

# Cooperating Robots for Search and Rescue

Jijun Wang  
School of Information Sciences  
University of Pittsburgh  
136 N. Bellefield Ave.  
Pittsburgh, PA 15260  
412-624-9426  
Jiw1+@pitt.edu

Michael Lewis  
School of Information Sciences  
University of Pittsburgh  
136 N. Bellefield Ave.  
Pittsburgh, PA 15260  
412-624-9426  
ml@sis.pitt.edu

Paul Scerri  
Robotics Institute  
Carnegie Mellon University  
5000 Forbes Ave.  
Pittsburgh, PA 15213  
(412) 268-2145  
pscerri@cs.cmu.edu

## ABSTRACT

Many hypothesized applications of mobile robotics require multiple robots. Multiple robots substantially increase the complexity of the operator's task because attention must be continually shifted among robots. One approach to increasing human capacity for control is to remove the independence among robots by allowing them to cooperate. This paper presents an initial experiment using multiagent teamwork proxies to help control robots performing a search and rescue task.

## Categories and Subject Descriptors

D.1.7 : Computers in Other Systems

## General Terms

Multiagent Systems, Experimentation, Human Factors

## Keywords

Multiagent Systems, Multirobot Systems, Human-Robot Interaction.

## 1. INTRODUCTION

Many hypothesized applications of mobile robotics require multiple robots. Envisioned applications such as interplanetary construction [4] or cooperating uninhabited aerial vehicles [8] will require close coordination and control between human operator(s) and cooperating teams of robots in uncertain environments. Multiple robots substantially increase the complexity of the operator's task because she must continually shift attention among robots under her control, maintain situation awareness for both the team and individual robots, and exert control over a complex system. In the simplest case an operator controls multiple independent robots interacting with each as needed. Control performance at this task has been investigated both in terms of average demand on human attention [1] and for simultaneous demands from multiple robots that lead to bottlenecks [5]. In these approaches increasing robot autonomy allows robots to be neglected for longer periods of time making it

possible for a single operator to control more robots. Providing additional autonomy by enabling robots to cooperate among themselves extends automation to human control activities previously needed to coordinate the robots' actions. Automating this function should decrease the demands on the human operator to the extent that attention being devoted to a robot involved coordination with other robots. If substantial efforts were required for coordination automation should allow improvements in performance or control of larger teams.

## 1.1 Teamwork Algorithm

The teamwork algorithms used to coordinate the simulated robots are general algorithms that have been shown to be effective in a range of domains [10]. To take advantage of this generality, the emerging standard approach is to encapsulate the algorithms in a reusable software *proxy*. Each team member has a proxy with which it works closely, while the proxies work together to implement the teamwork. The current version of the proxies is called *Machinetta* [8] and extends the successful Teamcore proxies [7]. *Machinetta* is implemented in Java and is freely available on the web. Notice that the concept of a reusable proxy differs from many other "multiagent toolkits" in that it provides the coordination *algorithms*, e.g., algorithms for allocating tasks, as opposed to the *infrastructure*, e.g., APIs for reliable communication.

The *Machinetta* software consists of five main modules, three of which are domain independent and two of which are tailored for specific domains. The three domain independent modules are for coordination reasoning, maintaining local beliefs (state) and adjustable autonomy. The domain specific modules are for communication between proxies and communication between a proxy and a team member. The modules interact with each other only via the local state with a blackboard design and are designed to be "plug and play", thus, e.g., new adjustable autonomy algorithms can be used with existing coordination algorithms.

The coordination reasoning is responsible for reasoning about interactions with other proxies, thus implementing the coordination algorithms. The adjustable autonomy algorithms reason about the interaction with the team member, providing the possibility for the team member to make any coordination decision instead of the proxy. For example, the adjustable autonomy module can reason that a decision to accept a role to rescue a civilian from a burning building should be made by the human who will go into the building rather than the proxy. In

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. To copy otherwise, or republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee.

AAAMAS '06, May 8-12, 2004, Future University, Hakodate, Japan.  
Copyright 2004 ACM 1-58113-000-0/00/0004...\$5.00.

practice, the overwhelming majority of coordination decisions are made by the proxy, with only key decisions referred to human operators.

Teams of proxies implement *team oriented plans* (TOPs) which describe joint activities to be performed in terms of the individual *roles* to be performed and any constraints between those roles. Typically, TOPs are instantiated dynamically from TOP templates at runtime when preconditions associated with the templates are filled. Typically, a large team will be simultaneously executing many TOPs. For example, a disaster response team might be executing multiple fight fire TOPs. Such fight fire TOPs might specify a breakdown of fighting a fire into activities such as checking for civilians, ensuring power and gas is turned off and spraying water. Constraints between these roles will specify interactions such as required execution ordering and whether one role can be performed if another is not currently being performed. Notice that TOPs do not specify the coordination or communication required to execute a plan, the proxy determines the coordination that should be performed.

Current versions of Machinetta include state-of-the-art algorithms for plan instantiation, role allocation, information sharing, task deconfliction and adjustable autonomy. Many of these algorithms utilize a logical *associates network* statically connecting all the team members. The associates network is a *scale free network* which allows the team to balance the complexity of needing to know about all the team and maintaining cohesion. Using the associates network key algorithms, including role allocation, resource allocation, information sharing and plan instantiation are based on the use of *tokens* which are "pushed" onto the network and routed to where they are required by the proxies. For example, the role allocation algorithm, LA-DCOP [9], represents each role to be allocated with a token and pushes the tokens around the network until a sufficiently capable and available team member is found to execute the role. The implementation of the coordination algorithms uses the abstraction of a simple mobile agent to implement the tokens, leading to robust and efficient software.

We have recently integrated Machinetta [2] with the USARsim simulation to provide a testbed for studying human control over cooperating teams of robots. This paper reports our first tests of the system and does not yet fully exploit the richness and complexity of coordination that are available.

## 1.2 Experimental Task

In this experiment, participants were asked to control 3 simulated P2DX robots (Figure 1) to search for victims in a damaged building. Each robot was equipped with a pan-tilt camera with a fixed 45 degrees of FOV and a front laser scanner with 180 degree FOV and resolution of 1 degree. The participant interacted with the robots through our Robots Control System (RCS). Status information, camera video, laser scanning range data, and a global map built from that data were available from each robot. The participant controlled the robot to explore the building and search for victims by issuing waypoints or teleoperating the robot and panning/tilting the camera. Once a victim was identified, the participant marked its location on the global map.

Challenges to mobility encountered in real robotic search and rescue tasks were simulated in our experiment by obstacles including chairs, bricks, and pipes. Transparent sheets of plastic and mirrors were introduced to cause perceptual confusion and increase task difficulty. The camera's FOV was restricted to 45 degrees to reflect typical limitations. As with real robotic system, there are uncertainties and delays in our RCS. Range data had simulated errors, the map was based on probabilistic data and some obstacles such as a chair or desk might be lost on the map because of inaccuracies in laser detection. Walls especially thin ones were also subject to loss due to errors in range data. There are also slight delays in video feedback and response to commands.



Figure 1. P2DX Robot

The RCS could work in either auto or manual mode. Under auto mode, the robots could cooperate in a limited way to automatically explore the environment. In manual mode, the robots had no automatic exploration capabilities and stopped after completing their commands. The experiment followed a repeated measures design in which participants controlled in both manual and auto modes. Order of presentation was counterbalanced and participants explored the same sequence of environments. The robots' location, orientation and the users' actions were recorded and timestamped throughout the experiment. The final map with marked victims was also saved. Demographic information and posttest survey were also collected.

### 1.3 The Robot and Environment Simulation

In this experiment, we used USARSim [11], a high-fidelity simulation of urban search and rescue (USAR) robots and environments. USARSim supports human-robot interaction (HRI) by accurately rendering user interface elements (particularly camera video), accurately representing robot automation and behavior, and accurately representing the remote environment that

links the operator's awareness with the robot's behaviors. It was built based on a multi-player game engine, UnrealEngine2, and so is well suited for simulating multiple robots.

USARSim uses the Karma Physics engine to provide physics modeling, rigid-body dynamics with constraints and collision detection. It uses other game engine capabilities to simulate sensors including camera video, sonar, and laser range finder.

The experiment uses USARsim's model of the NIST Yellow Arena [3]. The victims are evenly distributed within the arena and may appear as partial or whole human bodies. Victims were designed and placed to make the difficulty of finding them roughly the same. Two similar arenas (Figure 2) are used in the experiment. The two arenas were constructed from the same elements but with different arrangements.

### 1.4 The Robot and Environment Simulation



a) Arena 1

b) Arena 2

Figure 2. The Arenas.

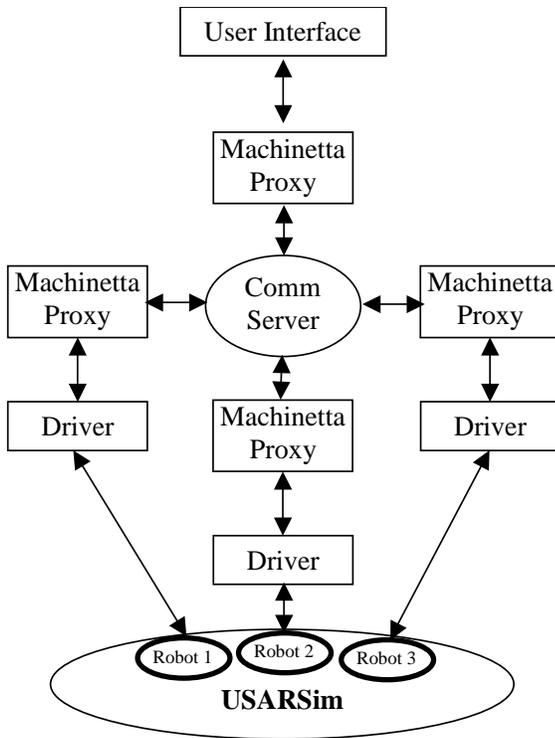


Figure 3. System Architecture.

The Robots Control System is based on Machinetta [2], a multiagent system based on teamwork proxies. The system’s architecture is shown in Figure 3. Each virtual robot connects with Machinetta through a robot driver. The driver parses the robot’s sensor data and transfers them to the Machinetta proxy. It also has limited low-level autonomy to interpret the proxy’s plan as robot commands; control the robot to avoid obstacles; and recover the robot when stuck. The user interface is connected to Machinetta as well to create a RAP (Robot, Agent and Person) system. There are embedded cooperation algorithms in Machinetta that can coordinate the robots and people through the Comm Server that exchanges information among the Machinetta proxies.

When the system works in manual mode, cooperation among the robots eliminated. When it runs in auto mode, the robot proxy is allowed to analyze the range data to determine what nodes the robot team needs to explore and how to reach those nodes from the current position (generating the paths). By exchanging these nodes and route information through Machinetta, a robot proxy can accept and execute a plan to visit a node by following a path (a series of waypoints).

Through the user interface, the operator can also directly control the robots’ cameras, teleoperate them or issue waypoints to the robots. Robots are controlled one at a time with the selected robot providing a full range of data while the unselected ones provide camera views for monitoring. On the user interface (figure 3), each robot is represented by a unique color. The control component’ background color is set to the currently selected robot’s color to help the users identify which the robot they are controlling. The components of the interface are:

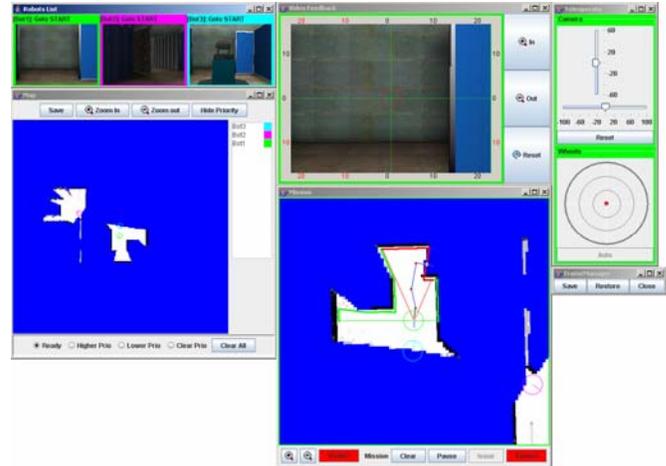


Figure 4. The Robots Control System.

- **Robots List (the upper left component)**

The Robots List was designed to help the user monitor the robots. It lists the robots with their names, states, camera video and colors. It is also used to select the controlled robot. Camera video for this component is updated at a low frame rate.

- **Map (left bottom component)**

This component displays the global map created by the robots. It is intended to help the user maintain situational awareness. On this component, blue indicates unexplored areas; white shows an unoccupied area that has been explored and black shows obstacles within an explored area. Areas with gray color may or may not contain objects. Dark gray indicates that an area contains an object with high probability.

- **Video Feedback (upper center component)**

The currently selected robot’s video is displayed on this component. The picture is updated frame by frame with high frequency. The camera’s pan and tilt angles are represented by the crosshair on the video. The ‘reset’ button re-centers the camera. The ‘zoom’ feature was disabled for this experiment to provide a fixed FOV.

- **Teleoperation (upper right component)**

This component includes two sub-panels. The “Camera” panel is used to pan, tilt or center the camera. The “Wheels” panel is a simulated joystick that controls the robot’s movement. When the user uses the joystick, the robot will automatically clear its exploring path and enter teleoperation mode. In the auto condition after the user finishes teleoperating, the robot will return to auto mode and attempt to generate a new path, in the manual mode the robot remains stopped. A teleoperation episode is terminated when the user clicks the “Auto” button or 6 seconds has passed without operator input.

- **Mission (bottom center component)**

This component displays the current exploration situation on a “you-are-here” style map. The upper direction of the map is always the camera’s direction. The range data is displayed as bold green line overlaid on the map. The red cone emitted from the

**Table 1 Sample Demographics**

	Age		Gender		Education	
	19	20~35	Male	Female	Currently Undergraduate	Complete Undergraduate
<b>Order 1</b>	1	6	1	6	4	3
<b>Order 2</b>	1	6	4	3	6	1
<b>Total</b>	2	12	5	9	10	4

**Table 2 Participants Experience**

	Computer Usage (hours/week)				Game Playing (hours/week)				Mouse Usage for Game Playing		
	<1	1-5	5-10	>10	<1	1-5	5-10	>10	Frequently	Occasionally	Never
Order 1	0	2	1	4	3	4	0	0	6	1	0
Order 2	0	0	6	1	3	3	1	0	2	5	0
Total	0	2	7	5	6	7	1	0	8	6	0

robot marks the area shown in the video feedback. Combining the cone with video feedback can provide the user with better situation awareness and sense of distances. The path the robot is trying to follow is also shown on the map. With this component, the user can create a new path by issuing a series of waypoints, modify the current path by moving waypoints, or mark a victim on the map. When the user begins to mark a victim, the robot pauses its action until the user finishes the mark operation.

**1.5 Procedure**

This experiment compared robot team control performance under auto and manual modes. Participant demographics were collected at the start of the experiment using an on-screen questionnaire. Standard instructions explaining how to use the interface were followed by a ten minute practice session in which participants following instructions practiced each of the operations available in the two modes and finished after searching for and finding a victim in auto mode. Order of presentation was counterbalanced with half of the participants assigned to search for victims in Arena-1 in auto mode and the other half in manual. After 20 minutes the trial was stopped. Participants were given brief instructions reminding them of significant features of the mode they had not used and then began a second 20 minute trial in Arena-2. At the conclusion of the experiment participants completed an online survey.

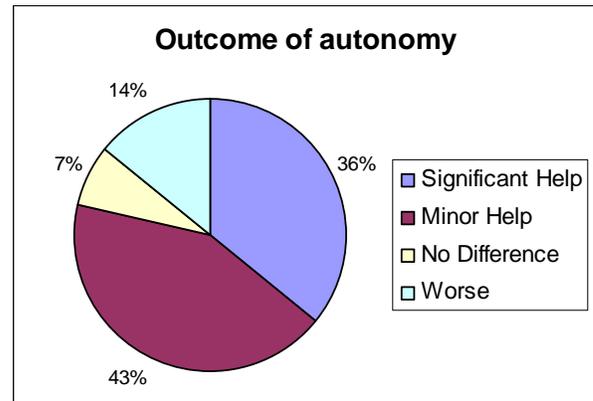
14 paid participants recruited from the University of Pittsburgh community took part in the experiment. The participants' demographic information and experience are summarized in tables 1 and 2.

**2. Results**

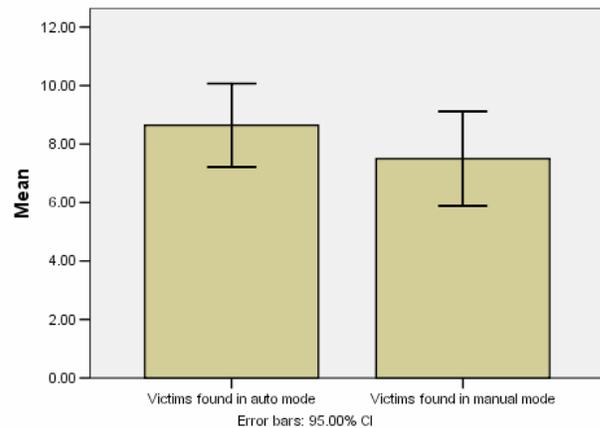
**2.1 Overall Measures**

*2.1.1 Subjective Measures*

Participants were asked to rate to what extent autonomy helped them find victims. The results show that most participants (79%) rated autonomy as providing either significant or minor help.



**Figure 5. Outcome of autonomy**



**Figure 6. Victims found by participants.**

Only 1 of the 14 participants (7%) rated autonomy as making no difference and 2 of the 14 participants (14%) judged autonomy to make things worse.

## 2.1.2 Performance Measures

### 2.1.2.1 Victims

Comparing the victims found by the same participant under auto mode and the victims found under manual mode using a one tail paired t test, we found that participants found significantly more victims in auto mode than in manual mode ( $p=0.044$ ) (Figure 6).

### 2.1.2.2 Explored Ratio

The explored ratio is the percentage of the area scanned by the robots. A one tail paired t-test was used to compare auto and manual modes. Participants were found to explore wider areas under auto mode than in manual mode ( $p=0.002$ ).

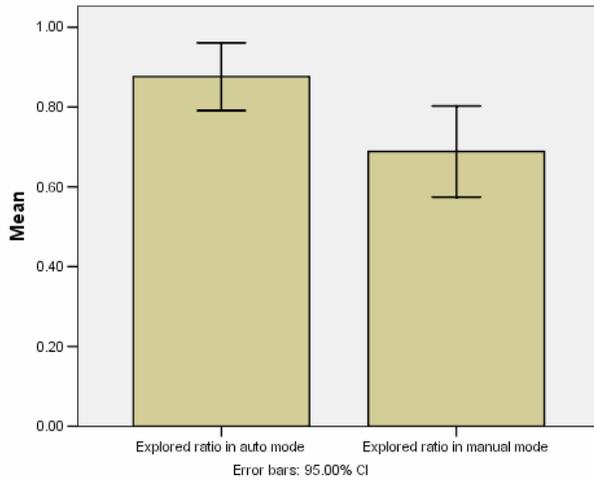


Figure 7. Explored Ratio.

## 2.2 Distribution of Attention among Robots

Measuring the distribution of attention among robots as the standard deviation of the total time spent with each robot no difference ( $p=0.232$ ) was found between auto and manual modes. However, we found that under auto mode, the same participant switched robots significantly more frequently than under manual mode ( $p=0.027$ ). The posttest survey showed that most

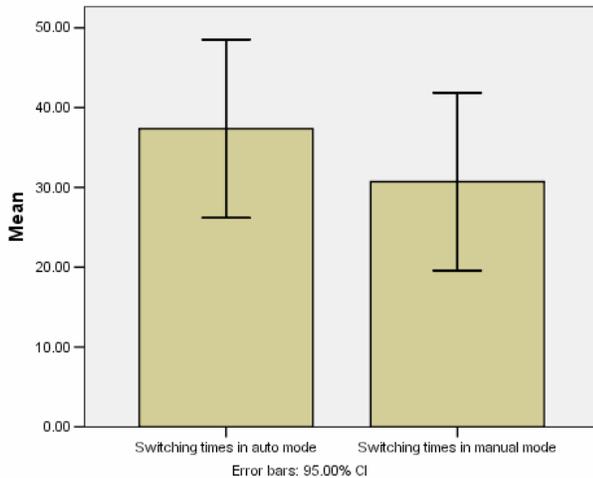


Figure 8. Switching Times.

participants switched robots based on the Robots List component. Only 2 of the 14 participants (14%) reported switching robot control independent of this component.

We also found that switches in control among robots led to finding more victims. Figure 9 shows the regression of victims found on the number of switches in attention among the robots ( $R^2=0.477$   $p=0.006$ ).

## 2.3 Forms of Control

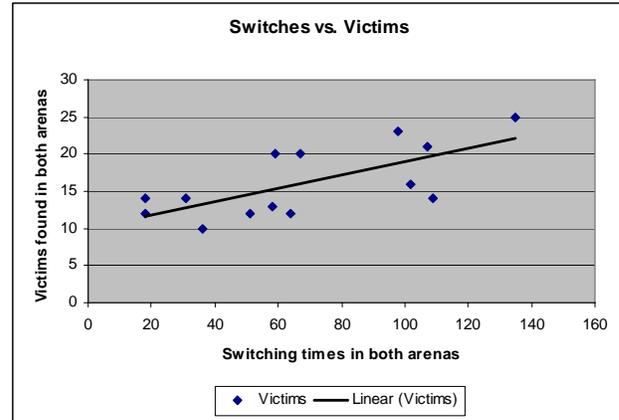


Figure 9. Switches vs. Victims.

Participants had three forms of control to locate victims: waypoints, teleoperation, and camera control. No difference was found between auto and manual modes in the use of these forms of control. However, in the auto mode, participants were less likely to control waypoints ( $p=0.004$ ) or teloperate ( $p=0.046$ ) during any single control episode.

Comparing the victims found with control operations (waypoints and teleoperation), we found an inverted U relationship between control operations and the victims found (figure 12). Too few or too much movement control led to fewer found victims.

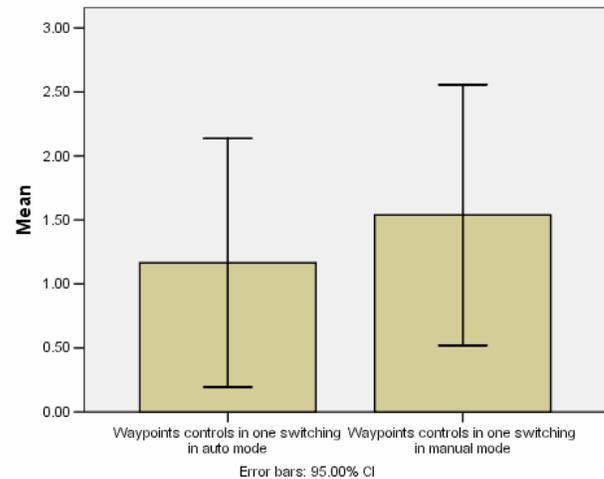


Figure 10. Waypoints controls in one switching.

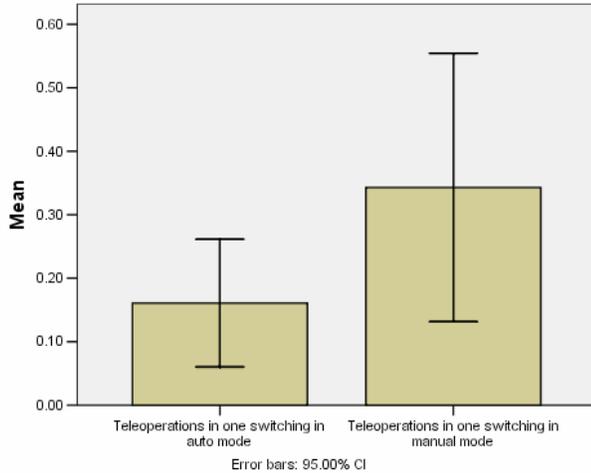


Figure 11. Teleoperations in one switching.

## 2.4 Trust and Capability of Using Interface

In the posttest we collected participants ratings of their level of trust in the system's automation and their ability to use the interface to control the robots. 43% of the participants trusted the autonomy and only changed the robot's plans when they had

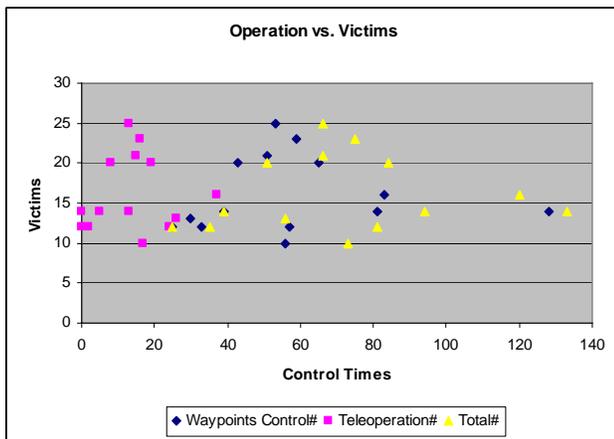


Figure 12. Robot Controls vs. Victims.

spare time. 36% of the participants reported changing about half of the robot's plans while 21% of the participants showed less trust and changed the robot's plans more often. A one tail t-test, indicates that the total victims found by participants trusting the autonomy is larger than the number victims found by other participants ( $p=0.05$ ). 42% of the participants reported being able to use the interface well or very well while 58% of the participants reported having difficulty using the full range of features while maintaining control of the robots. A one tail t test shows that participants reporting using the interface well or very well found more victims ( $p<0.001$ ) based on a one tail t-test. Participants trusting the autonomy reported significantly higher capability in using the user interface ( $p=0.001$ ) and conversely participants reporting using the interface well also had greater trust in the autonomy ( $p=0.032$ ).

## 3. Discussion

This experiment is the first of a series investigating control of cooperating teams of robots using Machinetta. In this experiment cooperation was extremely limited primarily involving the deconflicting of plans so that robots did not explore or re-explore the same regions. The presence of simple path planning capabilities and limited autonomy in addition to coordination in the *auto* condition prevents us from attributing our results solely to the presence of a coordination mechanism. In future experiments we intend to extend the range of coordination to include heterogeneity in sensors, mobility, and resources such as battery power to provide richer opportunities for cooperation and the ability to contrast multirobot coordination with simple automation.

Although only half of the participants reported trusting the autonomy or being able to use the interface well, the results showed that autonomy helped the operators explore more areas and find more victims. In both the conditions participants divided their attention approximately equally among the robots but in the auto mode they switched among robots more rapidly thereby getting more detailed information about different areas of the arena being explored.

The frequency of this sampling among robots was strongly correlated with the number of victims found. This effect, however, cannot be attributed to a change from a control to a monitoring task because the time devoted to control was approximately equal in the two conditions. We believe instead that searching for victims in a building can be divided into a series of subtasks involving things such as moving a robot from one point to another, and/or turning a robot from one direction to another with or without panning or tilting the camera. To effectively finish the searching task, we must interact with these subtasks within their neglect time[6] that is proportional to the speed of movement. When we control multiple robots and every robot is moving, there are many subtasks whose neglect time is usually short. Missing a subtask means we failed to observe a region that might contain a victim. So switching robot control more often gives us more opportunity to find and finish subtasks and therefore helps us find more victims. This focus on subtasks extends to our results for movement control which suggest there may be some optimal balance between monitoring and control. If this is the case it may be possible to improve an operator's performance through training or online monitoring and advice.

## 4. ACKNOWLEDGMENTS

This project is supported by NSF grant NSF-ITR-0205526.

## 5. REFERENCES

- [1] Crandall, J. and M. Goodrich. Characterizing Efficiency of Human Robot Interaction: A Case Study of Shared-Control Teleoperation. in proceedings of the 2002 IEEE/RSJ International Conference on Intelligent Robots and Systems. 2002.
- [2] Farinelli, A., P. Scerri, and T. M. Building large-scale robot systems: Distributed role assignment in dynamic, uncertain domains. in AAMAS'03 Workshop on Resources, role and task allocation in multiagent systems. 2003.

- [3] Jacoff, A., Messina, E., Evans, J. Experiences in deploying test arenas for autonomous mobile robots. in Proceedings of the 2001 Performance Metrics for Intelligent Systems (PerMIS) Workshop. 2001. Mexico City, Mexico.
- [4] Kingsley, F., R. Madhavan, and L.E. Parker. Incremental Multiagent Robotic Mapping of Outdoor Terrains. in Proceedings of the 2002 IEEE International Conference on Robotics & Automation. 2002.
- [5] Nickerson, J.V. and S.S. Skiena. Attention and Communication: Decision Scenarios for Teleoperating Robots. in Proceedings of the 38th Annual Hawaii International Conference on System Sciences. 2005.
- [6] Olsen, D. and M. Goodrich. Metrics for evaluating human-robot interactions. in Proc. NIST Performance Metrics for Intelligent Systems. 2003.
- [7] Pynadath, D.V. and Tambe, M., An Automated Teamwork Infrastructure for Heterogeneous Software Agents and Humans. *Journal of Autonomous Agents and Multi-Agent Systems*, 7, 2003, 71-100.
- [8] Scerri, P., et al., Coordinating large groups of wide area search munitions, in *Recent Developments in Cooperative Control and Optimization*, D. Grundel, R. Murphey, and P. Pandalos, Editors. 2004, Singapore: World Scientific. p. 451-480.
- [9] Scerri, P.; Farinelli, A.; Okamoto, S.; and Tambe, M.. Allocating tasks in extreme teams. In *Proc. of the fourth international joint conference on Autonomous agents and multiagent systems*, 2005.
- [10] Tambe, M., Towards Flexible Teamwork. *Journal of Artificial Intelligence Research*, 1997. 7: p. 83-124.
- [11] Wang, J., Lewis. L., Gennari J. A Game Engine Based Simulation of the NIST Urban Search & Rescue Arenas. in *Proceedings of the 2003 Winter Simulation Conference*. 2003. New Orleans.