An Agent-Based Approach to Balancing Teleoperation and Autonomy for Robotic 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

In highly complex domains such as disaster rescue, today's mobile robots simply do not have the ability to perform successfully on their own. Human teleoperation is strongly relied upon, but this also has problems: humans suffer from cognitive overload and have difficulties in constructing a representation of the space around a robot given information from its senses. 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.

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 robotics and robotic control technology. The environment is difficult to maneuver within, and unpredictable in that even a known building layout may have changed dramatically during the associated disaster. Basic robotics skills such as localization are strongly affected (for example, mobile debris causes wheel slippage, leading to more severe errors from wheel encoders), and sensing is much more difficult than any standard indoor domain. The wide range of skills necessary for adequate performance, 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 [6, 7]).

This reliance on teleoperation can also be seen in current research. Like other challenge problems such as robotic soccer, USAR research is evaluated in controlled conditions using a physical testbed (e.g. the NIST testbed [11], 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 [12], the vast majority of entries to such competitions are teleoperated. For example, at both AAAI-02 in Edmonton [4] and IJCAI-03 in Acapulco [5] we were one of only two entries running fully autonomously.

Beyond a desire as AI researchers to advance AI itself, there are good reasons behind a desire to avoid pure teleoperation. 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 [6, 7]. 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 [2]. Cognitive overload not only requires information presentation to be very selective, but strongly limits the number of robots that can be controlled by an individual.

We are interested in providing functional intelligent control to a team of mobile robots. Given the difficulty of operating within this domain, the state of the art in robotic control technology, and the problems associated with pure teleoperation, a combination of the two approaches (commonly known as a *teleautonomous* approach) is warranted. Ideally, an intelligent control mechanism should support a blend of teleoperation and autonomy that is blended as seamlessly as possible, allowing a teleoperator to focus attention on the problems that require the most assistance. A teleoperator should ideally only be interrupted when context suggests it is worth doing so; at the same time, the actions of any robot operating autonomously should be able to be overridden by an operator at any time.

We have developed an approach to blending teleoperation and autonomous control in behaviour-based mobile robotic robots [15]. This consists of three sets of facilities. First, a schema-based [1] autonomous control system for navigation and mapping that allows robots to perform autonomously (subject to the associated limitations that a domain as difficult as this one places). Second, support for direct teleoperation of robots, including a joystick-based interface as well as the ability to exert control at a higher level by setting waypoints. Finally, facilities for blending autonomy and teleoperation appropriately for each robot. These consist of a mediation agent that allows the blending of the desires of both teleoperator and robot for low-level robotic control, and an intervention recognition agent for recognizing both problematic (e.g. agent is stuck) and helpful (e.g. potential victim found) situations in which the operator should be interrupted.

In this paper, we focus on the facilities we have developed for blending teleoperation and autonomy appropriately. We describe these facilities and their implementation, and describe the results of an evaluation of the performance of these facilities on simulated Pioneer robots. This evaluation includes both the efficacy of the approach at both locating victims and general environmental coverage in domains of varying complexity, as well as the efficacy of the approach under varying operator loads. 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 [2]. Arkin and Ali describe two approaches for teleautonomy with respect to multiple robots. Both of these are schema-based [1] 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 having the operator act as a supervisor. The operator has access to the behavioural parameters of the society (e.g. the low level gains of each motor schema). The operator could effect the emergent behaviour of the society as a whole by adjusting individuals' behavioural parameters. 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 [3], but shares the same limitations of Arkin and Ali's original approach.

Crandall et al. [9] 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 can hinder performance, and can be due to the operator turning his or her attention away, or from delays between issuing commands and the robot receiving those commands. They also describe a robot control system consisting of a set of robot behaviours and a user interface for controlling the robots. Their systems use 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.

Trividi et al. [14] designed a system that is intended to allow robotic units to recognize traffic collisions and other accidents. This system is strictly a laboratory design and years away from being deployable, 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 takes the role of a point on the polygon. The operator supplies only location guidelines for the polygon forming activity, and the balance between autonomous ability and remote control has been fixed as well.

Murphy and Sprouse [13] 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 applying its senses to 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. [8] 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.

3. Design and Implementation

In our approach to blended teleautonomy, robotic agents are implemented using a schema-based [3] architecture with behaviours suitable for autonomous performance (navigation, mapping, victim identification) in USAR environments. Commands can be accepted from a teleoperator via a joystick facility for low-level direction of a selected individual robot, or via setting high-level waypoints. Central

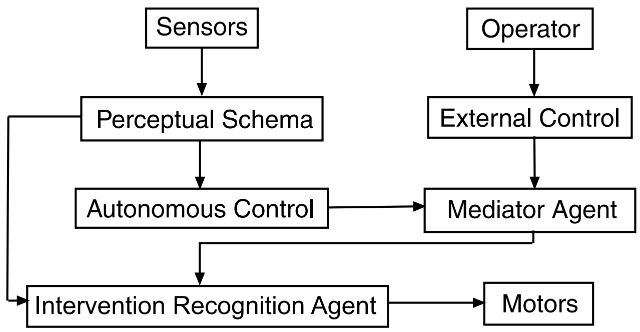

Figure 1: Overview of the components of the robot control system. Arrows indicate communication between components.

to this approach are two software agents running on each robot: 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 Figure 1). These two components are described in the remainder of this Section. Peripheral components, such as a user interface, are also necessary and are described fully in [15].

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, 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 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, such as being told to move forward when the operator is not necessarily aware the robot is on the edge of a drop. 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, as will be explained shortly. In contrast to this weighted mode, *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, allowing the operator to alter how the robot runs autonomously, 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 platform on a team and actually serves as the main control loop on the robot. It signals the perceptual system to refresh currently perceived information, requests an action vector 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 system that is responsible for identifying commands whose execution is dangerous or counter-productive in the current execution context. This is done by predicting the position of the robot if the command were executed, and whether that position would leave the robot in a negative situation from the standpoint of the knowledge in the command evaluator. Once commands are adjusted they are further blended depending on the degree of autonomy set by the operator if the weighted teleautonomy control mode is being used. This results in a control vector that is interpreted and sent to the robot's actuators.

The ideal command evaluation system would have enough knowledge to deal with any potential situation in USAR. Given the breadth of this problem, however, complete knowledge is unreasonable to expect. We are currently expanding our knowledge engineering efforts in this area, but have 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 a potential victim. The evaluation in Section 4 is based on this knowledge.

3.2. Intervention Recognition

Appropriately balancing operator instructions and autonomous abilities is only one part of this overall approach;

the other major component is the recognition of situations where operator intervention is required, in order to minimize the cognitive demands on the operator. Recognizing when robots require operator intervention in this approach requires examining specific situations in the form of an intervention recognition agent. The intervention recognition agent is ultimately responsible for indicating when the balance between autonomy and teleoperation of agents should be changed, by requesting a change of control from the operator (the infrastructure exists in this implementation for requesting assistance from other robots as well, but we have not yet implemented code in the autonomous control mechanism to deal with handling requests for such assistance). The intervention recognition agent 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. The intervention recognition agent is designed in an extendable manner so that specific scenarios of interest can be encoded in a knowledge-based fashion, resulting in a system that can be used in a wide range of environments.

For the purposes of encoding knowledge useful to USAR, we have currently implemented three specific scenarios within the intervention recognition agent 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 identifies when the robot is stuck and signals the operator. A stuck robot is defined as any robot that 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. None of the components in this approach contains an elaborate world model. However, the robot's autonomous control system is able to distinguish certain objects (*landmarks*) in the environment uniquely using the robot's sensors, and has limited ability for tracking such objects, which is used by the intervention recognition agent to support the identification of a lost or confused robot. 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. An intervention recognition agent uses this information to determine when a robot has returned to the same location too often or when a robot has returned to the same location too many times in a specified period of time. In either case, the operator should be notified so that the robot can be encouraged to explore different locations in the environment instead of spending too much time in the same area.

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 [12, 8], 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 sensing 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 [15]). Briefly, 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 the *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 as an array of perception instances. The perceptions stored in the perceptual memory do not attempt to create a representation of the world other than the basic landmarks described above: they are stored as raw perceptions that the perceptual schemas can use to identify interesting things in the environment.

There are three intervention event objects, polled regularly by the intervention recognition object to inquire if the state of the robot must be changed. These three objects address each of the three important conditions described above: *confused identifier*, *stuck identifier* and *victim identifier*. Each of these event objects contains a link to the current perceptions of the robot via the perception class. The stuck identifier object looks at the agent's current coordinates and compares them to the robot's location four cycles previous. If the current location and the location four cycles ago are the same, and there is a movement instruction being sent to the robot with a speed higher than 0, the agent is considered stuck. The victim identifier relies solely on the victim perceptual schema mentioned previously, while the confused identifier relies on counting the number of times a robot has been within perceptual range of any given landmark (a perceptual schema for landmark identification must be supplied for any domain in which the robot team is deployed). If the count surpasses the threshold, the robot is identified as confused. The system is extendible since new intervention events can be coded as additional intervention event objects.

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 [10]. 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 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 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.

Our evaluation of this approach is divided into two parts. First, an evaluation of the performance of the blended teleautonomous approach in comparison to autonomous and teleoperated approaches under varying environmental conditions, in order to demonstrate the utility of the approach as environments become increasingly complex. Second, a similar evaluation where the number of robots under control varies, in order to examine the benefits of the approach in terms of allowing an operator to control a larger number of robots effectively. Each of these evaluations is presented in separate subsections.

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. In all cases the same single human teleoperator was used. Extensive results are detailed in [15]; because of space limitations we present a subset of the results related to the coverage of the environment by a team of three robots, the time robots spent immobile, the number of victims found by a team of robots and the number of interactions between the operator and the control system.

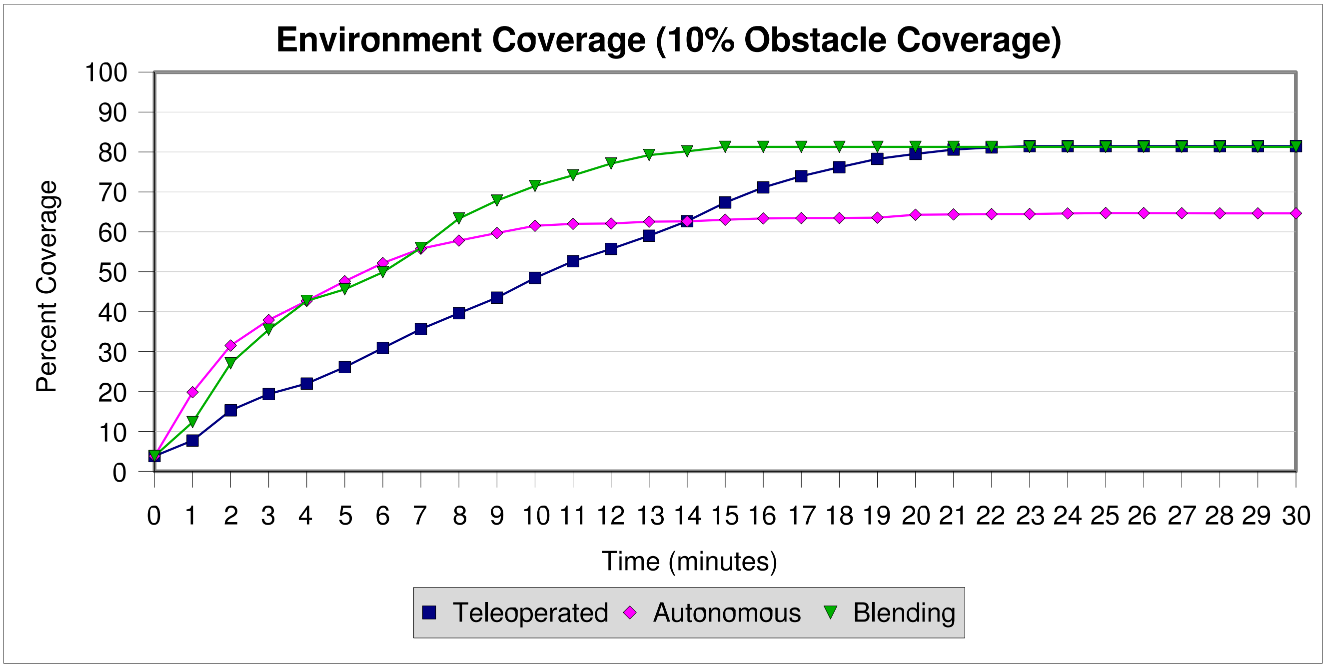

Figure 2: Average (n=5) environment coverage achieved by autonomous, blended teleautonomous and teleoperated robot control in environments where 10% was covered in obstacles.

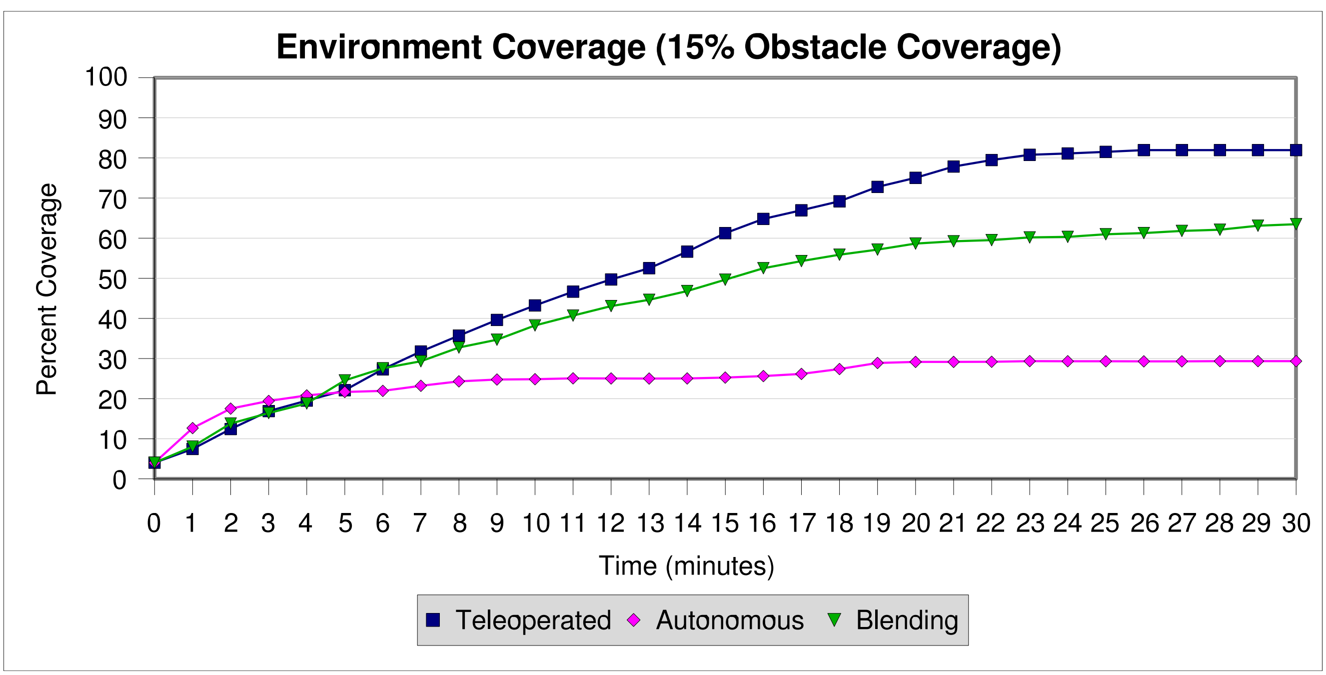

Figure 3: Average (n=5) environment coverage achieved by autonomous, blended teleautonomous and teleoperated robot control in environments where 15% was covered in obstacles.

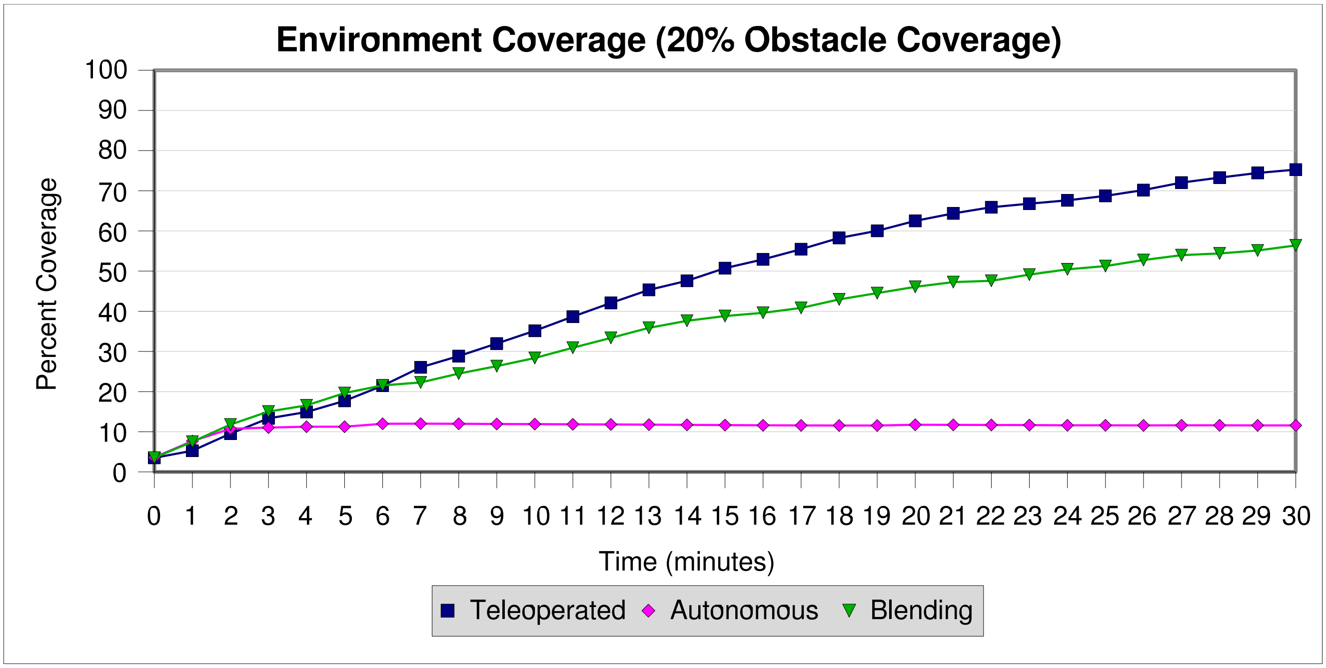

Figure 4: Average (n=5) environment coverage achieved by autonomous, blended teleautonomous and teleoperated robot control in environments where 20% was covered in obstacles.

Figures 2, 3 and 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 limitations). 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 then the autonomous control system could. Some unexplored areas were unlikely to be found by the autonomous robots because of the unique obstacle configurations in those unexplored areas. That is, while we check to ensure the robot can physically fit in any hallways formed, they may still be narrow enough that the robot's motor schema for avoiding obstacles would slow down the robot's progress. Teleoperated control performed slightly better than blending control at the 15% and 20% obstacle coverage levels (though it did not at 5% and 10% 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 an area that an operator just steered the robot away from.

The time each robot spent immobile with respect to autonomous versus blending robot control (see Figure 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 often able to free the robot. Since the operator was notified by an intervention recognition agent whenever a robot became stuck, the operator was often able to free the robot in a timely manner, reducing the amount of time any particular blending robot spent immobile. In the lower obstacle coverage trials (5% and 10% obstacle coverage), robots became stuck less overall. Moreover, when robots did get stuck, they tended to get stuck less severely, and therefore it was easy for the operator to get the robot mobile again. 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. In trials where the obstacle coverage was 20%, the time spent stuck for the blending control system was much higher, since robots were often difficult to get mobile, leading to robots being abandoned. Blending operator instructions with the autonomous instructions contributes to a significant increase in effectiveness of robots, which can be

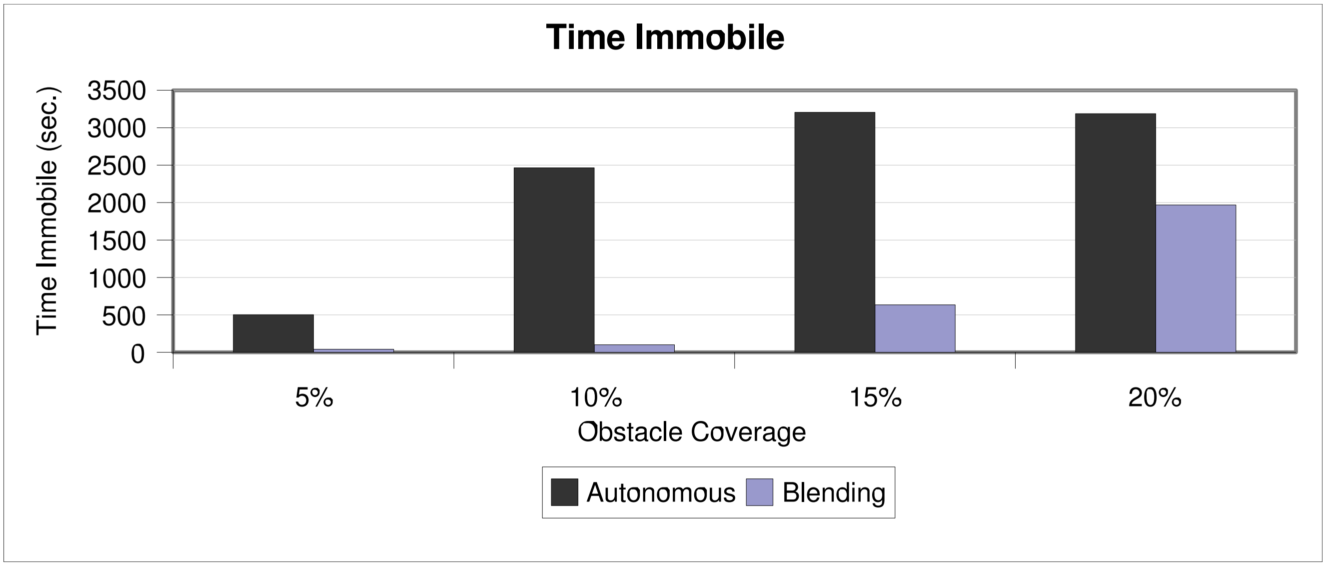

Figure 5: Average time in seconds spent immobile by environment difficulty, for blending and autonomous robot control.

observed by comparing the results of the autonomous trials and the blending trials.

With respect to successful victim identification, we found that the robots using blended teleautonomy had an advantage over both teleoperated control and autonomous control (see Figures 6, 7 and 8). At least a few victims in any experimental scenario were reasonably out in the open and easy enough to navigate to and find autonomously, and both blending robots and autonomous robots could take advantage of this. Correspondingly, only very little attention on the part of the operator was needed for blending robots. Later on in the trials, when the victims in the open were all located, the blending robots performed better then 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 Figure 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

The results above illustrate the effectiveness of our approach as environments become more complex. Not only can complex environments be covered more effectively than using either solely autonomous or solely teleoperated robots, but the number of human-robot interactions is significantly decreased over the use of teleoperated robots. All of the results above, however, use only three robots. To supplement these results, we investigated the effects of increasing team size, in order to examine the efficacy of the approach where a heavier operator load was involved.

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 Figure 10). Increasing the obstacle coverage level from 10% to 15% further increased the advantage gained by increasing the number of robots per team (see Figure 11). On average a team of three autonomous robots 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. Notice that a team of nine autonomous robots performed better in the earlier portion of the simulation (up to 15 minutes) then a blending team of three robots and has performance comparable to a blending team of six robots. In all cases the

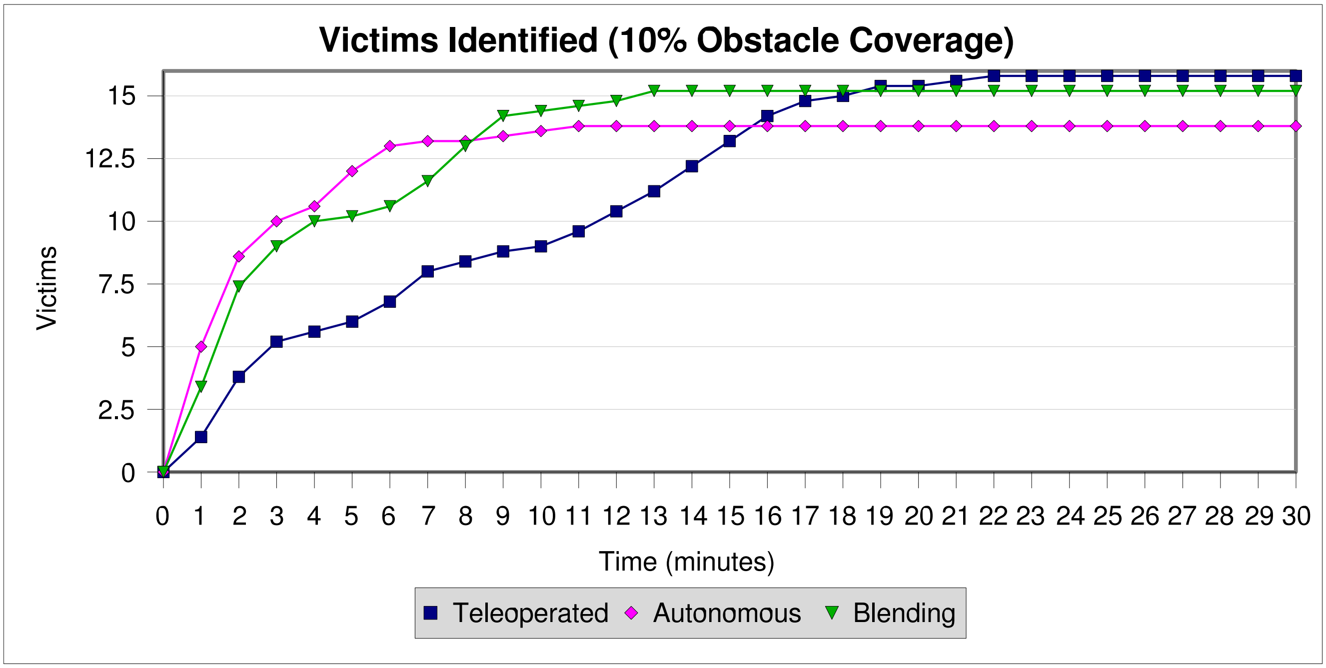

Figure 6: Comparison of number of victims identified in teleoperated, autonomous, and blending experiments in environments where 10% was covered in obstacles. All results are averages over 5 trials.

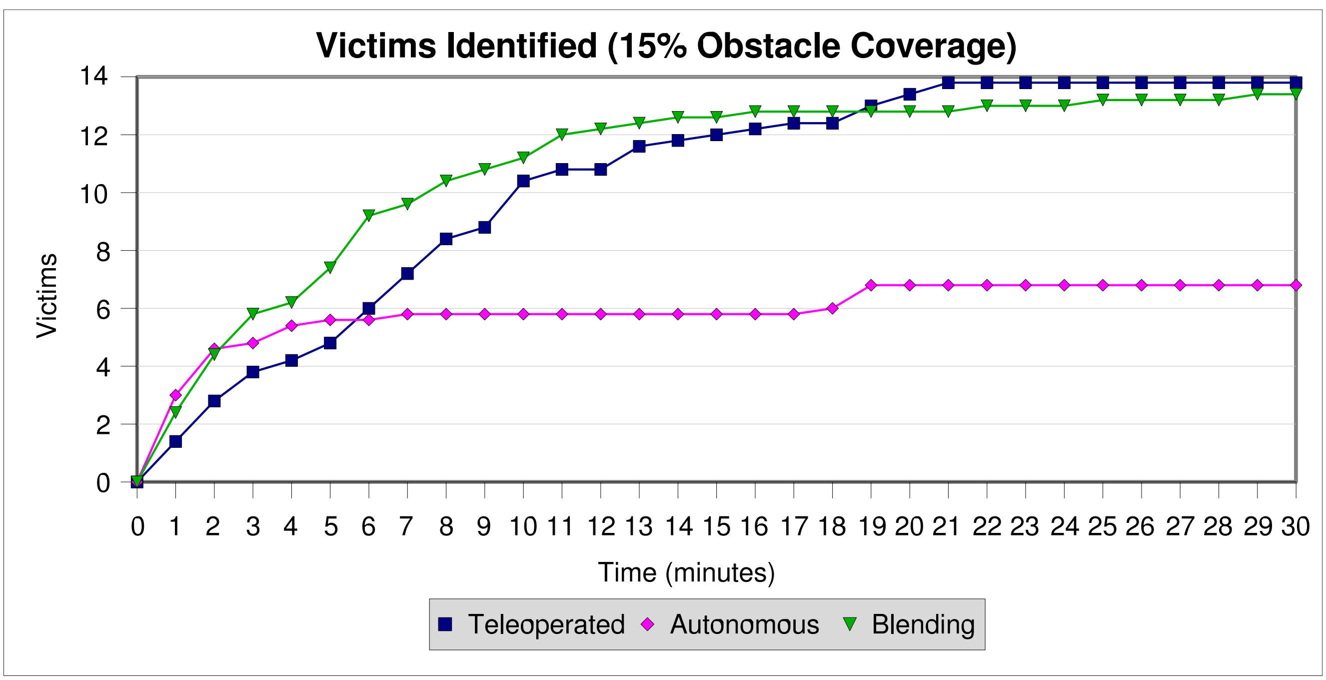

Figure 7: Comparison of number of victims identified in teleoperated, autonomous, and blending experiments in environments where 15% was covered in obstacles. All results are averages over 5 trials.

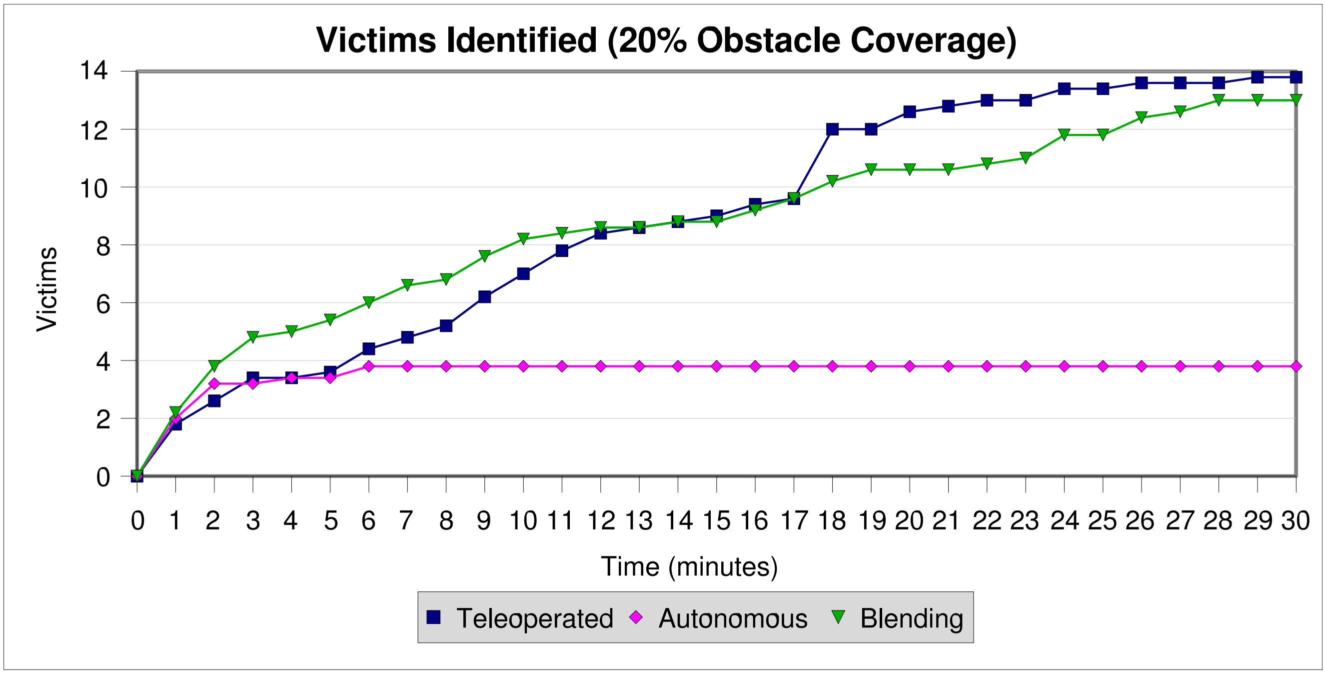

Figure 8: Comparison of number of victims identified in teleoperated, autonomous, and blending experiments in environments where 20% was covered in obstacles. All results are averages over 5 trials.

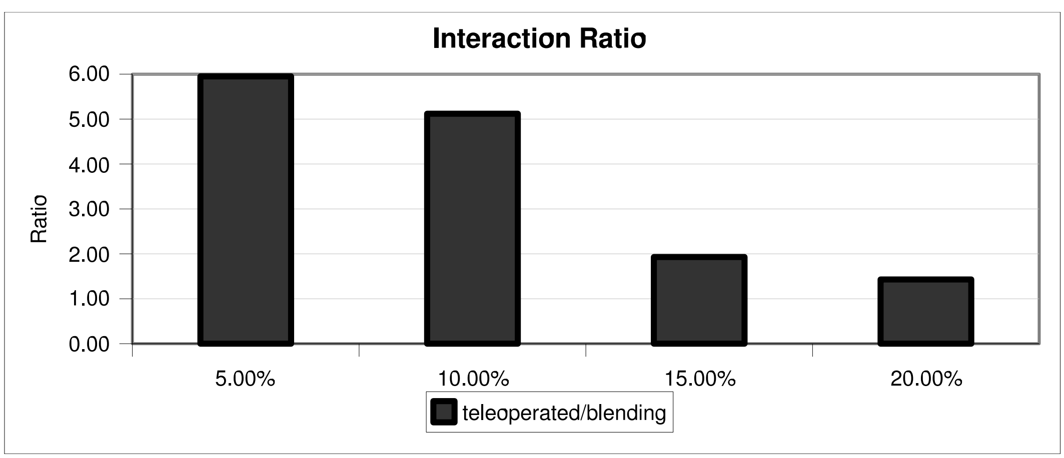

Figure 9: Average (n=5) ratio of operator interactions (teleoperated/blended teleautonomous in environments with 5%, 10%, 15%, and 20% obstacle coverage.

blending robots outperformed the autonomous robots with equal team sizes.

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 Figure 12). Increasing the obstacle coverage from 10% to 15% increased the performance improvements. The blending robots clearly have an advantage over autonomous robots (see Figure 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. The performance gain for blending robots was substantial when increasing the number of robots from three to six but almost negligible when increasing from six to nine.

Increasing the number of robots certainly increases the performance of the robot teams. However, the performance gain is more substantial when the number of robots increases from three to six, and further increasing the number of robots does not produce a significant amount of performance increase (i.e. increasing the team size from six to nine). In the cases described above, the performance gained by increasing the number of robots per team is not linear: as the number of robots increase, the performance gain decreases.

5. Future Work

There are a number of directions that future work in this area can profitably take. In terms of immediate further experimentation, we are currently examining the efficacy of this approach as the number of robots increases (preliminary results are presented above), both to examine how far the benefits obtained can be pushed, and to help identify future improvements that will allow the operator to better control larger teams of robots. We are also working on extending the knowledge bases used by the software agents, since the broader the range of situations these can recognize and deal with, the better the performance of this approach should be.

One of the most obvious extensions to this work is the application of the blending control system on physical robots. Since this work was done using the Player/Stage application suite, all code written to control the simulated stage robots is directly compatible with physical Pioneer mobile robot platforms. However, the code used in this thesis was not verified on a set of physical robots. Extending the blending control system to work with other mobile robot platforms is another goal of future work in this area. There are several issues that have to be addressed if this system is going to be applied to physical robots. First, on real robots, perfect localization is no longer a simple assumption. Odometry on real robots is likely to have at least some noise, and that noise will be cumulative. The application of vision and other sensor technology would have to be employed in order to have useful localization. Another assumption that has to be dealt with is the increase in sensor noise and environment complexity. Vision in particular will be a challenging problem in a real robot compared to a simulated one, and a more sophisticated method for handling errors will have to be developed. The approach and much of the code written for the simulated blending system will be applicable on the physical robots, but the underlying infrastructure will require much additional work.

In terms of the overall approach, the major planned extension of this work is to allow support from peer robots as well as human operators. The potential for this is already present in this approach in that an intervention recognition agent has the ability to send requests for advice or assistance to other robots and their agents. The mediation agent was also designed so that it could be extended to include instructions from peer robots instead of human operators. We intend to extend the autonomous control mechanisms to deal with requests for assistance or information from other robots, and integrate consideration of other robots into the knowledge bases used by the intervention recognition agents and command evaluators. This will allow research into the efficacy of making robots more proactive in terms of assisting one another in this domain (even volunteering information rather than being asked, for example).

In addition, there are opportunities for user modelling of the operator (and for modelling other robots when those extensions are complete). The blending system described in this research decreases the amount of risk to the robot, but a persistent operator is still able to greatly effect the robot's resulting actions. If the agent could model the operator and develop a level of trust, the decision of whether commands are blended and to what degree could be influenced by how much trust the robot's control system has with a particular operator. Consider an inexperienced operator who unintentionally persists in instructing the robot to perform very risky tasks that are entirely avoidable, such as moving too close to unstable ground. If the control system "knew" that the operator was inexperienced, the robot might blend the inexperienced operator's instructions differently. Imagine now that instead of an inexperienced operator, a malicious operator may try to take advantage of robots and hinder their progress for the operator's own means. In either situation, whether the operator is inexperienced or malicious,

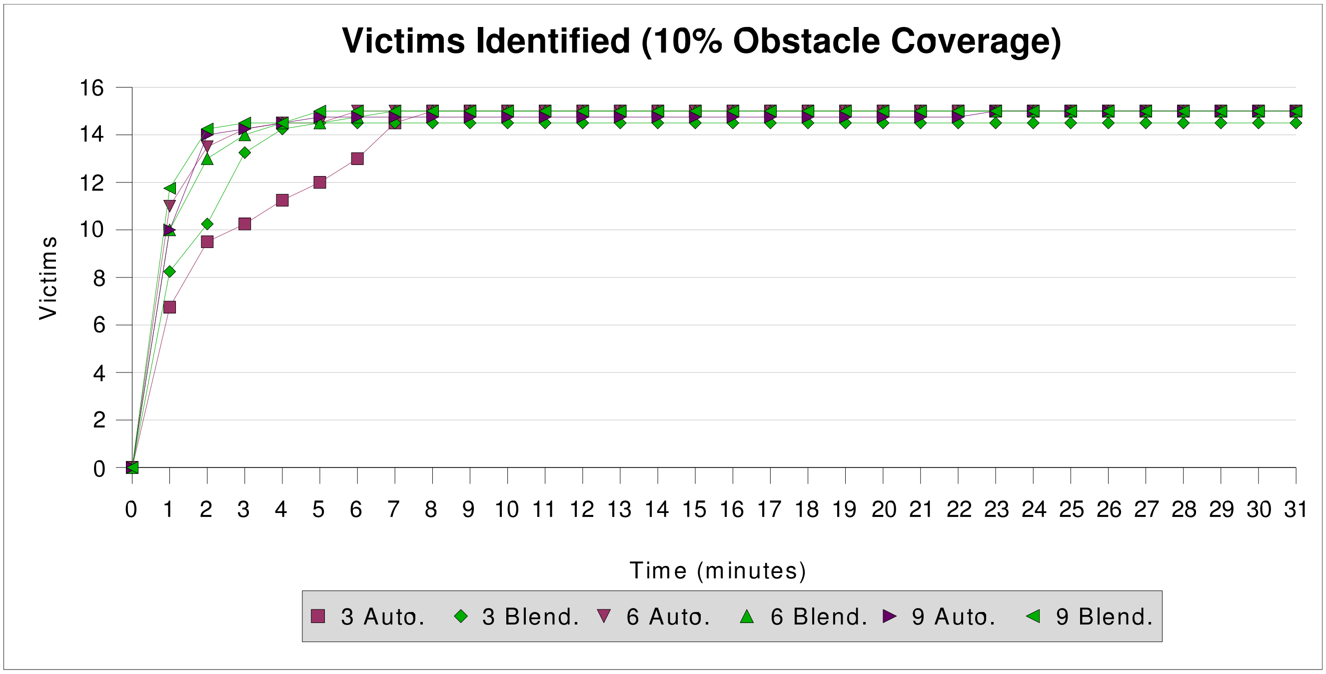

Figure 10: Comparison of number of victims identified in autonomous and blending experiments with different numbers of robots (3, 6 or 9) in environments where 10% was covered in obstacles. All results are average over 4 trials.

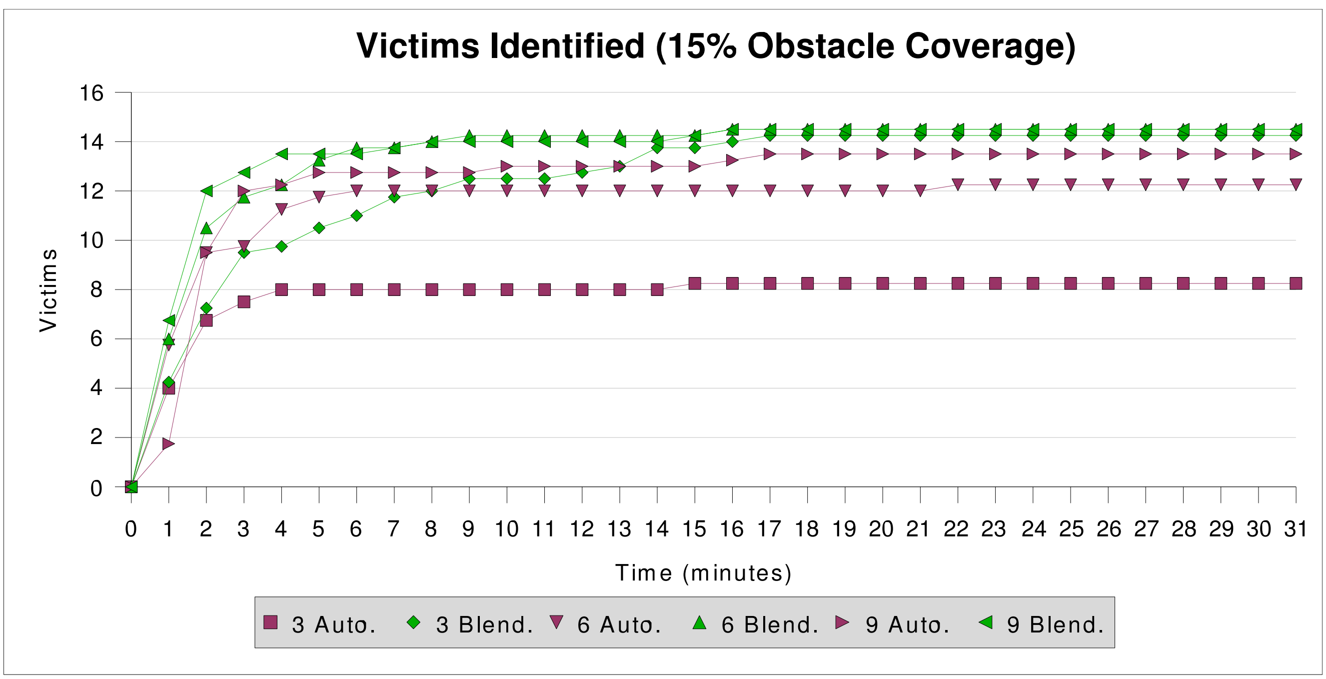

Figure 11: Comparison of number of victims identified in autonomous and blending experiments with different numbers of robots (3, 6 or 9) in environments where 15% was covered in obstacles. All results are average over 4 trials.

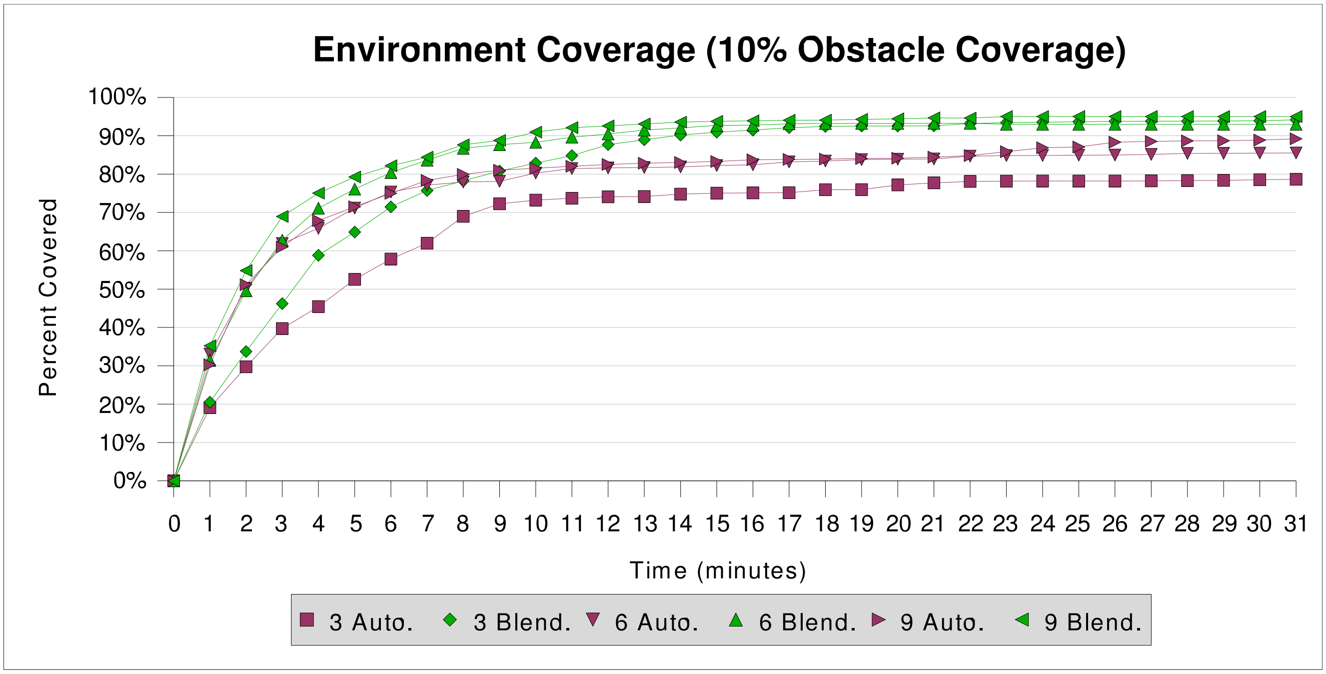

Figure 12: Average (n=4) environment coverage achieved by autonomous and blended teleautonomous robot control with different numbers of robots (3, 6 or 9) in environments where 10% was covered in obstacles.

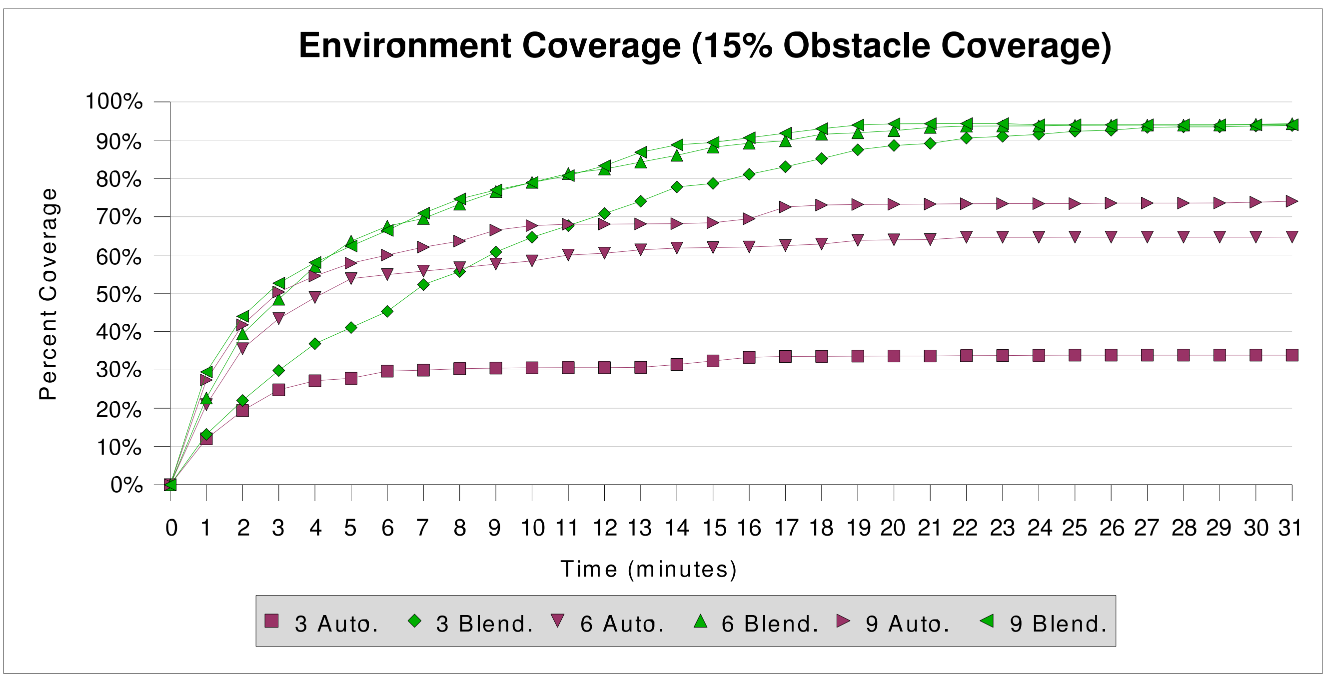

Figure 13: Average (n=4) environment coverage achieved by autonomous, blended teleautonomous and teleoperated robot control with different numbers of robots (3, 6 or 9) in environments where 15% was covered in obstacles.

the ability to model operators may have advantages. This type of research could blossom into multi-agent research, where robots keep track of operator reputations and share their reputations to benefit all members of a society.

Since there was only a single human operator for this research, the evaluation itself would benefit from a larger study involving a range of operators. This would involve recruiting a pool of subjects classified by experience with tools such as robots or remote-controlled vehicles (it would be useful to employ a psychological spatial-visualization test as a tool for categorizing subjects by their skills at visualizing remote situations). Additional criteria for measuring cognitive load would also have to be devised to supplement to the heuristic measurement used in this research. A usability study could provide a large amount of insight into the relationship between the human operator and the robot. This could lead to improvements in blending by taking advantage of some of the natural ways that human operators interact with robotic control software.

Finally, there is the opportunity for future work in user interface development. A well designed user interface can encourage easier interaction between the operator and a robot team. The user interface employed here is sufficient for this research, but it could be improved to be more operator friendly. It would be useful to experiment with changing the user interface to try to increase the immersiveness and allow the user to feel more "in control". Traditional user interfaces often seem passive, where the operator is watching the robot perform instead of being immersed with the robot in the environment.

6. Conclusion

This paper has described facilities for balancing autonomy and teleoperation effectively for a complex environment, Urban Search and Rescue. Further explanation of the other components that make up a complete teleautonomous navigation system (e.g. the user interface, the waypoint manager, etc.) may be found in [15].

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 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. The results have shown the blending supported by our control mechanism allows a human operator to more accurately control a group of robots, and the system to find victims faster and with fewer errors, than relying on autonomy or teleoperation exclusively. We have also indicated a number of directions for future experimentation and research.

7. References

[1] Ronald C. Arkin. *Behavior-Based Robotics*. Cambridge: MIT Press, Cambridge, MA, 1998.

[2] 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.

[3] 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.

[4] 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.

[5] 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.

[6] 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.

[7] 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.

[8] 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.

[9] 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.

[10] 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.

[11] 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.

[12] 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.

[13] 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.

[14] 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.

[15] Ryan Wegner. Balancing robotic teleoperation and autonomy in a complex dynamic environment. Master's thesis, University of Manitoba, July 2003. Department of Computer Science.