ShareCam Part I: Interface, System Architecture, and Implementation of a Collaboratively Controlled Robotic Webcam

Dezhen Song
IEOR Department
University of California,
Berkeley, CA 94720-1777, USA

Ken Goldberg
IEOR and EECS Departments
University of California,
Berkeley, CA 94720-1777, USA

***Abstract*— ShareCam is a robotic pan, tilt, and zoom web-based camera controlled by simultaneous frame requests from online users. Part II describes algorithms. This paper, Part I, focuses on the system. Robotic webcameras are commercially available but currently restrict control to only one user at a time. ShareCam introduces a new interface that allows simultaneous control by many users. In this Java-based interface, participating users interact from their remotely located browsers where users draw desired frames over a fixed panoramic image. User inputs are transmitted back to a pair of PC servers that compute optimal camera parameters, servo the camera, and provide a video stream to all users. We describe the system, online experiments, and compare results with two frame selection models based on user "satisfaction," one memoryless and the second based on satisfaction over multiple motion cycles. ShareCam is available online at: www.tele-actor.net/sharecam/**

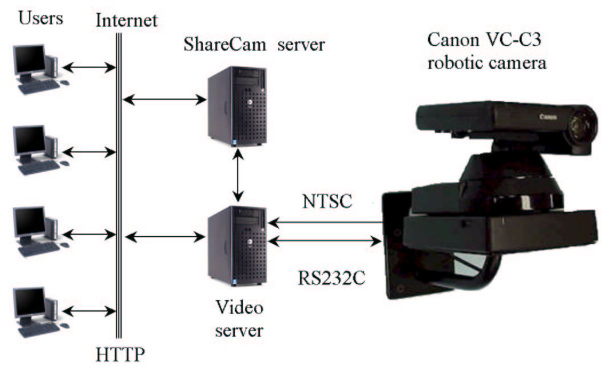

Fig. 1. *ShareCam System Architecture.*
http://www.tele-actor.net/sharecam/

I. Introduction

Robotic webcameras with pan, tilt, and zoom controls are now commercially available and are being installed in dozens of locations[1] around the world. In these systems, the camera parameters can be remotely adjusted by viewers via the Internet to observe details in the scene. Current control methods restrict control to one user at a time; users have to wait in a queue for their turn to operate the camera. In this paper we describe ShareCam, a new system that eliminates the queue and allows many users to share control of the robotic camera simultaneously.

As illustrated in Figure 1, the ShareCam system includes the camera and two servers that communicate with users via the Internet. Streaming video is captured at the camera server and streamed back to the remote users using a Java interface. User responses are collected at the ShareCam server and used to compute optimal camera positions, which are sent to camera server to control the camera.

ShareCam's Java-based interface includes two image windows, one fixed for user input and the other a live streaming video image. The interface collects requested camera frames (specified as desired rectangles) from n users, computes a single camera frame based on all inputs, and moves the camera accordingly. Below we describe system details and two frame selection models based on user "satisfaction".

This work was supported in part by the National Science Foundation under IIS-0113147, by Intel Corporation, by Microsoft Corporation, and in part by UC Berkeley's Center for Information Technology Research in the Interest of Society (CITRIS). For more information please contact dzsong@ieor.berkeley.edu or goldberg@ieor.berkeley.edu.

[1]See: http://www.x-zone.canon.co.jp/WebView-E/index.htm

II. Related Work

ShareCam is an example of Collaborative Telerobotics, in this case the telerobot is a camera with 3 degrees of freedom. In the taxonomy proposed by Tanie et al. [7], ShareCam is a Multiple Operator Single Robot (MOSR) system. Collaborative Telerobotics is motivated by applications such as education and journalism, where groups of users desire simultaneous access to a single robotic resource. Inputs from each user are combined to generate a single control stream for the robot.

The Internet provides a low-cost and widely-available interface that can make physical resources accessible to a broad range of participants. There are now thousands of webcams, dozens of "online robots", a book from MIT Press [14], and an IEEE Technical Committee on Internet and Online Robots.

Online robots, controllable over the Internet, are an active research area. In addition to the challenges associated with time delay, supervisory control, and stability, online robots must be designed to be operated by non-specialists through intuitive user interfaces and to be accessible 24 hours a day; see [17], [19], [22], [21], [24], [27], [30], [31], [38] for examples of recent projects.

Tanie, Matsuhira, Chong, et al. [7] proposed the following taxonomy for teleoperation systems: Single Operator Single Robot (SOSR), Single Operator Multiple Robot (SOMR), Multiple Operator Multiple Robot (MOMR). and Multiple Operator Single Robot (MOSR). Most online robots are SOSR, where control is limited to one operator at a time. Tanie et al. analyzed an MOMR system where each operator controls one robot arm and the robot arms have overlapping workspaces. They show that predictive displays and scaled rate control are effective in reducing pick-and-place task completion times that require cooperation from multiple arms.

A number of SOSR systems have been designed to facilitate remote interaction. Paulos and Canny's Personal Roving Presence (PRoP) telerobots, built on blimp or wheeled platforms, were designed to facilitate remote social interaction with a single remote operator [32], [33]. Fong, Thorpe and colleagues study SOSR systems where collaboration occurs between a single operator and a mobile robot that is treated as a peer to the human and modeled as a noisy information source [11]. Related models of SOSR "cobots" are analyzed in [1], [3], [11], [28], [40].

In an MOMR project by Fukuda, Liu, Xi, and colleagues [10], two remote human operators collaborate to achieve a shared goal such as maintaining a given force on an object held at one end by a mobile robot and by a multi-jointed robot at the other. The operators, distant from the robots and from each other, each control a different robot via force-feedback devices connected to the Internet. The authors show both theoretically and experimentally that event-based control allows the system to maintain stable synchronization between operators despite of variable time-lag on the Internet.

MOMR models are also relevant to online collaborative games such as *Quake*, where players remotely control individual avatars in a shared environment.

In SOMR systems, one tele-operator or process controls multiple robots. This bears some relation to Cooperative (behavior-based) robots, where groups of *autonomous* robots interact to solve an objective [2]. Recent results are reported in [5], [9], [37], [35].

One precedent of an online MOSR system is described in McDonald, Cannon and colleagues [29]. For waste cleanup, several users assist in waste cleanup using Point-and-Direct (PAD) commands [6]. Users point to cleanup locations in a shared image and a robot excavates each location in turn. In this Internet-based MOSR system, collaboration is serial but pipelined, with overlapping plan and execution phases. The authors demonstrate that such collaboration improves overall execution time but do not address conflict resolution between users.

In [12] Goldberg and Chen analyze a formal model of collaborative control and in [13] describe Internet-based MOSR system that averaged multiple human inputs to simultaneously control a single industrial robot arm. In [15], [16] we propose the "Spatial Dynamic Voting" (SDV) interface. The SDV collects, displays, and analyzes a sequence of spatial votes from multiple online operators at their Internet browsers. The votes drive the motion of a single mobile robot or human "Tele-Actor".

Research on controllable webcams or Internet cameras are focus on two perspectives: system architectures and applications. Desmet, Verkest, Mignolet et al. [8], [20], [43] designed webcams using reconfigurable hardware and embedded software. They implemented a secure VPN (Virtual Private Network) with 3DES encryption and Internet camera server (including JPEG compression). Brooks and McKee [4] implemented an automated camera which is placed during teleoperation using Visual Acts theory and architecture to provide operators with task relevant information in a timely manner. The applications of webcams is not limited to surveillance [23] or teleconferencing [25], [26], [34]. Schmid, Maule, and Roth [39] used a controllable webcam to perform all the tests for industrial robots given by ISO 9283 "Performance criteria and related test methods". Pollak and Hutter [36] installed a Phillips webcam on an Olympus BX60 light microscope to record movies of investigated samples. Zhang, Navab, and Liou [44] used webcams to creat an interactive sales model for web customers.

In independent work, Kimber, Liu, Foote et al describe a multi-user robot camera in [25], [26]. The application is designed for videoconferencing system. They use multiple cameras in the systems: panoramic cameras and a pan-tilt-zoom camera. Panoramic cameras generate a dynamic panoramic view of the conference site. Users control the pan-tilt-zoom camera by drawing on panoramic view. The system is well suitable for videoconferencing environment, where illumination condition is constantly good so that the image quality of panoramic view can be guaranteed. We believe multiple camera systems are good but not necessary for scenic sites where dynamic information is not necessary. The panoramic image can be generated by the same pan-tile-zoom camera resulting in less bandwidth requirement.

An earlier paper [42], published in the Workshop on Algorithmic Foundations of Robotics, formulated the ShareCam problem geometrically and reported initial results on exact algorithms: for n users and m zoom levels, the exact algorithm runs in $O(n^2 m)$ time. Har-Peled et al. [18] improved the exact algorithm to $O(mn^{3/2}\log^3 n)$ and proposed a near linear $\epsilon-$approximation algorithm. ShareCam Part II [41], the companion paper presented in this conference, describes approximate and distributed algorithms for solving the ShareCam frame selection problem.

Fig. 2. *This figure illustrates ShareCam's Java-based user interface, which currently runs on most Windows based PCs. Users view two windows. One (not shown) displays a live video stream as captured by the robotic camera. The second window, illustrated here, contains the user interface. The panoramic image is a fixed photo of the camera's reachable range of view. The snapshot above shows 6 active users listed in the scrollable window at the left. Each user requests a camera frame by positioning a dashed rectangle over the panoramic image. Based on these requests, the algorithm computes an optimal camera frame (shown with solid rectangle), and servoes the camera accordingly to displays the resulting live video stream. The horizontal bars indicate levels of user satisfaction as described below. The system is installed in our research lab at Berkeley but was moved outdoors in April 2003.*

III. ShareCam Interface

The ShareCam interface facilitates interaction and collaboration among remote users. Users register online to participate by selecting a characteristic color and submitting their email address to the ShareCam server, which stores this information in our database and immediately sends back a password via email. The server also maintains a tutorial and an FAQ section to familiarize new users with how the systems works.

The ShareCam interface contains two windows: The video window shows the current camera view. Figure 2 illustrates the panoramic window and the ShareCam user interface.

The interface also facilitates chat between users. Each user can type in a short sentence, which is displayed underneath his/her requested frame in the panoramic image. A clocklike timer is located at the bottom right of the interface indicating the time before the next camera movement (typically 5-10 seconds).

IV. Hardware

The ShareCam server is an AMD K7 950Mhz PC with 1.2GB SDRAM connected to a 100Mbs T3 line. The camera server is an AMD K7 950Mhz PC with 640MB SDRAM connected to an 100Mbs T3 line at the remote site. It has a video-capture card, which captures video at 320×240 resolution. It also serves as video server running InetCam [2] software to broadcast video.

We used the Canon controllable camera, model VC-C3. A comparable camera is available from Sony. The Canon camera has motorized pan, tilt and zoom with a 10x power zoom lens. It has PAL, composite, and S-video output with a resolution of 450 horizontal lines. It can communicate with a PC via a RS232C link at 14,400bps. Its pan, tilt, and zoom speed is 76 degrees per second at maximum and 0.5 degrees per second at minimum. It has an accuracy of 0.5 degrees and a 380,000 pixel CCD array.

[2] http://www.inetcam.com

V. Software

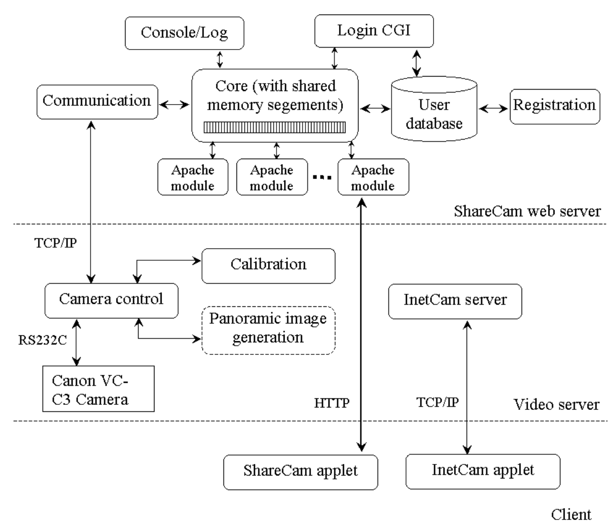

Fig. 3. *ShareCam system software diagram.*

As illustrated in Figure 3, custom software includes: (1) the ShareCam server, (2) the camera control software and video capturing package at the video server, and (3) the client side ShareCam Java applet.

The ShareCam server runs Redhat Linux 7.1 and the Apache web server 1.3.20. All modules are written in GNU C++ with optimization of running speed. The ShareCam server package consists of core process, Apache modules, communication process, user databases, registration module, console/log module, and login CGI script. The customized Apache module deals with communication between web clients and the server via HTTP. It accepts the requested frame from a client and sends him/her the requested frames of others every second. It can be viewed as a CGI script but with much higher scalability. The communication module connects to the video server via a socket link to send camera control commands. A console/log module allows us to monitor and record system status in real time.

The overall design emphasizes data sharing among all processes. Collaborative control requires that all clients are able to see each other's information in real time. This is achieved by sharing memory segments among all server processes. Therefore the shared memory segment managed by the core process is the key data structure.

Clients download two applets: the ShareCam applet and the InetCam applet. The ShareCam applet is a customized software, which is shown in Figure 2. Part of the frame selection computation is done at the client side, which is implemented in the ShareCam applet. The ShareCam applet is written in Java 1.1 to ensure the compatibility with most browsers. The InetCam applet is a third party software that functions as a video terminal.

The video server package includes camera control, InetCam server, calibration, and panoramic image generation. The camera control module written in Microsoft Visual C++ is the primary module. It accepts camera control commands from the ShareCam server and translates it into the RS232C protocol, which is built on packages provided by Lawrence Berkeley National Laboratory[3].

VI. EXPERIMENTS

In this section, we will present experimental results for two frame selection models. We begin with a review of definitions and notation. More details can be found in the companion paper: ShareCam Part II [41].

We consider two models for the optimal camera frame, the first is memoryless based only on the current set of frame requests. The second is a temporal model based on the history of frame requests with exponentially decaying weights.

A. Memoryless Frame Selection Model

In the ShareCam system, c is a vector of camera parameters that users can control. Let c define a camera frame $[x, y, z]$, where x, y specify the center point of the frame, which is corresponding to pan and tilt, and z specifies size of the frame, which corresponds to zoom level. c defines a rectangular camera frame (the camera has a fixed aspect ratio of 4:3). User i requests a desired frame r_i. Given requests from n users, the system computes a single global frame c^* that will best satisfy the set of requests.

We define a Generalized Intersection Over Maximum (GIOM) metric for user "satisfaction" $s(c, r_i)$ based on how the user's requested frame r_i compares with a candidate camera frame c. Each of n users submits a request. Let

$$s(c) = \sum_{i=1}^{n} s_i(r_i, c) \tag{1}$$

In the memoryless frame selection model, we want to find c^*, the value of c that maximizes $s(c)$ based only on the current set of requests:

$$\max_c s(c).$$

In each motion cycle, we servo the camera to this frame.

B. Temporal Frame Selection Model

An alternative frame selection model is based on the history of user frame requests over multiple motion cycles. We extend equation 1 using a weighted sum of the user satisfaction. In this case total satisfaction is a function of time t:

$$s(c, t) = \sum_{i=1}^{n} \alpha_i(t) s_i(r_i(t), c(t)) \tag{2}$$

where the weight $\alpha_i(t)$ for user i is a function of the user's previous "dissatisfaction" level: $u_i(t) = 1 - s_i(r_i(t), c(t))$. One candidate form for weights is

$$\alpha_i(t) = \sum_{k=0}^{t-1} \frac{u_i(k)}{2^{t-1-k}}$$

which yields the recursive formulation:

$$\alpha_i(t) = u_i(t-1) + \alpha_i(t-1)/2$$

If user i does not get satisfied by the camera frame computed during the current frame, his weight $\alpha_i(t)$, will increase over future motion cycles, eventually dominating the weights of other users to satisfy his desired frame request. In this sense fairness is guaranteed over time.

These frame optimization problems can be solved with exact algorithms [42] or fast new approximation algorithms [41].

Figure 4 shows four examples with the Memoryless Frame Selection model. Note that the optimal frame grows in image (b) after a large requested frame is added. In Figure 4(c), two more frames are requested. Since they can not compete with the central group of requested frames, the optimal frame remains unchanged. Figure 4(d) shows a case with all but two requested frames disjoint, the

[3]http://www-itg.lbl.gov/mbone/devserv/

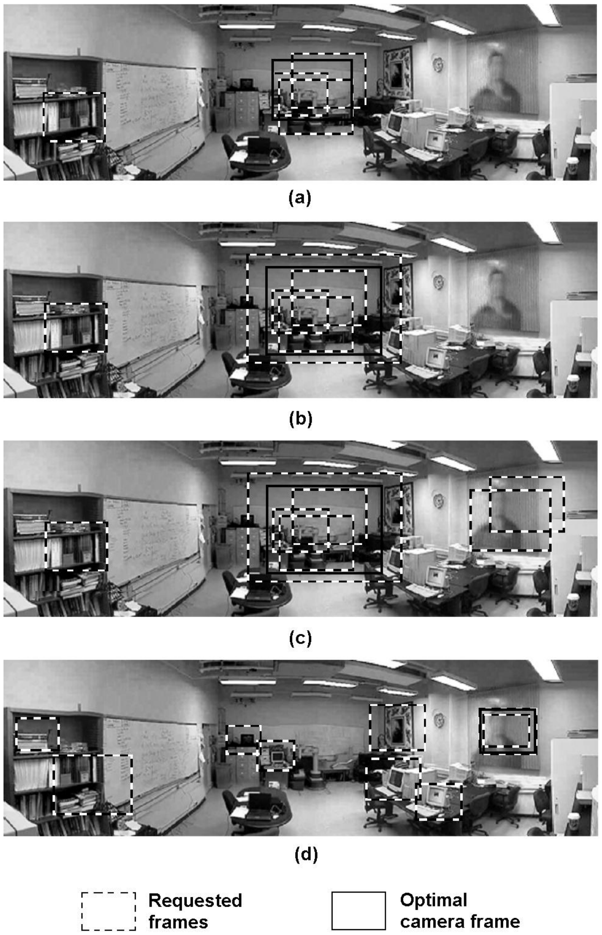

Fig. 4. *Examples using Memoryless Frame Selection model defined by equation 1. Four different sets of requested frames and the corresponding optimal frame are displayed. Note that the resulting frame is very different than what would be determined by simple averaging, and that some requests never get satisfied.*

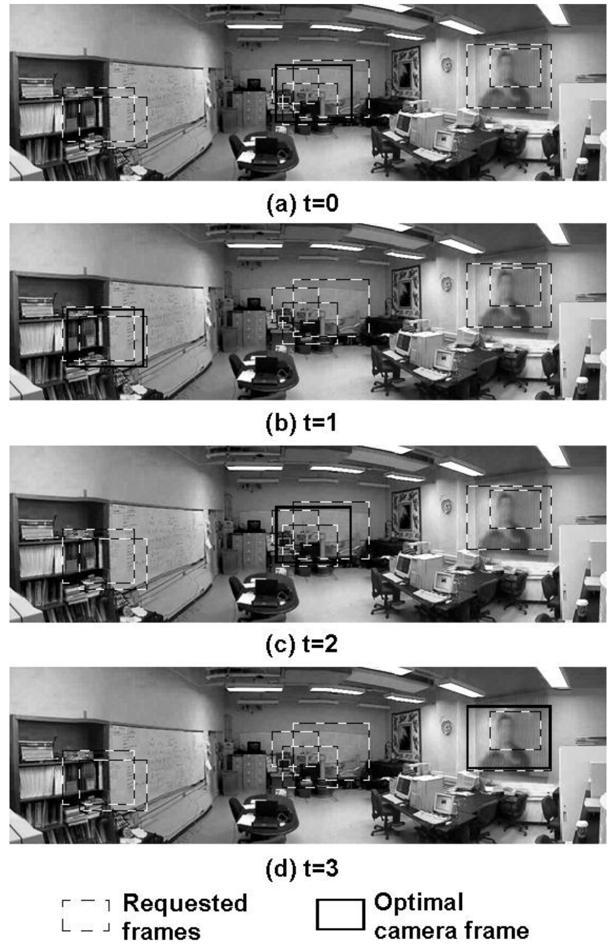

Fig. 5. *Examples with the Temporal Frame Selection Model defined by equation 2. The set of requested frames is held constant, but weights evolve so that the camera frame changes to facilitate "fairness".*

algorithm selects a frame that covers the two overlapping frames. Figure 4 also illustrates that some users can be starved indefinitely.

Figure 5 shows four examples with the Temporal Frame Selection model, where frame selection is based on user satisfaction over multiple motion cycles. A sequence of 4 motion cycles is illustrated with the same set of requested frames. Note that with this model, the camera frame changes to balance overall user satisfaction over time.

C. Online experiments

The ShareCam system went online in June of 2002 with the camera installed in our Alpha Lab from June 8, 2002 to February 2003 as shown in the previous figures. An illustration of the total requested frames is shown in figure 6.

Figure 6(a) displays all 4822 requested frames for the experiment duration. We are interested in how user interest is distributed in the panorama. To compute the interest distribution, we define $g(x, y)$ be the interest for point (x, y) in gray scale, i.e. $0 \le g(x, y) \le 255$, $r_j : 1 \le j \le 4822$ be the j^{th} requested frame, and an indicator variable,

$$I(x, y, j) = \begin{cases} 1 & \text{if } (x, y) \in r_j \\ 0 & \text{otherwise} \end{cases}$$

Say a darker point means more interest, the interest for point (x, y) is $g(x, y)$, and define $g_{max} = \arg\max_{(x,y)} g(x, y)$,

$$g(x, y) = 255(1 - \frac{\sum_{j=1}^{4822} I(x, y, j)}{g_{max}}).$$

We compute $g(x, y)$ for each point in the panorama and generate the figure 6(b). As shown in the figure, the most popular region is the center of the camera workspace, looking at the Adept robot arm in our lab, where one our colleague was often performing robot calibration tests.

(a) 4822 Requested frames

(b) Interest density distribution in grayscale

Fig. 6. *Data from June 8, 2002 to February 6, 2003.*

VII. Conclusions and Future Work

This paper describes the ShareCam, a MOSR teleoperation system that allows a group of Internet users to simultaneously share control of a pan, tilt, and zoom camera. We described the ShareCam interface, system architecture, and experiments with two frame selection models.

Currently, the panoramic image generation is done offline by hand using Photoshop. In future work we will develop an automatic procedure for creating and calibrating the panoramic image.

The Sharecam system was moved to an outdoor location on the UC Berkeley campus in June 2003, and is available online at http://tele-actor.net/sharecam.

VIII. Acknowledgments

We are grateful to Frank van der Stappen, Sariel Har-Peled, Vladlen Koltun, Dana Plautz, Anatoly Pashkevich, Cory Cox, Yong Rui, J. Foote, Q. Lui, and Eric Paulos for insightful discussions and feedback. Thanks to M. Faldu and W. Guan for their contributions to the ShareCam system, and I.Y. Song, K. "Gopal" Gopalakrishnan , Ron Alterovitz, A. Levandowski, M. Mckelvin, A. Ho, and J. Vidales for helpful suggestions.

IX. References

[1] H. Arai, T. Takubo, Y. Hayashibara, and K Tanie. Human-robot cooperative manipulation using a virtual nonholonomic constraint. In *IEEE International Conference on Robotics and Automation*, April 2000.

[2] R. Arkin. Cooperation without communication: Multiagent schema-based robot navigation. *Journal of Robotic Systems*, 9(3):351–364, 1992.

[3] C. Bernard, H. Kang, S. K. Sigh, and J.T. Wen. Robotic system for collaborative control in minally invaive surgery. *Industrial Robot: An international Journal*, 26(6):476–484, 1999.

[4] B.G. Brooks BG and G. T. McKee. The visual acts model for automated camera placement during teleoperation. In *IEEE International International Conference on Robotics and Automation (ICRA 2001), Korea*, 5 2001.

[5] Z. Butler, A. Rizzi, and R. Hollis. Cooperative coverage of rectilinear environments. In *IEEE International Conference on Robotics and Automation*, April 2000.

[6] D. J. Cannon. *Point-And-Direct Telerobotics: Object Level Strategic Supervisory Control in Unstructured Interactive Human-Machine System Environments*. PhD thesis, Stanford Mechanical Engineering, June 1992.

[7] N. Chong, T. Kotoku, K. Ohba, K. Komoriya, N. Matsuhira, and K. Tanie. Remote coordinated controls in multiple telerobot cooperation. In *IEEE International Conference on Robotics and Automation*, volume 4, pages 3138–3343, April 2000.

[8] D. Desmet, P. Avasare, P. Coene, S. Decneut, F. Hendrickx, T. Marescaux, J-Y. Mignolet, R. Pasko, P. Schaumont, and D. Verkest. *Design of Cam-E-leon, a run-time reconfigurable Web camera*. Springer-Verlag, Berlin, Germany, 2002.

[9] B. Donald, L. Gariepy, and D. Rus. Distributed manipulation of multiple objects using ropes. In *IEEE International Conference on Robotics and Automation*, April 2000.

[10] I. Elhaji, J. Tan, N. Xi, W. Fung, Y. Liu, T. Kaga, Y. Hasegawa, and T. Fukuda. Multi-site internet-based cooperative control of robotic operations. In *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 2000.

[11] T. Fong, C. Thorpe, and C. Baur. Collaboration, dialogue, and human-robot interation. In *10th International Symposium of Robotics Research*, 2001.

[12] K. Goldberg and B. Chen. Collaborative control of robot motion: Robustness to error. In *International Conference on Intelligent Robots and Systems (IROS)*, volume 2, pages 655–660, October 2001.

[13] K. Goldberg, B. Chen, R. Solomon, S. Bui, B. Farzin, J. Heitler, D. Poon, and G. Smith. Collaborative teleoperation via the internet. In *IEEE International Conference on Robotics and Automation (ICRA)*, volume 2, pages 2019–2024, April 2000.

[14] K. Goldberg and R. Siegwart, editors. *Beyond Webcams: An Introduction to Online Robots*. MIT Press, 2002.

[15] K. Goldberg, D. Song, Y. Khor, D. Pescovitz, A. Levandowski, J. Himmelstein, J. Shih, A. Ho, E. Paulos, and J. Donath. Collaborative online teleoperation with spatial dynamic voting and a human "tele-actor". In *IEEE International Conference on Robotics and Automation (ICRA)*, volume 2, pages 1179–84, May 2002.

[16] K. Goldberg, D. Song, and A. Levandowski. Collaborative teleoperation using networked spatial dynamic voting. *The Proceedings of The IEEE*, 91(3):430–439, March 2003.

[17] K. Han, Y. Kim, J. Kim, and S.Hsia. Internet control of personal robot between kaist and uc davis. In *IEEE International Conference on Robotics and Automation (ICRA)*, 2002.

[18] Sariel Har-Peled, Vladlen Koltun, Dezhen Song, and Ken Goldberg. Efficient algorithms for shared camera control. In *19th ACM Symposium on Computational Geometry, San Diego, CA*, June 2003.

[19] H. Hu, L. Yu, P. W. Tsui, and Q. Zhou. Internet-based robotic systems for teleoperation. *Assemby Automation*, 21(2):143–151, May 2001.

[20] J-Y. Mignolet J-Y, S. Vernalde, and M. Engels. A low-power reconfigurable internet appliance camera. *Revue Hf. Electronique, Telecommunications*, (4):4–11, 2000.

[21] S. Jia, Y. Hada, G. Ye, and K. Takase. Distributed telecare robotic systems using corba as a communication

architecture. In *IEEE International Conference on Robotics and Automation (ICRA)*, 2002.

[22] S. Jia and K. Takase. A corba-based internet robotic system. *Advanced Robotics*, 15(6):663–673, Oct 2001.

[23] T. Kaga, S. Nojima, E. Nanma, M. Tamura, and H. Nakagane. Internet camera system. *Matsushita Technical Journal*, 44(5):66–73, 1998.

[24] J. Kim, B. Choi, S. Park, K.Kim, and S. Ko. Remote control system using real-time mpeg-4 streaming technology for mobile robot. In *IEEE International Conference on Consumer Electronics*, 2002.

[25] D. Kimber, Q. Liu, J. Foote, and L. Wilcox. Capturing and presenting shared multi-resolution video. In *SPIE ITCOM 2002. Proceeding of SPIE, Boston*, volume 4862, pages 261–271, Jul. 2002.

[26] Q. Liu, D. Kimber, L. Wilcox, M. Cooper, J. Foote, and J. Boreczky. Managing a camera system to serve different video requests. In *Proceedings of IEEE International Conference on Multimedia and Expo (ICME), Lausanne, Switzerland*, volume 2, pages 13–16, Aug. 2002.

[27] R.C. Luo and T. M. Chen. Development of a multibehavior-based mobile robot for remote supervisory control through the internet. *IEEE/ASME Transactions on Mechatronics*, 5(4):376–385, 2000.

[28] K. Lynch and C. Liu. Designing motion guides for ergonomic collaborative manipulation. In *IEEE International Conference on Robotics and Automation*, April 2000.

[29] M. McDonald, D. Small, C. Graves, and D. Cannon. Virtual collaborative control to improve intelligent robotic system efficiency and quality. In *IEEE International Conference on Robotics and Automation*, volume 1, pages 418–424, April 1997.

[30] T. Mirfakhrai and S. Payandeh. A delay prediction approach for teleoperation over the internet. In *IEEE International Conference on Robotics and Automation (ICRA)*, 2002.

[31] L. Ngai, W.S. Newman, and V. Liberatore. An experiment in internet-based, human-assisted robotics. In *IEEE International Conference on Robotics and Automation (ICRA)*, 2002.

[32] E. Paulos and J. Canny. Ubiquitous tele-embodiment: Applications and implications. *International Journal of Human-Computer Studies/Knowledge Acquisition*, 1997. Special Issue on Innovative Applications of the World Wide Web.

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

[34] M. Perry and D. Agarwal. Remote control for videoconferencing. In *Proceedings of 2000 Information Resources Management Association International Conference. Anchorage, AK, USA*, 2000.

[35] P. Pirjanian and M. Mataric. Multi-robot target acquisition using multiple objective behavior coordination. In *IEEE International Conference on Robotics and Automation*, April 2000.

[36] C. Pollak and H. Hutter. A webcam as recording device for light microscopes. *Journal of Computer-Assisted Microscopy*, 10(4):179–83, 1998.

[37] I. Rekleitis, G. Dudek, and E. Milios. Multi-robot collaboration for robust exploration. In *IEEE International Conference on Robotics and Automation*, April 2000.

[38] R. Safaric, M. Debevc, R. Parkin, and S. Uran. Telerobotics experiments via internet. *IEEE Transactions on Industrial Electronics*, 48(2):424–31, April 2001.

[39] D. Schmid, B. Maule, and I. Roth. Performance teletests for industrial robots by the internet. In *First IFAC-Conference on Telematics Applications in Automation and Robotics TA 2001. Weingarten, Germany. IFAC*, 7 2001.

[40] R. Siegwart and P. Saucy. Interacting mobile robots on the web. In *IEEE International Conference on Robotics and Automation (ICRA)*, 1999.

[41] D. Song, A. Pashkevich, and K. Goldberg. Sharecam part II: Approximate and distributed algorithms for a collaboratively controlled robotic webcam. In *IEEE/RSJ International Conference on Intelligent Robots (IROS)*, Nov. 2003.

[42] D. Song, A. F. van der Stappen, and K. Goldberg. Exact and distributed algorithms for collaborative camera control. In *The Workshop on Algorithmic Foundations of Robotics, December*, Dec. 2002.

[43] D. Verkest, D. Desmet, P. Avasare, P. Coene, S. Decneut, F. Hendrickx, T. Marescaux, J-Y. Mignolet, R. Pasko, and P. Schaumont. Design of a secure, intelligent, and reconfigurable web cam using a c based system design flow. In *Thirty-Fifth Asilomar Conference on Signals, Systems and Computers, IEEE, Piscataway, NJ, USA*, volume 1, pages 463–467, 2001.

[44] X. Zhang, N. Navab, and S-P Liou. E-commerce direct marketing using augmented reality. In *Proceedings of International Conference on Multimedia and Expo. New York, NY, USA.*, 7 2000.