Proc. of IEEE International Conference on Intelligent Robots and Systems (IROS), Sendai, Japan, 2004.

Tightly-Coupled Navigation Assistance in Heterogeneous Multi-Robot Teams

Lynne E. Parker, Balajee Kannan, Fang Tang, Michael Bailey
Distributed Intelligence Laboratory, Department of Computer Science, The University of Tennessee, Knoxville
Email: {parker, balajee, ftang, mbailey}@cs.utk.edu

Abstract*— This paper presents the design and results of autonomous behaviors for tightly-coupled cooperation in heterogeneous robot teams, specifically for the task of navigation assistance. These cooperative behaviors enable capable, sensor-rich ("leader") robots to assist in the navigation of sensor-limited ("simple") robots that have no onboard capabilities for obstacle avoidance or localization, and only minimal capabilities for kin recognition. The simple robots must be dispersed throughout a known, indoor environment to serve as a sensor network. However, because of their navigation limitations, they are unable to autonomously disperse themselves or move to planned sensor deployment positions independently. To address this challenge, we present cooperative behaviors for heterogeneous robots that enable the successful deployment of sensor-limited robots by assistance from more capable leader robots. These heterogeneous cooperative behaviors are quite complex, and involve the combination of several behavior components, including vision-based marker detection, autonomous teleoperation, color marker following in robot chains, laser-based localization, map-based path planning, and ad hoc mobile networking. We present the results of the implementation and extensive testing of these behaviors for deployment in a rigorous test environment. To our knowledge, this is the most complex heterogeneous robot team cooperative task ever attempted on physical robots. We consider it a significant success to have achieved such a high degree of system effectiveness, given the complexity of the overall heterogeneous system.

I. Introduction

The most common use of heterogeneous multi-robot teams in the literature is to achieve functionally-distributed missions, in which the mission tasks require a variety of capabilities not possessed by any single robot team member. In these applications, team members must decide which robot should perform which task, based upon the unique capabilities of each robot. However, these applications typically do not enable robots to help each other toward accomplishing their individual goals through the sharing of sensory information (except in the form of map-sharing, which is indeed a common practice in multi-robot teams).

Our research goals are aimed at developing techniques that allow heterogeneous robot team members to assist each other in tightly-coupled tasks by providing information or capabilities that other teammates are not able to generate or perform on their own. In particular, this paper addresses the issue of cooperative assistive navigation. We present heterogeneous autonomous behaviors for assisting the navigation of a set of sensor-limited robots using a more sensor-capable leader robot.

Our particular application of interest is deploying a large number (70+) of simple mobile robots that have microphone sensors to serve as a distributed acoustic sensor network. However, due to cost and power considerations, our simple robots have no sensors for localization or obstacle avoidance, and minimal sensing for robot kin recognition (using a crude camera). The objective is to move the simple mobile robots into deployment positions that are optimal for serving as a sensor network. Because these sensor-limited robots cannot navigate safely on their own, we have developed complex heterogeneous teaming behaviors that allow a sensor-rich leader robot, equipped with a laser scanner and camera, to guide the simple robots (typically, 1-4 of these simple robots at a time) to their planned destinations using a combination of robot chaining and vision-based marker detection for autonomous teleoperation.

While this paper addresses the specific tightly-coupled task of heterogeneous robot navigational assistance, we believe more generally that these navigational assistance techniques can provide the foundation for enabling any type of heterogeneous robot to assist other robot team members through the exchange of sensory or command and control information.

The following sections provide the details of our approach to autonomous navigation assistance. We provide an overview of our approach in Section II. Section III describes the robot states and messages that enable the coordination of the multiple robots on the deployment team. The *Long-Dist-Navigation* mode using chain (i.e., follow-the-leader) formation-keeping is discussed from the perspective of both the leader robot and the sensor-limited robots in Section IV. Section V describes the leader robot's method of assisting the simple robots during the *Short-Dist-Navigation* mode. In Section VI, we give details of the implementation of this approach on a team of physical robots, followed by a discussion of the results of our approach in Section VII. Section VIII contains a discussion of related work. We present our concluding remarks in Section IX.

II. Overview of Approach

Since our simple robots have very limited navigation capabilities and cannot even disperse themselves, we autonomously plan the entire set of desired deployment positions for the simple robots at the beginning of the robot team deployment mission, using a known map of the environment. The planning of the sensor deployment positions involves satisfying a number of geometric constraints, including minimizing doorway

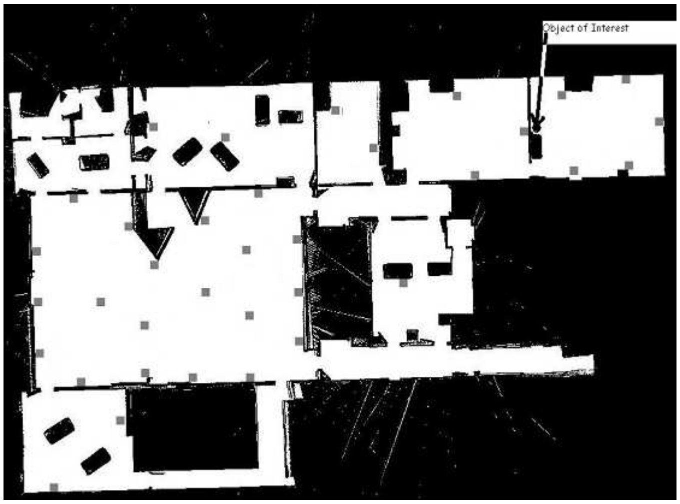

Fig. 1. Example result of autonomous planning of sensor robot deployment positions, showing 36 planned sensor positions (small gray squares).

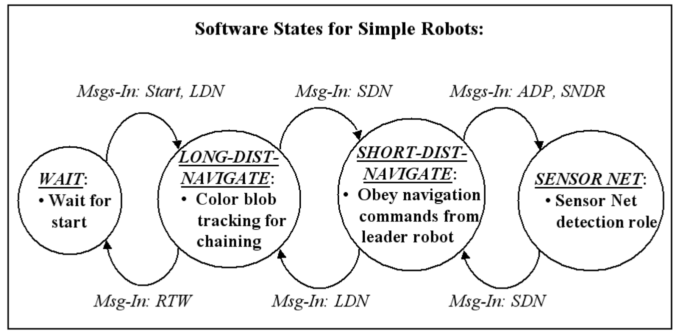

Fig. 2. State diagram of simple robot.

and pathway obstruction, maintaining line of sight, satisfying minimal inter-robot deployment positions, ensuring sufficient operating room for deployment by the leader robot, and so forth. Depending on the environment, up to several dozen deployment positions may be generated. Figure 1 shows an example plan of the deployment positions in one of our experimental environments. The robots are then autonomously grouped into deployment teams by assigning a leader robot a set of n simple robots and deployment positions. These deployment positions are grouped to minimize the turning angles required to execute the paths, to facilitate the multi-robot chaining by avoiding sharp turns as much as possible. The details of our deployment algorithm and deployment team groupings are provided in [8].

Our approach involves two modes of navigation. The first – *Long-Dist-Navigation* – involves the leader robot using its laser-based localization capability to lead the sensor-limited robots in a chain formation to the vicinity of the goal destination of the first simple robot. During this navigation mode, the simple robots use a crude camera and a color blob tracking algorithm to follow the robot ahead of it, which is outfitted with a rectangular red blob. This mode of navigation is used when the simple robots are far from their desired destination (greater than approximately 2 meters).

The second mode of navigation – *Short-Dist-Navigation* – involves the leader robot autonomously teleoperating the first simple robot into position using color vision to detect a fiducial on the simple robot. This fiducial provides the ID and relative pose of the simple robot. Once the first robot is guided into its exact deployment position, the leader robot then successively visits the deployment destinations of the remaining simple robots until all of the robots have been deployed. The leader robot then returns to its home position to pick up another set of simple robots to deploy. Once the simple robots are in position, they switch state to their primary role of forming a distributed acoustic sensor network for intruder detection.

III. Multi-Robot Coordination

The behavior state diagrams in Figures 2 and 3 illustrate more details of the behavior organization of the leader robot and the simple robots. In this multi-robot coordination process, several messages are passed between the robots, as defined in Table I.

The simple robots have three main states, as shown in Figure 2: *Long-Dist-Navigate*, *Short-Dist-Navigate*, and *Sensor Net*, in addition to the *Wait* state. The simple robot begins in the *Wait* state until it receives a “Start” message from the leader robot. The simple robot then transitions to the *Long-Dist-Navigation* state, effectually beginning the chain formation-keeping behavior[1]. Section IV elaborates on how this behavior is achieved. The simple robots remain in this state until they receive either an “SDN” or “RTW” message from the leader robot, causing them to either switch to the *Short-Dist-Navigate* state or return to the *Wait* state.

In the *Short-Dist-Navigate* state, the simple robot receives navigation control commands from the leader robot to assist the robot in reaching its “exact” destination position. Once the simple robot reaches its destination position, the leader robot sends an “SNDR” message to the simple robot instructing it to enter the *Sensor Net* state. In our application, the simple robot then forms part of a distributed acoustic sensor network to detect the location of “intruders” navigating through the area (see [7] for more details on our design and implementation of the distributed acoustic sensor network). The simple robot remains in the *Sensor Net* state until a leader robot returns to move the robot to another location. In our application, this occurs when the simple robot’s power level falls below a threshold and needs to return to a recharging station. The leader robot becomes aware of this need through messages from the simple robots, and returns to assist the simple robot back to the recharging station.

Figure 3 illustrates the state transitions of the leader robot. The leader robot also has three main states: *Navigate*, *Assist*, and *Transition*, as well as a *Wait* state. Once the leader robot receives a “Start” message (from the human operator), the

[1] While an ultimate objective of our research is to enable the robots to autonomously form themselves in the proper physical configuration to enter the *Long-Dist-Navigate* mode, for now we assume that the robots are manually initialized to be in the proper front-to-back orientation for successful chaining.

TABLE I

MESSAGES DEFINED TO ACHIEVE INTER-ROBOT COORDINATION AND COOPERATION.

Message ID	*Description*	*Sender*	*Receiver*
Start	Start mission	Human op. or leader robot	Leader or simple robots
LDN	Initiate Long-Dist-Navigation mode	Leader robot	Simple robot
SDN	Initiate Short-Dist-Navigation mode	Leader robot	Simple robot
ADP	At Desired Position	Leader robot	First simple robot
SNDR	Initiate Sensor Net Detection Role	Leader robot	First simple robot
RTW	Return to Wait	Leader robot	Simple robot

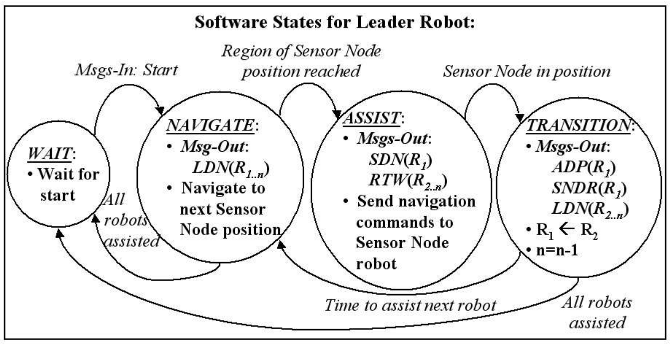

Fig. 3. State diagram of the leader robot.

leader robot enters the *Navigate* state. In this state the leader robot plans a path to the desired (or actual) location of the first simple robot on its team. It then uses its laser scanner to localize itself and avoid obstacles while it navigates to the goal position. Once the leader robot reaches the goal position, it changes states to the *Assist* state and sends a message to the first simple robot to enter the *Short-Dist-Navigate* state. The leader robot also sends an "RTW" message to the other simple robots on the deployment team to cause them to wait while the first leader robot is being assisted. At this point, the leader robot's goal is to autonomously navigate the first simple robot into its deployment position. The leader robot detects the current distance and pose state of the simple robot and then communicates velocity and steering commands to enable it to reach its deployment position. Once the first simple robot is in position, the leader robot sends it an "ADP" message to let it know that the desired position is reached, followed by an "SNDR" message to cause the simple robot to initiate the sensor net detection role. Finally, the leader robot sends an "LDN" message to the remaining simple robots, causing them to reinitiate their chaining behavior. The process is then repeated until all of the simple robots on the deployment team have reached their desired positions.

IV. LONG DISTANCE NAVIGATION MODE

A. Leader Localization

The leader robot is given a set of deployment coordinates and plans a path to those positions using a dual wavefront path planning algorithm [7]. As the leader robot moves to its desired position, it localizes itself to a map of the environment using an adaptive Monte Carlo localization technique that combines laser and odometry readings (similar to [9]).

B. Chaining Behavior

In our previous work [6], the simple robots did not have the ability to perform color blob tracking. Thus, the leader robot had to operate in the *Assist* mode (i.e., autonomous teleoperation) the entire time, even when the distances were large. Under this prior technique, all simple robots had to be maintained in a line-of-sight formation. However, in our current work, we added a simple vision system (the CMUCam) to each simple robot, enabling color blob tracking. In this approach, we mount a red "blob" on the back of each robot in the deployment team. In this mode, the simple robot keeps the red blob within view and moves towards the centroid of the blob. If the blob is lost, the simple robot tries to reacquire the blob by continuing its previous action or by panning itself from side to side. The effect of this blob tracking when multiple robots are front-to-back with each other is a follow-the-leader chaining behavior.

V. SHORT DISTANCE NAVIGATION MODE

A. Color Fiducial for Detection of Robot ID and Pose

Our approach to navigation assistance in the *Short-Dist-Navigation* mode is dependent upon the leader robot's ability to detect the identity, relative position, and orientation of the simple robots. Additionally, since we want to operate in a system that may have up to 70 simple robots, we need to have a unique marker for each robot. After extensive tests, we designed a striped cylindrical color marker for each simple robot as shown in Figure 4. The actual height of the marker is 48 cm, and the circumference is 23 cm. The marker is composed of four parts: a START block, an ID block, an Orientation block and an END block. The START block is a combination of red and green stripes at the bottom of the marker. The END block is a red stripe at the top of the marker. The START and END blocks make the marker unique in a regular environment. Adjacent to the END block is the Orientation block. The relative orientation of the robot is calculated by the width ratio of black and white in the Orientation block. The ID block is composed of 7 black or white stripes, where black represents 1 and white represents 0. This block provides $2^7 = 128$ different IDs and is easy to be extended to identify more robots if needed.

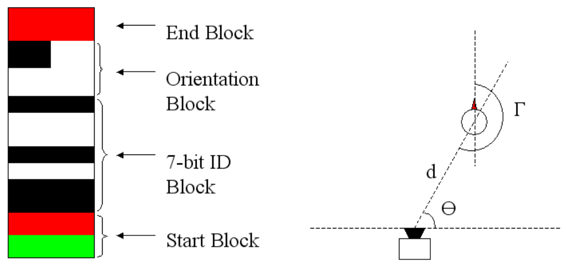

Fig. 4. Cylindrical marker design to provide unique visual ID, relative position, and orientation information for the simple robots.

Once a marker is recognized in the camera image, the marker detection algorithm determines the identity and the relative position of the marker in terms of the following parameters, as shown in Figure 4:

- d: the distance between the leader's camera and the center of the simple robot
- Γ: simple robot orientation – the angle between the heading of the simple robot and the center of the camera.
- Θ: the angle between the center of the simple robot and the plane containing the leader's camera.

Suppose that a marker of height h is located at (x, y) in the image plane of (r, c) pixels, the edges of the marker are (l, r), and the delimitation is located at column k. Then the above parameters are calculated by the leader robot as follows:

$$d = \frac{C_1}{h \times C_2}$$

$$\Gamma = 180 \times \frac{k - l}{r - k}$$

$$\Theta = FOV + \frac{x}{c} \times (180 - 2 \times FOV)$$

where FOV is the field-of-view of the camera, and C_1 and C_2 are constants defined by the size of the real marker.

B. *Autonomous Teleoperation*

In the *Assist* mode, the leader robot uses autonomous teleoperation to assist the simple robot in navigating to its desired position. We define *autonomous teleoperation* as the process of the leader robot transforming the relative information about the simple robot into steering and control commands that are communicated to effect the motion of the simple robot. The autonomous teleoperation approach requires the leader robot to convert its own global position (known using laser localization), as well as the known relative location of the simple robot (obtained from visual marker detection) into velocity and steering control commands communicated to the simple robot to guide it to its desired global position.

Once the leader robot calculates the proper motion commands and communicates them to the simple robot, the simple robot executes the received commands for a short time period (s seconds). The leader robot then recalculates the simple robot pose information and sends the next set of control commands, repeating until the simple robot is in position. The value of s is calculated experimentally and is optimized for the specific set of simple robots (typically 0.5 to 3 seconds).

VI. Physical Robot Implementation

Our approach to assist sensor-limited robots in navigation and deployment has been implemented on a team of physical robots. In these experiments, we had 4 leader robots, which were Pioneer 3-DX research robots. These robots have a SICK laser range scanner, a pan-tilt-zoom camera, and a wireless mobile ad hoc networking capability. On these robots, the laser range scanner faces forward, while the pan-tilt-zoom camera faces to the rear of the robot. This allows the robot to move forward while localizing, and then to provide navigation assistance to the simple robots without having to turn around.

The simple robots consist of a team of up to 70 AmigoBot robots (without sonar) that have a CMUCam camera for color blob tracking and an iPAQ running Linux for computation and a low-fidelity microphone for simple acoustic sensing. The AmigoBot robots are also able to communicate using wireless mobile ad hoc networking. We implemented our approach on all of these robots in C++ interfaced to the Player robot server [4].

VII. Results and Discussion

A. *Experiments*

The experiments reported in this paper were performed in the sample environment and deployment plan shown in Figure 1. The experiments consisted of repeated deployments of 1-2 simple robots per team. The experiments were tightly controlled by a set of human evaluators who were not the system developers. Additionally, the experiments were run by human controllers that were allowed access only to laser feedback from the leader robots and a stream of text messages from the robot team members to determine the state of the system. If a deployment failed on one experiment (for example, if a simple robot got caught on an obstacle when trying to follow the leader through a sharp turn), the consequences of that failure were not corrected unless the human controllers could determine from the leader robot's laser feedback that some robot had blocked a passageway. Thus, the data reported here incorporates propagation of error from one experiment to the next. In these experiments, a total of 61 simple robot deployments were attempted.

B. *Chaining Behavior*

The color blob tracking algorithm using the CMUCam on the simple robots is quite robust when operating in uncluttered environments. This algorithm can successfully follow a leading robot as it moves in the environment, up to a 90-degree turn angle. We have successfully demonstrated 5 simple robots robustly following a leading leader robot in a chaining behavior. The main limitation to scalability is the tendency of the following behavior to "shave off" corners. Thus, as this tendency propagates through many robots, eventually some simple robot will become lost or caught on obstacles by cutting

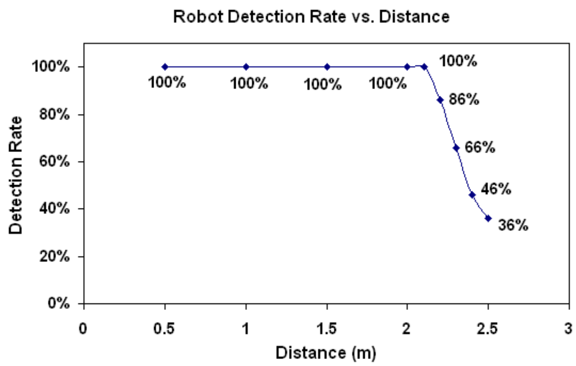

Fig. 5. Marker Detection result with various inter-robot distances.

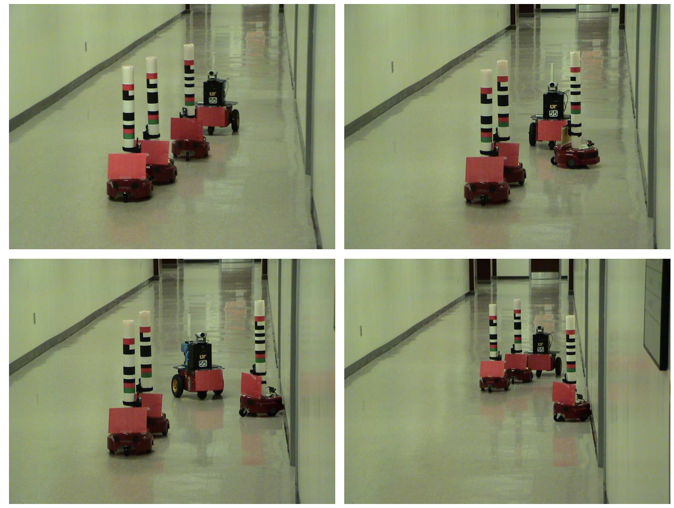

Fig. 6. These snapshots show our assistive navigation approach in operation (read left to right, top to bottom).

corners too close. In our experiments in cluttered spaces, it was difficult for the simple robots to follow the leader when it took many sharp turns in complex environments. The CMUCam also requires a fair amount of light to work properly. We were able to operate very well in open spaces with typical office lighting. Because of the requirement to make several sharp turns in our test environment, the chaining behavior was only found to be robust for teams of 1-2 simple robots in cluttered environments.

C. Color Vision-Based Marker Detection

The color vision-based marker detection behavior has been tested independently from the rest of the behaviors, to determine its robustness and accuracy as a function of distance and relative position in the leader's field of view. For these independent evaluations, we selected 10 different positions with various lighting and background conditions. Distances between the leader's camera and the simple robot marker varied from 0.5m to 2.5m. The resolution of the image is 160 x 120. When the leader can detect the marker, the determination of the relative pose of the simple robot is extremely high, with an average error in estimated distance of 6cm. The primary difficulty is in the leader robot not being able to find the marker, due to distance, unfavorable lighting conditions, or a cluttered visual background.

The results of these tests of robustness are shown in Figure 5. The performance is quite good until a distance of about 2.1 meters is reached, due to the limits of the size of the marker and the image resolution. The ability of the leader to detect the marker falls off quickly beyond this distance.

Our algorithm has several parameter settings; thus, when the algorithm is used for a new environment, we typically calibrate the camera to work properly given that lighting environment. These parameters include the range of RGB components and their correlation functions.

D. Autonomous Teleoperation

Figure 6 shows a series of snapshots of the leader robot autonomously teleoperating a simple robot to its planned deployment position. Our experimental results show that our technique for autonomous teleoperation provides accuracy of final simple robot positioning of approximately 30 centimeters, compared to the original planned waypoint positions. Since the typical distance between deployed simple robot positions is 2 meters or more, this level of accuracy is suitable for our purposes.

We also collected data on the time it takes to deploy a simple robot from the time that the team transitions to the *Short-Dist-Navigate* mode until the simple robot is successfully deployed. Over a set of 36 successful trials, the average time for deployment is 132 seconds, with a standard deviation of 45 seconds. The fastest deployment achieved was 65 seconds, while the slowest deployment time was 250 seconds. The variation is typically due to the leader occasionally losing the simple robot's visual marker and having to slowly pan its camera to find it again.

E. Overall System Evaluation

Our system is clearly composed of several modules. The successful completion of the entire deployment process depends upon the successful completion of all of the system modules. Additionally, the success of any given module is typically dependent upon the success of other modules. For example, the completion of the marker detection process is dependent upon the successful execution of the chaining behavior. Additionally, we made our system execution especially challenging by forcing the system to deal with the consequences of prior deployment failures. Thus, subsequent robot team deployments had to deal with situations such as partially blocked doorways if a prior deployment resulted in a simple robot being caught on the doorway. If all the test runs were independent, the overall system success rate would certainly have been higher.

To analytically evaluate the system's expected probability of success, we determined component interdependencies and estimated the probability of success of each of the component modules. Here, we identified the component modules to be

TABLE II

OVERALL SYSTEM PROBABILITY OF SUCCESS FOR WORST-CASE CONDITIONS.

Module	Success Probability	Subsystem Success Rate	Experimental Success Rate
localization	p_1	.83	
path planning	p_2	(est .99)	
navigation	p_3	(est .95)	
chaining	p_4	(est .78)	
marker detection	p_5	.98	
communication	p_6	.91	
complete system	$\prod_i p_i$	(est .54)	.67 (2-robot deplymnt) .48 (1-robot deplymnt) .59 (avg. of all trials)

localization (with p_1 probability of success), path planning (p_2), navigation (p_3), chaining (p_4), marker detection (p_5), and communication (p_6). In some cases, we can experimentally evaluate the success rate of the component modules; in other cases, it was not possible to isolate certain modules from the overall system. In the latter case, we derived an approximate calculation of the subsystem probabilities based upon our overall experimental observations. As shown in Table II, the complete system success probability is estimated to be $\prod_i p_i$, which is approximately 54%. Our actual experiments showed that the success rate for 2-robot deployments was 67%, while the success rate for 1-robot deployments was 48%. Over all 61 trials, the success rate was 59%.

The most error-prone part of the system was the chaining behavior in real-world environments that involved moving through tight doorways and making sharp turns. The most difficult positions tended to be single deployment assignments because they typically involved sharper turns. Thus, the success rate for single-robot deployments is much lower than for two simple robot deployments. For some of the positions, we had multiple failures. For example, for one deployment position, we tried (and failed) three times to deploy a simple robot to one of the more challenging positions. This also figures into our success rate, creating a reasonable "worst case" expectation of success. Future improvement to the chaining behavior should lead to improved overall system success, as well as increased ability on the leader's part in dealing with problems experienced by the following simple robots.

Because we recognized that many possible types of failures could occur in this system, we incorporated a significant amount of behavior fault tolerance in the leader robots to ensure that the leader robot could at least make it back home, even if the deployment of a simple robot was not successful. This was especially important in our experiments, in which we had only 4 leader robots compared to up to 70 simple robots. A critical phase of fault tolerance is the ability of the system to diagnose the correct failure state. Table III shows the set of base failure states identified for this system and the implemented recovery action.

Using these methods of behavior fault tolerance, the success rate of the leader robots making it back home autonomously was 91% out of 45 trials. The primary mode of failure for the the leader robot was losing communication, which caused the robot system to hang when it attempted to report back to the operator control unit on a non-existent communications channel. An improved method of implementing the communication between the robot and the human operator would remove this system dependence on maintaining a communications connection.

Of course, this rule-based approach to extending the fault tolerance of the system will only work properly if the human designers of the system correctly anticipate all possible modes of failure. Despite thorough consideration, it is not realistic to expect that all such failure modes will be adequately predicted. Indeed, if such an unanticipated failure mode were to occur in the current design, the leader robot would most likely not be able to return home, and would subsequently be unavailable for future use in the mission. Therefore, in our ongoing research, we are designing learning techniques that allow leader robots to learn from their previous experiences, or those of other leader teammates, with the objective of improving the overall team success rate. These learning techniques should enable a leader robot to adapt its future actions based upon these prior experiences, and therefore to successfully respond to, or recover from, events that were not foreseen by the human designers.

VIII. RELATED WORK

Several areas of related work apply to our research, including formation control, robot-assistance, and vision-based robot detection. Space does not allow the mention of many of these prior approaches, so we mention a few approaches that are especially relevant. In the area of formation-keeping, Balch and Arkin [1] list the advantages and disadvantages of different formations under various environmental constraints. Experiments conducted by Balch and Arkin [1] indicate that column formation optimizes performance in an obstacle rich environment. These prior experiments validate our decision to use chaining formation control in cluttered environments rather than other formations.

Other algorithms have been implemented that use vision-based formation control. For example, Noah Cowan, et al. [3] discuss one such approach for vision-based follow-the-leader formation-keeping. Their work utilizes two different controllers for maintaining formation using an omni-directional camera.

In the area of vision-based robot detection, several previous authors describe the use of fiducials similar to our approach. For example, Cho, et al. [2] present a fiducial consisting of circles and triangles in six colors with fast and robust detection. Malassis and Okutomi [5] use a three-color fiducial to provide pose information. Walthelm and Kluthe [10] measure marker distance based on concentric black and white circular fiducials. Our previous work in [6] utilized another design of a color marker, which was relatively more sensitive to current

TABLE III

IDENTIFIED FAILURE STATES DETECTED BY THE LEADER ROBOT AND IMPLEMENTED RECOVERY ACTIONS.

Failure Type	**Fault Recovery Action**
Can't reach waypoint	Re-plan path.
Lost simple robot	Leave lost robot in wait state and move on to next robot in chain.
Leader robot camera failure	Leave simple robot(s) in wait state, send camera failure feedback to human operator and return home.
Simple robot motor failure	Check if simple robot is close enough to goal; if so, change simple robot state to sensor detection and proceed as if successfully deployed; else, leave simple robot in wait state and proceed to the next simple robot.
Localization drift	Check if simple robot is close enough to goal; if so, change simple robot state to sensor detection and proceed as if successfully deployed; else, leave simple robot in wait state and proceed to the next simple robot.
Can't detect marker	Check if simple robot is close enough to goal; if so, change simple robot state to sensor detection and proceed as if successfully deployed; else, leave simple robot in wait state and proceed to the next simple robot.
Communication failure	Return home.

lighting conditions than our current marker design, which has been more experimentally robust.

IX. CONCLUSIONS AND FUTURE WORK

In this paper, we have outlined a general approach for enabling more capable robots to assist in the navigation of sensor-limited robots. In this approach, we use cooperation among teams of heterogeneous robots that involves a leader robot guiding a set of simple robots to their desired positions. The leader robot uses a laser scanner for localization, along with a vision system for autonomously teleoperating the simple robots into position. The simple robots make use of a crude vision system for color blob tracking to achieve the chaining behavior over long distances.

We have successfully implemented this approach on a team of physical robots and presented extensive testing results of the implementation in a rigorous experimental setup. Our future work is aimed at incorporating increased fault tolerance and learning into our system, so that if the simple robots fail during the deployment process, the leader robot explores more options for assisting the recovery of the simple robots.

To our knowledge, this is the most complex heterogeneous robot team cooperative task ever attempted on physical robots. We consider it a significant success to have achieved such a high degree of system effectiveness, given the complexity of the overall heterogeneous system.

We believe that these techniques can provide the foundation for enabling a wide variety of heterogeneous robot team members to assist each other by providing information or sensory data to assist other robots in accomplishing their individual goals. Our future work is aimed at facilitating this sensor-sharing capability in heterogeneous robot teams.

ACKNOWLEDGMENTS

The authors thank Chris Reardon, Ben Birch, Yifan Tang and Yuanyuan Li for their valuable discussions regarding this research. The authors also thank Andrew Howard and his team of mobile robots at the University of Southern California for generating the maps used in this research. This research was sponsored in part by DARPA/IPTO's Software for Distributed Robotics program, through Science Applications International Corporation, and in part by the University of Tennessee's Center for Information Technology Research. This paper does not reflect the position or policy of the U.S. Government and no official endorsement should be inferred.

REFERENCES

[1] T. Balch and R. Arkin. Behavior-based formation control for multi-robot teams. *IEEE Transactions on Robotics and Automation*, December 1998.

[2] Y. Cho, J. Parker, and U. Neumann. Fast color fiducial detection and dynamic workspace extension in video see-through self-tracking augmented reality. In *Proceedings of the Fifth Pacific Conference on Computer Graphics and Applications*, pages 168–177, 1997.

[3] N. Cowan, O. Shakernia, R. Vidal, and S. Sastry. Vision based follow-the-leader. In *Proceedings of the IEEE International Conference on Intelligent Robots and Systems*, October 2003.

[4] B. Gerkey, R. Vaughan, K. Stoy, and A. Howard. Most valuable player: A robot device server for distributed control. In *Proc. of 2001 IEEE/RSJ International Conference on Intelligent Robotics and Systems*, pages 1226–1231, 2001.

[5] L. Malassis and M. Okutomi. Three-color fiducial for pose estimation. In *Proceedings of the Asian Conference on Computer Vision*, 2000.

[6] L. E. Parker, K. Balajee, X. Fu, and Y. Tang. Heterogeneous mobile sensor net deployment using robot herding and line-of-sight formations. In *Proceedings of IEEE International Conference on Intelligent Robots and Systems (IROS '03)*, October 2003.

[7] L. E. Parker, B. Birch, and C. Reardon. Indoor target intercept using an acoustic sensor network and dual wavefront path planning. In *Proceedings of IEEE International Conference on Intelligent Robots and Systems (IROS '03)*, October 2003.

[8] Y. Tang, B. Birch, and L. E. Parker. Mobile sensor net deployment planning using ray sweeping and wavefront path planning. In *Proceedings of IEEE International Conference on Robotics and Automation (ICRA '04)*, 2004.

[9] S. Thrun, D. Fox, and W. Burgard. A probabilistic approach to concurrent mapping and localization for mobile robots. *Autonomous Robots*, 5:253–271, 1998.

[10] A. Walthelm and R. Kluthe. Active distance measurement based on robust artificial markers as a building block for a service robot architecture. In *Proceedings of the Fifth IFAC Symposium*, pages 325–330, 2001.