Role-based autonomous multi-robot exploration

de Hoog, J.; Cameron, S.; Visser, A.

Publication date
2009

Document Version
Author accepted manuscript

Published in
Computation world: Future computing, service computation, cognitive, content, patterns: ComputationWorld 2009, proceedings: 15-20 November 2009, Athens, Greece

Role-Based Autonomous Multi-Robot Exploration

Julian de Hoog, Stephen Cameron
Computing Laboratory
University of Oxford
Oxford, UK
{julian.dehoog, stephen.cameron} @comlab.ox.ac.uk

Arnoud Visser
Intelligent Systems Laboratory
Universiteit van Amsterdam
Amsterdam, NL
a.visser@uva.nl

Abstract—**Thanks to advances in both computer science and engineering, the divide between robotics and multi-agent systems is shrinking. Robots are capable of performing an ever wider range of tasks, and there is an increasing need for solutions to high-level problems such as multi-agent coordination. In this paper we examine the problem of finding a robust exploration strategy for a team of mobile robots that takes into account communication limitations. We propose four performance metrics to evaluate and compare existing multi-robot exploration algorithms, and present a role-based approach in which robots either act as explorers or as relays. The result is a complete exploration of the environment in which information is efficiently returned to a central command centre, which is particularly applicable to the domain of rescue robotics.**

***Keywords*-multi-agent systems; robotic exploration; role-based behaviour; limited communication; search and rescue robots**

I. Introduction

In the past, robots have typically been used primarily in industry and by the military, but steady advances in robotics, mobile communication networks and multi-agent systems mean that it is now possible to develop robot systems for an ever wider range of tasks. Such tasks include surveillance and reconnaissance, robotic search-and-rescue, underwater and planetary exploration, and bomb disposal.

While every robotics application has its own challenges, certain problems are common to a wide range of applications. One such problem is robotic exploration: how can a robot or a team of robots be made to explore and map a previously unknown environment? There has been significant progress in exploration by single robots, but multi-robot exploration remains a young field of study having several open problems, such as how to coordinate robots, how to merge information obtained by several robots, and how to deal with limited communication [1].

In this paper, we focus on the communication problem. As an example, consider search-and-rescue robots that must enter and explore wreckage left behind by large disasters such as earthquakes. In such environments, rubble, steel, and various building materials can lead to very poor communication ranges. A robot or team of robots exploring under such conditions must therefore employ an exploration strategy that is robust to communication drop-out and failure.

We propose a set of performance metrics to evaluate and compare various exploration algorithms, and we describe a novel exploration strategy that allows robots to explore efficiently, independently and irrespective of connectivity to teammates. The results are applicable not only to mobile robots but to any application involving communication-limited agents.

Our paper is structured as follows: in Section II we discuss related approaches to multi-robot exploration. Section III describes the problem we are hoping to solve, introduces our simulation environment and its communication model, lists and describes our proposed performance metrics and outlines our approach to multi-robot exploration. In Section IV we present our results, and in section V we discuss the strengths and weaknesses of our experiments and results, along with their implications for future work in this area.

II. Related Work

Among existing multi-robot exploration approaches, only a small number explicitly take the possibility of limited communication into account. The ones that do either explicitly plan on keeping robots within range of one another or apply an underlying strategy that encourages robots to stay in range.

Early approaches impose a line-of-sight constraint between robots [2], [3]. Robots enter the environment one at a time until they reach the limit of the line-of-sight constraint. In a variation of this approach, robots reactively choose a direction that will most likely keep them within sight of the rest of the team [4].

Several authors propose multi-robot exploration strategies based on market principles, in which robots place bids on sub-tasks of the exploration effort [5]–[8]. These bids are typically based on values such as expected information gain and travel cost to a particular location in the environment, and may be assigned in a distributed fashion among team members, or by a central agent. When strength of communication is factored into the bids, robots avoid areas outside of communication range.

Another common strategy for robotic exploration is to use frontiers [9], which can easily be extended to multiple robots [1], [10]–[12]. Similar to bids described above, utilities of individual frontiers may include a factor related to likelihood of communication success, so robots are less likely to explore areas that take them out of the team communication range.

Further approaches include the use of energy fundamentals to maintain network connectivity [13], results from graph theory to keep individual robots in comfort zones [14] and

† This conference paper is published in the following proceedings:
COMPUTATIONWORLD '09, pp.482-487, 15-20 Nov. 2009, doi: 10.1109/ComputationWorld.2009.14
The original publication is available at http://ieeexplore.ieee.org/ 978-0-7695-3862-4/09 $26.00

the application of synthetic spring forces to keep robots close to one another [15]. In related work, attempts have been made to limit the bandwidth requirements of a robot team [16].

While several of these approaches have proven successful in maintaining team connectivity during the exploration effort, they are usually limited by the constraint of having to keep team members within communication range. Even if members of a team are dispersed to the maximum extent that their communication ranges allow, in large and complex environments unexplored areas will remain.

A solution to this problem is to allow robots to autonomously explore beyond communication range limits. This can be implemented in terms of "robot pack" or clustering behaviour, in which groups of robots stay close together as they explore the environment [1], [8], [11].

However, for many applications (e.g. robotic search-and-rescue), information gathered by the robot team must be returned to a single point or command centre. To the best of our knowledge, there is no existing approach that explicitly plans on autonomous exploration of far reaches of the environment with a goal of returning all attained knowledge to a particular location. We hope that the method we propose here is a first step in this direction.

III. Our Approach

A. Problem Description

The exploration problem we hope to solve is as follows:

- Agents represent physically embodied, mobile robots having a range sensor.
- There is no prior knowledge of the environment. Agents only know what they have sensed themselves and what their teammates communicate to them.
- Agents may only communicate when they are within range of one another, as specified by our communication model (see section III-C).
- Agents enter the environment in the same location and are initially aware of one another's locations.
- At the point of entry there is a command centre, and the goal of the exploration effort is to relay as much information as possible to this command centre. In robotic search-and-rescue, this corresponds to the human responders' point of entry.

B. Simulation Environment

To evaluate and compare our role-based exploration algorithm, we developed MRESim, a JAVA-based simulation environment[1]. Environments, including walls and obstacles, may be specified by the user. MRESim takes an environment as input and simulates robot movement, collisions, sensor data and communication. At each time step, this is performed as outlined in Algorithm 1.

Currently we assume perfect sensor data and localisation. This is not realistic, but for now it is sufficient for purposes of comparison of exploration algorithms. In future work we intend to relax this assumption (see also section V-B).

[1]Available upon request from the authors

```
foreach agent do
    nextLoc = requestDesiredLocation(agent);
    if isValid(nextLoc) then
        move(agent, nextLoc);
        sensorData = simulateSensorData(agent, nextLoc);
        sendData(agent,sensorData);
    end
end
foreach agent do
    foreach agent2, agent2 != agent do
        if isInRange(agent, agent2) then
            communicateData(agent, agent2);
        end
    end
end
updateGUI();
```

Algorithm 1: Steps taken to simulate movement, sensor data and communication in one time step of MRESim

C. Communication Model

We have implemented and tested a variety of communication models in our simulations. For experiments reported here we use a standard path loss model with a wall attenuation factor as described in [17]:

$$S = P_{d_0} - 10 \times N \times log_{10}(\frac{d_m}{d_0}) \begin{cases} nW \times \text{WAF} & nW < C \\ C \times \text{WAF} & nW \geq C \end{cases}$$

where P_{d_0} is the reference signal strength, N is the path loss rate, d_m is the distance, d_0 is the reference distance, nW is the number of obstructing walls, WAF is the wall attenuation factor and C is the maximum number of walls to consider. This model is widely used in simulation, including the popular USARSim simulator [18]. A typical communication range for an agent is diplayed in Figure 1.

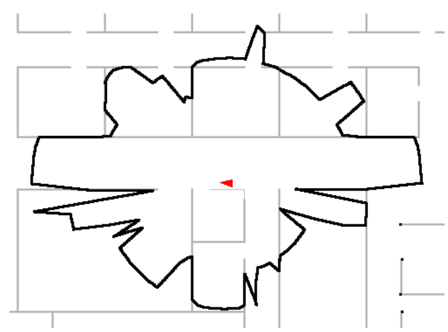


Fig. 1. Typical communication range for an agent using the communication model described in section III-C

D. Performance Metrics

For the evaluation and comparison of individual exploration algorithms, it is helpful to have a common set of easily measurable performance values that indicate the relative success of each approach. We propose the following four metrics:

1) *Total area explored (maximise).* We use the union of the areas explored by all robots of the team. With 'explored area' we mean the area that has been sensed by an agent (in our simulations, using the range scanner).
2) *Total knowledge of the environment at the command centre (maximise).* In applications such as robotic search-and-rescue, knowledge gathered by members of the team is only useful if it reaches the human responders. While some robots may know about far reaches of the environment, they may not have an opportunity to communicate this knowledge back to the command centre.
3) *Percentage of full exploration effort known to individual agents (maximise).* For a robot to explore a part of the environment, it is helpful to know what its teammates are doing and what other parts of the environment have been explored. The more any robot knows about the other robots' actions, the easier it is to efficiently coordinate the team effort.
4) *Time since last contact with the command centre for individual agents (minimise).* Human responders will want to have control over the robot team (for example if an environment is deemed of extreme danger or low priority in the middle of an effort and the robots need to be moved to a different environment). It is not desirable to have robots out of the range of the command centre for a long time.

E. Role-based Exploration

An overview of our role-based exploration approach is provided in Figure 2. We assign each agent one of two roles:

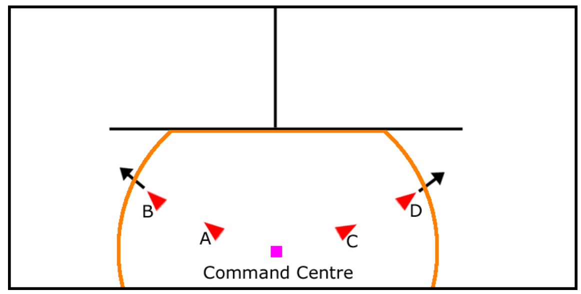
1) *Explorer*. Explorers are meant to explore the farthest reaches of the environment. To communicate their findings, they return periodically to previously agreed upon rendezvous points where they pass their knowledge to a relay.
2) *Relay*. Relays act as links between explorers and the command centre. The primary purpose of a relay is to communicate explorers' findings back up the communication chain. If a relay discovers information about the environment while relaying, this is added to the team knowledge, but exploration is only a by-product of the relay's movement.

In our current implementation, roles are assigned prior to the start of exploration and do not change. Consequently the team hierarchy may be represented by a tree.

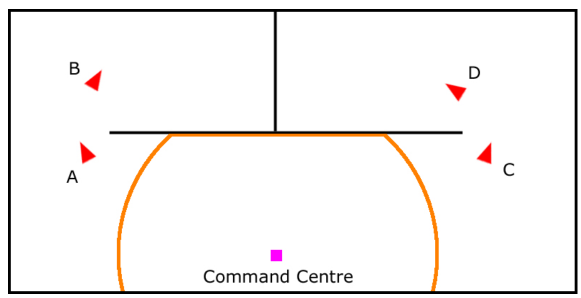
Explorers use frontier exploration to explore. When there are multiple explorers active, frontiers are assigned to explorers using the algorithm outlined in [12], *i.e.* each explorer calculates the best explorer-frontier pairing for all explorers in range, and thereby knows which frontier is best for it to explore. When an explorer is in range of its parent relay, it passes all known information to this relay. Furthermore, it calculates the time it will take the parent relay to reach its own parent relay and return to the rendezvous. As a result, an explorer knows exactly how much time it has to continue exploring before it must return to rendezvous once again with its parent relay.

Rendezvous points are dynamically set by an explorer during the exploration process. When it is time for an explorer to return to a previously set rendezvous point, it saves its current location and sets this to be the next rendezvous. As a result, the exploration effort is pushed deeper and deeper into the environment, and with each subsequent meeting relays are forced to come further to meet the explorer once again.

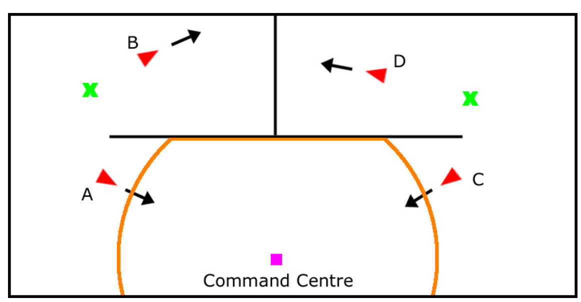
State transition diagrams for explorers and relays are presented in Figure 4.



(a) The command centre is represented by the magenta square with the orange line its communication range. B is an explorer with A as its parent relay; D is an explorer with C as its parent relay. Initially the explorers set out to explore using frontier-based exploration, and the relays follow them.



(b) All robots are now out of range of the command centre. The relays continue to follow the explorers for a short time period, but soon decide to relay all current knowledge back to the command centre. An explorer and its parent relay agree on the explorer's current location as the next rendezvous point.



(c) Rendezvous points are marked by the green crosses. The relays now return to the command centre, while the explorers continue to explore. Explorers can calculate how long it will take their parent relays to return to the rendezvous point, and can arrange to meet them there at the right time. Subsequent rendezvous points will be placed deeper and deeper in the environment.

Fig. 2. Role-based exploration: an overview.

IV. RESULTS

To compare role-based exploration with existing methods, we implemented four multi-robot exploration algorithms (corresponding uppercase letters are used throughout the rest of this paper):

A) Frontier-based, no exploration beyond the team's communication range limits
B) Frontier-based, exploration beyond the communication range limits, and robots return when there are no more frontiers left to explore, *i.e.* when the exploration effort is completed
C) Frontier-based, exploration beyond the communication range limits and regular periodic return by each robot to the command centre
D) Role-based exploration beyond communication range limits, as described above

In our tests, we used three types of environment: bare, cluttered, and room-based (see Figure 5 for a screenshot of the room-based environment). In each of the environments we ran each of the four algorithms once with two robots, and once with four robots. All tests were run until completion, meaning that either the environment was fully explored, or that it was not possible for the robots to explore any further. The total number of runs was thus (4 exploration algorithms) * (3 environment types) * (2 team configurations) = 24 runs. The results were reasonably consistent over the various runs, here we present a representative set from running the four algorithms in the room-based world using four robots.

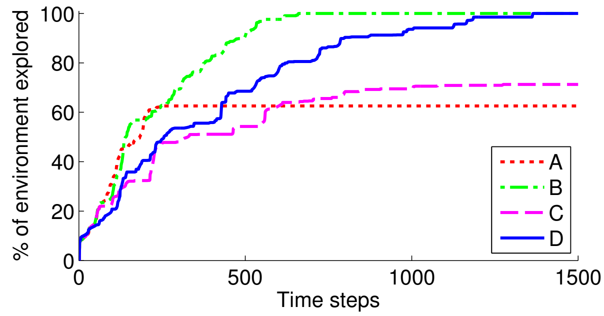
Returning to the performance metrics proposed in section III-D, a comparison of the four exploration algorithms is presented in Figure 3.

Only algorithms B and D were able to fully explore the environment – algorithm A was limited by communication range, and the return time interval in algorithm C was not long enough for robots to reach the far limits of the environment.

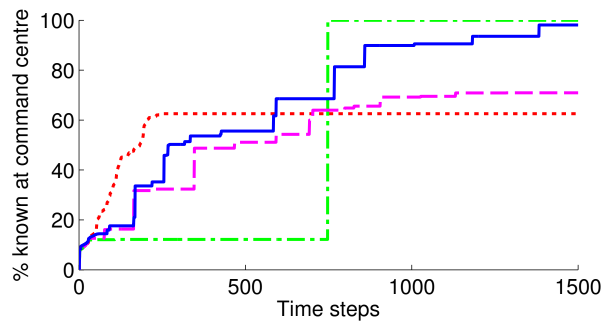
While algorithm B led to faster exploration, this would not have been known to anyone at the command centre – the full exploration effort only became available at the command centre all at once after 747 time steps when the exploring robots returned to the command centre's communication range.

For algorithm A, where all robots were always in range of one another, naturally the transfer of knowledge between agents and the time since last communication with the command centre were perfect. Consequently, however, only 62.5% of the environment was explored.

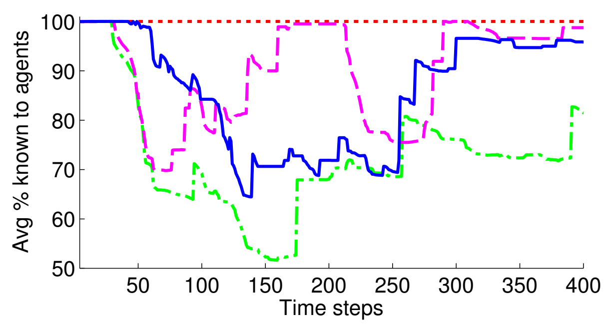
Among the other algorithms, B led to poor sharing of knowledge among agents and poor responsiveness to central commands. Algorithms C and D showed similar performance regarding sharing of knowledge and responsiveness. After approximately 800 time steps, algorithm C shows greater responsiveness to central commands, but this is due entirely to the fact that robots weren't reaching the farthest depths of the environment, while for algorithm D the exploration effort led to a complete exploration.



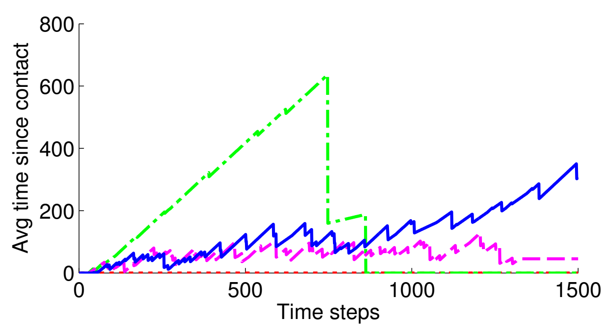
(a) Percentage of environment explored



(b) Percentage of environment known at command centre

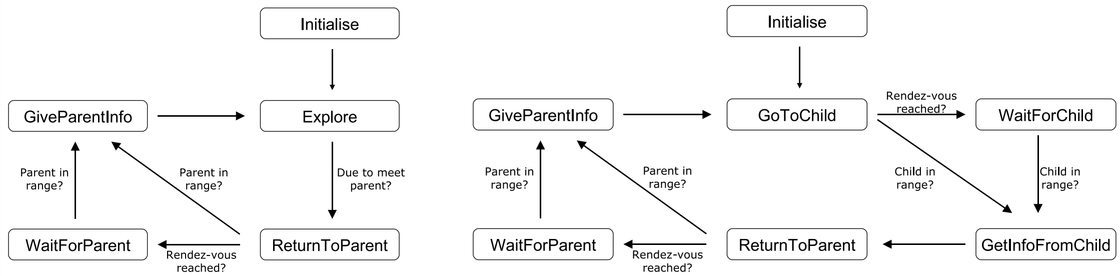


(c) Percentage of full exploration effort known to individual agents (average over all agents)



(d) Time passed for individual agents since last contact with command centre (average over all agents)

Fig. 3. Performance of the four exploration algorithms outlined in section IV using the four performance metrics proposed in section III-D. These results are based on exploration of a room-filled environment using four robots. In all graphs the x-axis represents simulation time steps.



(a) Explorers (b) Relays

Fig. 4. State transition diagrams

V. Discussion and Future Work

Our research on role-based exploration is a work in progress. However, initial results are encouraging, and provided certain adjustments are made, role-based exploration shows promise regarding applications such as robotic search-and-rescue.

A. Performance compared to purely frontier-based methods

As pointed out in section IV, role-based exploration is outperformed by non-limited frontier exploration (algorithm B) in terms of area covered, and by communication-limited frontier exploration (algorithm A) in terms of responsiveness to central commands. However, for an application such as robotic search-and-rescue, where it is of interest both (i) to explore as much of the environment as possible and (ii) to maintain the robot team in as tight a communication network as possible, role-based exploration presents a trade-off that in the authors' opinion would be preferable over the purely frontier-based approaches.

B. Potential weaknesses

There are a number of potential weaknesses and sources for error in role-based exploration. If a relay ceases to function, an entirely likely scenario in a potentially dangerous or toxic environment, it may be difficult for an explorer to find its way back up the communication chain. A possible solution would be for all robots to know all rendezvous points leading up to the root of the team hierarchy, and allowing only a limited wait at a given rendezvous before proceeding to the next highest (or lowest) rendezvous point.

Additionally, a dynamic environment may mean that certain locations believed to be reachable may become blocked. A possible solution would be a limited wait at rendezvous, followed by a replanning of the path. To ensure a meeting of child and parent, only the child should recalculate its path (otherwise they may miss each other).

In the results we present here, we assume perfect localisation by individual team members. This is not a realistic assumption, though it does not significantly affect the comparison of individual exploration algorithms. Nevertheless we intend to take noisy sensor data into account in future experiments.

C. Potential extensions

Several extensions and optimisations of the role-based approach could lead to much improved behaviour. An extension that the authors envision is the development of "territorial" exploration: as relays move back and forth between explorers and the command centre, the relays could contribute significantly to the exploration effort by exploring themselves. This would also support what is considered to be one of the main advantages of using multiple robots for exploration: a heterogeneous robot team. Small fast robots may be able to reach and provide information about the deepest parts of the environment quickly, while slower robots having more intricate sensors may be able to fully explore parts of the environment close to the command centre.

An additional extension of great interest is the use of stationary sensor nodes to aid the exploration effort. Stationary sensor nodes may be used to deposit or retrieve information at various locations in the environment, once they have been deployed by a robot. This has been used with success in various other applications [19], [20] and ties multi-robot exploration to the domain of sensor networks. It could be of great interest to deploy such nodes at rendezvous points, and inter-node communication could significantly reduce the robots' travel costs.

Finally, the current implementation is limited by the static team hierarchy. In certain situations, maintaining such a static hierarchy can lead to unnecessarily long travel to rendezvous points and inefficient exploration. A more robust and efficient approach would involve a dynamic team hierarchy in which robots may jump from one branch of the tree to another, or where the team is represented as a graph rather than a tree.

In future work, we hope to implement a territory-based multi-robot exploration algorithm, to explore a variety of approaches to dynamic team hierarchies, and to compare the resulting exploration efforts to a wider range of existing approaches.

VI. Conclusion

There is great scope for robots to be used for a wide range of tasks in the near future, including robotic search-and-rescue. For many such tasks, a team of robots must be able to explore a complex, communication-limited environment efficiently.

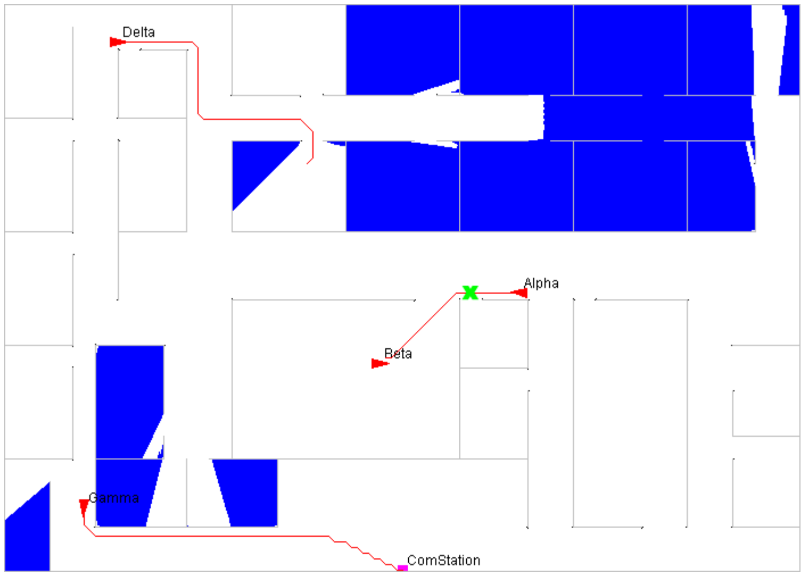


Fig. 5. A screenshot from the role-based exploration algorithm in the room-based environment after 605 timesteps. Blue area is not yet explored, white area has been sensed with a range scanner, red lines indicate robots' planned paths. Alpha and Beta are due to meet at the rendezvous point indicated by the green cross. Gamma and Delta have recently met, and Delta continues to explore while Gamma relays new information back to the ComStation.

Existing multi-robot exploration techniques typically plan for robots to stay in range of one another. In this paper we propose a new role-based approach in which robots take on one of two roles, explorer or relay. Explorers search the far reaches of the environment, but periodically return to meet with relays at previously agreed-upon rendezvous points. The relays in turn carry information about the far reaches of the environment back to a central command centre, equivalent to a human responders' point of entry. As a result, the environment is explored completely, and new knowledge is regularly brought to the command centre.

References

[1] D. Fox, J. Ko, K. Konolige, B. Limketkai, D. Schulz, and B. Stewart, "Distributed multirobot exploration and mapping," *Proceedings of the IEEE*, vol. 94, no. 7, pp. 1325–1339, July 2006.

[2] R. Arkin and J. Diaz, "Line-of-sight constrained exploration for reactive multiagent robotic teams," *Advanced Motion Control, 2002. 7th International Workshop on*, pp. 455–461, 2002.

[3] A. Howard, M. Mataric, and G. Sukhatme, "An incremental deployment algorithm for mobile robot teams," *Intelligent Robots and System, 2002. IEEE/RSJ International Conference on*, vol. 3, pp. 2849–2854 vol.3, 2002.

[4] M. Powers and T. Balch, "Value-based communication preservation for mobile robots," in *7th International Symposium on Distributed Autonomous Robotic Systems*, 2004.

[5] R. G. Simmons, D. Apfelbaum, W. Burgard, D. Fox, M. Moors, S. Thrun, and H. L. S. Younes, "Coordination for multi-robot exploration and mapping," in *Proceedings of the Seventeenth National Conference on Artificial Intelligence and Twelfth Conference on Innovative Applications of Artificial Intelligence*. AAAI Press / The MIT Press, 2000, pp. 852–858.

[6] M. B. Dias and A. T. Stentz, "A free market architecture for distributed control of a multirobot system," in *6th International Conference on Intelligent Autonomous Systems (IAS-6)*, July 2000, pp. 115–122.

[7] R. Zlot, A. Stentz, M. Dias, and S. Thayer, "Multi-robot exploration controlled by a market economy," *Robotics and Automation, 2002. Proceedings. ICRA '02. IEEE International Conference on*, vol. 3, pp. 3016 – 3023, 2002.

[8] W. Sheng, Q. Yang, J. Tan, and N. Xi, "Distributed multi-robot coordination in area exploration," *Robot. Auton. Syst.*, vol. 54, no. 12, pp. 945 – 955, 2006.

[9] B. Yamauchi, "Frontier-based exploration using multiple robots," in *AGENTS '98: Proceedings of the second international conference on Autonomous agents*. New York, NY, USA: ACM, 1998, pp. 47 – 53.

[10] W. Burgard, M. Moors, C. Stachniss, and F. Schneider, "Coordinated multi-robot exploration," *IEEE Transactions on Robotics*, vol. 21, no. 3, pp. 376 – 378, 2005.

[11] M. N. Rooker and A. Birk, "Multi-robot exploration under the constraints of wireless networking," *Control Engineering Practice*, vol. 15, no. 4, pp. 435–445, 2007.

[12] A. Visser and B. Slamet, "Including communication success in the estimation of information gain for multi-robot exploration," *Modeling and Optimization in Mobile, Ad Hoc, and Wireless Networks and Workshops, 2008. WiOPT 2008. 6th International Symposium on*, pp. 680–687, April 2008.

[13] B. Santos Pimentel and M. Fernando Montenegro Campos, "Multi-robot exploration with limited-range communication," in *Anais do XIV Congresso Brasileiro de Automtica (CBA'02)*, 2002.

[14] J. Vazquez and C. Malcolm, "Distributed multirobot exploration maintaining a mobile network," *Intelligent Systems, 2004. Proceedings. 2004 2nd International IEEE Conference*, vol. 3, pp. 113–118 Vol.3, June 2004.

[15] A. Mosteo, L. Montano, and M. Lagoudakis, "Multi-robot routing under limited communication range," *Robotics and Automation, 2008. ICRA 2008. IEEE International Conference on*, pp. 1531–1536, May 2008.

[16] D. Meier, C. Stachniss, and W. Burgard, "Coordinating multiple robots during exploration under communication with limited bandwidth," in *European Conference on Mobile Robots*, Ancona, Italy, 2005, pp. 26–31.

[17] P. Bahl and V. N. Padmanabhan, "Radar: an in-building rf-based user location and tracking system," *IEEE Infocom 2000, Tel-Aviv, Israel*, vol. 2, pp. 775–784, March 2000.

[18] S. Carpin, M. Lewis, J. Wang, S. Balakirsky, and C. Scrapper, "Usarsim: a robot simulator for research and education," in *Robotics and Automation, 2007 IEEE International Conference on*, April 2007, pp. 1400–1405.

[19] V. Ziparo, A. Kleiner, B. Nebel, and D. Nardi, "Rfid-based exploration for large robot teams," *Robotics and Automation, 2007 IEEE International Conference on*, pp. 4606–4613, April 2007.

[20] E. Ferranti and N. Trigoni, "Robot-assisted discovery of evacuation routes in emergency scenarios," *Robotics and Automation, 2008. IEEE International Conference on*, pp. 2824–2830, May 2008.