

AGENT-BASED SUPPORT FOR BALANCING TELEOPERATION AND AUTONOMY IN URBAN SEARCH AND RESCUE

Ryan Wegner and John Anderson
Autonomous Agents Laboratory
Department of Computer Science, University of Manitoba
Winnipeg, Manitoba, R3T 2N2, Canada
rwegner, andersj@cs.umanitoba.ca

Abstract

Today's artificial intelligence technology is insufficient to support autonomous task performance in complex domains such as urban search and rescue, forcing human teleoperation of robots to be extensively relied upon. This, however, also has problems: humans quickly suffer from cognitive overload, and have difficulties in constructing a representation of the space around a remotely-placed robot. In this paper we describe an approach to multi-robot control for such environments that focuses on combining the limited abilities of modern autonomous control systems together with human control. At the center of this approach is a pair of software agents running on each robot: one to recognize problems in the environment from the perspective of a robot, and another to mediate the interaction between a robot and a human controller. The intent of this approach is to allow a human to better control a team of robots, being interrupted only when the situation demands. We describe the implementation of this approach using simulated Pioneer robots, and evaluate the approach in comparison to autonomous and teleoperated mobile robots in a rescue domain.

Key Words: Urban Search and Rescue, Robotic Rescue, Teleautonomy, Agents, Multi-Agent Systems

1. Introduction

Urban search and rescue (USAR), the exploration of damaged or collapsed urban structures in search of disaster victims, is both a major potential application of AI technology and a current challenge problem for researchers in AI and robotics. USAR is an extremely difficult task for an autonomous mobile robot to perform adequately given the current state of the art in artificial intelligence. The environment is difficult to navigate, and unpredictable in that even a known building layout may have changed dramatically during the associated disaster.

Basic robotics problems such as localization are strongly affected (e.g. constant odometry errors from driving over unstable debris), and sensing is much less accurate than the common indoor domains in which today's robotic technology is mostly associated. The wide range of necessary skills, coupled with the unpredictability of the domain lead most existing efforts to rely heavily on human teleoperation of robotic units (including those whose use at the World Trade Center was widely publicized [1, 2]).

This reliance on teleoperation can also be seen in current research. USAR research is evaluated in controlled conditions using a physical testbed (e.g. the NIST testbed [3]), where the goal is to provide a map to the locations of human victims within an area of debris representing a collapsed structure. While these domains have been described as simplistic compared to real-world USAR [4], the vast majority of entries to such competitions are teleoperated. For example, at both AAAI-02 in Edmonton [5] and IJCAI-03 in Acapulco [6] our laboratory's team was one of only two entries running fully autonomously.

A strong reliance on teleoperation causes problems in practice. Casper and Murphy, for example, describe the operator fatigue that occurs very quickly in real-world rescue situations, and the associated errors in both control and in recognizing visual cues [1, 2]. There are also significant problems with providing situational awareness (that is, a functional mental view of the space within which a robot is operating) to a human operator, and teleoperators also suffer from cognitive overload in terms of processing information [7], limiting the amount of information that can be presented and the number of robots that can be controlled.

We are interested in providing functional intelligent control to a team of mobile robots. Given the state of the art in artificial intelligence, the logical application of current technology is to serve as an augmentation to human teleoperation, to allow agents to be autonomous to the degree the situation allows and to endeavor to minimize the problems associated with teleoperation. This combination, known as a *teleautonomous* approach, has been employed before in robotic domains, generally at a fairly coarse level (e.g. [8, 9, 10, 11, 12, 13]). However, because of the complexity, danger, and urgency of this domain, a seamless blend of teleoperation and autonomy is required, allowing a teleoperator to focus attention on the problems that require the most assistance at any time.

We have developed an approach to blending teleoperation and autonomous control in behaviour-based mobile robots [14] for USAR environments. This consists of three sets of facilities. The lower two of these are (1) a schema-based [15] autonomous control system for navigation and mapping that allows robots to perform autonomously (subject to the associated limitations of the USAR domain), and (2) support for direct teleoperation of robots, including an interface for low-level movements as the ability to exert control at a higher level by setting waypoints.

This paper describes the third and most sophisticated of these facilities, a system for blending autonomy and teleoperation appropriately for individual robots on a team. These facilities consist of two agent-based components: one for blending control, and the other for recognizing situations in which human intervention is likely useful. We describe these facilities and their implementation, and describe an evaluation of the performance of these facilities on simulated Pioneer robots. Before all this, we begin with a brief review of related literature.

2. Related Literature

The most well-known early work in combining teleoperation and autonomy is that of Arkin and Ali [7], who describe two approaches for teleautonomy with respect to multiple robots. Both of these are schema-based [15] approaches, where behaviours (wander, avoid, etc.) are encoded in the form of motor schemas, which are activated by perceptual schemas (defining perceptual items of interest) and interact at run time to produce output to robot effectors. The first approach has the operator’s control (input from a joystick) as a behaviour that influences the robots’ effectors just as any internal behaviour does. The second approach for teleautonomy involves giving the operator access to the behavioural parameters of the society (e.g. the low level gains of each motor schema in all agents). This work is limited by its simple blending and effect on an entire group of agents at once, but showed the utility of a teleautonomous approach. Blending an operator’s desires as a schema along with the robot’s desires was also implemented in the control of a teleautonomous Hummer [16], but shares the same limitations of Arkin and Ali’s original approach.

Crandall et al. [11] present the notion of *neglect* in teleoperated robots. They describe neglect as the amount of time during which the robot is not receiving some sort of instruction. They show that this down-time, which can be due to a lack of attention from the operator or delays in command reception, hinders performance. They also describe a control system consisting of a set of robot behaviours and a user interface for controlling the robots, using five levels of autonomy ranging from fully autonomous to dormant. However, they do not describe an implementation in their work to show that any balancing has been implemented beyond these crude layers.

Trividi et al. [13] designed a system intended to allow robotic units to recognize traffic collisions. This system is strictly a laboratory design, but makes use of teleautonomous robots that can form a perimeter around a collision. These robots specialize in forming a perimeter, and the remote operation provides very basic instructions to guide the robots to form perimeters around specific areas. This application of teleautonomy demonstrates the potential to have equipment constantly monitoring an area without the full attention of an operator, but is extremely simplistic: the robots have one purpose, and can achieve that fairly simply through a polygon-forming algorithm where each robot occupies one point on the polygon. The operator supplies only location guidelines for the polygon forming activity.

Murphy and Sprouse [10] describe a strategy for mixing robot and human control in the USAR domain by assigning a different search task to the operator than to an autonomous robot. The robot would perform a systematic search of an area, covering the entire area by splitting the area into sections and working on each section. The operator then performed the semantic search; in this case the operator directed the robot to semantically similar areas of interest. Murphy et al. [17] describe a paradigm for automating victim detection by a mobile robot, while the operators controlled the robot’s navigational system. They implement their strategy on a three-unit society architecture, where the robot, human and an Intelligent Assistant Agent together composed the society.

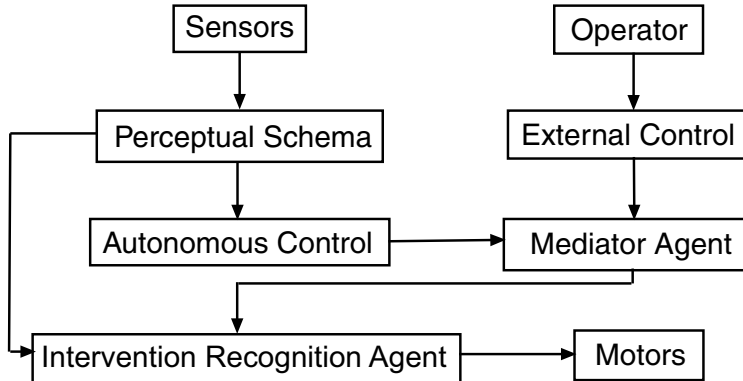


Figure 1: Overview.

3. Design and Implementation

In our approach to blended teleautonomy, robotic agents are implemented using a schema-based [16] architecture with behaviours suitable for autonomous performance (navigation, mapping, victim identification) in USAR environments. Commands can be accepted from a teleoperator via an interface for low-level direction of a selected individual robot, or via setting high-level waypoints. Peripheral components, such as a user interface, are also necessary and are described fully in [14].

The central part of this approach involves the use of two software agents running on each robot to effectively balance the teleoperated and autonomous components. A *mediation agent* is used to appropriately blend the commands from a teleoperator with a robot’s autonomous processing, while an *intervention recognition agent* recognizes situations in which an operator should be informed that intervention on his or her part is required (see Fig. 1). The remainder of this Section describes these two agents.

3.1. Mediation Agents

A mediation agent is responsible for smoothly integrating the operator’s commands with those of a robot’s autonomous control system. While previous approaches have focused on blending operator instructions directly with instructions from an autonomous controller or assigning goals to be carried out (see Section 2.), our approach is more flexible, allowing the agent to intelligently evaluate instructions before blending to ensure that instructions are safe and appropriate to execute. To blend autonomy and teleoperation appropriately, a mediation agent is capable of reasoning about commands that have been sent to the robot from the human operator. Some commands may be followed to the letter, while others are integrated with the robot’s own desires or completely refused. The latter allows the vetoing of actions that would put the robot in danger inadvertently. There may certainly be cases where putting a robot at risk may be deliberate (i.e. the value of information obtained is worth the potential loss of the robot), and so it is also possible for a mediation agent to allow the operator’s commands to be unquestioned.

A mediation agent operates in one of five modes that are set by a human operator. The most complex of these is *Weighted Teleautonomy*, intended to be the “normal” mode in which agents operate, where the mediation agent observes the current system, and weights inputs from the teleoperator’s interface and the autonomous control system. The user interface provides a slide control allowing a base ratio to be set. This sliding autonomy setting is only one component affecting the weight of autonomy vs. teleoperation, however - commands are also examined in their execution context and weighted based on effect (described below). In contrast to this weighted range, *Fully Autonomous* and *Fully Teleoperated* modes simply set the weight of one of the two sources to a zero value. In addition to these, we have developed two modes that allow a human operator to have more detailed control. In *Manual Behaviour Weight Modification*, the operator manually defines the internal weights an agent places on its behaviours (this weighting factor directly affects the influence that the behaviour’s recommended output has in the overall response of the robot’s control system [15]), while in *Manual Behaviour Switching*, the operator can switch through the behaviours that are implemented for the autonomous robot, and the robot runs autonomously using only the chosen behaviour. Together, these allow the subsumption of previous approaches to teleautonomy within a single architecture.

An instance of the mediation agent runs on each robot on a team and serves as the main control loop. It signals the perceptual system to refresh currently perceived information, requests an action from the robot’s autonomous control system, and if in a mode where an operator is participating, retrieves exterior control signals. These are passed to the *command evaluator*, a symbolic reasoning system responsible for identifying commands whose execution may be dangerous or counter-productive in the current execution context. This analysis is performed by predicting the future position of the robot if the command were executed, and whether that position would leave the robot in a negative situation to the best of the command evaluator’s knowledge. For example, if the robot were at the edge of a ledge and a forward motion command was issued, the system would predict a future position that was n cm forward, allowing to predict the likelihood of a fall. If the weighted teleautonomy control mode is being used, commands are further blended depending on the degree of autonomy set by the operator. This results in a control vector that is interpreted and sent to the robot’s actuators.

The knowledge necessary to reason about the propriety of commands is as broad as a robot’s domain. We have currently implemented knowledge for two particular situations that are intuitively useful in the USAR domain: moving too near an obstacle (which would potentially get the robot stuck or damaged through contact) and moving away from an object that appears to be a potential victim. The evaluation in Section 4. is based on this knowledge. We are currently broadening the system’s knowledge to encompass other situations in USAR.

3.2. Intervention Recognition

In addition to appropriately balancing commands from the operator and autonomous control system, we must also take note of specific situations in which human intervention is desired. This is the responsibility of the intervention recognition agent, which performs its task

through the use of a knowledge base estimating the degree of likelihood that a robot can or should carry on its current course of activity. An instance of the intervention recognition agent runs on each individual robot platform, analyzing the robot's perceptions, identifying specific scenarios indicative of the need for operator intervention, and separating these from the situations where progress is still likely without intervention.

While the intervention recognition agent is designed in an extendable manner so that specific scenarios of interest can be encoded in a knowledge-based fashion, we have currently implemented three specific scenarios within the USAR domain that we have found to be useful. The simplest problem to address, but the most common, is a robot becoming stuck or otherwise immobile. The intervention recognition agent considers a robot to be stuck when it is sending instructions to its actuators, but the actuators are not completing those instructions. If the robot's actuators are receiving commands, the robot will compare its current sensor readings to past sensor readings attempting to distinguish if there is any evidence supporting movement on the robot's part. If there is little or no evidence supporting movement within the last few perceive-act cycles, the robot is declared stuck.

In addition to becoming stuck, robots can become lost or unable to complete their goals. These are dealt with using the autonomous control system's ability to note unique landmarks in the environment. The robot's autonomous control system remembers how many times it has sensed a landmark and how much time has elapsed since the last time it has sensed the same landmark. The intervention recognition agent uses this information to determine when a robot has returned to the same location too often within a specified period of time. This allows the operator to be notified and the robot guided to a new location.

The most crucial event that can occur in the USAR domain, however, is the detection of victims. Victim identification is an extremely difficult task to perform well purely autonomously [4, 17], and so is one of the primary reasons why an operator would desire to be interrupted. In our approach, the intervention recognition agent is responsible for identifying when an object in the environment resembles a victim and notifying the robot's operator. The intent is for the operator to make a judgment whether a victim is at the location, since any autonomous system is likely to make errors in victim identification. An accurate model of victim identification is not the focus of this work, and for the purposes of evaluation, vision alone is used to identify objects resembling victims by their color using a single perceptual schema. For future deployment in competition, we intend to supplement this by searching for shapes indicative of partial human forms as well as other forms of perception, such as heat detection.

When an intervention recognition agent identifies a situation that requires the operator to intervene, the operator is notified through the user interface (described in more detail in [14]). The user interface contains a list of the current available robots and their states. When an intervention recognition agent identifies a situation where intervention is desirable, it changes the state of the current robot, updating the user interface. An operator working with the user interface can see that the robot requires assistance, along with a brief message describing the robot's current state, and is able to operate the agent by clicking on the robot's tab on the user interface.

Each robot's intervention recognition agent is implemented in Java as a package containing *intervention event* objects, the *intervention recognition* object and a *perception memory*

object. The perception memory stores snapshots of the robot’s perceptions for the past five perceptual cycles. These do not attempt to create a representation of the world other than the basic landmarks described above, and are stored as raw perceptions that are input to perceptual schemas. Three intervention event objects, representing the three types of interventions described above, are polled regularly by the intervention recognition object. Each of these accesses these stored perceptions looking for the conditions under which intervention would be required, and notifies the human controller in that case. For example, to identify a confused state, the appropriate event object counts the number of times a robot’s perceptual schemas have noted a given landmark. New intervention recognition objects can be defined to extend the system in a domain-dependent manner.

While the attempt to recognize situations where a human must intervene does require a continual devotion of computational resources, this is a much smaller set of resources than would be required to perform autonomously (since that would also require dealing with these problems autonomously as well as recognizing them). In addition, since one of the main problems in teleoperation is operators being overwhelmed by the ongoing process of monitoring robot perceptions looking for useful information and/or problems, the logical means of dealing with with is (barring complete robot autonomy) to give robots the ability to monitor themselves. The use of an agent to recognize and respond to unexpected situations has analogies to the work of Micacchi and Cohen [18]; their work, however, is done at a much higher level in the RoboCup Rescue simulator, and is intended for coordinating similar high-level decisions, as opposed to dealing with the lower-level perceptions and action consequences described here. Similarly, our current work in part involves extending this model to allow robots to monitor one another as opposed to just themselves.

4. Evaluation

In order to examine the performance of this approach, we placed robots in a controlled simulated USAR environment implemented using the Player/Stage simulation tool [19]. Player/Stage was chosen because it is widely used and allows development of code that operates directly on Pioneer robots. Each robot was a Pioneer II equipped with a differential drive, a SICK laser range scanner and a video (CCD) camera with a wide angle lens. The environment used was $20m^2$, and for the purposes of experimental trials environments were generated and evaluated for equal difficulty and for the purposes of repeatability. Environments were constructed for a desired degree of obstacle coverage (5%, 10%, 15%, and 20%), using $50cm^2$ obstacles to construct impassable walls with openings between them in order to approximate the structure of a collapsed building. Limits were set on the number of open areas (max. 20) that were generated as a result of this, their size ($100cm^2$ - $300cm^2$) and the number of openings to each area (at most 3). Single obstacles were then distributed throughout the environment to make up the desired obstacle coverage. Obstacles were distributed randomly except for two criteria. First, between every obstacle and every open area there was a minimum distance of 120cm, in order that multiple obstacles could not cluster too closely to open areas, thereby reducing the likelihood of creating areas of the environment that were completely inaccessible. While inaccessible areas will of course occur

in the real world, for the purposes of producing comparable domains we need to control for this. The distance between the center of any two obstacles in the environment could also not be less than 50cm, making it impossible for obstacles to physically overlap more than a few centimeters.

After generating sample environments, the number of local minima present was averaged, and environments were rejected that had local minima counts that were off more than a small range from that mean. Further, environments were also made consistent in difficulty by hand-verifying that there were no inaccessible areas of the domain, and that open areas did not get generated in such a manner that they formed hallways that were too narrow for the robot to physically occupy.

Each environment had 10 victims and 5 negative victims (objects that from a distance would appear to be victims). These were distributed randomly except for a proviso that the distance between the center of any real or negative victim from the next closest real or negative victim was at least 60cm.

For the purposes of the simulated environment, the visual schema for victim identification was implemented using colour blob detection, where it is possible to distinguish between actual victims and objects that only resemble victims by their color (negative victims) when they are within $3m$, while outside of $3m$ both victims and objects resembling victims are identified as objects of interest. While victim identification is not the focus of this work, this method serves to force the robot to move within a close proximity in order to make a victim identification, something that would be expected in the real world.

Potential landmarks in the simulated environment were labelled with bar codes that can be read using the laser range finder. While this is not consistent with the real world, the intent here was to allow a consistent and repeatable means of marking landmarks in the world to examine the operation of the confused identifier object.

We evaluated our approach, examining both performance as the environment became more complex, and as the number of robots increased. Each of these is described in the Subsections below.

4.1. Performance as Environment Complexity Increases

We examined the performance of the blended teleautonomous approach based on the components described in Section 3. in comparison to purely autonomous and purely teleoperated implementations using the same interface across varying degrees of obstacle coverage, examining the amount of time robots spent immobile, the number of victims found the amount of area covered, and the number of robot-operator interactions. In each case, all results presented are averages over five different trials.

Fig. 2, Fig. 3 and Fig. 4 show the performance of the three control systems in terms of area coverage over time for three of the four categories of obstacle coverage (5% coverage was similar enough to 10% coverage to omit, given space considerations). Teleautonomous robots performed significantly better than autonomous robots in terms of area coverage across all degrees of obstacle coverage. We attribute this performance to a human operator’s ability to recognize unexplored areas of the environment quickly and guide robots to unexplored areas more efficiently than the autonomous control system could. Teleoperated control performed

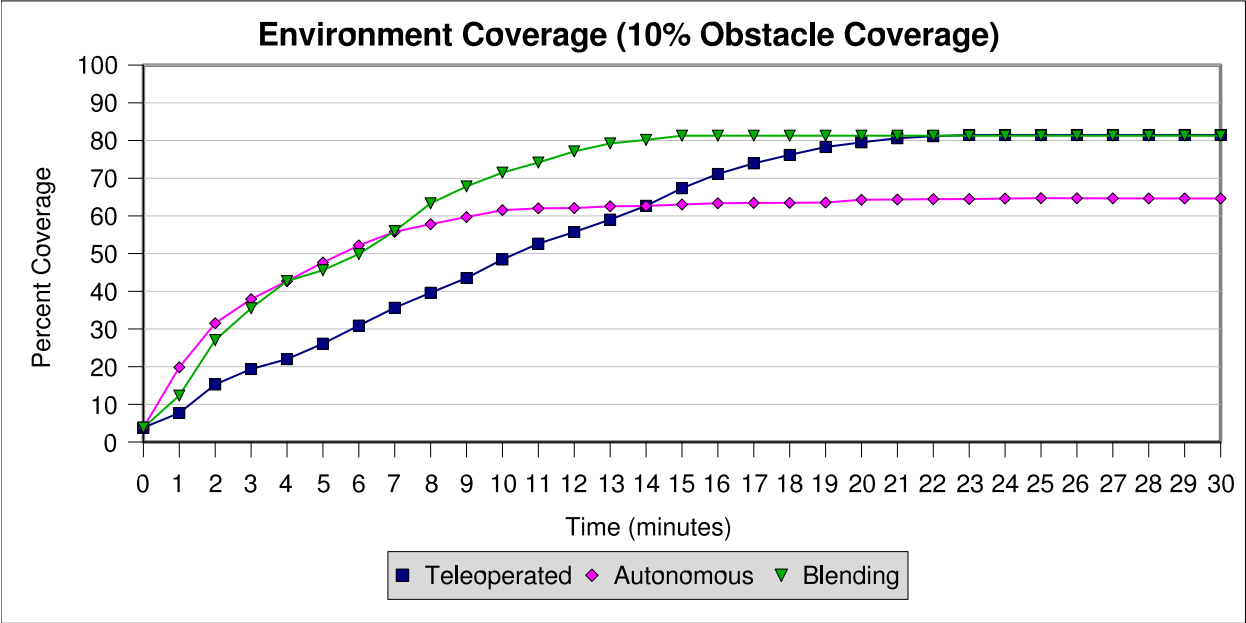


Figure 2: Average environment coverage achieved by autonomous, blended teleautonomous and teleoperated robot control in environments where 10% was covered in obstacles.

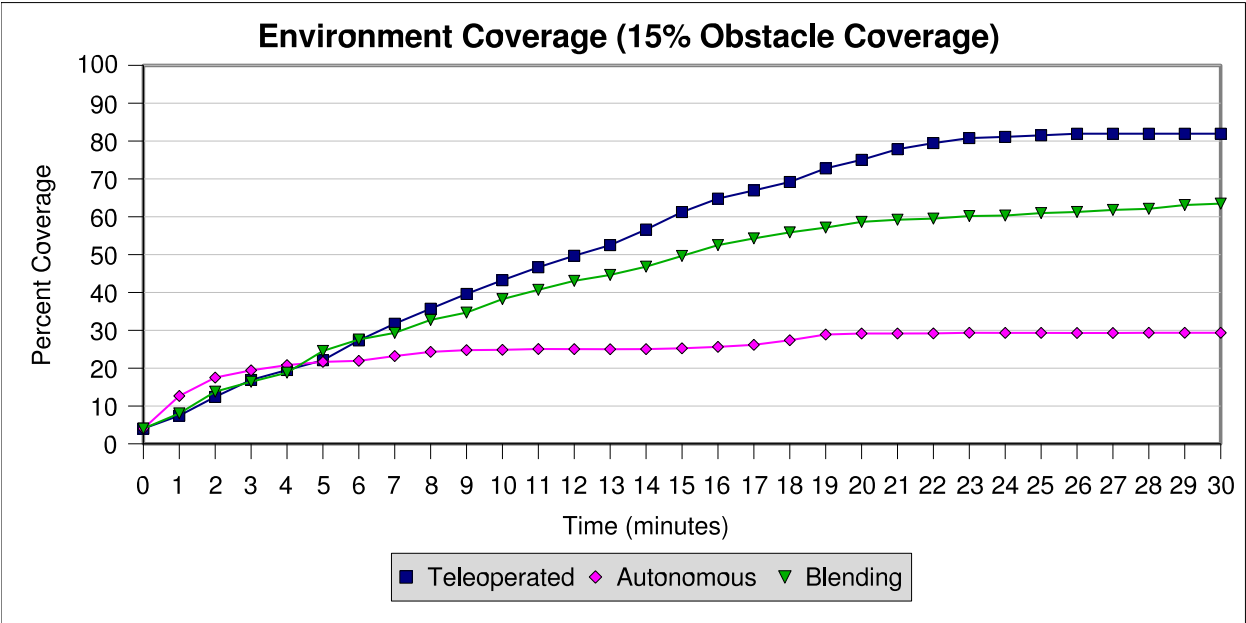


Figure 3: Average environment coverage achieved by autonomous, blended teleautonomous and teleoperated robot control in environments where 15% was covered in obstacles.

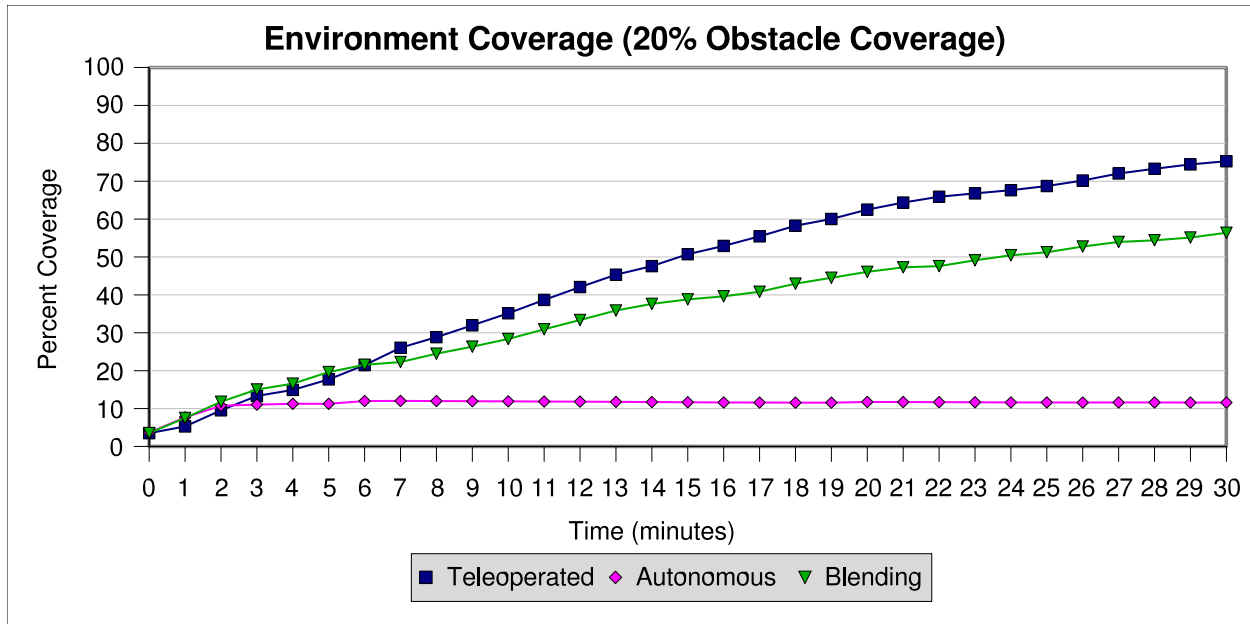


Figure 4: Average environment coverage achieved by autonomous, blended teleautonomous and teleoperated robot control in environments where 20% was covered in obstacles.

slightly better than blending control at the 15% and 20% obstacle coverage levels, since although the operator could guide blending robots into unexplored areas, once a robot was neglected (i.e. the operator shifted attention to another robot) the autonomous portion of the blending control system could guide the robot back to an explored area. This happened less at lower obstacle coverage levels: since there are fewer obstacles, there are fewer course changes necessary for robots to go around them when operating without benefit of an operator, and less likelihood of heading back toward a previously covered area.

The time each robot spent immobile with respect to autonomous versus blending robot control (see Fig. 5) is another indication of the gains associated with blending autonomy and teleoperation. Since the autonomous robots employ behaviour-based control, they are susceptible to local minima, often becoming stuck in difficult environments. When robots got stuck in autonomous trials, they would often remain stuck. In the blending trials, if a robot became stuck, the operator was informed and could usually quickly free the robot. In trials with higher obstacle coverage, the robots would get themselves stuck in much more complex ways, making it more difficult for operators to release them. Blending operator instructions with the autonomous instructions contributes to a significant increase in effectiveness of robots, which can be observed by comparing the results of the autonomous trials and the blending trials.

We also found that the robots using blended teleautonomy had an advantage over both teleoperated control and autonomous control in terms of victim identification (see Fig. 6, Fig. 7 and Fig. 8). At least a few victims in any experimental scenario were in open areas that were and easy enough to navigate to autonomously, and both blending robots and autonomous robots could take advantage of this. Correspondingly, only very little attention

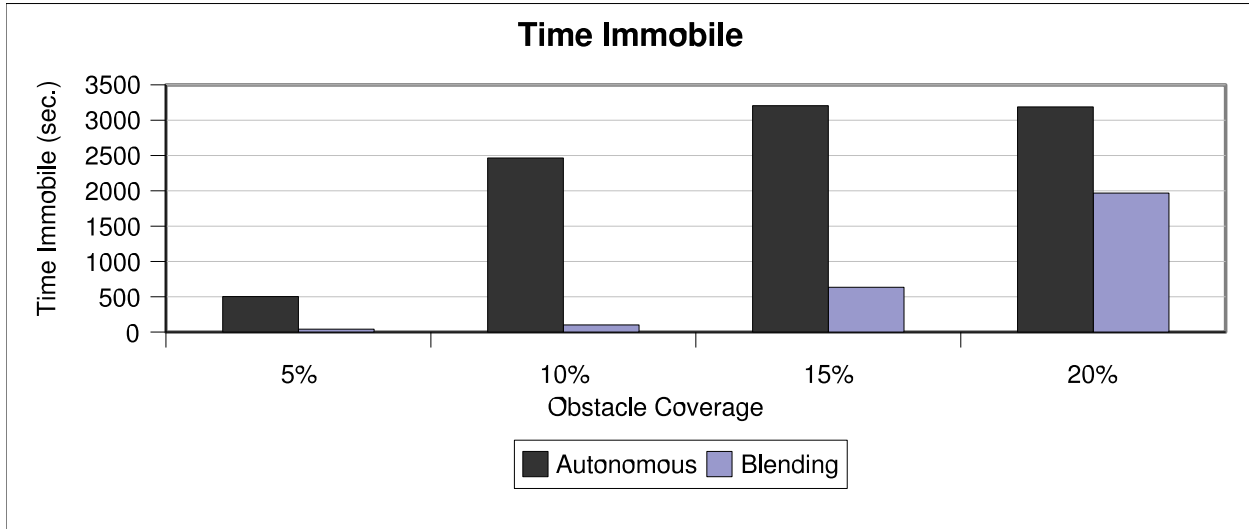


Figure 5: Average time in seconds spent immobile by environment difficulty, for blending and autonomous robot control.

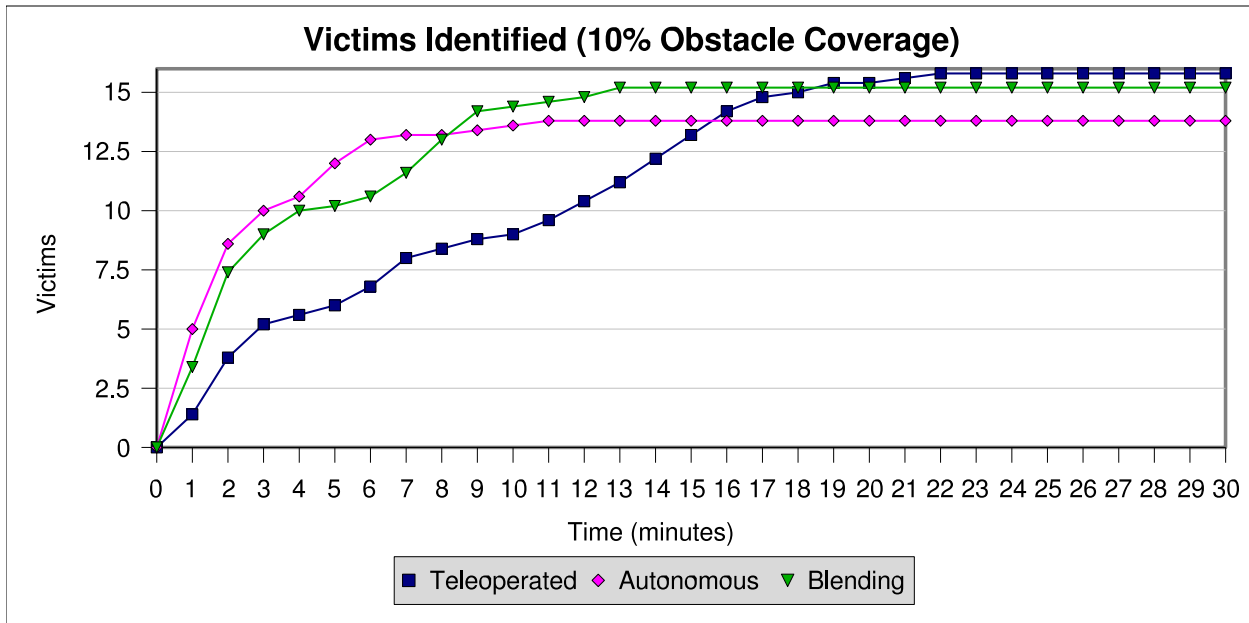


Figure 6: Comparison of number of victims identified in teleoperated, autonomous, and blending experiments in environments where 10% was covered in obstacles.

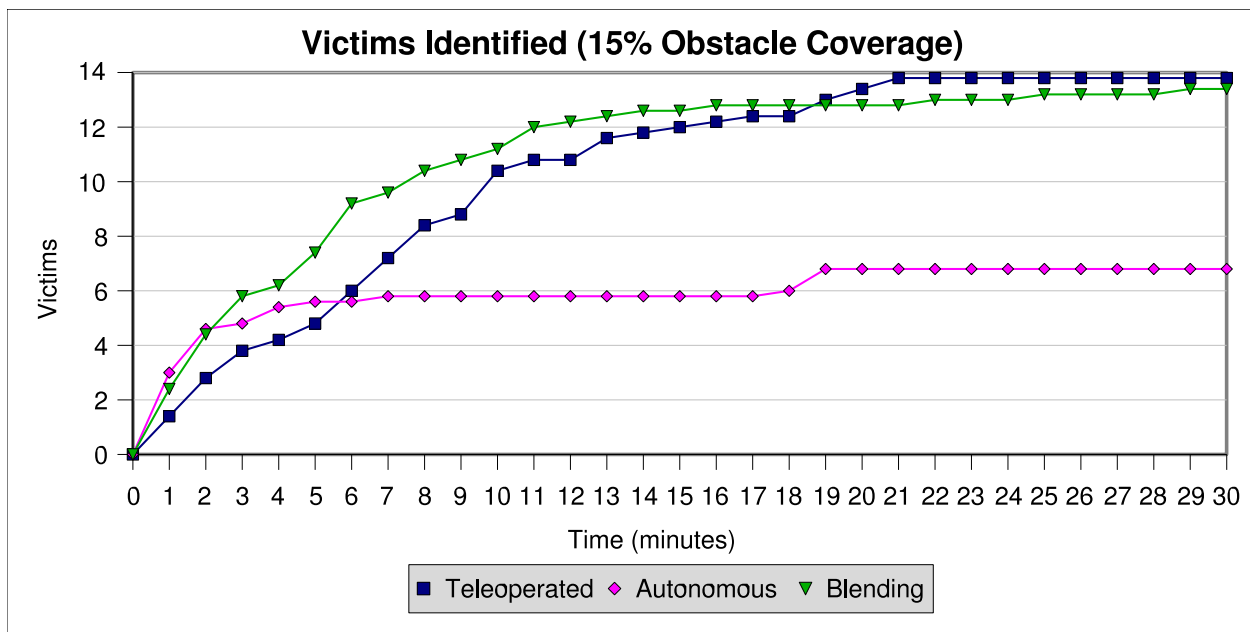


Figure 7: Comparison of number of victims identified in teleoperated, autonomous, and blending experiments in environments where 15% was covered in obstacles.

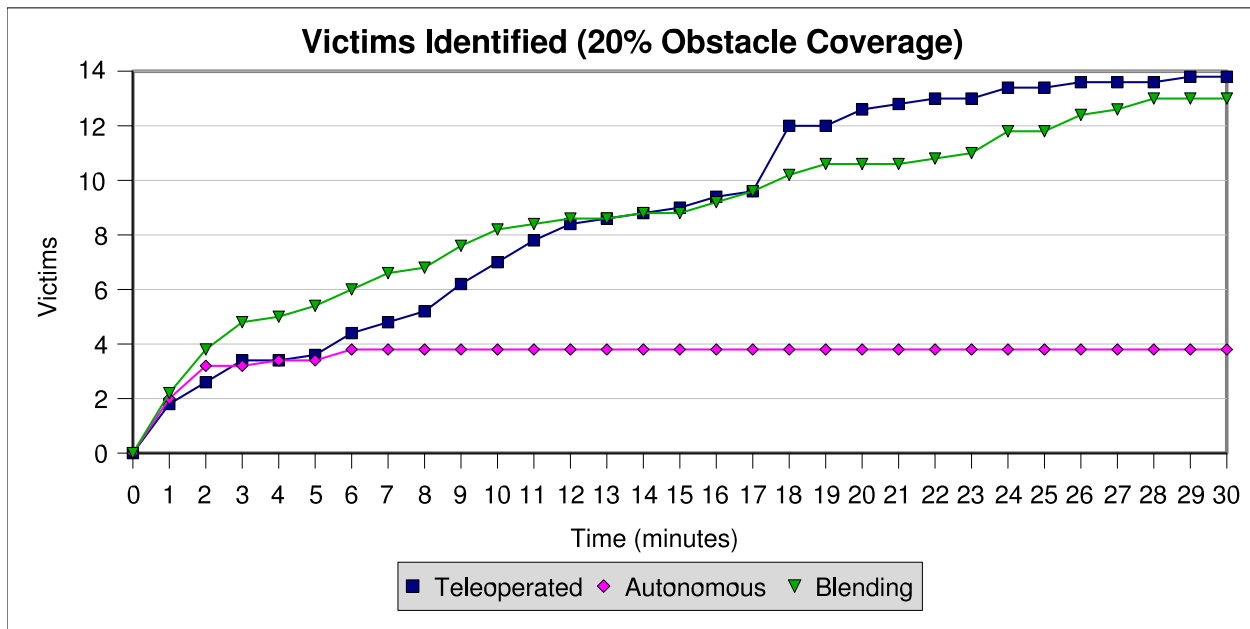


Figure 8: Comparison of number of victims identified in teleoperated, autonomous, and blending experiments in environments where 20% was covered in obstacles.

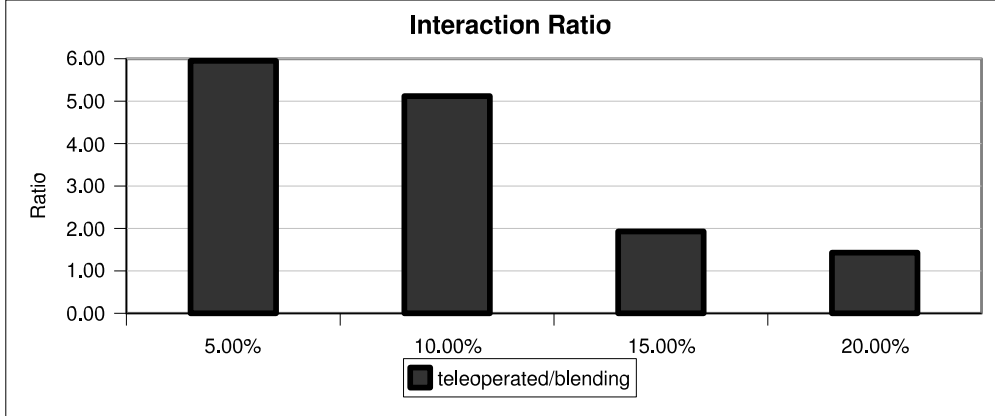


Figure 9: Average ($n=5$) ratio of operator interactions (teleoperated/blended teleautonomous in environments with 5%, 10%, 15%, and 20% obstacle coverage.

on the part of the operator was needed for blending robots. Later on, when the victims in the open were all located, the blending robots performed better than the autonomous robots, because the operator could guide the robots through the more difficult areas of the environment, encouraging the team to cover more area and discover more victims.

While there were cases (at the 15% and 20% obstacle coverage levels) where purely teleoperated robots could still outperform teleautonomous robots, there is a significant overhead being paid for this in terms of operator intervention and ultimately operator fatigue. Throughout all trials performed, the teleoperated robots required many more interactions to complete their task. This ranged from an average of 5.9 times more interactions than the blended control system for 5% obstacle coverage, to 1.4 times the number of interactions for 20% obstacle coverage (see Fig. 9). Even with the additional attention required by the more dense environments, the blending control system required less attention from the operator, which contributes to a lower cognitive load.

4.2. Performance with Larger Teams

We also investigated the effects of increasing team size, in order to examine the efficacy of the approach where a heavier operator load was involved. All results here consist of an average over four trials.

With respect to both the identification of victims and the coverage of the environment over time we found that increasing the number of robots from teams of three to teams of six and nine offered some improvement in performance. At the 10% obstacle coverage level a team of three autonomous robots required an average of seven minutes to identify all 15 victims in the environment, while teams of six and nine robots required approximately five minutes. Blending robots also performed better with a larger team of robots. However, the advantage for blending robots is much less pronounced (see Fig. 10). Increasing the obstacle coverage level from 10% to 15% further increased the advantage gained by increasing the number of robots per team (see Fig. 11). On average a team of three autonomous robots

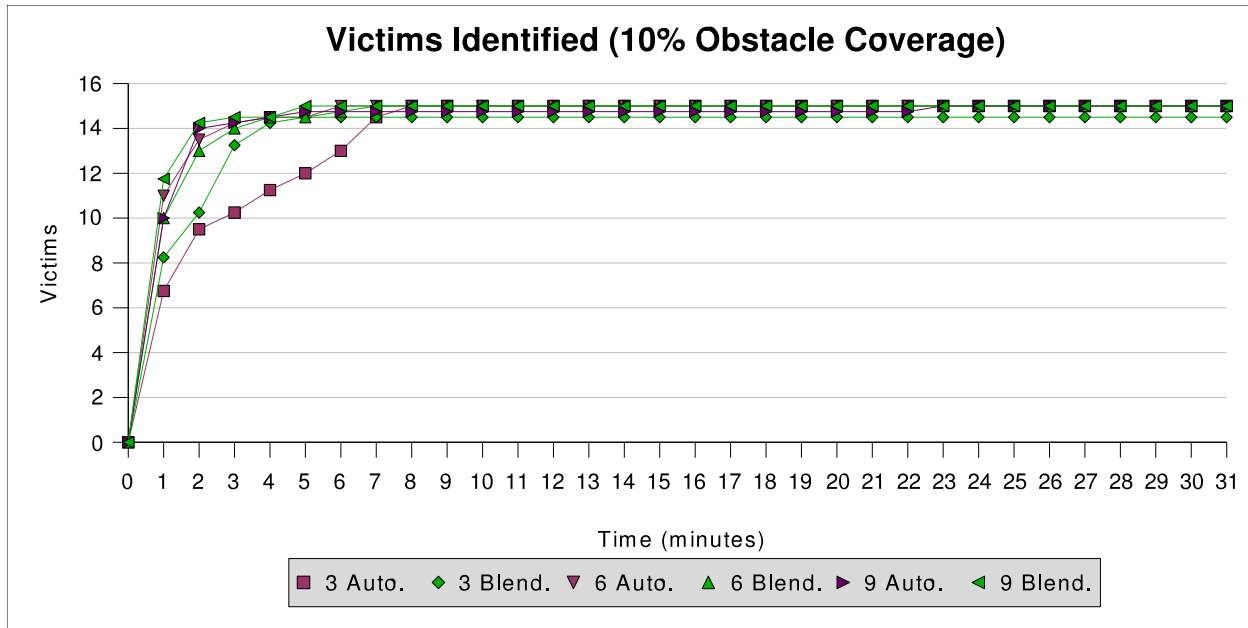


Figure 10: Comparison of number of victims identified in autonomous and blending experiments with different numbers of robots (3, 6 or 9) in environments where 10% was covered in obstacles.

was unable to identify more than eight victims, while a team of six robots could identify 12 victims and a team of nine robots could identify close to 14 victims.

With respect to environment coverage at the 10% obstacle coverage level teams of six and nine robots performed almost exactly the same, both reaching roughly 85% to 90% of the environment while teams of three robots could only reach an environment coverage of approximately 80%. Blending robots had a similar performance – however, the performance gained by increasing the number of robots was more important with respect to how quickly the environment was covered as opposed to how much of the total environment was covered (see Fig. 12). Increasing the obstacle coverage from 10% to 15% increased the performance improvements. The blending robots clearly have an advantage over autonomous robots (see Fig. 13). Teams of three robots were capable of covering an average of roughly 33% of the environment, but increasing the number of robots from three to six almost doubled the performance. Increasing the number of robots from six to nine produced a much smaller performance gain.

5. Discussion

This paper has described facilities for balancing autonomy and teleoperation effectively for an Urban Search and Rescue environment, through the use of agents that recognize untoward situations and mediate between robots and teleoperators. The experiments described in this paper demonstrate that this approach, through blending autonomy and teleoperation appropriately and notifying an operator when intervention is desirable, can significantly

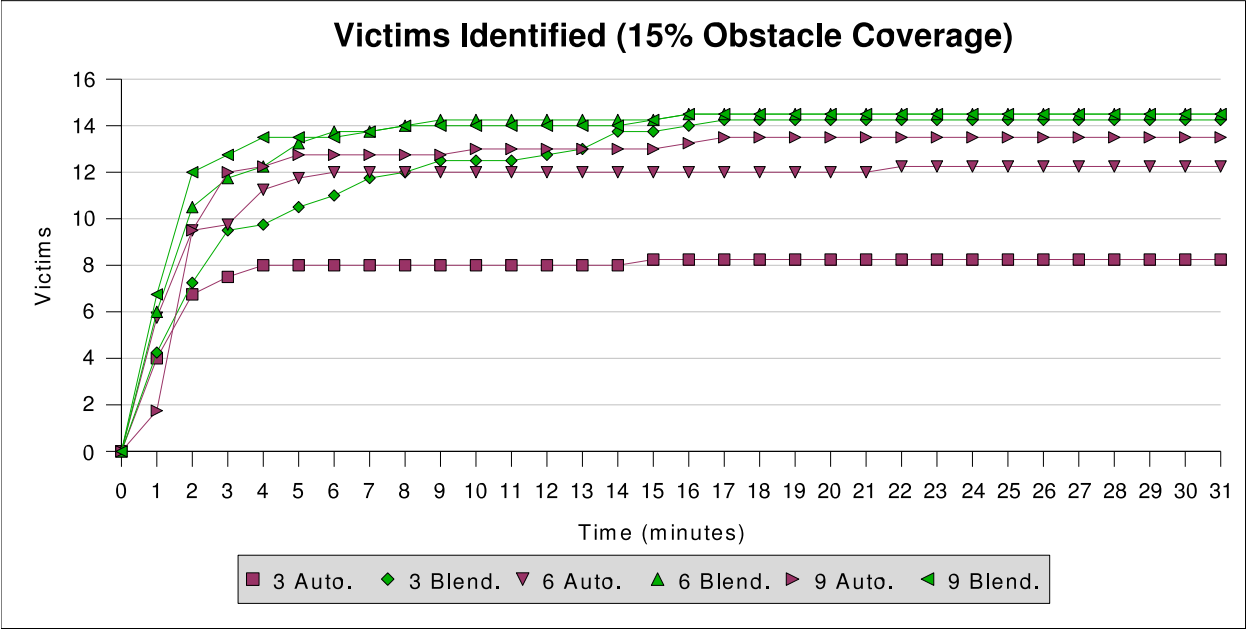


Figure 11: Comparison of number of victims identified in autonomous and blending experiments with different numbers of robots (3, 6 or 9) in environments where 15% was covered in obstacles.

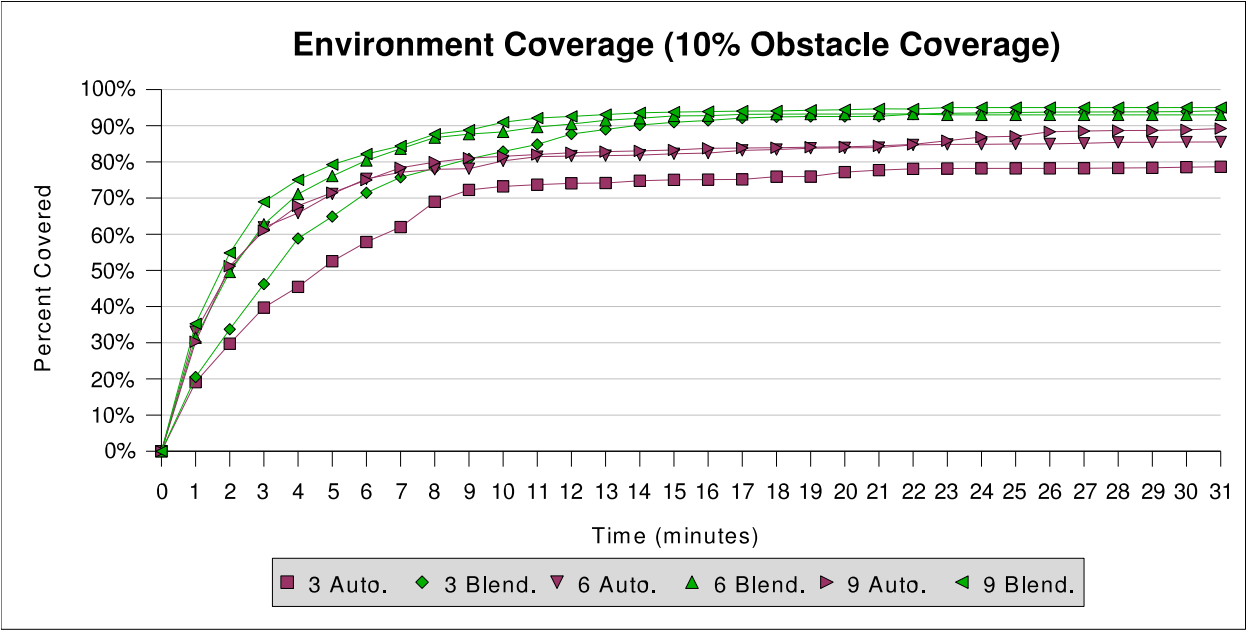


Figure 12: Average environment coverage achieved by autonomous and blended teleautonomous robot control with different numbers of robots (3, 6 or 9) in environments where 10% was covered in obstacles.

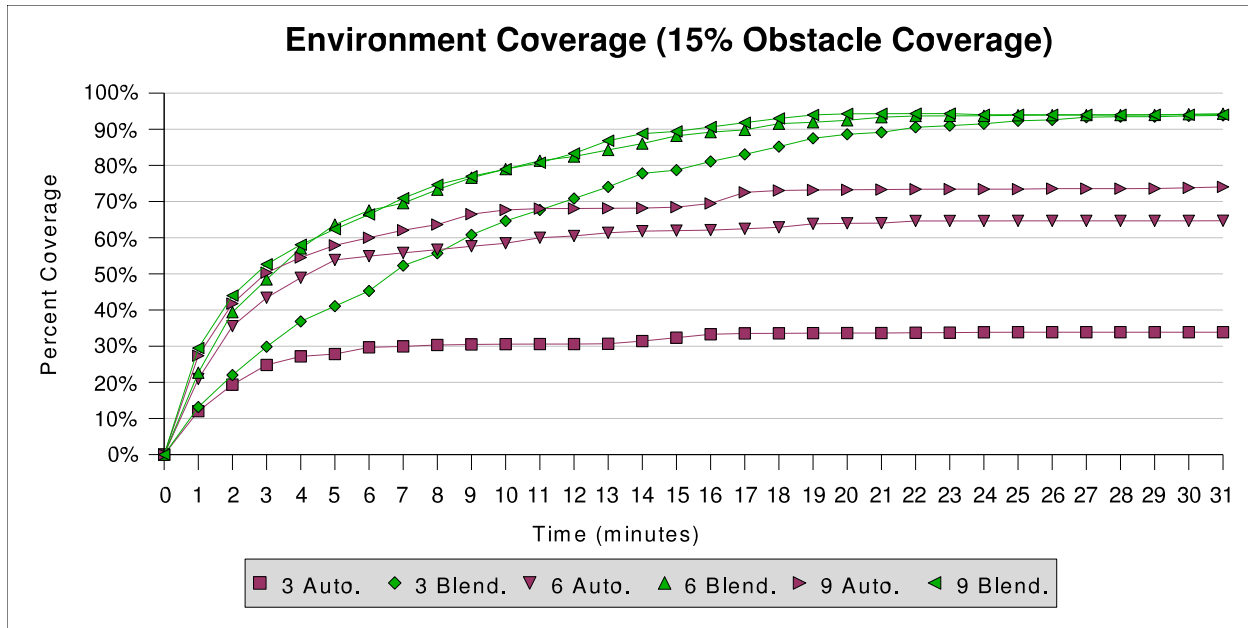


Figure 13: Average environment coverage achieved by autonomous, blended teleautonomous and teleoperated robot control with different numbers of robots (3, 6 or 9) in environments where 15% was covered in obstacles.

improve the effectiveness of a robot team. It is also evident that the blending of autonomy and teleoperation reduces the number of interactions between the operator and a collection of robots while still maintaining a comparable level of performance. We have also illustrated the efficacy of using this method while employing larger sizes of robot teams.

Future work on this project is proceeding in a number of directions. We are primarily working on porting this to a physical environment (maintaining the use of Pioneer robots, but with vision and sonar rather than laser) and on extending the knowledge bases used by the software agents to address a broader range of useful situations. We are also working on employing advice from peer robots as well as from human operators as another method of stretching the abilities of a human operator. There is a natural potential for this in our approach, in that the infrastructure for requesting operator intervention is similar to that for requesting intervention from other sources, and similarly, a mediation agent can receive advice over a network connection from another agent just as it can from a human operator. We have begun this work by implementing peer advice-giving (and the integration of that advice by the recipient) for stuck and confused agents, attempting to adapt the kinds of interventions a human operator supplies [20]. We do not yet attempt to weigh or blend human operator vs. agent advice, nor allow agents to directly request advice from peers as opposed to humans. Advice is simply given upon recognizing others in the environment and diagnosing their situation (by observation and a dialog with the robot) as being stuck or confused. Preliminary results indicate that that peer advice based purely on observation and agent interaction during the occasions when agents encounter one another in the environment can improve the performance of stuck and confused agents under conditions similar to the

experiments described here [20], and so we expect significant improvements when this is associated with a human operator. We are also currently working on extending autonomous control mechanisms to deal with deciding when to request and offer assistance to others, considering the value of potential advice over that of others that may be available (or that of an operator) before responding to a request. This will in turn allow research into the efficacy of making robots more or less proactive in terms of assisting one another in this domain.

While the ultimate goal of artificial intelligence is fully autonomous agents, we believe this goal, at least for domains as complex as USAR, remains in the distant future. In the meantime, research on profitably augmenting human control will help us leverage human abilities as much as possible while furthering understanding of the more difficult elements in domains such as USAR.

References

- [1] Jennifer Casper. Human-robot interactions during the robot-assisted urban search and rescue response at the world trade center. Master's thesis, University of South Florida, April 2002. Computer Science and Engineering.
- [2] Jennifer Casper and Robin Murphy. Workflow study on human-robot interaction in USAR. In *Proceedings of the International Conference on Robotics and Automation*, volume 2, pages 1997–2003, Washington, May 2002.
- [3] Adam Jacoff, Elena Messina, and John Evans. Experiences in deploying test arenas for autonomous mobile robots. In *Performance Metrics for Intelligent Systems Workshop*, Mexico City, Mexico, September 2001.
- [4] Robin Murphy, Jennifer Casper, Mark Micire, and Jeff Hyams. Assessment of the NIST standard test bed for urban search and rescue. In *Proceedings of AAAI Mobile Robotics Competition Workshop*, pages 11–16, Austin, TX, July 2000.
- [5] Jacky Baltes and John Anderson. A pragmatic approach to robotic rescue: The keystone fire brigade. In *Proceedings of the AAAI-02 Mobile Robot Competition Workshop*, pages 38–43, Edmonton, Alberta, July 2002.
- [6] Jacky Baltes and John Anderson. The keystone fire brigade 2003. In *Proceedings of the IJCAI-03 Mobile Robot Competition Workshop*, pages 30–35, Acapulco, Mexico, July 2003.
- [7] Ronald C. Arkin and Khaled S. Ali. Integration of reactive and telerobotic control in multiagent robotic systems. In *Proceedings of the Third International Conference on Simulation of Adaptive Behavior*, pages 473–478, Brighton, England, August 1994.
- [8] Khaled S. Ali and Ronald C. Arkin. Multiagent teleautonomous behavior control. *Machine Intelligence and Robotic Control*, 1(2):3–10, 2000.
- [9] Khaled S. Ali. *Muliagent Telerobotics: Matching Systems to Tasks*. PhD thesis, Georgia Tech, 1999. Computer Science and Engineering.

- [10] Robin Murphy and J. Sprouse. Strategies for searching an area with semi-autonomous mobile robots. In *Proceedings of Robotics for Challenging Environments*, pages 15–21, Albuquerque, NM, June 1996.
- [11] Jacob W. Crandall and Michael A. Goodrich. Experiments in adjustable autonomy. In *Proceedings of the International Conference of Systems, Man, and Cybernetics*, volume 3, pages 1624–1629, 2001.
- [12] Sean Graves and Richard Volz. Action selection in teleautonomous systems. In *Proceedings of the International Conference on Intelligent Robots and Systems*, pages 14–19, 1995.
- [13] Mohan Trivedi, Brett Hall, Greg Kogut, and Steve Roche. Web-based teleautonomy and telepresence. In *Proceedings of the 45th Optical Science and Technology Conference*, volume 4120, pages 81–85, San Diego, August 2000.
- [14] Ryan Wegner. Balancing robotic teleoperation and autonomy in a complex dynamic environment. Master’s thesis, University of Manitoba, July 2003. Department of Computer Science.
- [15] Ronald C. Arkin. *Behavior-Based Robotics*. Cambridge: MIT Press, Cambridge, MA, 1998.
- [16] Ronald C. Arkin and Tucker Balch. Cooperative multiagent robotic systems. In D. Kortenkamp, R. Murphy, and R.P. Bonasso, editors, *Artificial Intelligence and Mobile Robots*, pages 278–296. Cambridge: MIT Press, Atlanta, Georgia, 1998.
- [17] Jennifer Casper, Robin Murphy, Mark Micire, and Jeff Hyams. Mixed-initiative control of multiple heterogeneous robots for urban search and rescue. Technical report, University of South Florida, 2000.
- [18] Chris Micacci and Robin Cohen. A multi-agent architecture for systems operating in real-time environments. In *Proceedings of the Agents Meet Robots Workshop, 2004 Canadian AI Conference*, London, Ontario, May 2004. National Research Council.
- [19] Brian P. Gerkey, Richard T. Vaughan, Kasper Stoy, Andrew Howard, Gaurav S. Sukhatme, and Maja J. Mataric. Most valuable player: A robot device server for distributed control. In *Proceedings of Intelligent Robots and Systems*, pages 1226–1231, Wailea, Hawaii, October 2001.
- [20] Michael Gauthier and John Anderson. Peer instruction to assist a teleautonomous USAR system. In *Third International Conference on Computational Intelligence, Robotics, and Autonomous Systems*, Singapore, December 2005. (Submitted).

Biographies

Ryan Wegner is a doctoral student in the Department of Computer Science at the University of Manitoba, where he also completed his M.Sc. research on teleautonomy in urban search and rescue. He is interested in robotics and human-robot interaction, as well as applications of robotics to search and rescue and to security.

John Anderson is an Associate Professor in the Department of Computer Science at the University of Manitoba, and co-directs the Autonomous Agents Laboratory. His main research interests lie in autonomous agents, multi-agent systems, robotics, and the application of these to urban search and rescue and other complex problems. He is also interested in human-robot interaction and the use of robotics in education.