A collaborative search and engage strategy for multiple mobile robots with local communication in largescale hostile area

Xiaohui Cui
Wuhan University

Adel Said Elmaghraby
University of Louisville

A Collaborative Search and Engage Strategy for Multiple Mobile Robots with Local Communication in Large-scale Hostile Area

Xiaohui Cui, Rammohan. K. Ragade, Adel S. Elmaghraby
Computer Engineering and Computer Science Department
University of Louisville, KY 40292
X0cui001, rkraga01, adel@louisville.edu

Keywords: Collaborative, Robots, Exploring, Ad hoc, Large-scale

Abstract

This work considers the problem of using multiple small, low-cost robots, with a limited range of local communication ability, to collaboratively search and engage an indeterminate number of tasks in an unknown large-scale hostile area. In this paper, we provide an approach, which we refer to this schemas the gradual expansion-based exploring approach (GEBEA), to explore environment by the cooperating robots. We also propose a task allocation approach for solving the multi-robot multi-task allocation problem. The key idea in these approaches is to make the robots automatically build up and maintain a dynamically stable ad hoc network when they are exploring or engaging tasks. That makes the robots get the advantage of behaving both collaboratively and decentralized, which normally is only available in the system that has global communication. In this case, although each robot only has limited range of communication, by using the ad hoc communication network, all of the information obtained by any robot can be shared by the others. This allows robots to make decisions based on global information and reduce the possibility of overlap exploration. At the same time, the dynamic central control architecture generated by the robots, which finds the target task during the exploration period, can generate an efficient task solution.

1 INTRODUCTION

People prefer using robots to search and engage tasks in a hostile and hazardous environment. Many studies [1-5] indicated that those missions could be performed more effectively by employing large numbers of cheap, simple mobile robots as opposed to fewer numbers of expensive, complex robots. For collaboratively executing tasks by those robots, the control architectures for collaborative robotics need to be developed.

The traditional hierarchical control architectures are able to generate optimal or near optimal solution when the control modules have full information in real-time [6]. However, such architecture could easily fail under some hostile conditions, such as the battlefield, deep under-water, and some large-scale disaster environments. In these conditions, the global communication network is easy to be jammed. Sometimes it is also impossible to build a global communication because of the interference of the enemy in battlefield or just because of the special environment condition, such as in deep under-water area where radio waves can't go through. For the central or hierarchical control system, lost communication connection between controller and robots implies failure of the whole system. Some advanced communication methods, such as WLAN, can provide high bandwidth and high anti-interference communication in those hostile environments, but the communication range is very short. In addition to the communication problem, the highly dynamic changed environment and the mere presence of large number of mobile robots within the operational environment would make it impossible to process the information in real-time and generate an optimal mission planning solution [3,7].

Therefore, some distributed control approaches [2,3,8,9] have been proposed for systems that include a large number of robots, as well as for systems where the environment status is dynamically changed. In such distributed control architecture, each robot decides its own movements, and the robot-to-controller-and-back communication loop isn't necessary anymore. But all of those approaches still rely on global communication for sharing information among the robot team and exchanging cooperative information. The global communication is the key for improving the performance of multi-robot teams in a variety of tasks. However, to make each robots have global communication ability will largely increase the manufacturing cost. In addition, the global communication will consume a lot of energy resources of the robots and make the robot easily locatable by enemy in battlefield.

The problem being addressed in this paper is how to cooperatively search and engage an indeterminate number of tasks in a large-scale hostile area using multiple cooperating mobile robots with limited range communication. To solve the problem, we have developed a collaborative strategy for dynamically, automatically and quickly formatting small, lightweight, low-cost robots into functional teams to collaboratively search and engage tasks in a large-scale hostile area without central control and global communication.

2 PROBLEM STATEMENT

In this paper, we demonstrate our collaborative strategy by applying it in the design of the control architecture of the multiple heterogeneous robots, which are used for search and rescue in large-scale disaster environments, such as an earthquake. The goal of the robots in a search and engage process is to cover the whole disaster environment, find out the tasks and solve the tasks in a minimum amount of time [2]. For this large-scale disaster environment, we assume that:

1) The area affected by the large-scale disaster is very large compared to a single robot's local communication reach ability.
2) After the disaster, there are many kinds of rescue tasks that should be completed as soon as possible. Different kinds of rescue tasks need different rescue robots collaboratively working together.
3) The scales of the rescue tasks are varied at different places. For some tasks, maybe only one rescue robot is needed to solve the task. For other tasks, there maybe a need for a lot of different

kinds of rescue robots to collaboratively work together to solve the task.

4) After the disaster, the communication infrastructures in the disaster area are often badly damaged. No one can predict the actually number of the rescue tasks, the location of the tasks and the time the tasks will appear. The rescue robots have to search the whole disaster area to find the tasks.
5) The radio inference caused by the disaster and unmanaged radio usage makes the large area (global) radio communication channel jammed.
6) There are different kinds of robots deployed in the disaster area. And each kind of robot is specially designed for working on one kind of rescue task.
7) Some robots may fail during the rescue action because of damage or other mechanical problem and they cannot execute searching and rescuing mission anymore.
8) Each robot has basic mobile ability, self-location ability and a simple set of common reactive behaviors that enable it to avoid obstacles, search nearby rescue tasks, and engage on rescue tasks. Each robot is equipped with a limit-range sensor that can detect the rescue tasks nearby. It also has a larger but still limited-range transmitter/receiver for robot-to-robot communication capability. Each robot doesn't have global communication capability.

3 RELATED WORK

The problem of exploring an environment using multiple mobile robots has been considered for many years. In Yamauchi's frontier-based approach [9], all robots share an occupancy grid map. The occupancy grid map is incrementally built by all robots' sensor input during the robots' exploration. Yamauchi introduces the notion of the frontier, which means the regions on the boundary between open space and unexplored space. All robots follow a greedy strategy to move to the nearest frontier. Except to update the sensor information to the shared map, there is no explicit cooperation between each robot. That means there is a possibility that each robot will repeatedly cover same area and even interfere with other robots physically.

Burgard [2] suggested an algorithm for coordinating a group of robots while they use an occupancy grid map [9] to represent their environment. Instead of using a greedy algorithm to guide the robots move to the nearest unexplored area, this algorithm explicitly coordinates the robots by simultaneously considering the utility of the unexplored area and the cost to move to the area. The data communicated between robots include the sensor information and cooperating control message. All of those messages are transferred through a global communication method. If in some scenarios, the global communication is not available or not suitable to be used, the cooperative approach will fail. Most of the time, the robots that are equipped with some kind of wireless LAN transceiver, can only communicate with nearby robots. Under this circumstance, to exchange the information within those robots, the most efficient approach is employing an ad hoc wireless network [10].

Winfield [11] proposed a mechanism that uses broken ad hoc wireless networks to transfer the sensed data back to a collection point from a group of robots that are dispersed throughout the environment. Winfield argues that this mechanism, which relies on robots' continuous random motion to bring the robots into contact, is feasible and robust for some mission scenarios that do not require real-time data collection. However, the un-real time communication through the broken ad hoc networks make robots in this environment can not coordinate their action and may cause some robots to repeatedly explore same area while other areas are never explored.

Schlecht and Altenburg [1, 8] use local communication to maintain a parallel alignment of the UAVs (unmanned Air Vehicle) to search an area, orbit a circle around the target to conduct a synchronized multi-point attack. The parallel alignment helps UAV preserving continuous network connectivity through the local communication. Maintaining the alignment relies on the faster UAVs reducing their searching speed while waiting for slower UAV to catch their step. All of the UAVs have to circulate around the target and wait for the last UAV arriving before they start to attack the target. This parallel alignment control mechanism can get good performance in the application for exploring and engaging one task in a large-scale environment. It doesn't have good performance in the scenario where the robot's exploring speed may be different in different regions and large number of tasks is randomly dispersed in the unknown environment.

In most of the proposed exploring approaches, the goal of the algorithm is to use the robots' sensors to cover the whole environment in a minimum amount of time. The task solving time and task allocation method are not considered in these approaches. The significant different way of our approach to the other approaches is that we simultaneously consider the problem of multiple robots exploration and task allocation. In our approach, in addition to minimizing the time for robots covering the whole environment, the approach also needs to cooperate the robots to solve the tasks that robots find during their exploration and minimize the task solving time. In this paper, a sub-task period roaming approach is proposed to solve this multi-robot and multi-task allocation problem.

The task allocation approach described in this paper is similar to the dynamical role assignment methology described by Chaimowicz [12], who considers the problem of cooperative transportation task by multiple robots. In his approach, each robot performs a set of roles that define its actions. By dynamically changing roles according to the status of the environment, the robot team is able to complete the multiple tasks successfully. However, the robots must rely on the global communication that is not available in a hostile environment and the random search methods used by the robot cannot guaranty the robots' sensors covering the whole environment in minimum time.

4 PROBLEM SOLUTION APPROACH

In covering the whole large-scale disaster environment, finding the tasks and solving the tasks in a minimum amount of time and avoid overlap, it is essential that the robots keep track of which areas of the environment have already been explored and the locations where newest tasks are found. That means each robot in the environment needs to keep some kinds of communication channel with all other robots for exchanging the environment information. Since each robot has only limited-range communication ability and the environment needs to be explored is larger than the cover area of an individual robot's communication range, the best way for making the robots communicate to each other in this environment is by constructing an ad hoc communication network in this swarm of robots. The focus of this section lies in the question of how to coordinate the robots' action in order to maintain this ad hoc network during their search and engage tasks in the environment.

4.1 Represent the unknown environment by occupancy beehive map

The robots in the system use occupancy beehive maps to represent the environment. Each cell in the beehive map is a hexagonal block with unique identify number and each cell in the map has the same area. In each cell, no more than one robot will be deployed in this cell for exploring. The cell's area is dependent on the communication range of a robot. We assume each robot's communication range can only cover the immediate nearby cells of the cell that it is located in. Beyond the immediate nearby cells, the robots cannot directly communicate with each other. As showed in Figure 1, this is a part of the environment area that the robots will work in. Robot A's communication range will cover the cells surround cell number 102. It can communicate with the robots in the 6 cells that surround the cell number 102, such as the robot B, C and D in cell number 101, 104 and 106. The robots in other cells such as robot E, F and G cannot directly communicate with robot A because the distance between them has exceeded the communication range.

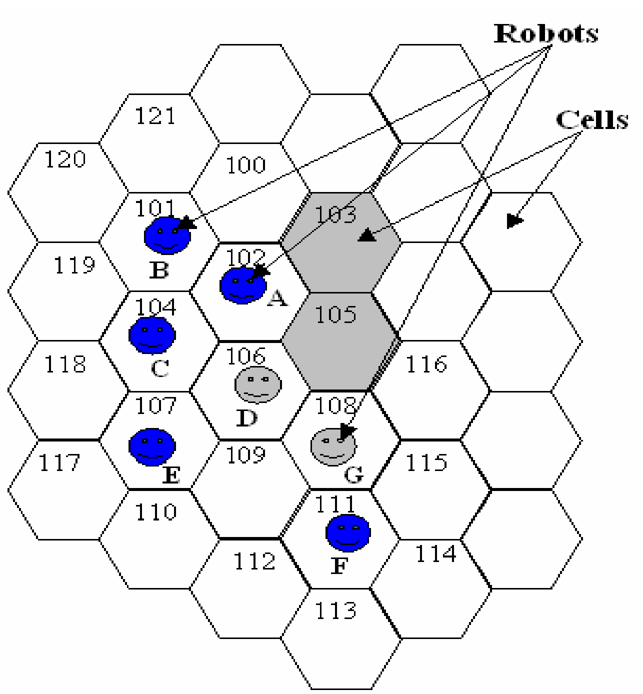

Figure 1: the occupancy beehive map

Each cell in such an occupancy beehive map is given a numerical value to represent the time the robot takes to completely explore it. The corresponding area of each cell has different geographical status and needs different times for a robot to complete the exploration. Each cell is an unknown area for the robot before it is explored. The robots cannot predict the time taken before it completes the exploration. Considering the complexity of searching a cell by a robot, we assume the time taken for exploring a cell is much more than the time taken for one robot moving from one cell to another cell.

4.2 Maintain the ad hoc communication network during search

Before the robots start their mission, they must form an ad hoc communication network for exchanging information that is required for collaborative search and engage in the environment. An ad hoc network can be formed in a group of robots if any robot in this group is covered by a least one other robot's communication range. We assume, at the initial time, a large number of robots are deployed at the environment and a subset of the robots can form an ad hoc network. For the robots that are accidentally deployed in an area where it is out of the communication range of the main ad hoc network group, they will execute an asterisk-shaped exhaustive search pattern [4] to find the main group. The center of this asterisk is the initial position of the robot and the radius of the asterisk is the estimated distance to the main group.

Each robot in the environment has a unique identifying number that makes it distinguished from other robots. After the mission start, each robot will broadcast its identifying number, its current status and its current location as a hello message to its nearby robots within a special time period, and the nearby robots will relay this hello message to other robots and at last every robot in the ad hoc network will receive this hello message. Each robot that receives this hello message will use it build a global table to represent the current status of robots in the group. This global table should be updated in a special time period, and if any robot's status information in this table does not receive update during the period, it will be considered as a failed or lost robot and will be de-listed from the global table. At the same time, each robot will use the occupancy beehive maps method to build its own environment status map as showed in Figure 1. According to the information of the global status table, a robot can determine which areas have already been explored by other robots and which cells are currently under exploration by other robots, and then choose to explore another unexplored cell.

To efficiently explore the unknown environment and at the same time keep the dynamic stability of the ad hoc network that is generated at the beginning of the mission, each robot will use the gradual expansion-based exploring approach to choose its next exploration cell and decide where to navigate. In this approach, the expansion is defined as the unexplored and un-occupied cell in the status map that has at least one robot located in one of its nearby six cells, and this robot cannot be the robot that uses this definition to find its expansion. That means, for different robots, their location will decide their specific views about the expansions in the environment. In figure 1, the expansions of the robot B in cell number 101 include cell number 100, 119, 118, 117, 110, 109, 112, 113, 114, 115 and 116. But for the robot E in cell number 107, the cell number 117 and 110 are not considered as expansions, because those two cells don't have other robots nearby except the robot E. At the same time, cell number 121 and 120, which aren't considered as expansions by the robot B, are considered as robot E's expansions. By using their own expansion list, each robot will always try to find the nearest expansion as its next navigate target for exploring. By following this law, we can assure that every time when a robot starts to navigate to a new un-explored cell, it can keep the connection with the ad hoc communication network of the main robots group.

4.3 Increase the robustness of the ad hoc communication network

The accuracy of the global environment information that each robot uses for making decisions is highly dependent on the stability of the ad hoc communication network that is built by the robots. To make this ad hoc network be more robust to failures of individual robots, the system should always be alert to the loss of individual robots. When robot A doesn't receive hello message from its nearby robot B in a special time period, the robot B will be considered as disabled robot and will be de-listed from robot A's global status map. If robot A finds that because of robot B's failure, it lost connection with the ad hoc network, it will move to the position that the robot B had stayed and rebuilt the connection with the ad hoc network. The algorithm that robot A uses to find out that it is lost connection with the network can be found in many papers about the routing protocols for ad hoc mobile wireless network [10].

Using gradual expansion-based exploration approach, if one robot is the only router for routing information for another robot or robots, its navigation to a new unexplored cell may cause another robot or robots to lose connection with the ad hoc network. In our implementation, if a robot planning move to a new location, the robot will first consult with its nearby robots. If it receives a connection broken alarm from nearby robots, it will wait in it original location until the alarm is relieved.

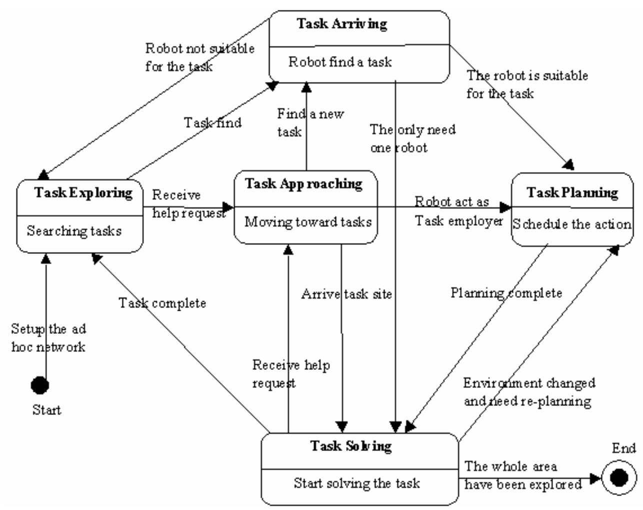

Figure 2: The state diagram and behaviors summary of robots

4.4 Task allocation

The gradual expansion-based approach helps robots maintain an ad hoc communication network during their exploring period. If one or more robots find tasks during exploring, the robots should try to solve the tasks as soon as possible. For solving the task allocation problem in this large-scale multi-task scenario, a new methodology, sub-task period roam approach, will be used for the task allocation between multiple heterogeneous robots in the large-scale environment. In this methodology, the sub-task period is composed of five periods: task exploring, task arriving, task planning, task approaching, and task solving. These sub-task periods indicate the different process sections for a robot to complete a task.

At any time, each robot should be located in one and only one sub-task period. During the whole robot's working period, each robot independently roams among those sub-task periods and all the robots in the environment compose the best robot swarm formation for optimizing the whole system performance. According to its surrounding environment and the information it received from other robots across the ad hoc communication network, each robot in the environment will independently decide which sub-task period it should be located in. When the global environment's change reaches a threshold, the robot will roam to other sub-task periods. Figure 2 shows the robot's task allocation state diagram for the robots using sub-task period roam approach in the simulated environment.

Each robot will start from the task-exploring period and will use the method mentioned in previous section to build an ad hoc communication network. When the events' changes have reached a threshold that is defined by the robot and causes the robot a need to change its status period, the robot will execute a set of rules to decide which sub-task period it should transit to. Part of the rules is list below:

WHILE (Environment events reach a threshold)
IF (AND((current, exploring), (task, appear), (utility, positive)))
ROAM(task arriving)
END-IF
IF (AND((current, exploring), (task, assignor attached), (utility, positive), (receive, help request)))
ROAM(task approaching)
………
………
………
END-IF
IF (other)
ROAM(task exploring)
END-IF
END-WHILE

By following the behavior rules, each robot can independently decide its sub-task period. After the robot transits to another sub-task period, it will act on the emergent behaviors that are specially designed for this period. Figure 3 gives an example about this scenario. In figure 3, robot E is the robot that finds the task. Although this robot is suitable for solving the task, to efficiently solve the task, more robots are needed to collaboratively work together. After finding the task, robot E attaches the task, transits to task planning period and acts as task assignor. According to robot E's plan, the task needs another three robots to work together. Robot A, robot B and robot D are suitable robots for this task. Robot C's ability is not the kind of robot that can solve the task. As the task assignor robot, robot E informs robot D, Robot B and robot A about the task.

After receiving the task information from the task assignor robot E, robot A, robot B and robot D transit from the task exploring period to the task approaching period, act as hired task assignees and start moving towards the task. After the hired task assignees arriving at the task site, the task assignor and assignees become a temporal task-solving group and the robot control model in this group become the central control architecture. In this whole task solving period, the task assignor will act as the leader of this temporal task-solving group and will direct all other task assignee robots' actions. If for some reason one task assignee needs to withdraw from this temporal group, the robot should first inform the group leader, the task assignor robot. The leader will try to find another robot to fill in the empty position. The leader will also monitor the process of task solving. If it finds some changes of the status of the environment, it will modify its plan for solving the task and directing other robots' actions.

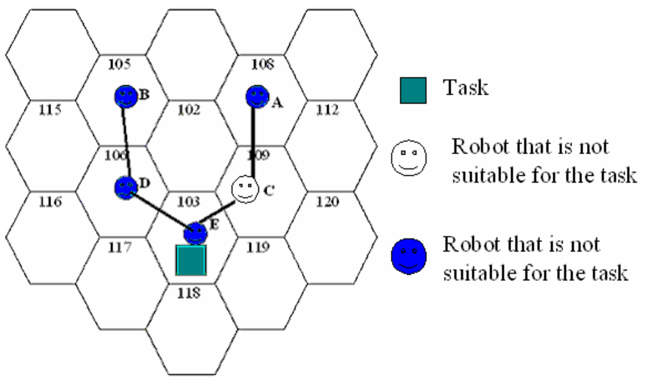

Figure 3: The task allocation by using local communication

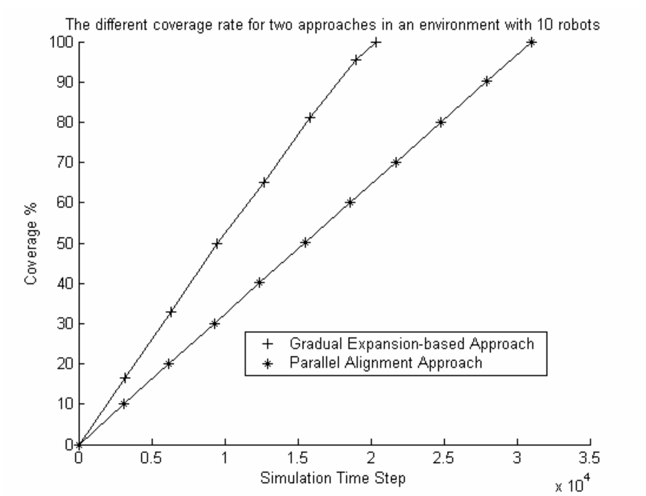

Figure 4: Comparing different ad hoc search approaches

5 SIMULATOR IMPLEMENTATION

To assess the performance of the proposed gradual expansion-based approach, we have conducted a simulation. In this simulation, the environment is a rectangular bounded area and the whole area is visualized as a grid type. The number of robots deployed in this simulated environment is n. The robot's limited wireless range can only cover its neighbor cells. The time for a robot to move from one cell to its neighbor cell equal 1 simulation time step. The time t that a robot consumed for exploring and engaging tasks in each cell is a random number between 1 to T time step. And this number is unknown to the robot before the robot moves into the cell. The key performance indicator is the length of simulation time steps for the robots completely exploring and for engaging tasks of all the cells in this bounded area.

For comparing the performance of our approach with other approaches, the parallel alignment approach proposed in [8] is also implemented in the simulation. This parallel alignment approach is the only cooperative task exploring and engaging approach we can find that uses local communication as the only communication method in a group of robots. It is similar with the task searching and solving method that a group of people generally uses for exploring a large area. Figure 4 shows the environment coverage against the simulation time steps, for two chosen approaches tested in an environment where robot number is n=10 and the cell exploring time range is T=30. As we expected, the coverage speed of the gradual expansion-based approach is faster than the parallel alignment approach. The time consumed for exploring and engaging tasks of all cells in the environment by using our approach is only 60% of the time consumed by parallel alignment approach.

6 CONCLUSION AND FUTURE WORK

This work considers the problem of using multiple small, low-cost robots, with limited range local communication ability, to collaboratively search and engage indeterminate number of tasks in an unknown large-scale hostile area. A collaborative search and engage strategy is proposed in this paper. The strategy includes a gradual expansion-based exploring approach for helping robots cooperatively search the environment and a subtask period roaming approach for solving the multi-robot multi-task allocation problem. A simulator has been implemented to test our approaches. According to the result of the simulation, the approach has a better performance than the parallel alignment approach [8] when they are tested in an unknown large-scale static environment that has indeterminate number of tasks. To test our collaborative strategy in a dynamical environment, our next step is to implement this collaborative search and engage strategy in a large-scale disaster rescue simulator. The implementation of the simulator is based on the RoboCup Rescue Simulator (RCRSS) [7], a computer simulation system providing a virtual environment where large-scale disaster such as earthquakes can be simulated and heterogeneous rescue agents, which simulate the real world rescue robots, can collaborate in the task of disaster mitigation.

References

[1] Altenburg K, Schlecht J, Nygard K. 2002. "An Agent-based Simulation for Modeling Intelligent Munitions". Proceedings of 2nd WSEAS Int. Conf. on Simulation, Modeling and Optimization, pages 60 – 65.

[2] W. Burgard, M. Moors, and F. Schneider. 2002. "Collaborative exploration of unknown environments with teams of mobile robots". In dagstuhl. Springer Verlag, pages 52 - 70.

[3] R. Cassinis, G. Bianco, et al. 1999. "Strategies for navigation of robot swarms to be used in landmines detection". Proceedings of. Eurobot '99, Zürich, pages 211 - 219.

[4] Kwok, K. and Driessen B. 1997. "Cooperative Target Convergence Using Multiple Agents", Proceedings of the SPIE, Sensor Fusion and Decentralized Control in Autonomous Robotic Systems, Vol. 3209, pages 67 – 75.

[5] Lynne E. Parker. 2000. "Current State of the Art in Distributed Autonomous Mobile Robotics", Proc. of Distributed Autonomous Robotic Systems, pages 3 –12.
[6] Y.U. Cao, A.S. Fukunaga, A.B. Kahng, F. Meng,1995. “Cooperative mobile robotics: antecedents and directions”, International Conference on Intelligent Robots and Systems, Volume 1, Pittsburgh, Pennsylvania, USA, pages 226 – 234.
[7] S. Tadokoro, H. Kitano, and others, 2000. “The RoboCup-Rescue Project: A Robotic Approach to the Disaster Mitigation Problem”. Proceedings of IEEE International Conference on Robotics and Automation, pages 87 - 96.
[8] Schlecht J, Altenburg K, Ahmed B, Nygard K, 2003. "Decentralized Search by Unmanned Air Vehicle using Local Communication" International Conference on Artificial Intelligence, Las Vegas, Nevada, 6 pages.
[9] B. Yamauchi. 1998. “Frontier-based exploration using multiple robots”, Proceedings of the Second International Conference on Autonomous Agents, pages 47 -53.
[10] E.M.Royer and C.Toh. 1999. "A review of current routing protocols for ad hoc mobile wireless networks", IEEE Personal Communications, pages 46 -55.
[11] A. F. Winfield. 2000. “Distributed sensing and data collection via broken ad hoc wireless connected networks of mobile robots”, In L. E. Parker, G.W. Bekey, and J. Barhen, editors, Distributed Autonomous Robotics Systems, volume 4, pages 273–282. Springer
[12] Chaimowicz, L., Campos, M., Kumar, V. 2002. “Dynamic Role Assignment for Cooperative Robots”, ICRA 2002 (ICRA02), Washington, D.C. pages 292 – 298.

Biographies

Xiaohui Cui (x0cui001@louisville.edu) received his M.S. degree in computer science from Wuhan University in August 2000, and his B.E. degree in Electrical Engineering from Wuhan Technology University of Surveying & Mapping in August 1992. He is current a PhD candidacy in the Computer Engineering and Computer Science Department at the University of Louisville. He is a research assistant of the Multimedia Laboratory and Bioinformatics Laboratory. His research interests include Collaboration of Robots, Multi-agent system and distributed computing.

Rammohan K. Ragade (ragade@louisville.edu) received his Ph.D. degree from IIT-Kanpur, India in 1968. He is a Professor of Computer Engineering and Computer Science at the University of Louisville. He is also the Coordinator for the Ph.D. Program in Computer Science and Engineering. He is a senior member of IEEE and a member of the ACM. His research interests include object oriented methodologies, knowledge engineering and expert systems, system simulation , wireless networks, computer security and Multi Agent Systems technology.

Adel S. Elmaghraby (adel@louisville.edu) Adel S. Elmaghraby is professor and Chair of the Computer Engineering and Computer Science Department and the Director of the Multimedia Research Lab at the University of Louisville. He has also held appointments at the SEI-CMU, and the University of Wisconsin–Madison. His research is in Intelligent Multimedia Systems, Neural Networks, PDCS, Visualization and Simulation. He is a member of editorial boards, and technical reviewer. He has been recognized for his achievements by several professional organizations including a Golden Core Membership Award by the IEEE Computer Society.