Towards Single-Operator Control of Tightly-Coordinated Robot Teams

Gal A. Kaminka and Yehuda Elmaliach *

The MAVERICK Group
Computer Science Department
Bar Ilan University, Israel
{galk, elmaley}@cs.biu.ac.il

Abstract

There is growing recognition that many applications of robots will require a human operator to direct multiple robots that collaborate to achieve the operator's goals. However, the bulk of existing work in this area assumes that robots are independent of each other, and thus ignores key challenges and opportunities in monitoring and operating tightly-coordinating teams. This paper takes steps to address these open issues. First, we introduce a graphical display that explicitly shows the coordination in the team, in terms of the robots' state with respect to each other. As a result, the operator can easily detect coordination failures, even before these cause overall failure in the task. We also take advantage of the robots' teamwork, to allow the robots to actively assist the operator in maintaining her control of the team. We evaluate these techniques in several multi-robot tasks, and show that they lead to significant improvements in task completion time, and number of coordination failures during execution.

1. Introduction

There is need for human control of robot teams. While robots can do many mundane or dangerous tasks for us, in many cases we can not leave all decisions to them, for example for safety reasons, or to make decisions which the robots are unable to make. While the autonomous capabilities of robots increase in quality every day, many future applications would still require a human operator to direct multiple robots that coordinate with each other to achieve the operator's goals. Examples of such applications include search and rescue operations [5], multi-rover planetary exploration, and multi-vehicle operation [3].

Previous approaches to human control of multiple robots treat the operator's attention as a centralized resource, which is time-shared between the robots[2, 3, 1, 9]. Robots that require operator's assistance initiate or are issued *call-requests,* which are queued for the operator. The operator switches control between robots, and uses single-robot teleoperation with individual robots to resolve the call requests in some (prioritized) sequence. This method works well in settings where the task of each robot is independent of its peers, and thus the resolution of call requests can be done in sequence, independently of other call-requests.

Unfortunately, these centralized methods face difficulties in *coordinated tasks*—tasks that require tight, continuous, coordination between the robots, i.e., robot teams where robots are highly inter-dependent. First, due to the coordinated nature of the task, robots depend on each other's execution of subtasks; thus a single point of failure (e.g., a stuck robot) will quickly lead to multiple call requests. Second, the coordination state of the robots must be monitored in addition to their individual state; but inferring the state of coordination from multiple individual robot reports can be difficult for the operator. Third, when the operator switches control to a robot, the other robots must wait for the resolution of the call-request, because their own decision-making depends on the results of the operator's intervention. As a result, robots wait idly while the call request is resolved.

Thus two key challenges are raised in controlling a robotic team in a coordinated task. The first challenge is to integrate information from multiple robots so as to allow the operator to monitor their coordination, in addition to their progress towards the goals (*coordination monitoring challenge*). The second challenge is to allow the operator to act on call-requests such that resolution time is minimized (*resolution challenge*)

For example, consider the task of controlling three robots moving in formation (a task requiring tight continuous coordination between robots), by teleoperating the lead robot, and allowing the others to maintain the formation autonomously. The monitoring challenge is raised because the operator must monitor the formation itself—slowing down or speeding up the lead as necessary—and not just

* This research was supported in part by BSF grant #2002401, and by MOD grant 2935762101

the distance to the destination. To do this, the operator must integrate incoming information from each robot (e.g., the robot's camera view), which can be difficult. The resolution challenge is raised when a robot gets stuck. Since the task requires moving in formation, the continual movement of the robot (as well as the formation they will take) depends on which robot failed, and whether the failure is catastrophic. Thus most robots will be idle while the the operator attempts to resolve the fault (for instance, by teleoperating one of the functioning robots to provide video imagery of the stuck robot).

This paper focuses on these two challenges within the context of tasks requiring tight coordination. To address the monitoring challenge, we develop a graphical *coordination display* that allows the operator to visualize the robots' coordination—their state with respect to each other—and thus visually identify coordination failures before they become catastrophic. To address the resolution challenge, we take advantage of the teamwork of the robots, to allow the robots to actively assist the operator in resolving a failure. We examine a *distributed* control methodology in which functioning members of the team, rather than switching to an idle mode of operation, actively seek to assist the operator in determining the failure.

We empirically evaluate these approaches in experiments with human operators. The experiments included monitoring and controlling robots in two robotics team tasks requiring tight coordination: Movement in formation (and attending to broken formations) and cooperative box-pushing. Our first set of results show that the use of the coordination display tool can lead to significant improvement in the number of failures in these tasks. We also evaluate the use of distributed robot-operator teamwork in resolving failures. Under failure conditions, if one robot is stuck, the others actively seek the stuck robot, in order to assist the operator to resolve the failure. A second set of results shows that such teamwork leads to shorter failure recovery times, when compared to both fully-teleoperated and a totally-autonomous approach.

2. Background and Motivation

The work we present in this paper focuses on both a visual monitoring interface, as well as on a distributed collaborative control paradigm. Previous work on visual interfaces for multiple robots attempt to immerse the operator within the environment of the teleoperated robot, while facilitating switching control between robots. For instance, Adams et al. [1] investigated the use of a three-dimensional GUI that has selectable operation modes to switch control between robots, teleoperate a robot, create a navigation plan for the robot, or replay the last few minutes of the robot's task execution (for diagnosis of failures). Our work contrasts sharply with this approach, as we focus on a display that abstracts away the details of the robots' local surroundings, focusing instead on displaying their relative state, not their absolute state with respect to some environment.

The bulk of existing work on controlling multiple robots put the operator in a centralized role in attending to robots, and do not often distinguish between different task types on the basis of the coordination involved. Indeed, many existing approaches implicitly assume that robots are relatively independent in their execution of sub-tasks. As a result, a centralized control scheme does not interfere with task execution. Fields [2] discusses unplanned interactions between a human and multiple robots in battlefield settings. Robots are mostly autonomous, but may send *call request* messages to the human operator to ask for assistance. These call requests are queued, and the operator resolves the problems one by one. Fong et al. [3] propose a *collaborative control* system that allows robots to initiate and engage in dialogue with the human operators. The robots employ user-modeling techniques to improve their interaction with the operator.

In contrast to the above centralized approaches, we believe that resolving call-requests is in the interests of all robots currently coordinating with the robot requiring assistance—and thus they should be actively collaborating with the operator to resolve the call request. Other work has also examined distributed paradigms for human/robot interaction. Tews et al. [9] describe a scalable client/server architecture that allows multiple robots and humans to queue service requests for one another. Scerri et al. [7] describe an architecture facilitating teamwork of humans, agents and robots, by providing each member of the team with a proxy and have the proxies act together in the team. Our work differs from both of these investigations in that we do not attempt to put humans and robots on equal ground with respect to control. In our current work, only the human can initiate the distribution of a task to resolve a call-request. However, once so selected, the task is carried out in tandem by all members of the robotic team and the operator.

3. Socially-Attentive Display

We focus on coordinated tasks—robotic team tasks that require tight coordination between the robots. The hypothesis underlying our investigation is that such tasks require monitoring of the coordination in the team, i.e., explicit monitoring of the team-members' state with respect to each other. Such monitoring is called *socially-attentive monitoring* because it focuses on inter-agent relations, rather than their goals [4]. For example, in formation maintenance tasks, socially-attentive monitoring may include information about the relative positions of the robots within the formation (without reference to where the formation is heading).

A corollary of our hypothesis is that when an operator controls robots in a coordinated task, she will need to infer socially-attentive information if it is not directly available. Unfortunately, existing displays are not necessarily well suited to display socially-attentive information. Instead, they provide information about the current state of each robot, individually. Thus the operator is burdened with inferring the socially-attentive information that is required.

The key to the monitoring approach we advocate is to provide the operator with a socially-attentive display that integrates the raw information coming from each individual robot into a coherent visual display of the social relationship within the monitoring system. Using this display, the operator can easily identify coordination faults (if there are any) within the monitored team, with little or no need for inferring this information from the other displays. This eases the cognitive load on the operator in coordinated tasks. The socially-attentive display must complement the monitoring display associated with the task.

Towards this end, we developed a display called *the relation tool*. The relation tool is a graphical display of multiple robot states on a two-dimensional (2D) plane. Colored dots denote different robots. The positions of the dots denote their states, and thus the shape they make up—their relative positioning—denotes their relative states. In principle, a target relative positioning of the dots must be defined for each application, which signifies correct coordination. Every application requires its own method of projecting robot state onto a 2D plane, and a target shape that defines normative coordination. The key is that the operator should be able to see, at a glance, whether the shape being maintained corresponds to correct coordinated execution.

We investigate the relation tool concretely in two popular coordinated tasks: Formation maintenance and cooperative pushing. Both of these tasks have been implemented in our lab using Sony AIBO robots. Each robot has an onboard video camera and a infra-red distance sensor pointing at the direction of the camera. They transmit their video and sensor readings to the operator's station for monitoring (the operator cannot see the AIBOs directly). The operator can intervene at any time in the operation of any of the robots. The software components utilized and extended the Tekkotsu AIBO control software [8].

We begin with the formation task. The objective is to navigate a simple triangular formation (three robots), through a short obstacle course. A simple algorithm for maintaining the formation with these robots has the lead (front) robot teleoperated by the operator, while the two follower robots maintain fixed angles and distances to this robot using their sensors. The robots do not utilize any communications for maintaining the formation. We do not assume that a global (world-view) camera exists, since in most realistic applications this assumption would not hold.

Figures 1 and 2 show this task in action. In both figures, sub-figure (a) shows the actual position of the robots on the ground; (b) shows a screen snapshot of the relation tool; and (c) shows the split-camera view from each of the individual robots. Figure 1 shows a failed formation situation, while Figure 2 shows an example of perfect formation.

While the split-camera view does indeed provide indication of whether the formation is maintained, it relies on inference by the operator to do so. Moreover, it is difficult to see from the split camera view to what degree the formation is maintained, and which robots are lagging behind or to the side.

In contrast, the relation tool makes it easy, at a glance, to see whether the formation is maintained. It integrates and processes incoming sensory information from individual robots, using an averaging window to smooth over any erroneous jumps in sensor readings. It then translates the smoothed data into a 2D coordinates. Here, the translation from the relative states to be maintained to a 2D plane is relatively simple, since the task itself is defined in terms of relative positions.

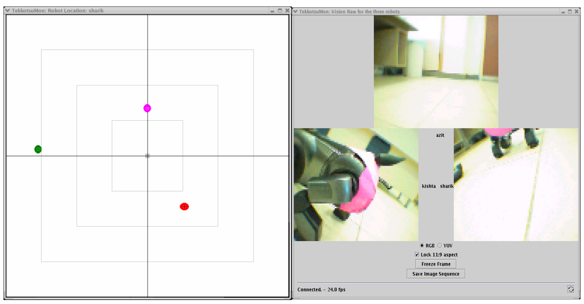
Initially, we set the coordinate system to the standard X, Y coordinate system. However, based on experiments with several operators, we have found that polar coordinates—angle and distance—improve the results. The X axis denotes the angle to the leader, while the Y axis denotes the distance to the leader. The position of the leader is always fixed. Using these coordinates, one can quickly determine whether the formation is breaking because a robot is lagging behind (distance too great), or its angle with respect to the leader is too sharp (e.g., because of a sharp turn).

We now turn to the cooperative pushing task. In this task, two AIBO robots jointly push a light-weight bar across the floor. The bar is color-marked, such that each robot can identify its position with respect to the bar. If the mark moves too much to the side, this would indicate a drift, i.e., the robot is either lagging behind or is pushing too quickly ahead. Again, the robots do not communicate with each other in this task.

Socially-attentive monitoring in the case of pushing involves only one dimension—the robots are to maintain equal speeds. We thus fixed the horizontal axis position of the two robots, and used the Y axis to denote relative velocity, based on the angle to the color mark.

The relation tool can be useful in tasks where the coordination between the robot is not spatial. For instance, given a set of sub-tasks which are to be assigned uniformly to different robots (e.g., using ALLIANCE [6]), the relation tool could display use the vertical axis to denote the load on each robot. The operator could then check whether the robots' load is balanced simply by noting the different heights of different dots (signifying different robots). It would also be trivial to use additional visual signs to show the operator the

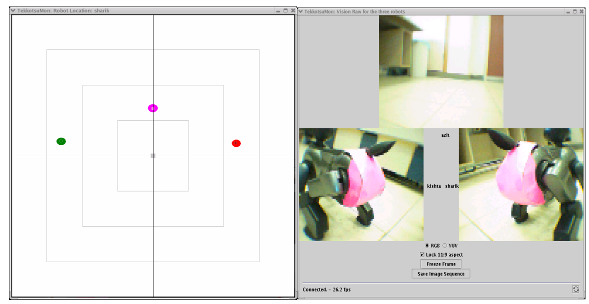
(a) Ground truth



(b) Relation tool view (c) Split-camera view

Figure 1: Failing Formation

(a) Ground truth



(b) Relation tool view (c) Split-camera view

Figure 2: Successful Formation

deviation of the shape from the ideal, etc.

The relation tool is a simple an effective tool. It has three main advantages. First, it significantly reduces the amount of inference needed by the operator to infer the relative state of robots—and thus the state of coordination between them. Second, it is not limited to reproducing a global view of the robots, but instead its dimensions can be used to directly provide the operator with information about what is going wrong (in case of failure), e.g., as in the formation case. Third, it can easily complement other types of displays useful for the task, such as any that show the heading or distance left to the destination.

4. Distributed Call-Request Resolution

As previously discussed, centralized resolution of call requests, by the operator, may work well when robots' tasks are independent of each other. However, in coordinated tasks, many robots may have to stop their task execution until a call request is resolved, because their own task execution depends on that of the robot that requires the resolution. In such cases, it is critical to minimize the time it takes to resolve a call request.

We propose a distributed control approach, whereby the robots who depend on the resolution of the call-request take active steps to resolve it, in collaboration with the operator. This approach takes advantage of the robot teamwork, by turning the resolution of the call-request into a distributed collaborative task for all involved. Moreover, the active robots (that do not require assistance) are involved in a coordinated effort with the robot requiring assistance, and thus may be in a better position to assist it.

We investigate distributed resolution in repairing broken formations. As described in the previous section, three robots were to be led in a triangular formation. However, we disabled one of the robots (to simulate a catastrophic failure), not letting it move or communicate. In accordance with previous approaches, a call-request was issued to determine the whereabouts of the failing robot. The prototypical method to resolve such a situation, based on previous approaches, would be for the operator to halt the operation of all remaining robots in the formation, and then teleoperate the other robots until the position of the failing robot was determined.

Instead, in distributed resolution, all affected robots actively seek the failing robot. If an operator switches control to one of these active robots, the others coordinate with it (for instance, to cover more ground in the search). However, even if the operator is not involved (e.g., because she may be handling other, unrelated, call-requests), the active robots will still seek out the failing robot. When the operator is ready, they would have hopefully found it.

The robots use their knowledge of the robot's role in the formation to attempt to locate it. They first head out directly towards where the robot would have been if it simply lagged too far behind in the formation. If they fail to find it there, they begin a general search pattern (spiral) that is guaranteed to find the robot, but may take relatively long time.

The key objective in this techniques is too speed up call-request resolution by distributing it, and by using organizational knowledge in the team. Although we tested this technique with triangular formation, the same principle (search at the position where robot should have been, before falling back to a general–but slow–search) can easily transfer to other formations and other spatial tasks.

5. Experimental Evaluation

To evaluate the techniques we presented, we tested their performance with novice human operators, and contrasted them with other approaches. We report on experiments with the socially-attentive display (Section 5.1) and with distributed call-request resolution (Section 5.2).

5.1. Evaluation of Socially-Attentive Display

We begin by evaluating the contribution of the socially-attentive display in the formation maintenance and cooperative pushing tasks. In the formation maintenance task, the operator leads the robots in a triangular formation towards a target destination, while avoiding obstacles. If the operator causes the lead robot to turn too sharply, or move too quickly, the formation may break. However, the operator seeks to minimize the time it takes to reach the destination. In the pushing task, the operator controls the velocity of one of the robots, while the other is pushing autonomously. The operator must be careful not to push too quickly for the other robot, nor to lag behind.

We conducted experiments contrasting different combinations (see below) of the socially-attentive display with individual robot display (i.e., their incoming video streams). We ran multiple experiments with novice operators on all combinations. Operators were given an approximate 25-minute training session in operating a single and multiple robot (including the formation and pushing), and only began the experiment when they reported they felt comfortable controlling the robots. Different operators were used for the different tasks (7 for pushing; 9 for formation), but every operator was tested with all combinations available in the task they operated. To avoid ordering effects, the order in which operators were exposed to the different settings was scrambled. In no setting were the operators able to see the robots while operating them.

In the formation maintenance task, we compared three interfaces. The first presented the operator with the split-view video streams from all robots (e.g., Figure 2-c). The second combined the this split-view with the socially-attentive display previously described. The final interface consisted of a single camera (the lead robot's) and the socially-attentive display. Each of the interfaces was tried with two different obstacle courses. The *simple* course consisted of an open space with no obstacles at all. The *hard* course consisted of a single obstacle that had to be by-passed while giving the follower robots sufficient leeway. For each of these trials, we counted the number of catastrophic and non-catastrophic formation failures. Non-catastrophic formation failures were measured as the number of times a follower robot has temporarily lost track of the lead. Catastrophic failures were measured by the number of times the robots reported on permanent tracking failures, which would then be fixed manually by the experimenter—not the test-subject operator—by moving the robot by hand until it regained its tracking.

Figure 3 shows the results of this experiment in terms of the average number of non-catastrophic failures per operator. The results reflect only 8 of the 9 subjects—one subject's results were excluded from all experiments since his failure rate was an order of magnitude greater than the others in controlling the robots, in all interfaces. Figure 4 measures the average number of catastrophic failures for the same experiments.

The results show that in both simple and hard settings, a combination involving a socially-attentive display outperforms monitoring using only individual displays, thus lending support to the hypothesis that socially-attentive display can significantly improve monitoring of robots in coordinated tasks.

However, surprisingly, different combinations vary significantly in performance, depending on the difficulty of the obstacle course. The split-view combination works best in the simple course, while the one-view combination works best in the hard course. We believe that the reason for this difference may be associated with the amount of information that the operator cognitively integrates when operating the robots. In the hard course, the operators had had to maintain very careful control over the formation, since bypassing the obstacle took some mental visualization of where the robots where with respect to the formation. The split-view display might have been distracting in such settings. On the other hand, in the simple course, the operators may have essentially ignored the relation tool, because they felt confident that the formation would not be broken due to any turns. Indeed, one of the operators that had several catastrophic failures in the simple course reported to us that he had not even looked at the relation tool in the one-view trial. Here, a split-view display might have helped by forcing the operators to look more closely at the formation (because there were more incoming streams of data than in the one-view combination). Further investigations of these phenomena are needed.

The second experiment examined the use of the rela-

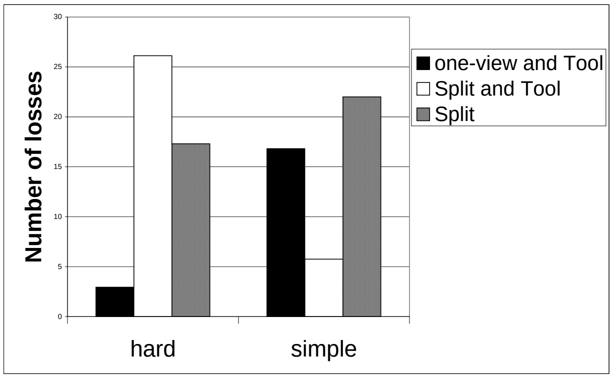


Figure 3: Formation: Non catastrophic failures

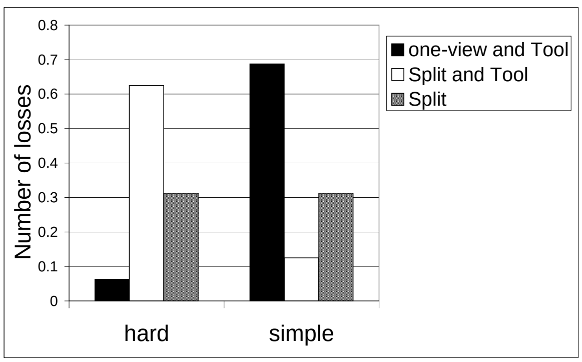


Figure 4: Formation: Catastrophic failures

tion tool in the cooperative pushing task, as previously described. Here, we contrasted three interfaces: a split-camera view only, a combination split-camera and relation tool, and the use of the relation-tool alone. Seven human operators (different than in the formation experiment) were tested on all three interfaces, again scrambling the order of their introduction to the different interfaces to prevent biasing the results due to human learning. Their performance was measured as the average absolute angle deviation from the imaginary horizontal line connecting the robots when they maintain ideal relative velocity. This angle was sampled at 20Hz during task execution. The results are averaged for all operators.

The results of this experiment are shown in Figure 5. The figure shows that both interfaces using the relation tool were clearly superior to the interface relying on camera alone. This is due to this task being essentially a pure-coordination task: The operator does not need to worry about where the pushed object is going, as long as the relative velocity of the robots is maintained at 0 (i.e., their velocities are equal). Thus even a socially-attentive display by itself is sufficient. On the other hand, the non-social split-camera view (by it-

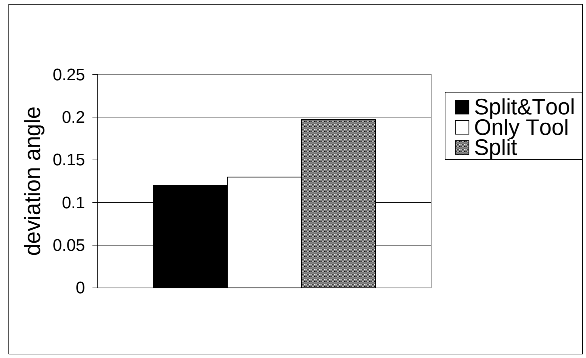


Figure 5: Cooperative Pushing: Total failures

self), is difficult to use for coordination. However, the results still show slight advantage to maintaining the split view in addition to the tool.

Thus in both monitoring tasks, a combination of socially-attentive interface (using the relation tool) with individual robot information proved significantly superior to an approach that did not use this tool. Based on the experiments with formation monitoring, it would seem, though, that the choice of how to combine socially-attentive monitoring with individual information is tricky. Different combinations vary significantly in their monitoring results. We leave exploration of this issue to future work.

5.2. Distributed Call-Request Resolution

We now turn to evaluating the distributed control technique. We set up three robot failure scenarios, and for each compare several control paradigms, with eight human operators. Again, the ordering of the scenarios was randomized between operators to prevent biasing the results. In all scenarios, the right follower robot was disabled, and color marked to allow its detection by the other robots (called *active robots*) and the operator. We distinguished two phases: The first phase of the resolution involved recognition of the disabled failure from any distance. The second phase involved its localization by another robot reaching within 35 centimeters of it. Each scenario began with the simulated disabling of the robot (and issuing of the call request), and ended with its localization by at least one robot—teleoperated or autonomous.

We compared several control schemes: The fully *teleoperated* scheme corresponded to the centralized control used in previous approaches. In this scheme, the operator would switch control from one active robot to the next, as deemed necessary, and manually teleoperated the controlled robot until the disabled robot was found. In the *autonomous* scheme, the active robots searched for their peer autonomously, using the searching behaviors described in Section 4. The teleoperated and autonomous approaches were compared to *distributed* and *semi-distributed* control approaches. In the *distributed* approach, the operator teleoperated one active robot, while the other active robot searched for the disabled robot autonomously. The operator was able to switch to controlling the other robot as needed. In the *semi-distributed* scheme, the active robots began to search by themselves, but once the operator recognized the disabled robot in the camera view of one of the robots (i.e., once the first phase of the search was over), she took over and the teleoperated the active robot until it localized the disabled robot.

A potential advantage of the distributed scheme is that it utilizes knowledge that the robots may have in locating the disabled robot. In particular, because the robots have moved in formation prior to the call-request, they may have an easier time guessing their peer's location than the operator (who needs to orient herself in space via the teleoperated camera).

To evaluate the importance of this advantage, we experimented with three locations for the disabled robot: *Easy* placed the disabled robot at approximately where it would be had it just stopped in its tracks prior to the team getting notification of its "disappearance", i.e., a bit farther behind its location within the formation; *Medium* placed the robot behind the left follower robot; and *Hard* placed the robot to the left of the left follower robot, and behind it (i.e., completely out of place compared to the formation). Thus the locations progress from a location easily predictable by the robots, to a location unpredictable to them.

For each of these locations and for each method, we measure the time that it would get the operator to recognize the disabled robot in any one of the cameras (the operator uses the split-view interface in this task), i.e., the duration of the first phase. We then measure the time that it takes for an active robot—autonomous or teleoperated—to reach the disabled robot, i.e., the duration of the second phase. Since the motivation behind the distributed control scheme is to reduce the time spent awaiting resolution, we prefer shorter durations.

Figure 6 shows the results of the different control schemes for the first phase, averaged across operators. The figure measures the average time (in seconds) it took the operator to recognize the disabled robot from afar, in the split-view camera display. Clearly, all approaches in which robots attempt to orient themselves towards the predicted location of the disabled are superior to a teleoperated (centralized) approach. Note that in all approaches, the operator recognizes the robot from afar. The active robots do not necessarily recognize the other robot from afar, and as we will see below, may end up searching for it in the wrong location. This significantly shorter initial recognition is a beneficial side-effect of the distributed approaches.

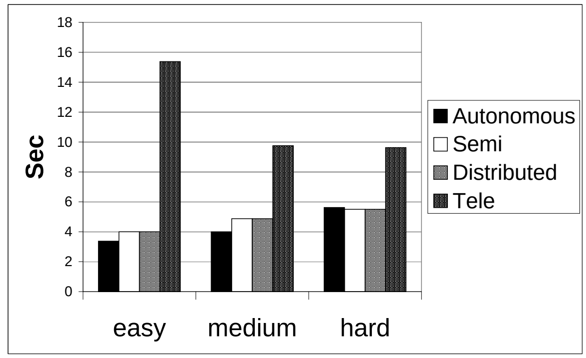


Figure 6: Phase 1 Duration

Figure 7 shows the results for the second phase, where the task is to arrive within the proximity of the disabled robot. Despite its poor performance in phase 1, the teleoperated approach does quite well in phase 2. This is easily explained—here the disabled robot is already recognized, and the teleoperated approach simply allows the operator to now drive the teleoperated robot as quickly as possible. The distributed approach performs well, because it essentially turns this phase into a race between a teleoperated robot and an autonomous robot, as to who gets to the disabled robot first. Moreover, unlike the semi-distributed approach, where there's an overhead of a few seconds while the operator takes over control (see the results for the medium location), here the transition from phase 1 to phase 2 is fairly smooth, because one active robot continues to search even while the operator is taking over control of the other. Thus the results are close to the teleoperated approach, and are better than the autonomous approach in the easy and hard locations.

Indeed, contrasting the results of the autonomous and distributed approaches is telling. As we move from the easy location to medium to hard, our expectations were that the autonomous and distributed approaches would work in a trade-off: *Autonomous* would be quicker than *distributed* in the easy location (since its search predictions are supposedly correct in this location), while in the medium and hard locations, *distributed* would be better, because despite the autonomous search predictions failing in these locations, the operator in *distributed* would make up for any lacking of the automated search. However, in the easy location, *distributed* did better than *autonomous*, while in medium, *autonomous* did better than *distributed*.

A closer look at the durations of the autonomous approach at both locations reveals that in the easy location, autonomous took 32.2 seconds, while in the medium location it only took 22.66 seconds. In a sense, our own expectations (which were not based on pilot experiments) were wrong. What we had believed to be an easy location for the auto-

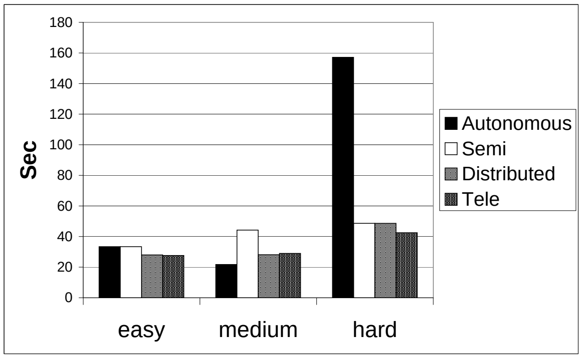


Figure 7: Phase 2 Duration

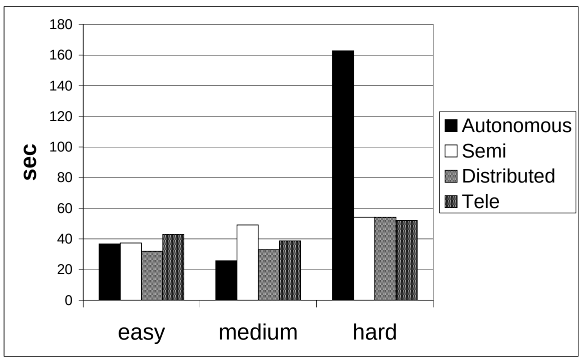


Figure 8: Total Time to Resolution

mated search was in fact of medium-level difficulty, while our medium location was in practice a good match for the predictions of the autonomous search.

Ultimately, the performance of the distributed approach vs. the teleoperated (centralized) approach is measured in the total time it takes to identify the location of the disabled robot—the total time for resolution—combining phases 1 and 2. Figure 8 shows the average total duration for the eight operators. The results show that in both *easy* and *medium* locations, the distributed approach is preferable to the centralized teleoperation approach. In the *hard* location, distributed trails teleoperation by a slight margin. Distributed does better than the semi-distributed approach in all locations, and better than the autonomous approach in the *easy* and *hard* locations. Overall, the distributed collaboration between the operator and active robots in the distributed approach proves to be a powerful technique for significantly reducing the time to complete the task of locating the disabled robot.

6. Summary and Future Work

This paper takes steps towards techniques that allow a single human operator to control a team of robots that are tightly coordinated. In particular we point at two challenges that are unique to controlling tightly-coordinating teams: Explicit monitoring of the coordination, and resolving calls for operator attention as quickly as possible. Existing techniques do not adequately address this challenge. In particular, in such settings, their centralization of the operator as the only entity capable of resolving calls for attention is inappropriate, because it does not take advantage of the teamwork of the robots, and their implicit or explicit organizational knowledge.

We demonstrate two techniques. First, we show that socially-attentive displays can significantly improve the failure rate in two tasks requiring maintenance of team coordination by the operator. Second, we show that a distributed control scheme, allowing teamwork between the operator and all robots, reduces the time of resolving failures (compared to the centralized and autonomous approaches), in many instances.

The results presented in the paper also raise questions for future work. For example, the results for the monitoring tool indicate that while a socially-attentive display can be a significant advantage, it is only so in particular combinations with regular displays—and these improved combinations change from task to task.

References

[1] Julie A. Adams. *Human Management of a Hierarchical System for the Control of Multiple Mobile Robots*. PhD thesis, University of Pennsylvania, 1995.

[2] MaryAnne Fields. Modeling the human/robot interaction in onesaf. In *Proceedings of the 23rd Army Science Conference*, 2002. (Poster).

[3] Terrence Fong, Charles Thorpe, and Charles Baur. Multi-robot remote driving with collaborative control. *IEEE Transactions on Industrial Electronics*, 50(4):699–704, August 2003.

[4] Gal A. Kaminka and Milind Tambe. Robust multi-agent teams via socially-attentive monitoring. 12:105–147, 2000.

[5] R.R. Murphy, J.L. Casper, M. J. Micire, and J. Hyams. Mixed-initiative control of multiple hetrogeneous robots for urban search and rescue. submitted to IEEE Transactions on Robotics and Automation, 2000.

[6] Lynne E. Parker. ALLIANCE: An architecture for fault tolerant multirobot cooperation. *IEEE Transactions on Robotics and Automation*, 14(2):220–240, April 1998.

[7] P. Scerri, L. Johnson, D. Pynadath, P. Rosenbloom, M. Si, N. Schurr, and M. Tambe. A prototype infrastructure for distributed robot, agent, person teams. In *AAMAS-03*, 2003.

[8] The Tekkotsu Homepage. www.tekkotsu.org, 2002–2003.

[9] Ashlley D. Tews, Maja J. Mataric, and Gaurav S. Sukhatme. A scalable approach to human-robot interaction. In *ICRA-03*, 2003.