into GIF format and returns it to Server A, which updates the most recent status image and returns it to the client.

To maintain compatibility with the widest possible set of user platforms, we stayed within the standard HTTP protocol. For example, although X windows permit live video feed, we sacrificed this feature for the sake of compatibility. We hope that future versions of the protocol will allow the server to connect to and update clients to avoid manual re-loading of images.

The major difficulty in implementing Server C was scheduling responses to the network, the local mouse, and the image capture board. Although we seriously considered a multi-tasking environment such as Linux, the Electrim camera was only compatible with DOS and the company would not part with any source code. Thus we hand-crafted our memory management and used the screen itself as a memory buffer. This enabled us to speed a custom GIF encoder down to a few microseconds per status image.

5.1 Random Tokens

Each time Server A returns a new status image to an operator or observer, it adds a large random number to its embedded URL for the update button. This random token prevents the client from caching the status image (otherwise repeated requests to update the image would simply reload the local image and not request an updated image from Server A).

The random token also allows Server A to identify and track clients. When an operator logs in with a verified password, Server A tracks the operator by maintaining a database of recent clients so that URL requests can be customized depending on the user's status. For example the queue is only visible to the operators and those on deck.

Servers A and B are at opposite ends of the USC campus and are connected via Ethernet. Each machine has its own IP address and resides in the usc.edu domain. Communication is achieved using a socket connection between the two machines. The implementation on Server A was done using the standard BSD socket functions provided with the SunOS 4.1 operating system and Perl. On Server C we used a publicly available socket package called Waterloo TCP and Borland C.

6 Performance

An off-center air nozzle caused sand to accumulate in one side of the box, so the sand had to be periodically re-groomed. Other than a few random downtimes, the robot was online 24 hours a day for 6 months.

Network throughput averaged 20 Kbytes/sec, which was poor compared with 500 Kbytes/sec that can be achieved

between two Sun workstations in close proximity on the campus network. We blame this largely on the MS-DOS operating system which forced us to implement busy/wait cycles to obtain concurrence between the robotic/camera operations and the networking duties.

When server C detects an error, it automatically resets the robot controller, recalibrates, and returns the robot to its previous position. Also, server A automatically sends email if any of the key servers stop functioning. This occurs on average twice a month usually due to re-starts of the primary usc server. Server A also sends mail to the project team if server C stops responding, which occurs about once a month.

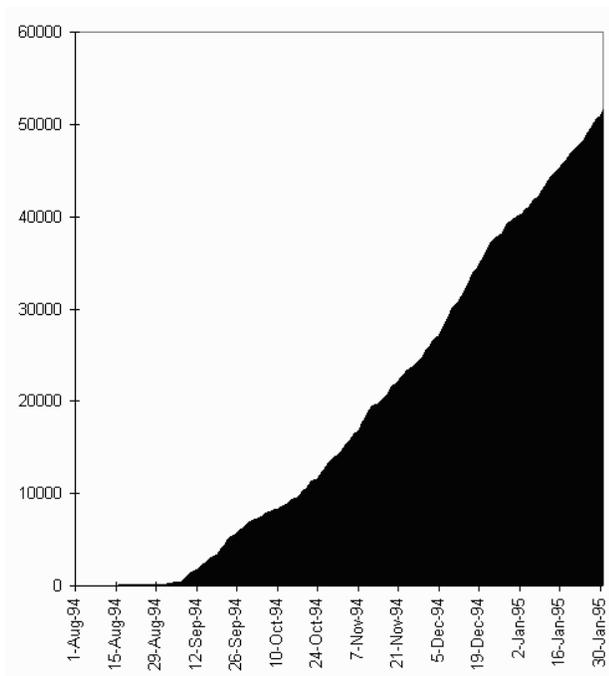


Figure 6: Cumulative number of unique (new) hosts accessing the project.

We monitor system usage with standard access_logs and with custom logs at Server B. In WWW parlance, a “hit” is a client request for a file from our system directory tree. In the period 1 Aug, 1994 through 1 Feb, 1995: 1,968,637 hits were made by 52,153 unique hosts (Figure 6). If we define “uses” as clusters of hits with less than half hour idle time, the system was used 87,700 times due to repeat visits. The daily average was 430 uses which generated approximately 1000 new images. In 1994, the Mercury Project accounted for roughly half of all requests to USC’s Web server during that period.

7 Operator Logs

Some samples from over 1000 pages of Operator’s logs:

From: Rex Kwok, rkwok@cs.su.oz.au
Date: Thu Nov 3 21:52:17 PST 1994

“FANTASTIC! It is amazing to operate a robot arm from Australia.”

From: Scott Hankin, hankin@osf.org
Date: Fri Sep 23 09:34:59 PDT 1994:

“...this site seems similar to the Internet. The search is analogous to trying to find something on the net, where you scan to locate areas of interest. Sometimes you’ll encounter a useful nugget of information like [the antique lantern]; other times you’ll discover information which seems valid but may be misleading, like the sample of “fool’s gold”. Some information is in different languages, like the scrap of paper with text in English and German which points to the multinational nature of the net.”

From: Dr. Steve M. Potter, spotter@gg.caltech.edu
Date: Thu Oct 27 23:30:09 PDT 1994

“What fun! Cool idea, folks. Best use of forms and clickable maps I have seen...I was wondering how I know this is not a clever laser-disk full of pictures you grabbed, with no robot, until i saw the time on the watch after blasting it. That was when my skepticism evaporated.”

And our favorite...

From: James Bryant, jcbryant@jax.jaxnet.com
Date: Sat Sep 10 08:54:11 PDT 1994

“I don’t believe I have seen a nicer application of science, or its money on the net.”

8 Discussion and Future Applications

The system design is an example of Minimalist Robotics [29]. The SCARA-type robot requires only 4 axes, is relatively inexpensive and robust, and it is easy to avoid singularities. The method of actuation we’ve used here is also about the minimum.

We viewed the Mercury Project as a feasibility study for a broad range of applications using the Internet to bring remote classrooms and scientific lab equipment to a much wider audience. Since the Internet Protocol offers no transmission guarantees, our interface design is not suitable for time-critical interactions such as remote assembly with force feedback. Our system also suffers by forcing operators to work sequentially and wait in a queue. In the Tel-e-garden, we used a faster Adept-1 robot so that many operators can be accomodated as the system is multi-tasked. We hope that widespread access to Internet Robots can permit the next generation of students and researchers to share experiences that can advance basic and applied science.

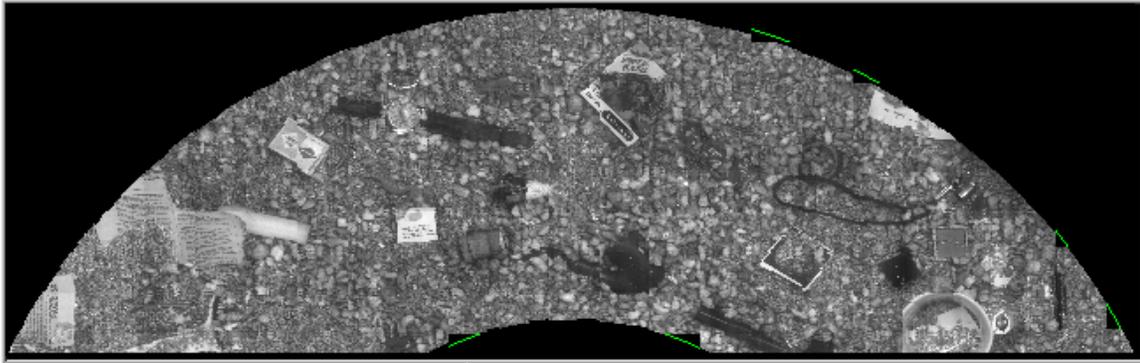


Figure 7: Composite image of workspace with artifacts such as miniature lantern, seed packet, etc..

Acknowledgments

We are grateful to the following for their support and helpful suggestions: Michael Mascha and Nick Rothenberg from USC's Dept of Anthropology, Howard Moraff of NSF, John Canny, Eric Paulos, Richard Wallace, George Bekey, Bob Farzin, Andy Fagg, Juergen Rossman of U. Dortmund, Peter Danzig, Rick Lacy, Mark Brown, Eric Mankin, Irene Firtig, Victoria Vesna, Peter Lunenfeld, Zane Vella, the Los Angeles Museum of Miniatures, the Laika Foundation, and everyone who participated by operating the robot.

References

- [1] Tim Berners-Lee, Robert Cailliau, Jean-Francois Groff, and Bernd Pollerman. World-wide web: The information universe. *Electronic Networking: Research, Applications and Policy*, 1(2), Westport CT, Spring 1992.
- [2] John Canny and Ken Goldberg. A risc approach to sensing and manipulation. *Journal of Robotic Systems*, 12(6):351–362, June 1995.
- [3] Raymond Goertz and R. Thompson. Electronically controlled manipulator. *Nucleonics*, 1954.
- [4] R. D. Ballard. A last long look at titanic. *National Geographic*, 170(6), December 1986.
- [5] A. K. Bejczy. Sensors, controls, and man-machine interface for advanced teleoperation. *Science*, 208(4450), 1980.
- [6] R. S. Mosher. Industrial manipulators. *Scientific American*, 211(4), 1964.
- [7] R. Tomovic. On man-machine control. *Automatica*, 5, 1969.
- [8] A. Bejczy, G. Bekey, R. Taylor, and S. Rovetta. A research methodology for tele-surgery with time delays. In *First International Symposium on Medical Robotics and Computer Assisted Surgery*, Sept 1994.
- [9] Matthew Gertz, David Stewart, and Pradeep Khosla. A human-machine interface for distributed virtual laboratories. *IEEE Robotics and Automation Magazine*, December 1994.
- [10] T. Sato, J. Ichikawa, M. Mitsuishi, and Y. Hatamura. A new micro-teleoperation system employing a hand-held force feedback pencil. In *International Conference on Robotics and Automation*. IEEE, May 1994.
- [11] Thomas B. Sheridan. *Telerobotics, Automation, and Human Supervisory Control*. MIT Press, 1992.
- [12] Search **yahoo.com** under 'Interesting Devices'.
- [13] Barney Dalton and Ken Taylor. A framework for internet robotics. In *International Conference On Intelligent Robots and Systems (IROS): Workshop on Web Robots*, Victoria, Canada, 1998.
- [14] <http://telerobot.mech.uwa.edu.au/>.
- [15] <http://found.cs.nyu.edu/cgi-bin/rsw/labcam1>.
- [16] <http://www.eia.brad.ac.uk/rti/>.
- [17] Eric Paulos and John Canny. Delivering real reality to the world wide web via telerobotics. In *IEEE International Conference on Robotics and Automation (ICRA)*, 1996.
- [18] E. Paulos and J. Canny. Designing personal teleembodiment. In *IEEE International Conference on Robotics and Automation (ICRA)*, Detroit, MI, 1999.
- [19] Yuichiro Akatsuka Steven B. Goldberg, George A. Bekey. Digimuse: An interactive telerobotic system for viewing of three-dimensional art objects. In

International Conference On Intelligent Robots and Systems (IROS): Workshop on Web Robots, Victoria, Canada, 1998.

and Automation. IEEE, May 1994. Also available via anonymous ftp from 128.125.51.19 under pub/Tech_Rept/Iris/1993/IRIS_93_315.ps.Z.

- [20] Reid Simmons. Xavier: An autonomous mobile robot on the web. In *International Conference On Intelligent Robots and Systems (IROS): Workshop on Web Robots*, Victoria, Canada, 1998.
- [21] Patrick Saucy and Francesco Mondada. Khep-on-the-web: One year of access to a mobile robot on the internet. In *International Conference On Intelligent Robots and Systems (IROS): Workshop on Web Robots*, Victoria, Canada, 1998.
- [22] Roland Siegward, Cedric Wannaz, Patrick Garcia, and Remy Blank. Guiding mobile robots through the web. In *International Conference On Intelligent Robots and Systems (IROS): Workshop on Web Robots*, Victoria, Canada, 1998.
- [23] Kam S. Tso Paul G. Backes and Gregory K. Tharp. Mars pathfinder mission internet-based operations using wits. In *IEEE International Conference on Robotics and Automation (ICRA)*, Leuven, Belgium, 1998.
- [24] Matthew R. Stein. Painting on the world wide web: The pumapaint project. In *International Conference On Intelligent Robots and Systems (IROS): Workshop on Web Robots*, Victoria, Canada, 1998.
- [25] Organized by Roland Siegwart. In *International Conference On Intelligent Robots and Systems (IROS): Workshop on Web Robots*, Victoria, Canada, 1998.
- [26] Predictive Simulation of Autonomous Robots for Tele-Operation Systems using the World Wide Web. Dirk schulz, wolfram burgard and armin b. cremers. In *International Conference On Intelligent Robots and Systems (IROS): Workshop on Web Robots*, Victoria, Canada, 1998.
- [27] Gunter Niemeyer and Jean-Jacques Slotine. Toward force-reflecting teleoperation over the internet. In *IEEE International Conference on Robotics and Automation (ICRA)*, Leuven, Belgium, 1998.
- [28] Kevin Brady and TJ Tarn. Internet-based remote teleoperation. In *IEEE International Conference on Robotics and Automation (ICRA)*, Leuven, Belgium, 1998.
- [29] John Canny and Ken Goldberg. "RISC" for industrial robotics: Recent results and open problems. In *International Conference on Robotics*