A Minimalist Telerobotic Installation on the Internet

Bobak R. Farzin, Ken Goldberg and Adam Jacobs
IEOR Dept., UC Berkeley

Abstract

As a feasibility study for simple, fast, reliable, relocatable telerobotic systems on the Internet, we describe an interactive online lightbox system. It is accessible to anyone with a graphical browser and provides a minimalist example of interactivity. The project uses motors and light switches and returns live images resulting from user specified combinations of devices:

http://www.dislocation.net/

1 Introduction

For the first time in history, telerobotics is now widely accessible to nonspecialists via the Net[1]. Web cameras are the most common devices, claiming to provide live remote images. Some websites use the mouse/imagemap protocol to permit remote viewers to direct the gaze via computer controlled pan/tilt mechanisms. Robotic devices provide a further level of interactivity by allowing participants to manipulate the remote environment which may include a model railroad, paintbrush, sculpture or garden.

Our goal was to create a simple, fast, reliable, relocatable telerobotic system on the Internet. This system must be accessible to anyone with a graphical browser and provide an interesting application of telerobotic simplicity. The project uses lighting illumination as its action and returns live images to the user to show the results of that lighting combination at that instant. This robust system needed to function unattended at all times, and cope with very low lighting conditions.

Our secondary goal was to provide a site that forces users to consider the WWW as an information medium. As an information medium, epistimological questions arise in direct application to the project. It was important to provide a site where the user questions what they are seeing, and are able to respond to that consideration. This paper focuses on the details of the implementation and a brief discussion of the impact of the project.

2 Related Work

A robot can be broadly defined as an electromechanical system where a mechanism is controlled by a computer. In telerobotics, the mechanism is remote. It sends back data and is generally controlled by a human at the other end.

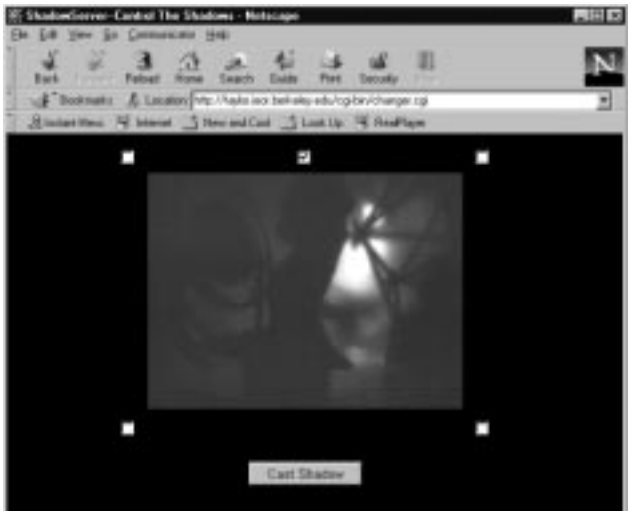

Figure 1: halogen light is activated by selecting the top center button

Goertz demonstrated one of the first “master-slave” teleoperators 50 years at the Argonne National Laboratory[2]. Remotely operated mechanisms have long been desired for use in inhospitable environments such as radiation sites, undersea [3] and space exploration [4]. At General Electric, Mosher [5] developed a complex two-arm teleoperator with video cameras. Prosthetic hands were also applied to teleoperation [6]. More recently, teleoperation is being considered for medical diagnosis [7], manufacturing [8] and micromanipulation [9]. See Sheridan [10] 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 “tele-presence”. Our objective was to provide widespread access by using only the “point-and-click” interface available under the standard HTML language.

3 Related Work

There are currently dozens of machines interfaced to the Internet; a good sample can be found at the Yahoo site [11].

To our knowledge, the Mercury Project was the first Web robot. It came online in August 1993 and allowed users to "dig" for buried objects in a sand-filled workspace using a four-axis IBM robot equipped with a black and white camera and compressed air nozzle [12]. A team at the University of Western Australia independently developed a WWW interface for a six-axis robot that came online in September 1994 [13]. The Australian Web robot allows users to input spatial coordinates to specify relative arm movements, directing the robot to pick up and stack blocks on a table. Several increasingly elegant user interfaces have been developed by the Australian team.

In October 1994, Wallace demonstrated a robotic camera [14] and Cox unveiled a system that allows WWW users to remotely schedule photos from a robotic telescope [15]. In 1995, Simmons [16] developed a system that allows users to control a mobile robot moving in 6 story building and to snap pictures after it autonomously navigates to a desired location. Like Cox's robotic telescope, the Xavier robot executes commanded motions in batches several times a day.

In 1995, Paulos and Canny[17] implemented an elegant user interface allowing users observe museum artifacts placed within the workspace of a 6 axis robot. In a similar vein, George Bekey and Steve Goldberg used a six-axis robot, rotating platform, and stereo cameras to allow remote scholars to closely inspect a priceless sculpture on display in a museum at USC [18]

The WWW provides an excellent platform for public access to expensive resources. Backes[19] described the Web Interface for Telescience (WITS) to allow the general public to plan and simulate missions using a NASA rover. This interface brings the expensive exploration of space to non-specialists using a simple visual interface.

Client-side applets using Java can be used to control web robots. citedalton98,hirakawa98. In 1998, Stein [20] demonstrated a Java interface that allows a user to select paint strokes before sending a set of stroke commands to a web robot with a paintbrush. Schulz [21], Slotine[**?**] and Tarn[**?**] consider how control theory used for teleoperation over dedicated lines can be adapted to the Internet, where delays vary unpredictably.

Paulos and Canny propose that users can communicate effectively using *tele-embodiment* over the Internet, using simple, inexpensive, PRoPs (Personal Roving Presences)[22]. These blimps and roving machines navigate user inhabited space, but have web interfaces that allow control from any remote location and provide modes of communication such as focus of attention and gesture.

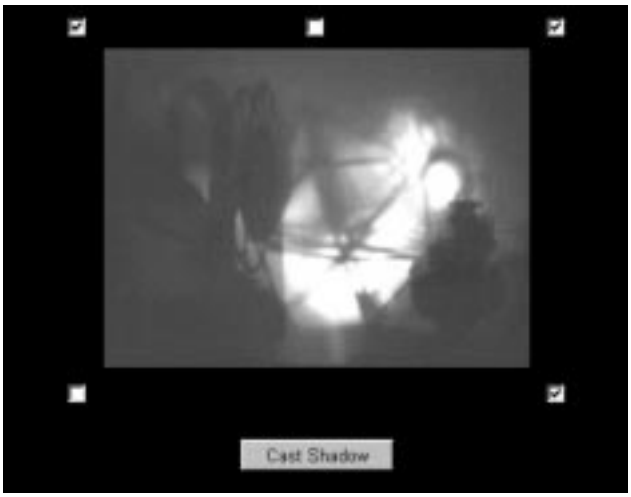

Figure 2: A Processed Shadow Request

Saucy[23] and Siegwart[24] developed small, highly reliable mobile web robots that users can direct through a maze. The latter project includes a VRML version so that viewers can switch back and forth between simulacral and the distal experiences.

A number of net art installations use telerobotics (cite (MIT Press book)) such as Fujihata's Light on the Net (http://light.softopia.pref.gifu.jp/), which allows web users to individually turn on and off lights in a 7×7 grid in the hall of the Gifu Softopia Center in Japan. Fujihata's project is similar in spirit to the interface described in this paper.

4 User Interface

Upon entry in our system, web clients are presented with a page featuring an image of Marcel Duchamp's 1916 work "Ball of twine (with hidden noise)". By clicking on the image, the client proceeds to an introduction page explaining the physical apparatus and giving a brief explanation of its operation. On this page the number of shadows "served" is reported and the client is presented with a "proceed" button to continue.

The proceed button invokes the server application and returns a web page, as seen in Figure 1, with one randomly chosen light in the on position. Lighting control is indicated with the check boxes, where a checked box indicates an illuminated light (top center.) The client is then able to activate or deactivate any of the check boxes and then by pressing the "cast shadow" button they submit the request to the server.

The server processes the request and returns the appropriate shadow image to the client with the unique image from the apparatus and the check boxes marked according to the request. The client can then select any other combination of

lights and repeat the process by clicking on the cast shadow button again. There is no limit to the number of requests a user can make. Although there are six buttons giving a possible 64 lighting combinations, because of random motion in the box, there is no limit to the number of shadows that can be formed.

On each request page, a link to a document that allows the user to e-mail feedback, read the "most promising inferences," and review the most current statistics of the site.

5 System Architecture

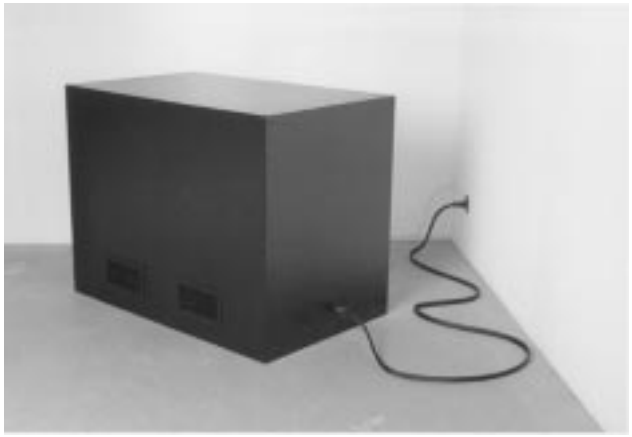

Figure 3: The Apparatus Exterior

5.1 Hardware

The lightbox (see Figure 3) sits 58 inches deep, 38 inches wide and 48 inches high. The Pentium PC computer systems run Red Hat Linux 5.0. Linux offers several advantages over Windows NT as a development environment for telerobotics. The Linux Apache 1.3.0 web sever is more stable than comparable NT systems, Linux allows greater access to the system memory and devices and is easily adaptable to the special needs that telerobotics presents (Including custom TCP/IP software and controller hardware.)

5.1.1 Physical Apparatus

The lightbox is constructed on a wood base with surrounding metal panels. The black metal exterior stands as a basic box with only one cable connection to the outside world. We have provided a very powerful cooling and ventilation system to keep the inside cool even over long periods of operation and many rapid requests.

Figure 4 illustrates the inside of the apparatus box. At the near end is the control box containing the cooling system that moves air over the power transformers operating

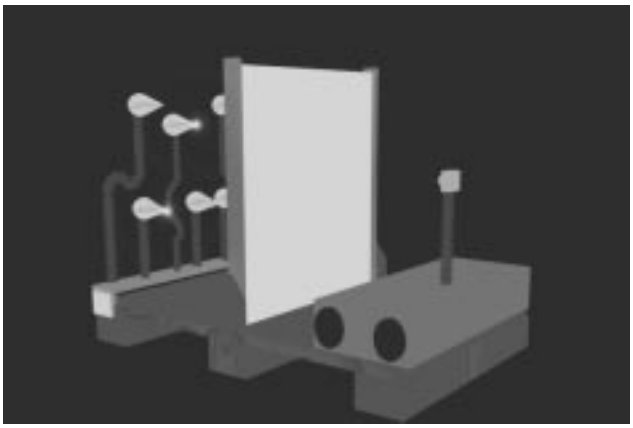

Figure 4: Inside the Lightbox

the lights and one of the computer systems. The air is directed out the lightbox through four vents that surround the exterior.

In the middle of the box is a scrim made of Acrylite FF, a thin semi-transparent acrylic that servers as a screen for the projected shadows. The original screen was paper, but to avert fire hazard we switched to a the semi-transparent acrylic panel that is inexpensive, durable and of the proper opacity to allow a crisp shadow.

At the far end are the lights that illuminate the objects which are placed between the lights and the scrim. The lights are switched from the control box and with custom circuitry that toggles a set of six solid state relays. The custom circuit uses the MAX233AC to interpret the serial line input and then passes the data onto a PIC16c73 EEPROM that controls the relays.

5.1.2 Lighting

We use commercially available 20 watt halogen desk lamps. We chose these lamps for their power, small size, and long lamp life. The intention was to use the stock lamps right out of the box, allowing for easy replacement and adjustment. However, the bare lamps are too bright, and more than one or too lights floods the entire screen to a solid white. Therefore the lamp light is controlled with aluminum cones that focus the beam down to a point light source. The lamps also do not provide the directional control that is necessary to produce sharp shadows, so Loc-Line was used to provide flexible control of the lamp heads.

5.1.3 Video Capture

In the first attempts to construct the Shadow Server we used the popular QuickCam because we were able to gain drivers for it over the web [1] and because of its low cost. A

[1] http://www.cs.virginia.edu/ patrick/cqcam/

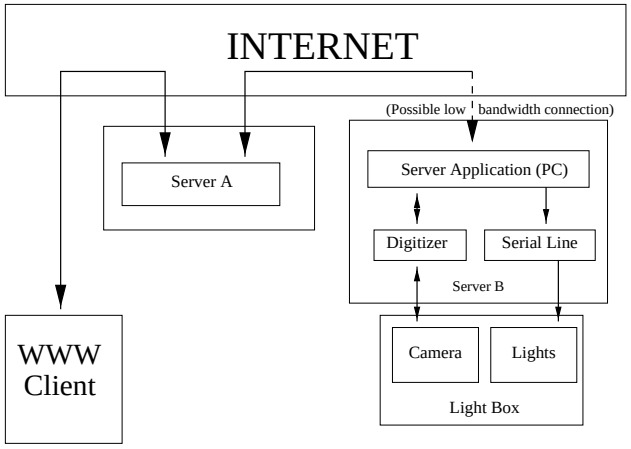

Figure 5: System Architecture

simple command line syntax made capturing the image simple to incorporate into the C code. However, the QuickCam often took up to twenty seconds to capture a single image. The driver that we were provided took excellent fast images in normal lighting conditions, but needed to take up to ten images with processor intensive comparisons between them to obtain the brightness and contrast settings in low light. This twenty second wait became a significant hindrance when there were hundreds of people in the queue.

Most users of the Internet are not willing to wait more than 30 seconds per request (Ref. Khep on the web???). To reduce the time involved in capturing we used the WinTV video capture board that contains the BT848 chipset, a common standard that has Linux code available[2]. After patching the code to use new memory addresses and function with the hardware on our machines, we were able to get excellent results with capture times of less then .25 seconds.

5.1.4 Camera

This application required a camera to work in low light conditions. The small CCD "spy" type cameras offer low light capability and have the added advantage of automatic exposure. The PC18-XS [3] camera we selected takes exposures at 0.5 lux and is easily adjusted to the the brightness, contrast and saturation that each lighting combination requires with software. The camera also offers higher resolution at 380 TV lines of resolution with 512 X 492 pixels. The camera weighs only 28g and is a 1.5 inch cube.

5.2 Software

A graphical representation of the system architecture is shown in Figure 5. Figure 6 illustrates the logic of a server request.

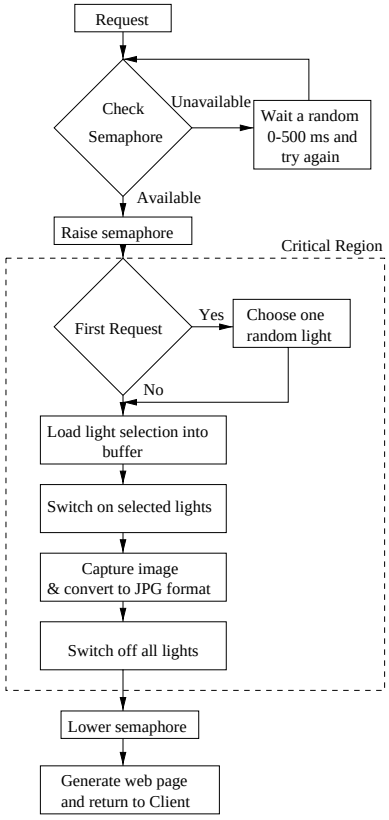

Figure 6: Query Logic Diagram of Server Systems

There are two server machines, call them Server A and Server B. The WWW client query begins with a request to Server A for the shadow control page via the URL. That request passes over the web and arrives at Server A in the ALPHA (Assembly Line Parts Handling Algorithms) lab which is running a custom CGI program that processes requests and negotiates with Server B.

Often there are many WWW clients making a request at once. Because there is only one lightbox, and only one image can be taken at a time, the pages must be processed serially from the queue on Server A.

Server A then determines if the semaphore is currently raised (see 4.3.3), indicating a currently active request, and a queue to be completed. If the semaphore is raised, the process sleeps for a random 0-500ms and then again attempts to gain the semaphore. Once the semaphore is lowered, Server A enters the critical region to control the hardware.

Then, Server A determines if this is the first image being requested by the particular user via an embedded value in the URL. If so, then one light is randomly chosen to be

[2] http://www.thp.uni-koeln.de/ rjkm/linux/bttv.html
[3] http://www.scx.com/page3.html

switched on, and the light status information is placed in a temporary buffer. Server A then queries Sever B to process the request in the lightbox.

Server B may have a low bandwidth connection, as slow as a dial-up phone connection. When the query arrives from Server A, the lights are switched, illuminating objects and projecting a shadow onto the scrim. The projected image is then captured on a small CCD camera, compressed in JPEG format and sent back to the waiting Server A. Server A then takes the image and incorporates it into the HTML document, associates the web page with the appropriate random token for the image (see 4.3.4), lowers the semaphore, and returns the page to the WWW client that had made the request. Once the page is sent off, the next request in the queue is processed and passed onto Server B.

5.3 Subsystems

The software on Server A and Server B is further explained below with the details of their operation including the use of a random token and semaphore to guarantee that the user receives the image that he or she requested.

The entire operation of switching, capture, and page generation takes an average 1.593 seconds. At that point the speed of the users Internet connection determines the download time of the 1.6k HTML document and 10K image. Even under a fast connection, a request takes approximately 5-10 seconds because of network traffic. Thus, the switching, capture and page generation operation takes only a fraction of the users request time. Thus, even with a queue of users, it is transparent to any particular user that there is time spent in the queue waiting for Server B to process the request in the lightbox.

5.3.1 Server A

Server A is a 200MHZ Pentium II PC that resides at a static IP address in the ALPHA Lab. This machine runs the Apache web server under Red Hat Linux 5.0 and is the WWW server to the web client and the exclusive *client* of Server B. Server A does several tasks in addition to serving web pages. These include keeping track of access statistics, e-mailing the maintainers in the case of an operation failure, generation of the random token and communication with Server B and lightbox.

Although Application on Server A can be called in parallel, only one user can be processed at a time through the critical region. This is accomplished by the use of a semaphore that creates a queue where only one user has access to Server B. The Application then switches the lights and captures the image whose name is determined by the random token.

5.3.2 CGI

Communication between the server and client is done with the Common Gateway Interface (CGI) protocol under HTTP/1.0. Through the use of HTML forms, we can easily pass the request from the client to Server A. Frequently PERL is used for CGI programming because of its simplicity and excellent database interface. However, PERL is slow because it is an interpreted, not compiled, language. For this project, speed is critical, so a compiled language such as C is preferred. C also offers the advantage of fast computation which is critical to the proper operation of the semaphore, light switching and camera capture.

5.3.3 Semaphore

A Semaphore is classically used for queuing of processes in uniprocessor and multiprocessor operating systems. Much like a uniprocessor system, we needed to serialize the use of a single resource by many users. To accomplish this we used a semaphore that alternately creates and destroys a file named “lock” in a particular directory.

The semaphore allows the execution of a critical region of code that requires exclusive use of the lightbox. In our first efforts we attempted to use the UNIX procmail v3.10 lockfile() command. However this command forks a child process, establishes a new shell and then writes out the “lock” file. This shell is kept alive until the critical region is completed and the lock is released. Although this system seemed to work well under a light load, when we got upwards of 1000 hits a day, the server would crash because the spawning and killing of shells was OS intensive and takes up to 1 second, where the file manipulation within C takes 2-5 milliseconds.

The method that we implemented use simply checks for a lock file to become available, and does not require and spawning of a child process. When a request arrives, Application checks for the presence of the “lock” file, that is a raised semaphore. If the file is present, it waits for a random 0 to 500ms, and then trys again to access the “lock” file. This random waiting time provides a staggering of two simultaneous requests. That is, if two applications request the lock file at the same time, and are both put to spin for, say, 200ms then they will wake up and again request the lock throwing them into a deadlock, where neither will ever get a lock on the system. The randomness of the sleep time averts this difficulty.

Because this is all coded into the application only a single application is open at any one time, there are no child processes. Also, the maximum wait time between checking for a free lock is 500ms. In the original solution, the wait time was The lockfile() command requires at least 1 second before it trys to gain the lock again. Thus it may be several seconds before a lock is established. This difference alone

cuts the wait time and number of processes on the server in half, more than doubling the overall speed of the system.

5.3.4 Random Token

When a client requests an image a 10 digit random number is generated and creates a random token identifier for the request. The random token allows us to synchronize multiple near simultaneous requests without mixing the images among the requesting clients. The random token is embedded in the HTML document, and used as the file name of the captured image. When the web client receives the document, it requests the image and receives the unique image associated with that user and that request. The random token also prevents the client from caching the previous image and forces the browser to load the new shadow image.

This implementation circumvented the use of a database for the project because each request has no relation to the user or their previous requests, if any. Each request is memoryless, that is we do not need to associate the random tokens with with clients for more than that one request, so after the request is serviced, the random token is discarded. Each night Server A looks at the date on the image files, and the oldest ones are deleted until they fall below a quota value of 5MB. This prevents the server's hard drive from filling with old images, and crashing the system.

5.3.5 Server B

Server B is a 100MHZ Pentium PC located inside or near the sealed light lightbox. It has a modem connection with which to communicate with Server A via a custom socket connection using TCP/IP. When Server B receives a request from Server A, it switches the appropriate lights projecting a shadow image on the screen. The CCD camera then snaps an image and the PPM output is compressed to JPEG format and then returned to Server A.

The server application is specially designed to handle loss of communication with Server A. Server A and Server B handshake at 30 second intervals to assure that the connection is still open between them. Server A maintains a static IP address in the ALPHA Lab, and is assumed to be available at all times. Because Server B is on a modem connection, it is likely that it may temporarily loose its network connection, and need to reconnect with a different IP address. Thus, if the Server B fails for three consecutive connections, it will try to ping a known host off of the Berkeley campus to check its modem connection. If it receives no response, it will assume that the network connection has been lost, and will sleep for five minutes allowing the modem connection to return, before trying again to make a connection.

However, if the connection to an outside server *is* successful, then Server B will assume that the Server A is unreachable, and send e-mail to the appropriate parties involved, informing them of the trouble and then shutting itself down. The system can then be re-activated remotely after the problem is located and fixed.

¿From the other end, if Server A does not hear from Server B on its regular 30 second handshake, then it will assume that Server B's connection has gone down, and will wait for two minutes before trying to re-connect to Server B, hoping that Server B has established a new modem connection. If the attempt to reconnect is unsuccessful three consecutive times, then Server A will assume that something has gone wrong with Server B and it's network connection. The Server A will send out e-mail, and put up a page stating that the site is experiencing technical difficulty. This page will reroute the user to stored sample shadows. Server A will then wait and will try to establish a connection every 20 minutes, hoping that the system can regain a connection. If the connection is re-established, then the program will return the system to its normal state, and will send e-mail to that effect.

6 Performance

The system went online 21 Apr 1997 and from that date to 3 Aug 1998, there have been 350 thousand successful requests from 30 thousand unique hosts. The server averaged 180 requests per day, about 8 per hour, but has received two major spikes in usage. The first came the week of 7 Sep 1997 on the announcement of the Shadow Server project on WiReD news online, Yahoo! as a pick of the week, and listed on Cool site of the day for 10 Sep, 1997. This generated a total of 27 thousand hits on the site. The second surge of users came with an announcement of the project in the New York Times online, generating 13 thousand hits on the server the week of 22 Mar 1998. The average user requested 3 shadow images per visit, and the majority of the traffic was refered from news stories and search engine listings.

The system then went off line for serveral days while it was moved to an exhibition in San Francisco. At that time, the project went back on line and from 3 Aug 1998 to 26 Jan 1999 it has recieved 45 thousand hits, from over 8 thousand unique hosts. This exhibition and printed news stories were the centeral points of distribuing the new URL for the project. Therefore it is not surprising that few references from search engines appeared in the referer logs. Instead, the majority of references came from inidividuals directly typing in the URL and recieving the page. From the traffic rates, it is apparent that this promotion is significantly less effective then on-line news stories and search engine listings.

Over the course of the project we received over 70 e-mail responses. Three of them are listed below and other

responses or note can be found at:

http://taylor.ieor.berkeley.edu/shadowserver/topten.html

7 Select Inferences

Some samples from the over 70 responses we received as of 3 August, 1998:

From: Al Liguori aml1vil@feist.com
Date: Wed, 10 Sep 1997 23:17:55

"From one viewpoint it looked like a collection of media dispensing tools, a projector, headsets, and mics. From another, as I added more and more lighting sources I started to wonder if a comment was being made about the media, that is as more and more "light" is shed on a subject less and less truth is revealed, things, as in your demo, get blurred and fuzzy. The truth is hidden in the alcove and only those with enough perseverance, wit, or sheer pig-headeness find it. Then I went outside and had a smoke. Thanks."

From: Scott Peace-Miller spm@gsconnex.com
Date: Sun, 8 Feb 1998 23:51:07

" I thought that the Shadowserver was more of a statement on perception in general. We only have one way to perceive the objects in the box – through the internet. We can fiddle with the variables, but we are always limited to a few different pictures. There is no way to get "beyond" these pictures. Analogous to this is our interaction within our world. "

Perhaps our most interesting corespondence was with Mike Bujnowski:

¿From hauntedchicago@excite.com
Date: Wed, 30 Dec 1998 08:16:33 PST
Subject: Dislocation of Intimacy

"Ken,
You, sir, and your dislocation of intimacy page are frauds. Is your name Ken, or is that someone you made up? What sort of a random timer do you have the fans in your empty black box hooked up to? And where did you take the pictures that are on your website that you are passing of as an interactive session with your stupid box? Fake."

From: Ken Goldberg ¡goldberg@ieor.berkeley.edu¿
To: hauntedchicago@excite.com
Subject: Re: Dislocation of Intimacy

"Dear Sir,
Thanks for your email. In fact you have identified the precise issue we are interested in, what we call: telepistemology. If you want to know more, please let me know and I'll send you some details on a forthcoming book on this subject. happy new year.
Ken"

From: hauntedchicago@excite.com
To: Ken Goldberg ¡goldberg@ieor.berkeley.edu¿
Date :Mon, 4 Jan 1999
Subject: Apology

"Ken,
I was pretty harsh, wasn't I. I apologize. This actually sounds like a great area of research. I read a little bit about telepistemology and I am actually fascinated by the implications. I guess you could say that myannoyance with the discovery of the true nature of the "light box", which I thought I would be able to control, was itself an interesting reaction. I'd like to know more about your studies, and about the book. Thank you.
Mike Bujnowski"

From: Ken Goldberg ¡goldberg@ieor.berkeley.edu¿
To: hauntedchicago@excite.com
Date :Mon, 4 Jan 1999
Subject: Re: Apology

"Hi Mike,
Actually I appreciated your indignation! And you are quite right that your reaction was itself interesting.
Regards,
Ken"

Acknowledgments

We are grateful to the following people for their assistance in the construction and administration of the Shadow Server project: Eric Paulos, Josh Postamentier, Eric Lee, Steven Bui, Soyen Shih, Yaron Oren.

References

[1] David Kushner. Web robots offer hnads on experience from afar. *The New York Times*, November 1998.

[2] Raymond Goertz and R. Thompson. Electronically controlled manipulator. *Nucleonics*, 1954.

[3] R. D. Ballard. A last long look at titanic. *National Geographic*, 170(6), December 1986.

[4] A. K. Bejczy. Sensors, controls, and man-machine interface for advanced teleoperation. *Science*, 208(4450), 1980.

[5] R. S. Mosher. Industrial manipulators. *Scientific American*, 211(4), 1964.

[6] R. Tomovic. On man-machine control. *Automatica*, 5, 1969.

[7] 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.

[8] Matthew Gertz, David Stewart, and Pradeep Khosla. A human-machine interface for distributed virtual laboratories. *IEEE Robotics and Automation Magazine*, December 1994.

[9] 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.

[10] Thomas B. Sheridan. *Telerobotics, Automation, and Human Supervisory Control*. MIT Press, 1992.

[11] **http://dir.yahoo.com/Computers_and_Internet/ Internet/Interesting_Devices_Connected_to_the_Net/**.

[12] K. Goldberg, M. Mascha, S. Gentner, N. Rothenberg, C. Sutter, and Jeff Wiegley. Beyond the web: Manipulating the physical world via the www. *Computer Networks and ISDN Systems Journal*, 28(1), December 1995.

[13] **http://telerobot.mech.uwa.edu.au/**.

[14] **http://found.cs.nyu.edu/cgi-bin/rsw/labcam1**.

[15] **http://www.eia.brad.ac.uk/rti/**.

[16] Reid Simmons. Xavier: An autonomous moboile robot on the web. In *International Workshop On Intelligent Robots and Systems (IROS)*, Victoria, Canada, 1998.

[17] E. Paulos and J. Canny. Ubiquitous tele-embodiment: Applications and implications. *International Journal of Human-Computer Studies/Knowledge Acquisition*, 1997.

[18] Yuichiro Akatsuka Steven B. Goldberg, George A. Bekey. Digimuse: An interactive telerobotic system for viewing of three-dimensional art objects. In *International Workshop On Intelligent Robots and Systems (IROS)*, Victoria, Canada, 1998.

[19] Kam S. Tso Paul G. Backes and Gregory K. Tharp. Mars pathfindes mission internet-based operations using wits. In *IEEE International Conference on Robotics and Automation (ICRA)*, Leuven, Belgium, 1998.

[20] Matthew R. Stein. Painting on the world wide web: The pumapaint project. In *International Workshop On Intelligent Robots and Systems (IROS)*, Victoria, Canada, 1998.

[21] Predictive Simulation of Autonomous Robots for Tele-Operation Systems using the World Wide Web. Dirk schulz, wolfram burgard and armin b. cremers. In *International Workshop On Intelligent Robots and Systems (IROS)*, Victoria, Canada, 1998.

[22] 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.

[23] Kevin Brady and TJ Tarn. Internet-based remote teleoperation. In *IEEE International Conference on Robotics and Automation (ICRA)*, Leuven, Belgium, 1998.

[24] E. Paulos and J. Canny. Designing personal tele-embodiment. In *IEEE International Conference on Robotics and Automation (ICRA)*, Detroit, MI, 1999.

[25] Patrick Saucy and Francesco Mondada. Khep-on-the-web: One year of access to a mobile tobot on the internet. In *International Workshop On Intelligent Robots and Systems (IROS)*, Victoria, Canada, 1998.

[26] Patrick Garcia Remy Blank Roland Siegward, Cedric Wannaz. Guiding mobile robots through the web. In *International Workshop On Intelligent Robots and Systems (IROS)*, Victoria, Canada, 1998.