

# Unsupervised Scoring for Scalable Internet-Based Collaborative Teleoperation

Ken Goldberg, Dezhen Song, In Yong Song,  
Jane McGonigal, Wei Zheng, UC Berkeley  
Dana Plautz, Intel

**Draft date:** November 14, 2003

*Abstract*—Our group is studying the potential for scalable Internet-based Collaborative (Multiple Operator Single Robot) Teleoperation, where many users simultaneously share control using browser-based point-and-click interfaces. For applications such as distance learning, automated methods are required to evaluate user performance; this provides individual assessment/reward and incentive for active participation. In this paper we describe “unsupervised scoring”: a numerical measure of individual performance based on clustering and response time. We have implemented a new user interface incorporating this metric using Java. We describe a distributed algorithm for rapidly computing and displaying user scores. In the last section, we demonstrate how user scores can be correlated with task performance.

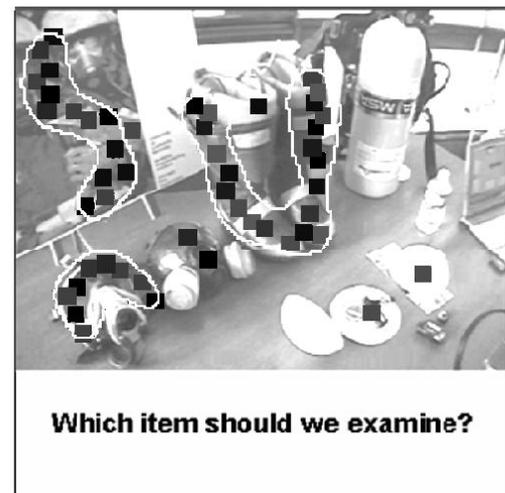
## I. INTRODUCTION

Remote-controlled machines and teleoperated robots have a long history [35]. Networks such as the Internet provide low-cost and widely-available interfaces that makes such resources accessible to the public. In almost all existing teleoperation systems, a single human remotely controls a single machine. For applications ranging from entertainment to remote exploration to distance learning, we consider Collaborative teleoperation (CT) systems that allow many users to simultaneously share control. In a taxonomy proposed by Tanie et al. [4], these are Multiple Operator Single Robot (MOSR) systems, in contrast to conventional Single Operator Single Robot (SOSR) systems.

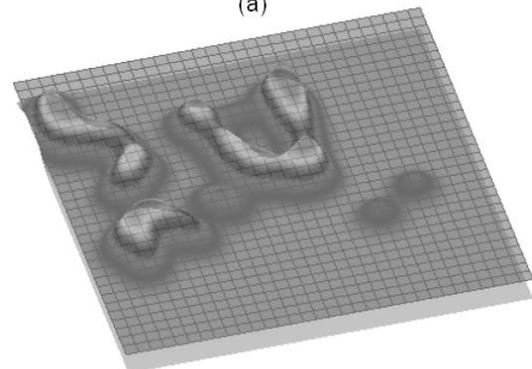
In MOSR systems, inputs from many participants are combined to generate a single control stream. There can be benefits to collaboration: teamwork is a key element in education at all levels [32], [6], [5] and the group may be more reliable than a single (possibly malicious) participant [13]. In [16], [17], we introduced the “Tele-Actor”, a system involving a skilled human with cameras and microphones connected to a wireless digital network, who moves through the remote environment based on live feedback from online users.

In Tele-Actor field tests with students ranging from middle and high-school we discovered that students would often become passive, simply watching the video without participating. Instructors asked for a mechanism to quantify student engagement for grading purposes. We realized we could address both

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



(a)



(b)

Fig. 1. Above, (a) illustrates a sample view from our scalable user interface, where spatial inputs from users are represented with colored square markers, in this case indicating points of interest in a live video image. The unsupervised scoring metric assigns a scalar value to each input. Below, (b) illustrates the aggregate (consensus) density distribution that is used to automatically compute scores based on vote arrival time.

issues by introducing a scoring metric that would continually assess students and provide a competitive element to the interactions.

This paper describes an unsupervised scoring metric and algorithms for rapidly computing it. The metric is “unsupervised” in the sense that it does not rely on a human expert to continuously evaluate performance. Instead, performance is based on “leadership”: how quickly users anticipate the decision of the majority.

As illustrated in Figure 1, the unsupervised scoring metric is based on clustering of user inputs. For  $n$  users, computing scores runs in  $O(n)$  time. This paper presents problem formulation, distributed algorithms, and experiment results.

## II. RELATED WORK

Networked robots, controllable over networks such as 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. There are now dozens of Internet robots online, a book from MIT Press [15], and an IEEE Technical Committee on Internet and Online Robots. See [20], [33], [21], [22], [30], [18], [31], [24], [27], [26] examples of recent projects.

Tanie, Matsuhira, Chong, et al. [4] proposed a useful taxonomy for teleoperation systems: Single Operator Single Robot (SOSR), Single Operator Multiple Robot (SOMR), and Multiple Operator Multiple Robot (MOMR).

Most networked robots are SOSR, where control is limited to one human 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 [4].

In an MOMR project by Elhajj, Fukuda, Liu, Xi, and colleagues [8], [9], 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. MOMR models are also relevant to online collaborative games such as *Quake* and *The Sims Online*, where players remotely control individual avatars in a shared virtual environment.

One precedent for an online MOSR system is described in McDonald, Cannon and colleagues [29]. For waste cleanup, several users to assist remotely using Point-and-Direct (PAD) commands [3]. 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 [14], [13], Goldberg, Chen, et al present an Internet-based MOSR system that averaged multiple vector inputs to control the position of an industrial robot arm. In [36], [37], we present another MOSR system that allow a group of users to simultaneously share control of a single robotic camera. We formulate MOSR problem as resource allocation problem and develop algorithms for camera control.

Outside of robotics, the notion of MOSR is related to a very broad range of group activities including compute gaming, education, social psychology, voting, etc. Although there currently are no standard "best practices" for the design of scoring systems in the computer games industry, some tentative efforts recently have been made to identify the key

features of an effective scoring and ranking model. GamaSutra, the flagship professional resource for digital game designers from around the world, has proposed four characteristics of a successful scoring system. It should be reproducible (consistent), comprehensive (incorporating all significant aspects of game play), sufficiently detailed (so that players understand how it is calculated), and well-balanced (allowing for players to stay competitive even when playing from a disadvantage) [23]. Kriemeier suggests that a failure to address each of these four principles will negatively affect user participation and motivation. Proposed customizations to further increase player motivation in a system with these four standard features include introducing non-zero sum (cooperative) scoring elements and a score decay algorithm that will penalize players for a lower rate of participation in the game. The Tele-Actor unsupervised scoring model aims to incorporate these features, and to investigate their value and effectiveness from a user-perspective.

Two different areas in educational psychology investigate the usefulness of games in learning. The first addresses the element of "fun" in games as a powerful motivational tool [12], [19]. In particular, cooperative play models are thought to provide particularly powerful evidence of games as tools of engagement; as Sutton-Smith notes [38], children and young adults are so motivated to be accepted in such play, they make sacrifices of egocentricity for membership in the group, a claim that may be tested by the unsupervised scoring model.

Csikszentmihalyi's theory of "flow" [7], a confidence-building state in which a participant becomes absorbed in and highly effective at a particular activity, argues for the importance of feedback as the primary criterion for achieving flow. According to Csikszentmihalyi, flow creates a sense of motivation that is intrinsic to the activity, rather than relying on extrinsic rewards (prizes, grades, acclaim), and intrinsic motivation is ultimately more important for long-term success in any activity. For example: Gee [11] argues that students today are used to experiencing flow and achieving a sense of mastery in their at-home game play, but not in school. Many educators are considering adopting more game-like scenarios for learning as a way to incorporate students' proclivity for structured play.

Group decision-making is sometimes modelled as an n-person (multi-player) cooperative game [39], [1]. We can view the unsupervised scoring system as a nonzero-sum multi-player game, in which users compete and cooperate to increase their individual leadership scores. The open-endedness of the n-person, non-zero sum scenario makes its outcome the most theoretically challenging to predict in game theory. The data collected on user behavior in the form of leadership scores may be useful in evaluating a variety of predictive models for n-person games and the associated likelihood and structures of cooperative behavior in real-life [2], [28], [34].

A related question is, how do players learn game strategies and adapt their game play behavior accordingly over time? The study of adaptive learning models in games attempts to develop a theory of "game cognition" that explains why people do not always discover or follow optimal strategies in game play [25]. Erev and Roth [10], for instance, argue for



Fig. 2. The Spatial Dynamic Voting (SDV) interface as viewed in the browser by each online voter. Low framerate live video (approx 1 fps) is displayed in the left window. Users vote on spatial preferences using their mouse to position a small marker (votel) in the middle (voting) window, over either a prestored or live image. Users view the position of all votels and can change their votel positions based on group dynamics. As described below, votel positions are processed to assign scores to each user. The list of active voters, ranked by score is displayed in the right window. The lower right window displays a plot of voter score, overall average, and scores for the top three voters.

the importance of the presence or absence of reinforcement and feedback within a repetitive game structure as the most important factor in predicting player action. By providing feedback in the form of a leadership score and simultaneously tracking changes in scores over time, we can quantify how well scores correlate with group performance.

### III. PROBLEM DEFINITION

Figure 2 illustrates the user interface.

In this section, we propose an unsupervised scoring metric based spatial distributions of votes.

#### A. Inputs and assumptions

Consider the  $k$ th voting image. The server receives a response from user  $i$  in the form of an  $(x, y)$  mouseclick on image  $k$  at time  $t$ . We define the corresponding *votel*:  $v_{ik}(t) = [x_{ik}(t), y_{ik}(t)]$ .

Each votel represents a user's response ("vote") to the voting image. We model such responses with a *voter interest function*, a density function based on the bivariate normal distribution:

$$f_{ik}(x, y) \sim N(v_{ik}(t), \Sigma_{ik}(t))$$

where  $v_{ik}(t)$  is the mean vector and  $\Sigma_{ik}(t)$  is a  $2 \times 2$  variance matrix, such that,

$$\iint_{\sigma} f_{ik}(x, y) dx dy = 1$$

where  $\sigma$  is the area of the voting image. Since  $\sigma$  is a bounded 2D region, the voter interest function is a truncated bivariate normal density function with mean at  $v_{ik}(t)$  as illustrated in Figure 3(a).

1) *Majority Interest Function*: When voting on image  $k$  ends at stopping time  $T$ , the last votel received from each of  $n$  active voters determines  $V_k$ , a set of  $n$  votels. We define the *ensemble interest function* for voting image  $k$  as the normalized sum of these voter interest functions.

$$f_k(x, y) = \frac{1}{n} \sum_{i=1}^n f_{ik}(x, y).$$

2) *Consensus Regions*: As illustrated in figure 3(c), we can extract spatial regions by cutting the ensemble interest function using a horizontal plane at a height proportional to the overall volume. Let  $z_k$  be the cutting threshold. The  $z_k$  value satisfies the condition that the ratio between the partial volume of the ensemble interest function above the horizontal plane and the total volume of the ensemble interest function is constant  $r$ . We use a value of 0.10 (10% of the volume lies above the plane). The cutting plane defines an iso-density contour in the ensemble interest function that defines a set of one or more closed subsets of the voting image,

$$S_k = \{(x, y) | f_k(x, y) \geq z_k\}.$$

As illustrated in Figure 3(c), we refer to these subsets as *consensus regions*:

$$S_k = \{C_{1k}, C_{2k}, \dots, C_{lk}\}.$$

Since there are  $n$  voters,  $l \leq n$  is number of consensus regions.

3) *Majority Consensus Region*: Given  $S_k, V_k$ , the *majority consensus region* is the region with the most votels (breaking ties arbitrarily). Let

$$I_k(i, j) = \begin{cases} 1 & \text{if } [x_{ik}(T), y_{ik}(T)] \in C_{jk} \\ 0 & \text{otherwise} \end{cases}$$

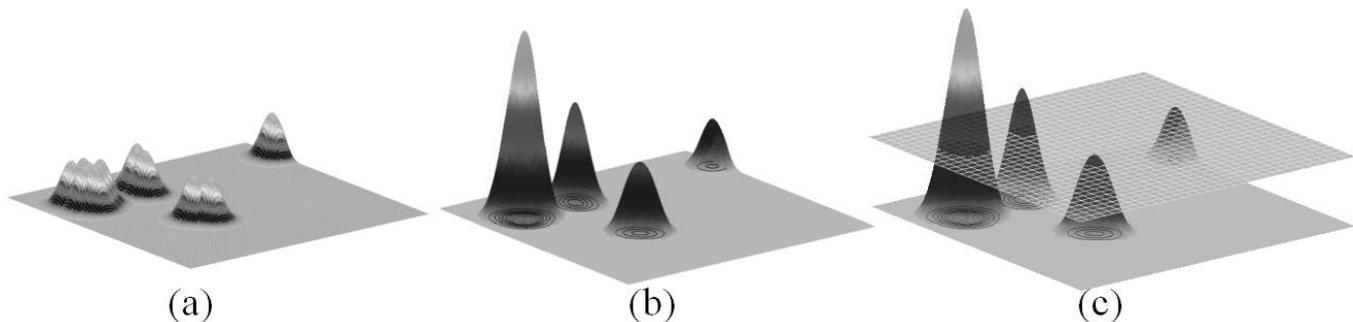


Fig. 3. An example of voter interest functions, the corresponding majority interest function, and an illustration of consensus region generation for the voting image in Figure 2.

The count

$$n_{kj} = \sum_{i=1}^n I_k(i, j)$$

is the number of votels inside consensus region  $j$  of voting image  $k$ . Breaking ties arbitrarily, let  $C_k^*$ , the majority consensus region, be any  $C_{jk}$  with max  $n_{kj}$ .

### B. Unsupervised Scoring Metric

We measure individual performance in terms of “leadership”. By definition, a “leader” anticipates the choices of the group. In our context, a leader is an individual who votes early in a position consistent with the majority consensus region. Define  $I_s$  is an outcome index for voter  $i$  and voting image  $s$ :

$$I_{s,i} = \begin{cases} 1 & \text{if } [x_{i,s}(T_s), y_{i,s}(T_s)] \in C_s^* \\ 0 & \text{otherwise} \end{cases}$$

Define  $t_{s,i}$  as the duration of the time that voter  $i$ 's votel stays in the majority interest region for the  $s^{\text{th}}$  voting image,  $T_s$  is the total voting time for voting image  $s$ . Therefore term  $\frac{T_s - t_{s,i}}{T_s} I_{s,i}$  characterizes how well the voter anticipated the majority consensus region for the voting image  $s$ .

To smooth out rapid changes in user scores, we pass the term to the following low pass filter to get a stabilized “Leadership score”:

$$L_{k+1,i} = (1 - \alpha)L_{k,i} + \alpha \frac{T_k - t_{k,i}}{T_k} I_{k,i}$$

where the initial value  $L_{0,i} = 0$  for each voter  $i$ . The value of filter factor  $\alpha$  is set to 0.1 in our experiments to allow smooth fluctuation in user scores.

Note that this scoring metric depends only on the spatio-temporal pattern of votes and does not require a human expert.

## IV. DISTRIBUTED ALGORITHM

To compute the unsupervised score for each user, we start by maintaining the ensemble interest function with a grid. We partition the voting image into  $160 \times 160$  regular cells. For each voter interest function  $f_{ik}(x, y)$ , we discretize it into a 2D array with respect to the same lattice resolution. Depending on the variance of the Gaussian function and accuracy threshold, each element of the 2D array only contains constant number of non-zero entries. Therefore, to compute the ensemble interest function for  $n$  votes, we add those  $n$  2D arrays into the cells.

This operation takes  $O(n)$ . Figure 3(b) shows the shape of the ensemble interest function for the voter interest function in 3(a).

As illustrated in figure 3(c), we can extract spatial regions by cutting the ensemble interest function using a horizontal cutting plane. The next step is to compute the height of the cutting plane  $z_k$  for the given volume ratio  $r$ . Define  $r(z)$ ,  $z > 0$  be the volume ratio between the partial volume of the majority interest function above the cutting plane with height  $z$  and the total volume of the ensemble interest function. As  $z$  increases, the horizontal plane rises. Hence  $r(z)$  is a monotonic decreasing function of  $z$ . A binary search can find  $z_k$  in  $O(\log(1/\varepsilon))$  steps with error  $\varepsilon$ . Since we need to compute the partial volume for each step, which takes at most  $O(m)$  for  $m$  cells in the grid. Therefore, the complexity for computing the threshold is  $O(\log(1/\varepsilon)m)$ . If we consider  $\varepsilon$  as a constant, then it is  $O(m)$ .

The second step is to perform the threshold for the ensemble function and find connected components, each of which forms a consensus region. The connected components algorithm makes two passes through the 2D array processing a row of cells at a time, from left to right. During the first pass, each cell is assigned the minimum label of its neighbors or if none exists, a new label is assigned. If neighboring cells have different labels, those labels are considered equivalent and entered into an equivalence table. The second pass uses the equivalence table to assign the minimum equivalent label to each non-zero entry. Since, this step also takes  $O(m)$  time, the total computation time for computing the consensus regions for a given ensemble interest function is  $O(m)$ .

To determine the majority consensus region, we need to count the number of votels inside each consensus region. For this purpose, each cell maintains a votel count that is updated during votel insertions. In a single pass, we can sum the number of votels in the cells belonging to each consensus region. Thus, this step takes  $O(m)$  time.

If we add computation time of algorithms for ensemble interest function, consensus regions, majority consensus regions, and leadership score together, the total computation time is  $O(n + m)$ .

The algorithm has been implemented and runs on the client side using Java. We tested the algorithm using a 750Mhz PC Laptop with 256Mb RAM. Figure 4 illustrates the linear

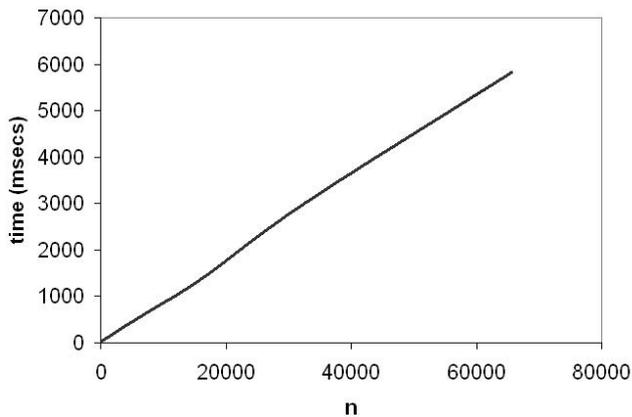


Fig. 4. Processing time for computing the unsupervised scoring metric as a function of the number of voters based on trials using random voter positions.

scalability of the algorithm in terms of number of voters. The Tele-Actor server is primarily responsible for distributing voting data to all users. Each client sends the user's voting data and receives updated voting data from the Tele-Actor server every 1.0 seconds. For every server update, at most  $n$  new voter interest functions will need to be inserted. Using the updated interest functions, the consensus regions and the majority consensus region are recomputed. When a voting cycle ends, each voter computes only the voter's new leadership score, thus distributing the scoring calculations among the clients.

## V. THE "TELE-TWISTER" APPLICATION

To understand how the unsupervised scoring metric works with groups of participants and to test software reliability, we developed a collaborative teleoperation system based on a popular party game. The result is a sequence of multi-player non-zero sum games embedded inside a sequence of two-player zero sum games.

Twister was the first board game where human bodies are the board pieces. In this classic game, human players interact over a large horizontal playing board. Players sequentially place their hands and feet on colored circular targets chosen randomly (eg, Left foot: GREEN). The challenge for players is to maintain placement of hands and feet in increasingly difficult combinations without falling over.

In our version, Tele-Twister, there are two human players called "twisters". One twister is dressed in red, the other in blue, as illustrated in figure 5. Remote participants ("voters"), download the Java applet and are assigned to one of two teams, red or blue. In Tele-Twister, random target selection is replaced by the teams, who view game status using the low framerate video and vote using the interface to collectively decide on the targets and compete to win: having their opponent fall over first.

A typical voting pattern is illustrated in Figure 5(a). Vote positions for all online voters are updated every second and displayed at each voter's browser. Consensus regions are computed and updated continuously by each browser. voters are free to change their vote position at any time during the voting cycle, but when it ends, the majority cluster determines the next move for the human twisters.

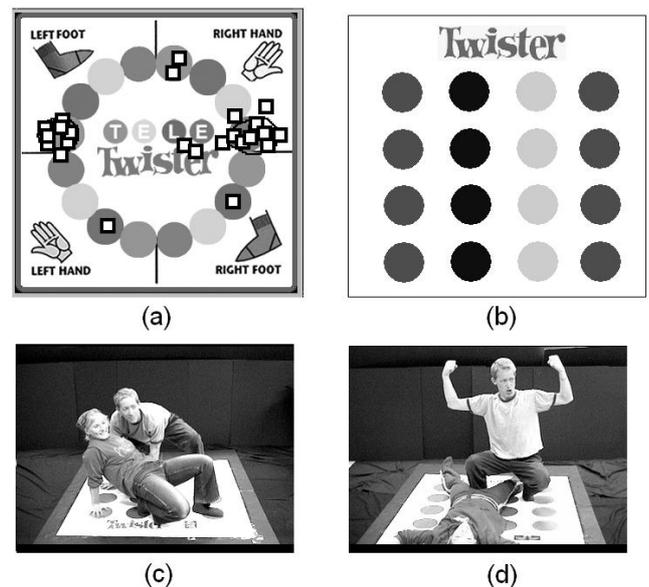


Fig. 5. Tele-Twister: a collaborative teleoperation system where online voters direct the movements of human players.

Tele-Twister is thus strategic: the red team chooses targets that are easy for the red twister and difficult for the blue twister, and vice versa for the blue team. Tele-Twister encourages active collaboration within teams and competition between teams. In this context, the unsupervised scoring metric rewards active participation and collaboration (voters who don't vote or are outside the majority region at the end of a voting cycle receive a zero score).

Figure 5(b) shows the board layout, with 16 circles: red, blue, yellow, and green. Since August 12th, 2003 we have conducted public "field tests" on Fridays, from 12-1pm Pacific Time. These field tests attract 10-25 online participants, who vote in alternating 30 second voting periods until one twister falls, ending the round. Typically, each round continues for 10-20 voting periods, so we conduct 4-5 rounds during each field test. Figure 5(c) and (d) are two snapshots from a typical round.

Figure 6 shows scoring data from the field test on Friday September 26th, 2003. All players begin the field test with a score of zero, so average score per team climbs during the initial voting cycles: the blue team wins the first round after approx 23 voting cycles.

How do user scores correlate with task performance (winning the round)? Note that in the four subsequent rounds, the team with the highest average score consistently wins the round: the red team wins rounds 2,3 and 5 and the blue team wins round 4. During round 4, members of the red team had difficulty agreeing on the appropriate next move, reducing the score of many red voters and in turn the average red team score. The lack of consensus (ie. "split votes") resulted in a loss during that round.

A team has higher average scores when the team collaborates, reaching consensus faster. This does not always correlate with success: it can lead to short-term snap decisions that may appear strong but are strategically weak.

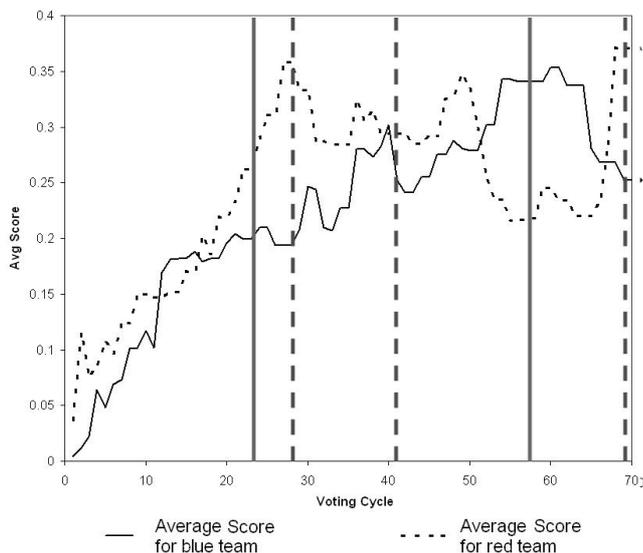


Fig. 6. Plot of unsupervised scoring metric from the Sept 26 2003 field test with two teams of 21 online voters, for five rounds of the Tele-Twister game. The figure plots average score for members of each team during sequential voting cycles. Vertical bars indicate the end of a round and which team wins. A solid vertical line indicates that the blue team won that round and a dashed vertical line indicates that the red team won.

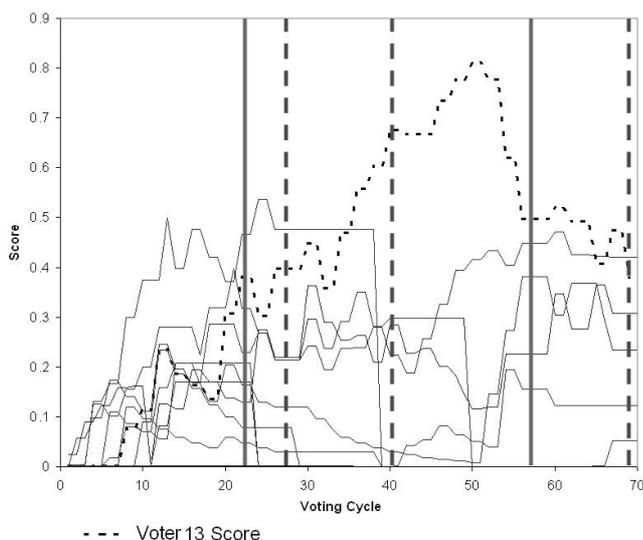


Fig. 7. From the same Sept 26 2003 field test, plot of unsupervised scoring metric for seven individual voters from the Blue team.

Figure 7 plots individual voter scores from the same field test. Note that the score of voter 13 is consistently higher. In this case voter 13 is a member of our lab who has played the game during many previous rounds and has developed skill at picking the next moves. Other players follow his moves, resulting in a high score.

More experiments will be conducted in next few months. We will report these results in the revision of this paper if accepted. To experiment with the interface, please visit: [www.tele-actor.net](http://www.tele-actor.net)

## VI. FUTURE WORK

This paper describes an unsupervised scoring metric for collaborative teleoperation that encourages active participation and collaboration, and a distributed algorithm for automatically computing it. To understand how the scoring metric works with groups of participants and to test software reliability, we developed a collaborative teleoperation system based on a sequence of multi-player non-zero sum games embedded inside a sequence of two-player zero sum games. Initial results suggest that the metric encourages active participation and correlates reasonably with task performance.

## ACKNOWLEDGMENTS

Thanks to T. Shlain, E. Paulos and D. Pescovitz for valuable input on initial experiments, the students who have participated in the project: J. Wang, K. "Gopal" Gopalakrishnan, our sponsors the National Science Foundation and Intel Corporation, and to research colleagues: R. Bajcsy, J. Canny, P. Wright, G. Niemeyer, R. Siegwart, D. Greenbaum, and T. Shlain.

## REFERENCES

- [1] R.J. Aumann and J.H. Dreze. Cooperative games with coalition structures. *International Journal of Game Theory*, 3:217–237, 1974.
- [2] R.J. Aumann and M. Maschler. The bargaining set of cooperative games. pages 443–476, 1964.
- [3] 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.
- [4] 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.
- [5] P. Coppin and W. Whittaker et al. Eventscope: Amplifying human knowledge and experience via intelligent robotic systems and information interaction. In *9th IEEE International Workshop on Robot and Human Interactive Communication (Ro-man)*, 2000.
- [6] C. H. Crouch and E. Mazur. Peer instruction: Ten years of experience and results. *American Journal of Physics*, 69(9):970–977, September 2001.
- [7] Mihaly Csikszentmihalyi. *Beyond Boredom and Anxiety*. Jossey-Bass Publishers, 1982.
- [8] 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.
- [9] I. Elhaji, J. Tan, N. Xi, W.K. Fung, Y.H. Liu, T. Kaga, Y. Hasegawa, and T. Fukuda. Multi-site internet-based tele-cooperation. *Integrated Computer-Aided Engineering*, 9(2):117–27, 2002.
- [10] I. Erev and A. E. Roth. Predicting how people play games: Reinforcement learning in experimental games with unique, mixed strategy equilibria. *American Economic Review*, 88(4):848–81, September 1998.
- [11] James Paul Gee. High score education. *Wired*, 11(5), May 2003.
- [12] Erving Goffman. *Strategic Interaction*. University of Pennsylvania Press, 1969.
- [13] 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.
- [14] 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.
- [15] K. Goldberg and R. Siegwart, editors. *Beyond Webcams: An Introduction to Online Robots*. MIT Press, 2002.
- [16] 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.
- [17] 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.

- [18] 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.
- [19] W. Hartmann and B. Rollett. Positive interaction in the elementary school. pages 195–202, 1994.
- [20] H. Hu, L. Yu, P. W. Tsui, and Q. Zhou. Internet-based robotic systems for teleoperation. *Assembly Automation*, 21(2):143–151, May 2001.
- [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] 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.
- [23] Bernd Kreimeier.
- [24] 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.
- [25] R. Marimon and E. McGrattan. On adaptive learning in strategic games. 1995.
- [26] R. Marin, P.J. Sanz, and J.S. Sanchez. Object recognition and incremental learning algorithms for a web-based telerobotic system. In *IEEE International Conference on Consumer Electronics*, 2002.
- [27] R. Marin, P.J. Sanz, and J.S. Sanchez. A very high level interface to teleoperate a robot via web including augmented reality. In *IEEE International Conference on Consumer Electronics*, 2002.
- [28] A. Mas-Collell. An equivalence theorem for a bargaining set. *Journal of Math and Economics*, (18):129–139, 1989.
- [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] B. Rogoff, E. Matusov, and C. White. *Models of teaching and learning: Participation in a community of learners*. Oxford, England: Blackwell, 1996.
- [33] R. Safaric, M. Debevc, R. Parkin, and S. Uran. Telerobotics experiments via internet. *IEEE Transactions on Industrial Electronics*, 48(2):424–31, April 2001.
- [34] P. Shenoy. On coalition formation: A game-theoretical approach. *International Journal of Game Theory*, 8:133–164, 1979.
- [35] Thomas B. Sheridan. *Telerobotics, Automation, and Human Supervisory Control*. MIT Press, 1992.
- [36] D. Song and K. Goldberg. Sharecam part I: Interface, system architecture, and implementation of a collaboratively controlled robotic webcam. In *IEEE/RSJ International Conference on Intelligent Robots (IROS)*, Nov. 2003.
- [37] 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.
- [38] Brian Sutton-Smith. *The Ambiguity of Play*. Harvard University Press, 1997.
- [39] J. von Neumann and O. Morgenstern. *The Theory of Games and Economic Behavior*. Princeton University Press, 1953.