Peer Instruction to Assist a Teleautonomous USAR System

Michael Gauthier and John Anderson
Department of Computer Science, University of Manitoba, Winnipeg, Canada R3T2N2
umgaut03,andersj@cs.umanitoba.ca

Abstract

While the goal of Urban Search and Rescue is the autonomous mapping of a damaged building coupled with the location of human victims, artificial intelligence is not yet advanced enough to fully support such a rich and complex domain. Human teleoperation of robots is still relied upon extensively in this and many other domains, but suffers because of operator fatigue and problems with situational awareness. These two factors have led to recent research on combined teleautonomous approaches. The work described here is intended to supplement teleautonomous approaches in multi-robot settings by allowing robots to recognize specific situations in which they can assist their peers, thus allowing a teleoperator's limited attention to be better spread around a team of agents. We overview the techniques we employ to assist others in the robotic rescue domain, and describe how they integrate into an existing teleautonomous robot control system for the robotic rescue. We also evaluate the performance of these facilities in a simulated robotic rescue domain.

1 Introduction

Robotic Urban search and rescue (USAR)involves exploring collapsed structures to map the environment, locate human victims, and warn human rescue personnel of potential dangers. This domain represents both an important technological application as well as a significant research challenge for robotics and artificial intelligence. Despite the recognition of recent technological advances in autonomous systems, including a new fully-autonomous USAR league at RoboCup, the current state of the art in artificial intelligence (AI) severely limits success in an application as complex as this one. The environment makes it difficult for robots to maneuver, while debris makes sensing and localization difficult and prevents robots from taking advantage of the structure inherent in most indoor domains. Infrastructure damage can also make communication sporadic and error-prone. The task itself is also demands a very broad range of skills from an autonomous agent (in perception, mapping, and commonsense reasoning, for example). Because of a these factors, human teleoperation is still extensively relied upon in this and many other robotic tasks. For example, at the NIST robotic USAR competition at IJCAI in 2003, our entry [2] was one of only two running fully autonomously.

Teleoperated robotic control is obviously much preferred over having human rescuers in harm's way. However, teleoperation is limited not only by the number of operators available, but by the skills of that operator and their deterioration over time. Casper and Murphy [3, 4] describe the operator fatigue that occurs very quickly in real-world USAR situations, such as their work at the World Trade Center in 2001, as well as the errors in control and properly recognizing visual cues that arises from this. Operators also have a difficult time reconstructing the robot's perceptual space while also processing information in their own perceptual space, a factor commonly called cognitive overload [1]. Cognitive overload is an even greater factor in the case where a team of robots must be controlled, since this multiplies the perceptual information that must be processed.

One of our research goals in robotic USAR is to bridge the gap between these two approaches, in addition to the obvious goal of improving autonomous processing. We do this in two ways: first, by enhancing the information provided to a human operator to deal with the problems of cognitive overload (e.g. [7]), and second, by attempting to combine an appropriate balance of autonomous and teleoperated processing in a teleautonomous approach, in order to make the best use of an operator's limited attention.

In work toward the latter goal, we have previously described an approach to dynamically balance autonomy and teleoperation in a multi-robot system for a USAR domain [9, 10]. This approach is based on adding a knowledge-based component to each robot that allows for the recognition of situations of interest in the USAR domain, allowing an operator to be interrupted only when necessary. Experimentation with this domain has shown that the approach is both more functional than teleoperation or teleautonomy alone, and allows an operator to successfully control a larger group of agents than would otherwise be possible [10]. This paper extends this work by adding the ability to allow agents on a multi-robot team to intelligently assist one another, removing more of the operator's workload and allowing the operator's attention to be better shared.

We begin with an overview of previous work in this area, including details of our own previous work. We then describe the agent observation and advice-giving mechanisms, and evaluate this approach by adding it to the approach of [10] and examining performance in the same domains. This evaluation shows that agents can assist one another while still achieving better environment coverage than without these mechanisms.

2 Related Work

Combining teleoperation and autonomy was done in a very basic manner in the 1990's by Arkin and Ali [1]. Their approach involved integrating these using a behaviour-based approach, and having a teleoperator integrated as one particular behaviour. By weighting the influence of the teleoperative behaviour, a group of agents could be influenced by an operator as well as their own desires. Trividi et al. [8] use teleautonomous robots to form a perimeter around a traffic accident, but only in a very limited scope: the robots' autonomous abilities involve forming a polygon around a set of points, and the operator simply supplies the points to be formed (i.e. a specific context for their autonomous behaviour). Crandall et al. [6] describe a system with five specific levels of autonomy for robots, but do not describe an implementation to illustrate balancing or adaptively selecting the most appropriate level for the current context. Murphy et al. [5] attempt to divide tasks between robot and operator (as has some of Murphy's previous work) - in this case, using an automated victim detection system while operators controlled robot movement.

The previous work with which this is closely associated is that of Wegner and Anderson [10]. Wegner and Anderson present an approach that achieves smoothly blended teleautonomy - allowing agents to be as autonomous as possible given the circumstances surrounding them - by starting with behavior-based autonomous agents with basic behaviours for robotic USAR (navigation, mapping, and victim identification). They implemented basic robotic teleoperation through a joystick control and manual operation of agent behaviors, and also high level settings such as waypoints for path planning. Blending is allowed through a multi-level setting, which incorporates the sliding scale of [6], but these settings are implemented by instructing the robots internally how heavily to weight autonomous or operator commands. The standard operating mode (Weighted Teleautonomy) lets agents function as autonomously as possible, judge themselves when this autonomy must be reduced by requesting operator intervention [9].

This approach is implemented using two software agents on each robot: one to appropriately mediate autonomous and operator instructions, and the other to recognize situations of interest. The former of these the *Mediation Agent* evaluates effectiveness the commands an agent generates and receives, using a knowledge-based system (the *Command Evaluator*) to predict the outcome of actions and analyze the consequences. Instructions are weighted based on predicted outcome, and further weighted based on the mode set by the operator: taking control of the agent completely, for example, is implemented by telling the mediation agent to completely discount agent instructions. As a result, agent or operator instructions may be followed to the letter, treated as two action vectors to be blended, or ignored. The command evaluator currently contains knowledge for moving too near obstacles and moving away from potential victims.

The second agent, the Intervention Recognition Agent, is intended to look for specific situations where operator attention is required, ultimately indicating when the balance of autonomy and teleoperation should be changed. This is also a knowledge-based component, and contains information about a specific subset of USAR tasks. Three tasks were implemented to deal with the most common requirements for operator intervention: stuck or immobile agents (recognized through a lack of progress despite executing movement commands), confused agents (a confused agent is one that repeatedly wanders through the same general area making no real progress, and can be recognized by repeated recognition of noted landmarks), and the detection of victims using a basic perceptual schema. When a strong likelihood of one of these situations is detected, the mediation agent is informed and the operator is signalled.

This approach has been implemented for pioneer robots and tested using USC's Stage simulation software. It has proven to be extremely functional, allowing an operator to control a larger set of robots and to achieve higher environment coverage and locate more victims per unit time than either teleoperation or autonomy alone. The continuing work presented here takes advantage of the fact that this is a multi-robot system, extending this approach to allow advice to be given by other agents, in addition to a human operator, so that intervention requests can be reduced and an operator's limited attention better utilized.

3 Peer Assistance

Robots in a USAR domain will encounter one another repeatedly in the environment, given a large enough team and a reasonable operating area. Such encounters are used regularly in multi-robot systems to exchange maps and localize, for example. Here, we take advantage of these encounters to observe the current state of robots in the environment, offering third-party assistance as necessary. Since the framework of [10] already allows instruction to be blended, additional sources of instruction fit naturally into this framework, and so we have designed this instruction software as an addition to this approach and employed Pioneer robot models with only camera and sonar sensors (i.e., no laser scanner). The intent of this is to supplement teleautonomy with additional assistance, and as such we have attempted to design our approach to add functionality without strong cost, in order that situations that prevent peer instruction from being useful do not negatively affect the entire system. Our peer assistance approach consists of four phases: *recognition*, *diagnosis*, *prescription*, and *active monitoring.* This section explains each of these phases in turn.

In order to be able to offer assistance to others, we must be able to recognize when we have encountered another robot. Our approach scans a camera feed for a red region that approximates the color and size of a Pioneer robot. A region that is too small could be a robot that is too far away to be of real help, or simply a small red object, and is currently ignored. A region suitable for a reasonably close robot must still be verified as such, and this verification involves polling robots of their locations to determine if this is plausibly a teammate. This necessitates a common coordinate system. Teammates answering with coordinates are compared to the apparent location of the red region, based on the observer's location, and anything within a small error range is considered a positive match. It is possible that the robot cannot answer due to communication interference, in which case no assistance will be offered, but there is little lost from the original approach save a brief moment to check this. It is also possible that two robots (or one robot and another red object) are close enough that they are misidentified. In this case, advice will not likely be as useful, depending on the degree of error. The original model, however, can deal with inappropriate advice to the degree the command evaluator can recognize the consequences.

Once an agent is recognized, the observer stops (to make it less likely that the observed agent will disappear from view) and requests data from the observed agent. This is responded to by the state of the agent (whether it is observing, for example - so that two agents do not both stop and observe one another forever), its location, and four of the eight sonar readings. These readings are stored in two forms: first, a short term sample (the most recent four snapshots, if four are available), designed to be enough information to solve immediate problems, such as being stuck in a hard-to-maneuver location. Problems also exist that cannot be solved with a small set of data points, such as wandering aimlessly. For these situations, we store a long term sample consisting of the last 30 records obtained from each individual agent.

Once data is available, an observing agent can attempt to diagnose any problem that exists. Here we are restricting ourselves to the same knowledge used in the original teleautonomous system, and are attempting to give advice on stuck and confused agents. When a short term sample is complete, the observer will attempt to diagnose a stuck agent. Sonar readings that are small and unchanging can be indicative of the robot's being stuck against debris on one or more sides, while a simple lack of change in position can be diagnosed as stuck on debris or in a corner. When a long-term sample is available the observer can analyze for signs of confusion. This is diagnosed by examining the reported coordinates, and any signs of repeatedly visiting the same location (temporally disparate records in the long-term sample showing coordinates within a specific distance from one another). If confusion is diagnosed, the observer will study the orientation of the observed robot, since if the robot is no longer heading in the same direction, it may already be over its previous confusion.

After diagnosis, advice can be prescribed to the observed agent. To determine appropriate prescriptions, we ran many trials correcting stuck and confused agents, in an attempt to note repeated patterns. For a stuck agent, we attempt to narrow the problem down to the particular side of the robot using sonar, and prescribing a shifting movement in a reciprocal direction. While the agent itself can of course access its own sonar and diagnose from which side it might be stuck, the observing agent has the advantage of noting obstacles around the stuck robot that may not be readily apparent to the robot itself, making it less likely that a prescribed course of action will put the robot in a worse position. If a stuck side cannot be determined, the robot is assumed to be stuck from the front, and a similar reverse movement can be prescribed (this assumption was made because under experimentation, in situations where a stuck side cannot be determined, turning and moving often got robots in worse positions than they were originally).

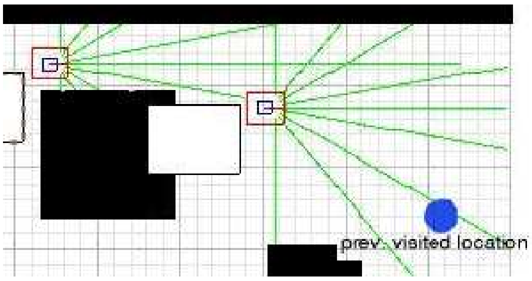

Figure 1: Advising a confused agent

For a confused agent that is heading towards a location at which it has previously been observed, the direction of approach is noted. Simply prescribing the opposite direction is too simplistic a solution, because it does not take into account obstacles such as walls. For example, an agent approaching a previous location from the left, as shown in Figure 1, may have a wall to its own left, and cannot simply be told to move left rather than right. Our advice thus takes into account perceived obstacles as well, and would advise the agent to move right in the above case. This introduces the need for active monitoring, as situations such as the above require more than one piece of advice in order to solve the problem.

When agents receive advice, it currently overrules the agent's own autonomous processing. We found that blending autonomous instructions with advice, where advice did not take priority was ineffective because the agent's autonomous systems were not helping at this point (or it would not be stuck/confused), and blending caused the outcome of the blending no longer properly followed the advice. While the agent's autonomous processing is overridden, the other components of the teleautonomous approach, such as the command evaluator, can still be used to evaluate outcomes and blend with operator instructions if desired. This also deals with the case where advice is being given and the operator is independently issuing instructions.

The final phase, active monitoring, is used to deal with situations such as Figure 1, where sequences of instructions are necessary, as well as the more general problem of when to stop giving advice. Once an agent begins receiving advice and its autonomous system is overruled, it continues listening for further advice, until the observing agent ceases providing it, allowing sequences of instructions to be carried out. Similarly, the observing agent monitors the progress of the agent it is assisting. Records collected during active monitoring are not stored, but are examined as they are received in order to provide a continual stream of advice in real time. In stuck situations, records can be examined to see if an agent is reversing into a wall, for example. If the advice is not having the intended effect, it can be re-diagnosed and new advice offered (in this case, altering the suggested direction of movement).

With confused agents, active monitoring is beneficial because in telling an observed agent to turn away from a previously visited location, the observed agent may then turn towards another previously visited location (which can be noted because the observer also has the long term sample of records). When the observed agent is satisfied the observed agent is out of difficulty, it ceases instruction and moves on. A time limit is placed between short-term record gathering to allow the observed agent to move out of sensor range so that it is not observed again immediately. A timeout is also used in the case of communication failures, so that an observer will eventually give up assisting a robot that is not responding.

4 Evaluation

In order to evaluate the efficacy of these peer assistance techniques, we used the precise test domains (simulated USAR arenas 20 meters on a side) of Wegner and Anderson [9, 10], and integrated the peer assistance techniques described above into the autonomous robot code used by Wegner and Anderson, thus allowing a direct comparison to their autonomous and teleautonomous approaches. One of these domains is shown in Figure 2. Sonar traces are shown on each robot, and the square near each robot shows relevant information captured by the robot's camera. Here, one robot is assisting another on the middle right, and both are seen by a third agent above. The assisting robot ultimately becomes stuck as well, and the third robot arrives and assists both.

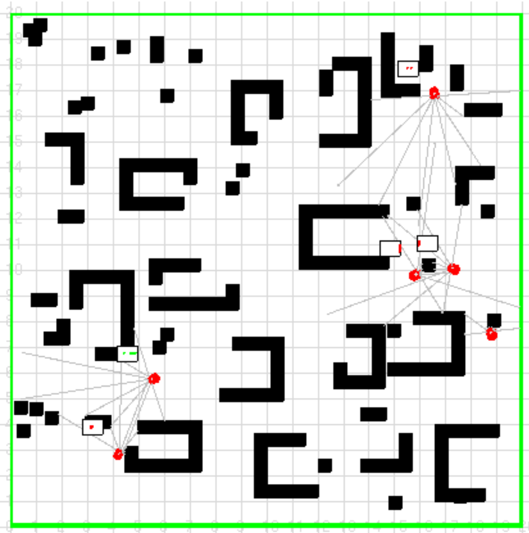

Figure 2: Sample implementation environment

In order to measure agent performance, we used total environment coverage over time as a metric - that is, the percentage of the environment successfully mapped. Wegner and Anderson's [10] original studies were performed over domains with 5, 10, 15, and 20% obstacle coverage, and showed a dividing line in performance between 10% and 15% coverage domains, so we chose the 15% coverage domains to replicate. Here (Figure 3), a team of 6 robots using our peer advice approach and no operator intervention achieves 83% environment coverage on average over five different domains verified to be of equal difficulty in terms of navigation (in terms of the number of local minima and domain accessibility). This compared to 65% for autonomous agents over the same average, and 93% for agents using blended teleautonomy.

This is a very positive result, showing that with these techniques we can achieve results close to that of employing a human teleoperator. It is especially positive in that without the assistance of a teleoperator, all agents eventually become stuck in situations that were beyond the scope of other agents to assist (this situation occurred from 9 to 29 minutes into a trial, and averaged 20.5 minutes). Agents are thus performing their useful work in a smaller time frame than the approaches they are being compared against. Moreover, they derive most of their benefit from helping one another early (when the majority of agents are not in trouble, and so can observe and assist). This is shown in the faster environment coverage at early points in the trial - even faster than those assisted by a teleoperator. The addition of a teleoperator to these agents would only further enhance this approach. Figure 3 also verifies our approach in that greater environment coverage is achieved over autonomous agents even considering that agents are not doing useful work when they are stopped assisting others.

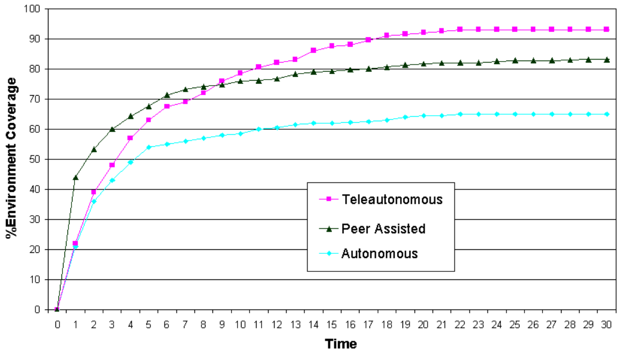

Figure 3: Percentage environment coverage over 30 minutes for each control scheme

While this would appear to show that a teleoperator could be dispensed with, there are important caveats to this. First, such a positive result is dependent on a subtle balance between the size and complexity of the domain and the number of agents in it. If the the population is not high enough, agents will become distressed before anyone can observe and assist them, and the entire population can become immobile earlier. An equilibrium between robots becoming stuck and those able to give assistance must be achieved. This equilibrium ultimately broke down in each trial we ran - however, a great deal of useful work was accomplished before this occurred. However, given that our goal is to allow a teleoperator to control a larger population of robots, these techniques show that with such a population available the teleoperator will ultimately be interrupted less frequently. Indeed, a teleoperator could focus a good portion of the time simply keeping enough agents running in order that others could be assisted. We have not yet run an experiment to illustrate the maximum number of robots controllable by a teleoperator with these results integrated.

5 Conclusions and Future Work

This paper has illustrated the techniques we employ to allow robots to assist one another in a robotic

USAR domain. We have evaluated these techniques against the approach of Wegner and Anderson and have shown that when enough robots are available we can approach the abilities of a human teleoperator in this setting.

These are only preliminary results, because we are examining only a subset of the potential difficulties that can emerge in a domain as complex as USAR. An agent being stuck, for example, will never be overturned or on a severe incline, which can happen easily in even the easiest USAR arenas. Similarly, subtle problems, such as an electrical cord blocking one wheel, cannot be noticed with sonar alone, and are difficult for even a teleoperator to diagnose remotely. Other difficulties beyond being stuck and confused also occur routinely – these were chosen because performance data for teleoperated, autonomous, and teleautonomous approaches for dealing with these problems was available. However, so long as the work required to give advice does not take strongly away from the time devoted to dealing with this domain, which our data supports, these are useful to deploy.

Much potential also remains to be exploited by these techniques. Currently most assistance is based on data from the robots themselves, and must be communicated. Value added by the third party observer is mainly in the form of knowing about obstacles in the environment that are not obvious to the agent in trouble. By employing more data available only to a third party – realizing through vision that a stuck agent is at a different orientation than it thinks it is, for example – much greater value can be captured in advice. This will require much more sophisticated vision than we currently employ. These results are also only verified within Stage, and must be deployed on real robots. We currently have two pioneer robots running this software, but have yet to examine its performance in the real world. This will ultimately require more physical robots, as the likelihood of one robot encountering the other in trouble in the real world repeatedly is not large.

Finally, the assumption of a shared coordinate system needs to be removed. We are independently working on identifying shareable landmarks in the environment and basing orientation and distance elements in instructions from these, which will remove this assumption in future. This same work also involves identifying useful features such as doorways or branches in paths through the environment, which can both serve as landmarks and on improving advice (advising an agent to move away from a path that the observer has already explored, for example).

References

[1] Ronald C. Arkin and Khaled S. Ali. Integration of reactive and telerobotic control in multiagent robotic systems. In *Proc. of the 3rd International Conference on Simulation of Adaptive Behavior*, pages 473–478, Brighton, August 1994.

[2] 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.

[3] Jennifer Casper. Human-robot interactions during the robot-assisted USAR response at the world trade center. Master's thesis, University of South Florida, April 2002.

[4] 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.

[5] Jennifer Casper, Robin Murphy, Mark Micire, and Jeff Hyams. Mixed-initiative control of multiple heterogeneous robots for USAR. Technical report, University of South Florida, 2000.

[6] Jacob W. Crandall and Michael A. Goodrich. Experiments in adjustable autonomy. In *Proc. of the IEEE Systems, Man, and Cybernetics Conference*, pages 1624–1629, 2001.

[7] Brian McKinnon, Jacky Baltes, and John Anderson. A region-based approach to stereo matching for USAR. In *Proceedings of RoboCup-05*, Osaka, July 2005.

[8] Mohan Trivedi, Brett Hall, Greg Kogut, and Steve Roche. Web-based teleautonomy and telepresence. In *Proceedings of the 45th Optical Science and Technology Conference*, pages 81–85, San Diego, 2000.

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

[10] Ryan Wegner and John Anderson. An agent-based approach to balancing teleoperation and autonomy for robotic search and rescue. In *Proceedings of the Agents Meet Robots Workshop, 17th Conference of the Canadian Society for the Computational Studies of Intelligence (AI-04)*, pages 1–15, London, ON, May 2004.