Collaborative Teleoperation via the Internet *

Ken Goldberg, Billy Chen, Rory Solomon, Steve Bui,
Bobak Farzin, Jacob Heitler, Derek Poon, Gordon Smith
Alpha Lab, IEOR Department, UC Berkeley

We describe a system that allows a distributed group of users to simultaneously teleoperate an industrial robot arm via the Internet. A java applet at each client streams mouse motion vectors from up to 30 users; a server aggregates these inputs to produce a single control stream for the robot. Users receive visual feedback from a digital camera mounted above the robot arm. To our knowledge, this is the first collaboratively controlled robot on the Internet. To test it please visit:

http://ouija.berkeley.edu/

1 Introduction

In most teleoperation systems, a single user input controls a single robot. Several researchers have considered the problem of one user controlling multiple robots; for a review of research in cooperative teleoperation see [1]. Here we consider the inverse problem: combining multiple user inputs to control a single industrial robot arm. We proposed the term "collaborative teleoperation" to describe such a many-one control architecture.

Consider the following scenario in space or undersea: a group of users are working together to control a telerobot. Each user monitors a different sensor and submits control inputs appropriate to that sensor information. All of these inputs must be combined to produce a single control input for the telerobot. If the inputs can be put into vector form, one solution to this "control fusion" problem is to average all inputs. Since each user has access to a different noisy sensor, The Central Limit Theorem (Appendix A) suggests that the average may yield a more effective control signal than that from any individual source.[1]

In this paper, we describe a client-server system on the Internet that facilitates many simultaneous users cooperating to share a single robot resource. The closest precedent we have found is Cinematrix, a commercial software system that allows a roomful of participants to collaboratively control events projected on a screen [2]. Collaborative control is achieved using two-sided color-coded paddles and a computer vision system that aggregates the color field resulting from all paddle positions. The developers report that groups of untrained users are able to coordinate aggregate motion, for example to move a cursor through a complex maze on the screen.

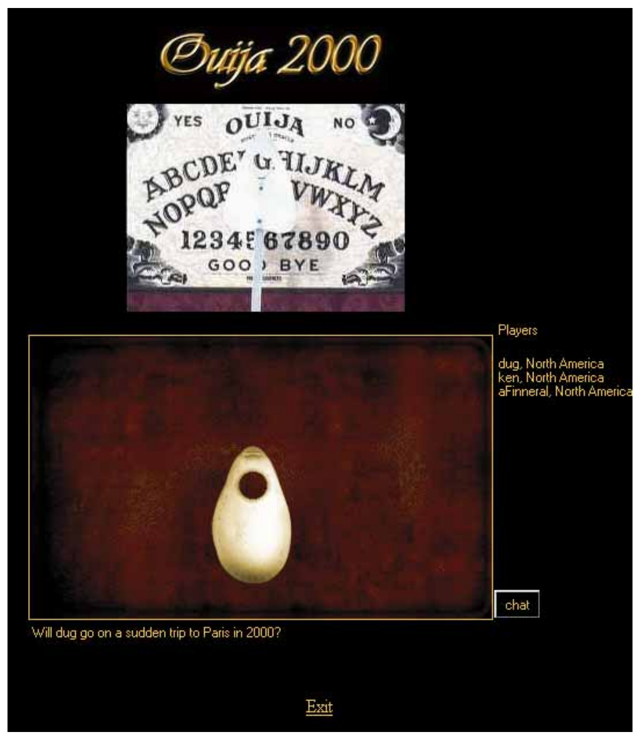

Figure 1: *User interface on the Internet. The top window displays live video of the robot and Ouija board in our laboratory. The bottom window displays user force inputs corresponding to mouse movements.*

Since 1994, our lab has implemented five systems that allow users to interact remotely via the Internet to control a physical robot. In the 1994 Mercury Project [3], users queued up for 5 minute individual control sessions. In the 1995 Telegarden project [4], user motion requests were interleaved so that each user seemed to have simultaneous control of the robot.

These two control models are analogous to the batch and

*IEEE International Conference on Robotics and Automation, April 2000, San Francisco, CA. For more information please contact goldberg@ieor.berkeley.edu.

[1] Of course the CLT assumes independence and zero-mean noise, which may not be satisfied in practice.

multi-tasking approaches to mainframe computing. They suffer from the standard latency and throughput problems associated with time sharing. In this paper we experiment with a control model where motion commands from multiple simultaneous users are *aggregated* into a single stream of motion commands.

2 Related Work

Goertz demonstrated one of the first "master-slave" teleoperators 50 years at the Argonne National Laboratory[5]. Remotely operated mechanisms have long been desired for use in inhospitable environments such as radiation sites, undersea [6] and space exploration [7]. See Sheridan [8] for an excellent review of the extensive literature on teleoperation and telerobotics.

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. The Mercury Project's Internet robot [3] went online in August 1994. In Sept 1994, Ken Taylor, at the University of Western Australia, demonstrated a remotely controlled six-axis telerobot with a fixed observing camera [9]. 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 [10]. Also in October 1994, Wallace demonstrated a robotic camera [11] and Cox put up a system that allows WWW users to remotely schedule photos from a robotic telescope [12]. Paulos and Canny have implemented several Internet robots with elegant user interfaces [13, 14]. Bekey and Steve Goldberg used a six-axis robot, rotating platform, and stereo cameras to allow remote scholars to closely inspect a rare sculpture [15]. Since then there have been a number of Internet-controllable mobile robots and other devices; see [16, 17] for surveys.

Online games like Quake [18] have quite complex collaboratively controlled environments over the Internet. Generally each user controls an independent element or avatar so that the system as a whole is collaboratively controlled.

3 Application

As an application of collaborative control, we selected the Ouija [19] board game, familiar to many from their youth. Several users play the game together by placing their hands together on a sliding plastic disk known as a "planchette". The planchette is free to slide over a board marked with letters and messages such as Yes and No. The group of users poses a question. As each user concentrates the planchette slides across the board to indicate an answer. Although we do not address the claim the planchette is influenced by supernatural powers, it is in many cases influenced by the conscious and unconscious movements of all participants. In this way the planchette aggregates information from a group of noisy receptors.

4 User Interface

Figure 1 illustrates the interface at the client's Internet browser. One goal in designing the graphical user interface (GUI) is to remain faithful to the user's experience playing the original game. A list of "players" (active clients) is updated and displayed to the right. At the bottom, the current "question" is displayed. The question is randomly selected from a file and is parameterized to include the name of one player. Past experience suggests that if we allow users to type in questions, the question display would quickly degenerate into graffiti. New clients register by providing an active email address, a userid, and password. They can then re-enter the system at any time using this password. After a few screens of instruction, the user arrives at the screen displayed in Figure 1.

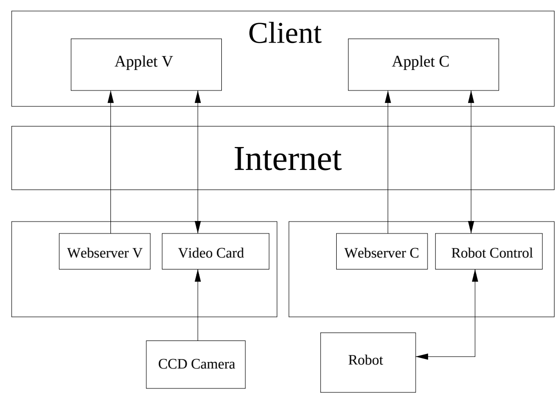

Figure 2: *System architecture: each client runs two applets, Applet V for video feedback and Applet C for control*.

Figure 2 describes the system architecture. Each client gets two applets; each communicates with a unique server in our lab. Applet and Server V provide live streaming Video feedback of the robot and planchette motion. Applet and Server C coordinate control of the robot.

The user's mouse serves as a local planchette. The user is asked to place both hands lightly on the mouse. Subtle mouse motions are monitored by Applet C, resulting in small motion of the displayed planchette icon in Applet C's window. These motions are continuously sampled by the client's applet to produce a motion vector that is sent back to Server C at periodic intervals, currently once every 5 seconds.

Server C collects motion vectors from all active clients, averages, and then sends a global motion command to the robot. The resulting planchette motion is observed by all active clients through Applet V.

5 Hardware

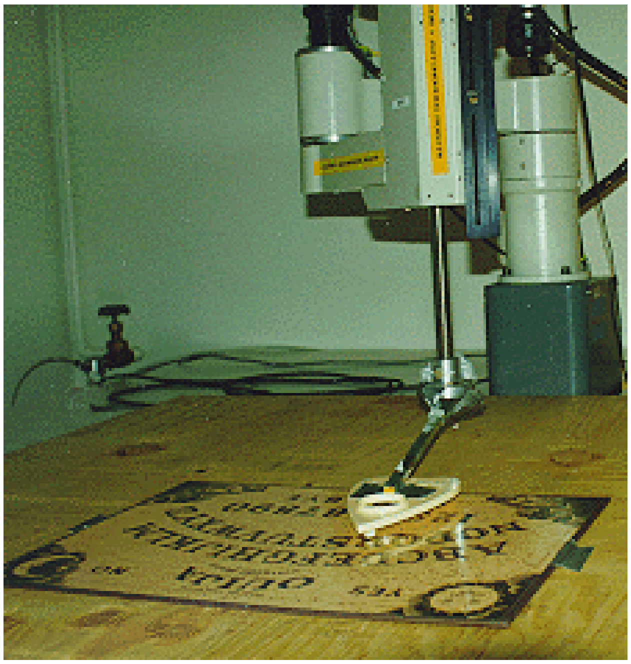

Figure 3: *Robot with Planchette*.

System hardware includes of two PCs, a CCD camera, and an Adept 604-S 4 axis robot arm.

Webserver V is a Pentium II 266 running Windows NT 4.0 SP5. Server V runs the Apache 1.3.3 server and a commercial software package called InetCam which handles live video. Server V also holds a FlyVideo '98 windows-compatible video card that is connected to a color CCD camera.

The video card is capable of either full motion capture, at 30 frames per second, or single still image captures. For our purpose, we used the full motion capture capabilities of the card. Resolution of video captures can be set up to a maximum of 640x480 with 12, 24, or 32-bit color resolutions. It supports connections of the following types 75 ohm IEC coaxial input (cable TV), composite video input (RCA), S-Video input (SVHS), audio input, and line audio output. The supplied driver allows us to set resolution rates and color resolution as well as configure optimal settings for hue, contrast, color, and brightness.

Webserver C is an Intel Pentium II, 266 MHz, running Red Hat Linux version 5.2. This machine runs an Apache Web server version 1.3.3. Webserver C handles HTML client requests. The Robot Server also runs on this machine. The Robot Server is attached through an RS-232 serial connection to a controller for an Adept 604-S robotic arm with four degress of freedom. The Adept controller runs the V+ operating system and programming language.

6 Software

After a user registers and enters the system, the client downloads one 25 KB java archive that includes all classes for Applets V and C. Initially, we kept each Applet separate, but discovered that for slow (14.4Kbps) connections, Applet V would load first and then begin streaming video, which would consume the bandwidth and prevent loading of Applet C.

6.1 Applet V: Streaming Video

In our initial design, we implemented two server-side scripts: one that took a snapshot from the capture card once-per-second and converted it to a black-and-white GIF, and another CGI script which then pushed the image to the client's web browser once-per-second. This had two drawbacks. First, we wanted to use inter-frame compression to reduce bandwidth usage since each image was roughly 70 kB. Second, Microsoft Internet Explorer, as of version 5, does not implement support for the server-side push function.

We chose a commercially available streaming video package, Inetcam version 2.1.0 [20]. Currently, Inetcam runs only under Windows 95/98/NT. We configured the InetCam server with frame resolution = 240 x 180, compression rate = 65 % , maximum frame rate = 5 per second, and max number of simultaneous users = 30.

6.2 Applet C: Planchette Control

Applet C displays a small window with a representation of the planchette. Applet C also displays two text panels: one listing current players and another listing the question being considered. When Applet C has been downloaded, it establishes communication to Server C through a socket connection. Through this connection, the client sends desired force vectors to Server C every 3 seconds. Server C aggregates force commands from all active users and controls the robot. Server C also transmits information about the current question being asked and the player list back to each Applet C.

To control the robot from Server C, we developed code based on Java I/O streams that can communicate with Adept's V+ programming-language (which has several functions for handling serial I/O). Testing revealed that every 120 bytes or so, one or two bytes would be mysteriously lost. We fixed this problem by slowing down the connection to 4800 bits per second. We originally wrote applet C using

Java 1.1, but found that as of June 1999 few browsers supported that version so we re-wrote it in Java 1.0 using the Java Development Kit (JDK) and the Abstract Windowing Toolkit (AWT).

7 Planchette Motion Model

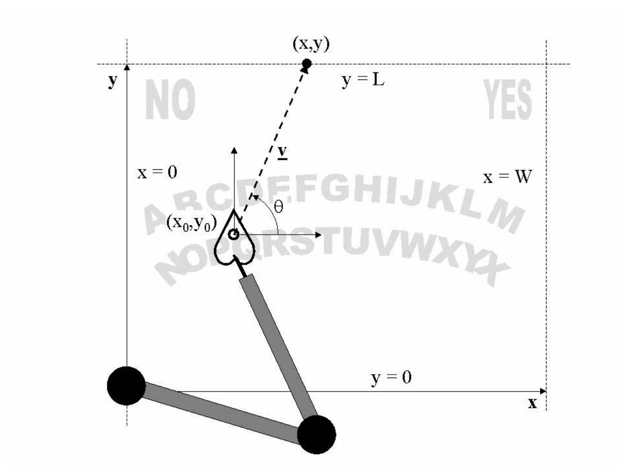

Figure 4: *Planchette Coordinates*.

As described above, Applet C at each client sends a desired motion vector to Server C every 3 seconds. At the client, the user's mouse position is read by a local java applet and a ”virtual” planchette is displayed in the lower window as it tracks the user's mouse. To make the interface more realistic, planchette motion is based on an inertial model. We treat the vector from the center of the planchette screen to the current mouse position as a force command. Consider the coordinate frame defined in Figure 4. User i specifies desired acceleration (a_{ix}, a_{iy}).

We model frictional drag on the planchette with a constant magnitude and a direction opposite the current velocity of the planchette. If the current velocity of the planchette is $\underline{v}_0 = (v_{0x}, v_{0y})$ and the magnitude of the constant frictional acceleration is a_f then $a_{fx} = a_f \frac{-v_{0x}}{\sqrt{v_{0x}^2+v_{0y}^2}}$, and $a_{fy} = a_f \frac{-v_{0y}}{\sqrt{v_{0x}^2+v_{0y}^2}}$. So that $\underline{a}_f = (a_{fx}, a_{fy})$. The resulting velocity of the planchette is: $\underline{v} = v_0 + (\underline{a} + \underline{a}_f)\Delta t$. The virtual planchette is updated locally 30 times a second, so $\Delta t = .03$ seconds.

One suggestion was that we should “normalize” the input force vector from the user after polling cycle. This has the positive effect of treating no motion as a zero force input, but has the negative effect of causing confusion due the difference in cycle time between the local planchette motion and the remote planchette motion. For example if the remote planchette is in the upper left and the user wants to move it to the lower right, he or she will position the local planchette in the lower right, but after one cycle this position would be treated as zero and the user would be unable to move the planchette any further in the desired direction. Thus we do not normalize the input force vector.

Summing the inputs from all users gives us the net desired acceleration of the planchette, $a_x = \sum_i a_{ix}$, and $a_y = \sum_i a_{iy}$. Then $\underline{a} = (a_x, a_y)$. The physical robot accepts commands in the form of a desired goal point and speed. To avoid the robot moving outside the viewable region, we calculate a goal point on the boundary of the region. For example, with a region defined by $0 < x < W$ and $0 < y < L$, we project the robot's current position in direction $\underline{v}$ until it hits the boundary. Let $\theta = \tan^{-1}(\frac{v_y}{v_x})$. To calculate the goal point, the following equation for y corresponds to each of the four possible regions of θ:

$$
\begin{array}{rl}
0^\circ \le \theta < 90^\circ & y = \min(L,\, y_0 + (W - x_0)\tan\theta) \\
90^\circ \le \theta < 180^\circ & y = \min(L,\, y_0 + (-x_0)\tan\theta) \\
180^\circ \le \theta < 270^\circ & y = \max(0,\, y_0 + (-x_0)\tan\theta) \\
270^\circ \le \theta < 360^\circ & y = \max(0,\, y_0 + (W - x_0)\tan\theta)
\end{array}
$$

Then $x = x_0 + \frac{(y-y_0)}{\tan\theta}$. We send the robot controller a move command toward goal point (x,y) with speed $\|v\| = \sqrt{v_x^2 + v_y^2}$. This procedure is repeated every 3 seconds.

8 Experimental Data and Analysis

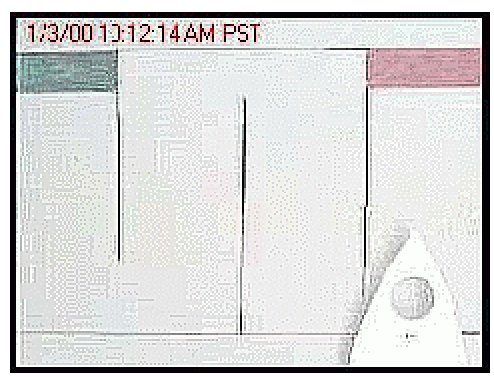

Figure 5: *Test Maze M1*.

To test collaborative control performance in the presence of noise, we replaced the Ouija board with a diagram of a maze and asked test subjects to use our interface to navigate through the maze as shown in Figure 5. Figure 6 illustrates the camera image after it is corrupted by noise (we turn down brightness and resolution differently at each client). We detect maze boundaries in software to prevent the planchette from moving across a maze wall or from leaving the maze itself. The test subjects were two female

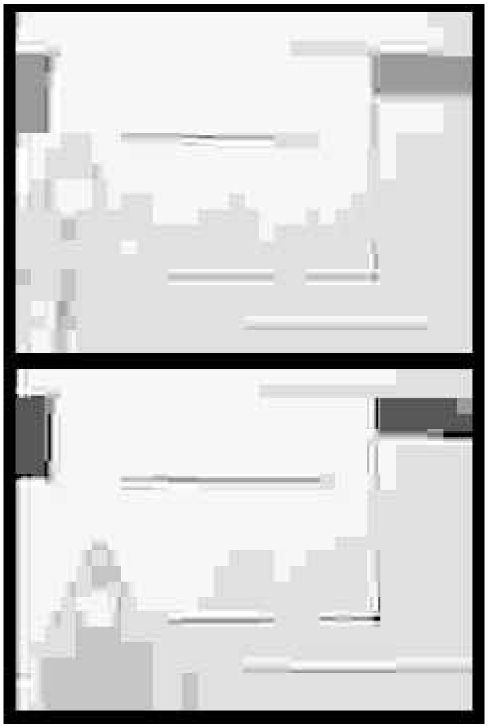

Figure 6: *Two views of the Maze corrupted by digital noise.*

undergraduates. Neither had any previous experience with the system. For each trial, we recorded their total navigation time in seconds, first individually (A and B), and then when working collaboratively (A+B):

Subject	Trial 1	Trial 2	Trial3
A	146	139	105
B	177	141	175
A+B	65	71	72

As described in the Appendix, the Central Limit Theorem suggests that the average of all controls may yield better performance than that from any single source. Since each player has different noisy information about the maze and the current location of the planchette, the last line in the table suggests that performance improves when the players collaborate.

9 Summary and Future Work

We have described a system for collaborative teleoperation on the Internet using a shared control model using vector averaging.

The notion of collaborative control is relevant to a broad range of systems including market prices, voting, and the shared development of software systems such as Linux. The research field of Computer Supported Cooperative Work (CSCW) studies such systems usually outside the domain of robotics. See [21, 22, 23] for a sample of the literature in this area. The well-known "Delphi Method" [24] for group decision making developed in the 1960's by Olaf Helmer and Norman Dalkey at the RAND Corporation is built on a similar model of input averaging.

As of February, Ouija 2000 has 1000 registered players. Past experience suggests that user load (and complaints!) will force us to refine our implementation. We will continue maze experiments with different test subjects and will also test the effect of differing time delays. We also plan to experiment with other control models, for example winner-take-all or weighting the average based on prior performance by each user.

Appendix A

In statistics, the Central Limit Theorem describes how independent random variables can be combined to yield an estimate that gets more accurate as the number of variables increases.

Let X_i be an input from user i. Consider that each X_i is an iid random variable with mean μ and variance σ^2. Let χ_n denote the arithmetic average of the inputs and χ^* denote the normalized random variable

$$\chi^* = \frac{\chi_n - \mu}{\frac{\sigma}{\sqrt{n}}}$$

The Central Limit Theorem states that as n approaches ∞, χ^* approaches a normal distribution with mean zero and unit variance.

Acknowledgments

We thank Brian Carlisle, Gil Gershoni, Michael Idinopulous, Hiro Narita, Tiffany Shlain, Joan Sulaiman, and Heidi Zuckerman-Jacobson for their contributions to this project. Ouija 2000 will be included in the Whitney Biennial in New York.

References

[1] O. Khatib, K. Yokoi, K. Chang, D. Ruspini, R. Holmberg, , and A. Casal. Coordination and decentralized cooperation of multiple mobile manipulators. *Journal of Robotic Systems*, (13(11)), 1996.

[2] Rachel and Lauren Carpenter. **http://www.cinematrix.com** .

[3] 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. Archives can be viewed at http://www.usc.edu/dept/raiders/.

[4] K. Goldberg, J. Santarromana, G. Bekey, S. Gentner, R. Morris, C. Sutter, and J. Wiegley. A telerobotic garden on the world wide web. Technical report, SPIE OE Reports, June 1996. Visit: http://telegarden.aec.at.

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

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

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

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

[9] Barney Dalton and Ken Taylor. A framework for internet robotics. In *IEEE International Conference On Intelligent Robots and Systems (IROS): Workshop on Web Robots*, Victoria, Canada, 1998.

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

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

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

[13] 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.

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

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

[16] Organized by Roland Siegwart. In *IEEE International Conference On Intelligent Robots and Systems (IROS): Workshop on Web Robots*, Victoria, Canada, 1998.

[17] Organized by Ken Goldberg and Eric Paulos. In *IEEE International Conference On Robotics and Automation (ICRA): Workshop on Current Challenges in Internet Telerobotics*, Detroit, MI, 1999.

[18] **http://www.idsoftware.com** .

[19] Parker Brothers Inc. The ouija board game.

[20] **http://www.inetcam.com**.

[21] Garret Hardin. The tragedy of the commons. *Science*, (162), 1968.

[22] Thomas Malone and Kevin Crowston. The interdisciplinary study of coordination. *ACM Computing Surveys*, 26(1), 1994.

[23] Eric S. Raymond. The cathedral and the bazaar. *First Monday*, **http://www.firstmonday.dk**, (3(3)), 1998.

[24] H. Linstone and M. Turoff. *The Delphi Method: Techniques and Applications*. Addison-Wesley, 1975. An excellent review can be found in: http://eies.njit.edu/ turoff/Papers/delphi3.html.