Communicative Exploration with Robot Packs

Martijn N. Rooker and Andreas Birk

School of Engineering and Science
International University Bremen
Campus Ring 1, D-28759 Bremen, Germany
a.birk@iu-bremen.de

Abstract. Exploration is a core challenge for RoboCup Rescue. So-called communicative exploration is a novel strategy for multi-robot exploration that unlike other approaches takes the limits of wireless communication systems into account. Here, previous results that where achieved for a team of robots linked to a basestation are significantly extended to also cover robot packs, i.e., multi-robot teams that are not permanently tied to an operator's station. Unlike teams that are constrained by the immobility of a basestation, packs can explore arbitrarily large regions. Results from experiments with packs of 4, 5 and 6 robots are presented. The new strategy constantly maintains the communication between the robots while exploring, whereas the commonly used frontier-based exploration strategy, which is used in the experiments as comparison to our approach, leads to a rapid loss of communication.

1 Introduction

Exploration is a core issue for many robotics applications [Zel92, MWBDW02] including especially RoboCup Rescue. Obviously, the usage of multi-robot systems is a very interesting option for exploration as it can lead to a significant speed-up and increased robustness, which both are very important for rescue missions. A popular basis for multi-robot exploration is the frontier-based Exploration algorithm introduced by Yamauchi [Yam97], which was extended by himself [Yam98] as well as by Burgard et.al.[BFM+00] to deal with multiple robots. These extensions suffer the drawback that perfect communication between the robots is assumed. When it comes to real multi-robot systems, communication is based on wireless networks typically based on the IEEE 802.11 family of standards, which is also known as WLAN technology [OP99]. WLAN links suffer from various limitations [PPK+03]. Especially, they have a limited range posing a severe limit on the usefulness of the aforementioned exploration algorithms. In [RB05] we presented a new exploration strategy that takes the range limits into account and that is therefore more suited for real application scenarios. This previous work was limited to robots tied to a basestation, i.e., there was a hard limit to the maximum area that could be explored. Here, we extend the previous work to robot packs, i.e., the constraint of a basestation is dropped. This is especially of interested when working with autonomous robots, which is one of the next big challenges within RoboCup Rescue.

A. Bredenfeld et al. (Eds.): RoboCup 2005, LNAI 4020, pp. 267–278, 2006.

Communicative exploration builds in general on the Frontier-Based Exploration algorithm [Yam97], where a frontier is defined as regions on the boundary between open space and unexplored space. A robot moves to the nearest frontier, which is the nearest unknown area. By moving to the frontier, the robot explores new parts of the environment. This new explored region is added to the map that is created during the exploration. In the multi-robot approach different robots are moving stochastically over to the frontier [Yam98], respectively in a coordinate manner such that multiple robots will not move to the same position [BFM$^+$00]. When we assume a realistic communication model for a multi-robot system, there is a limit to the communication range of each robot. This is not taken into account in previous approaches where nothing prevents the robots from moving further and further away from each other.

We extend the frontier-based exploration such that exploration takes place while the robots maintain a distributed network structure which keeps them in contact with each other through ad-hoc networking [Per00], assuming some underlying dynamic routing [JMH04, RT99, JW96]. This *communicative exploration* algorithm is based on a utility function, which weights the benefits of exploring unknown territory versus the goal of keeping communication intact.

The rest of this paper is structured as follows. Our line of research is motivated in more detail in section 2. The concrete communicative exploration algorithm is introduced in section 3. The experiments and results are presented in section 4. Section 5 concludes the paper.

2 Exploration by Robots Packs

The approach presented in this paper is based upon the frontier-based exploration, which is described in [Yam97, Yam98]. As mentioned in the introduction, the basic idea of this algorithm is simply to move to the boundary between explored and open space. As illustrated[1] in figure 1, robots tend to drift apart and communication is lost.

At the International University Bremen (IUB), a team is working since 2001 in the domain of rescue robots [BKP03, BCK04, BKR$^+$02](Figure2). Rescue robots shall assist first responders in urban disasters scenarios ranging from earthquakes to gas or bomb explosions [RMH01, Sny01]. A typical mission tasks is the detection and localization of victims. Exploration combined with the constraints of real-world communication systems is an obvious topic of interest in this application scenario. We were hence interested in overcoming the limitations of the frontier-based approach.

We describe in [RB05] a first step where the issue of keeping in constant contact with a base-station while exploring is addressed (figure 3). The transmission of data to an operators station is crucial in rescue missions as it can

[1] Cells with explored space are colored dark green, unexplored ones are bright gray, the frontier cells are yellow, obstacles are dark gray. Robots are red spots, their communication ranges red circles. Active links are indicated by red lines.

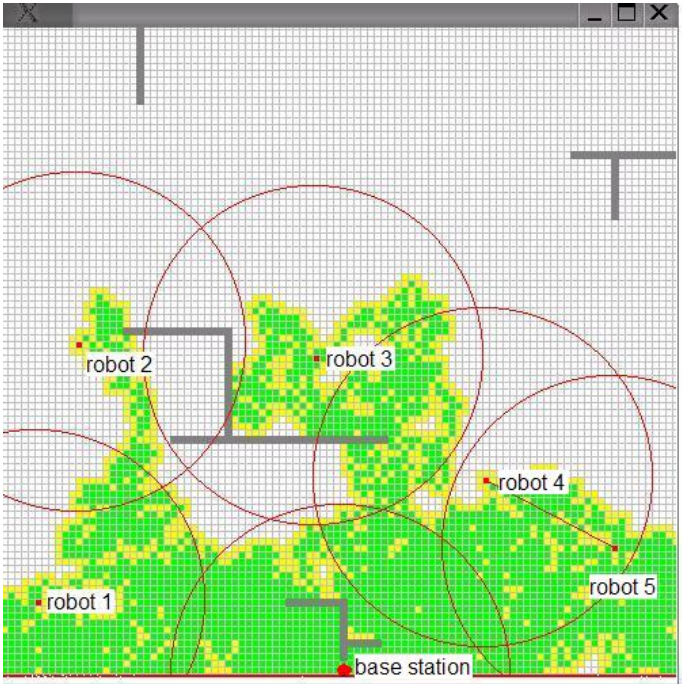


Fig. 1. The frontier-based exploration algorithm does not put any constraints on the spread of the robot pack. The robots soon drift far apart and communication is easily lost. In the above screenshot from a typical simulated run, only robots 4 and 5 are in each others cell and hence capable of communicating with each other.

Fig. 2. Two of the IUB rescue robots at the RoboCup Rescue competition

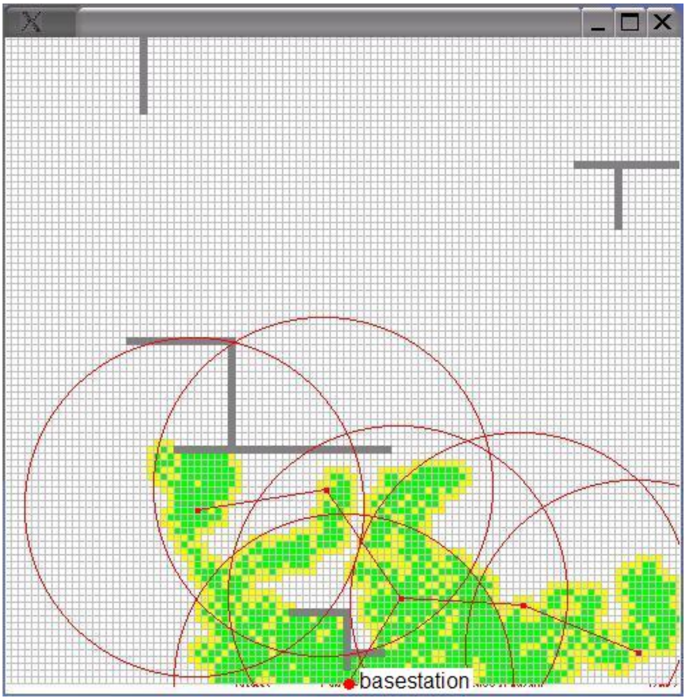


Fig. 3. A screen-shot from a typical simulated run of the basic communicative exploration algorithm where the robots constantly keep in contact with an immobile base-station. For this purpose, an utility function penalizes the motion that lead to a loss of a link in the ad-hoc network formed by the robots. Though being important for some application scenarios, this approach limits the maximum operation area of the robots.

not be assumed that the robots return to the spot where they are deployed, in contrary, their total loss during a mission is a likely risk. Therefore, they have to deliver all crucial information, like victims and hazards found or map-data, ideally on-line to an operators station, which is at a secured position. For this purpose, the robots either have to be in direct contact with the base-station or to use other robots as relays.

In this paper, an extension of our previous results to robot packs is presented. This means that the constraint of keeping contact with an immobile base-station is dropped. Instead, the *communicative exploration* algorithm keeps the network structure in a robot pack intact, which can move freely to do the exploration. This allows to apply the results to arbitrary robot teams. Note that the algorithm can be highly beneficial independent of exploration tasks. Though there is some work dealing with cooperative robots without communication [Ark92], typical architectures for coordinating multi-robot systems like ALLIENCE [Par02] require proper communication structures.

3 The Communicative Exploration Algorithm

The following section describes the *communicative exploration* algorithm in detail. Note that to make a guaranteed "optimal" exploration, i.e., to cover the largest possible area without communication loss, a proper motion-planning for the aggregate of all n robots would have to be done in *every* step, which is for complexity reasons infeasible. The basic idea is therefore to use an utility function that penalizes moves that lead to a loss of communication links. The utility is then used to select a best possible candidate from a random population.

At time t, every robot i has a position $P_i(t) = (x_i, y_i)$, which represents the position in the world W. W is represented by a grid, where every cell can contain four different values, namely *unknown, frontier, visited, obstacle.*

Configuration of n robots at time t is defined as:

$$cfg(t) = \{P_1(t), P_2(t), \cdots, P_n(t)\}$$

For moving to a new configuration at time $t+1$, a configuration change cfg_c is calculated. A configuration change is defined as follows:

$$cfg\_c(t) = \{m_1(t), m_2(t), \cdots, m_n(t)\}$$

with $m_i(t)$ being the movement of robot i at time t. For every robot, the following movements are defined:

$$m_i(t) \in M = \{N, NE, E, SE, S, SW, W, NW, R\}$$

with R representing no movement and the other values a movement in one to the surrounding grid cells, if possible.

With this definition for a configuration change, there are in total 9^n configuration changes possible for n robots. Unfortunately, all possible configurations are not always possible. For example, it could happen that in a specific configuration a robot moves into an obstacle or that multiple robots in a configuration move to the same position, which will result in a collision and maybe damage of the robots. Furthermore, the exponential number of possible new configurations makes it impossible to calculate all of them. Hence, the decision has been made to generate a limited amount of random calculated configurations per time-step. Instead of considering all the 9^n possible configurations, k configurations are calculated with $k \ll 9^n$.

For every new calculated configuration, a *utility value* is calculated. This utility value represents the usefulness of the new calculated configuration. For the calculation of the utility of a configuration, the different possible locations where the robots can move to are taken into consideration. Every robot in the configuration adds a certain value to the total configuration, depending on its position in the calculated configuration. The following possibilities for a new position can occur:

- **Impossible position:** This position can happen in two different situations:
  - Two or more robots want to move to the same position.
  - A robot wants to move to a position that is occupied by an obstacle.

  Configurations with these position should be avoided, therefor a negative value is assigned to these positions.
- **Loss of communication:** As mentioned before, the idea behind the approach is to maintain communication between all robots during the exploration process. The maintenance of communication can be direct or indirect. To check if communication is maintained in a configuration, the whole configuration at this point and not only a specific robot. As soon as one robot in the configuration is out of communication range, the configuration should be avoided. In this case a negative value is assigned to the total utility value of this configuration.
- **Frontier cell:** A frontier cell represents the boundary between explored and unexplored areas of the world. So for exploring new areas, the robots should move to frontier cells. Configurations where robots move to frontier cells are the ones that are favored above all, therefor a positive value is assigned to every robot that moves to a frontier cell.
- **Other:** The last possibility where a robot can move to is a cell that has already been explored. This could also mean that a robot maintains its position in the configuration. Although this position is not optimal, it is not a position that has to be avoided. This position will not add anything to the utility of a configuration, there a "neutral" value is assigned to this position.

The utility value of a single robot position is calculated with the following values:

$$U(P_i(t+1)) = \begin{cases} -100 & \text{if impossible position} \\ 1 & \text{if frontier cell} \\ 0 & \text{otherwise} \end{cases}$$

With these values the total utility value of a configuration change can be calculated with the following formula:

$$U(cfg\_c_i) = \delta + \sum_{i=1}^{n} U(P_i(t+1))$$

with δ being the assigned value for the maintenance or loss of communication. In the experiments performed the value for losing communication is set to -10.

The decision if communication is maintained in a configuration can be done in a rather simple manner. Every robot has a neighbor list. This list contains robots that are directly connected, which means that two robots i and j, $i \neq j$, are within each others communication range. In the approach presented here, it is assumed that if two robots are within communication range of each other, there is a communication link between these two robots.

So every robot within communication range is stored in the neighbor list. These neighbor lists can be seen as adjacent list, which can be used to represent a graph. In this graph the robots are the vertices and if two robots are within each others communication range (and thus in each others neighbor list) an edge consists between these two vertices. Now that a graph is available, the communication maintainability can be calculated. To check if two robots are connected with each other (direct or indirect) there should be a path on the graph between these two robots.

To check if all the robots are connected with each other, it is enough to check if every robot is connected to one specific robot. For this, one robot i is taken as temporary base-station and for every robot j, $i \neq j$ is checked if a connection exist with i. If every robot j is connected with robot i, it also means that every robot in the graph is connected with the other robots. Checking if a path exist between two robots can be done by using well-known graphs algorithms, like DFS or BFS.

As mentioned, there are 9^n different configurations for n robots. The exponential number of possible new configurations makes it impossible to check all of them. Hence, we generate a limited number of random new configurations per time-step and choose the one with the best utility. So, instead of considering all the 9^n new configuration, only k configurations are considered, with $k << 9^n$. In the experiments presented here k is set to 50, leading to an extremely fast evaluation of the possible configurations.

4 Experiments and Results

For the experiments performed, we define a world with some obstacles placed in it, as can be seen in figure 4. The world is represented as a classic evidence grid [ME85], whereby every grid cell can have one of four different values (*unknown, frontier, visited, obstacle*). At the begin of the experiments, all the cells are initialized to unknown. Only the cells that contain obstacles are initialized different.

The robots that are used in the experiments are homogeneous. They have the possibility to explore a certain region. At a certain position, the current position will be marked as *visited* and the surrounding cells will be marked as *frontier*. Every robot has the ability to communicate. If two robots are within a certain range of each other, a communication link is created between them. In the experiments, it is assumed that a communication link is always created when two robots are within communication range. Furthermore, every robot has the ability to move around in the environment. The robots are moving from one grid cell to the other, thereby having the possibility to move vertical, horizontal, diagonal or to remain at their current position.

The frontier-based exploration is used for comparison to the communicative exploration. For testing how well the different exploration approaches perform, the following measurements are taken. For both approaches it is calculated how many grid cells are explored during a run. Each run consists out of 2500 time steps, whereby a time step is defined as every robot making one movement. A movement can in this case also imply that a robot remains at its current position. Furthermore, for the frontier-based approach, the *communication level* is calculated. The communication level indicates how many robots are connected to each other. If there is a connection level of 100% all the robots are connected with each other.

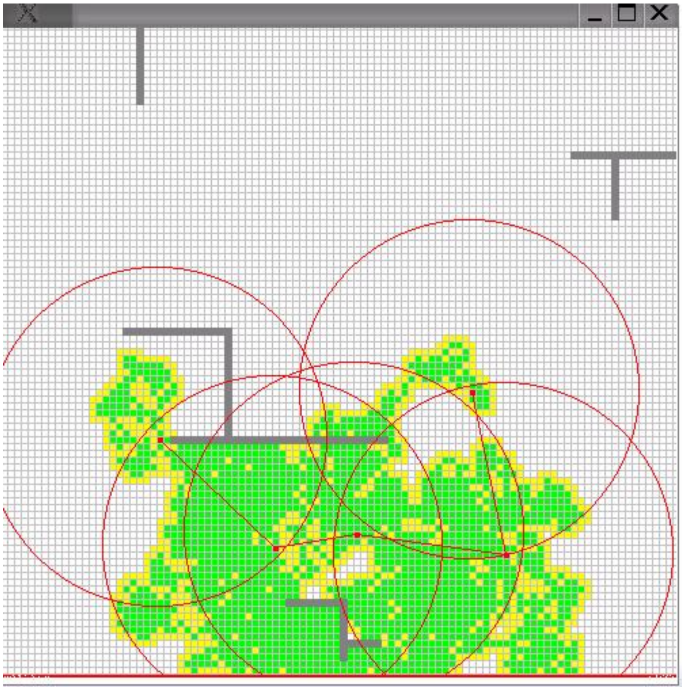
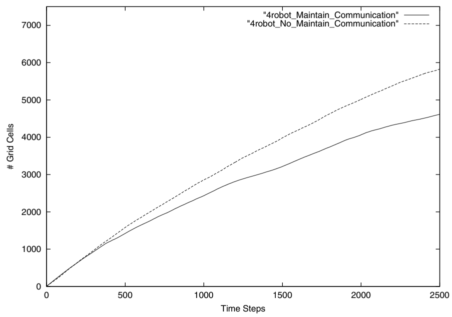


Fig. 4. A pack of 5 robots exploring while maintaining communication with each other

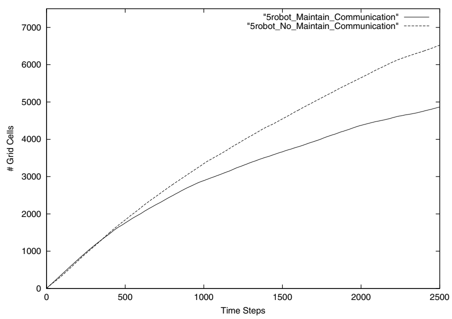
Experiments with different amounts of robots are performed. One experiment uses 4 robots, the second 5 robots and the last experiment 6 robots. Every experiment is performed 10 consecutive times.

The results in terms of exploration speed are shown in figure 5. It can clearly be seen that frontier-based exploration is performing better than communicative exploration in this respect. This can be explained in an easy way. The advantage of the first approach is that the robots do not have a "limitation" on their movement, as do the robots in the communication-based approach have. Therefore they can spread out very fast.

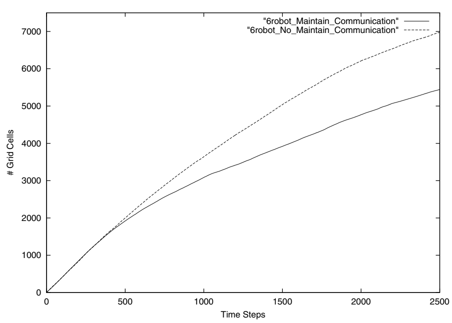
Although communicative exploration moves on slower, its most important task is accomplished. During the whole process of exploration, communication between all the robots is maintained, i.e., the communication level is constantly 100%. This can not be said from the other exploration approach. As can be seen in figure 6, full communication is only established in the beginning of the exploration process, but deteriorates after a few time steps amazingly rapid and never reaches a level of full communication again. The reason for this is exactly the same as the reason why this approach is exploring faster. As the movements of the robots are not bounded by the communication threshold, robots travel rapidly out of each other communication range and are not able to restore the communication link between each other again.



(a) 4 robots

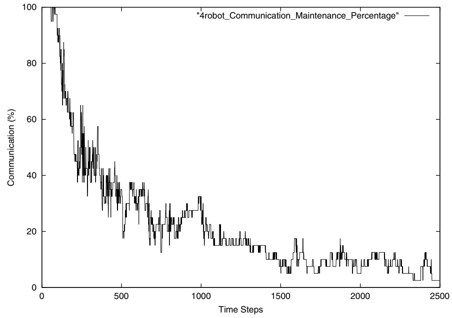


(b) 5 robots

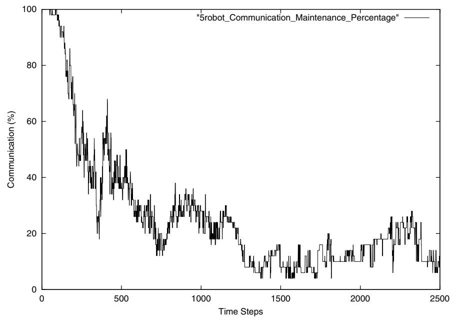


(c) 6 robots

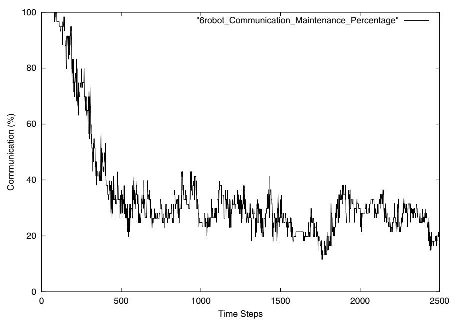
Fig. 5. The amount of grid cells explored with frontier-based and communicative exploration for packs of 4, 5 and 6 robots. The graphs are based on averages of 10 runs. The frontier-based exploration (upper, dashed line) outperforms communicative exploration (lower, solid line). But the communicative exploration maintains communication links between all robots.



(a) 4 robots



(b) 5 robots



(c) 6 robots

Fig. 6. The percentage of communication between the robots during frontier-based exploration for packs of 4, 5 and 6 robots. 100% communication means that every robot is in contact with each other. Again, an average of 10 runs is used. Note that the communication between the robots is lost very fast. Communicative exploration maintains in contrast a constant level of 100%.

5 Conclusions

An extension of the frontier-based approach for exploration [Yam97, Yam98] was presented. In the original algorithm, all robots operate at the borderline to the unexplored space. They hence move further and further away from each other. This leads to communication loss when a realistic network model based on cells with limited ranges is applied. The novel approach of communicative exploration manages to maintain sufficient links between the robots such that a proper network structure is kept. This is achieved by a simple utility function that penalizes the threat of communication losses. In previous work of ours, the approach was limited to robot teams that are constraint by a basestation. Here, we extend the approach to freely moving robot packs. Experiments are presented with packs of 4, 5 and 6 robots where frontier-based exploration leads to a rapid loss of communication. Communicative exploration manages to constantly keep the communication between the robots in the pack intact while exploring. This feature is bought at the expense of somewhat slower progress in the exploration process.

References

[Ark92] R. C. Arkin. Cooperation without communication: Multiagent schema-based robot navigation. *Journal of Robotic Systems*, 9(3):351–364, 1992.

[BCK04] Andreas Birk, Stefano Carpin, and Holger Kenn. The IUB 2003 rescue robot team. In D. Polani, B. Browning, A. Bonarini, and K. Yoshida, editors, *RoboCup 2003: Robot Soccer World Cup VII*, volume 3020 of *Lecture Notes in Artificial Intelligence (LNAI)*. Springer, 2004.

[BFM+00] W. Burgard, D. Fox, M. Moors, R. Simmons, and S. Thrun. Collaborative multi-robot exploration. In *Proceedings of the IEEE International Conference on Robotics and Automation*, 2000.

[BKP03] Andreas Birk, Holger Kenn, and Max Pfingsthorn. The iub rescue robots: From webcams to lifesavers. In *1st International Workshop on Advances in Service Robotics (ASER'03)*. 2003.

[BKR+02] Andreas Birk, Holger Kenn, Martijn Rooker, Agrawal Akhil, Balan Horia Vlad, Burger Nina, Burger-Scheidlin Christoph, Devanathan Vinod, Erhan Dumitru, Hepes Ioan, Jain Aakash, Jain Premvir, Liebald Benjamin, Luksys Gediminas, Marisano James, Pfeil Andreas, Pfingsthorn Max, Sojakova Kristina, Suwanketnikom Jormquan, and Wucherpfennig Julian. The IUB 2002 rescue robot team. In Gal Kaminka, Pedro U. Lima, and Raul Rojas, editors, *RoboCup-02: Robot Soccer World Cup VI*, LNAI. Springer, 2002.

[JMH04] D. B. Johnson, D. A. Maltz, and Y.-C. Hu. The dynamic source routing protocol for mobile ad hoc networks (dsr), July 2004. IETF Internet Draft, draft-ietf-manet-dsr-10.txt.

[JW96] D. B. Johnson and D. A. Waltz. Dynamic source routing in ad-hoc wireless networks. In T. Imielinski and H. Korth, editors, *Mobile Computing*, pages 153–181. Kluwer Academic Publishers, 1996.

[ME85] H. Moravec and A. Elfes. High resolution maps from wide angle sonar. In *Proceedings of the IEEE International Conference on Robotics and Automation*, pages 116–121, 1985.

[MWBDW02] Alexei A. Makarenko, Stefan B. Williams, Frederic Bourgault, and Hugh F. Durrant-Whyte. An experiment in integrated exploration. In *IEEE/RSJ Intl. Workshop on Intelligent Robots and Systems*, 2002.

[OP99] Bob O'Hara and Al Petrick. *The IEEE 802.11 Handbook: A Designer's Companion*. Standards Information Network IEEE Press, 1999.

[Par02] Lynne E. Parker. ALLIANCE: an architecture for fault tolerant multirobot cooperation. *IEEE Transactions on Robotics and Automation*, 14(2):220 – 240, 2002.

[Per00] C. E. Perkins. *Ad Hoc Networking*. Addison Wesley Professional, 2000.

[PPK+03] Jin-A Park, Seung-Keun Park, Dong-Ho Kim, Pyung-Dong Cho, and Kyoung-Rok Cho. Experiments on radio interference between wireless lan and other radio devices on a 2.4 ghz ism band. In *The 57th IEEE Semiannual Vehicular Technology Conference*, volume 3, pages 1798 – 1801, 2003.

[RB05] Martijn Rooker and Andreas Birk. Combining exploration and ad-hoc networking in robocup rescue. In Daniele Nardi, Martin Riedmiller, and Claude Sammut, editors, *RoboCup 2004: Robot Soccer World Cup VIII*, volume 3276 of *Lecture Notes in Artificial Intelligence (LNAI)*, pages pp.236–246. Springer, 2005.

[RMH01] M. Micire R. Murphy, J. Casper and J. Hyams. Potential tasks and research issues for mobile robots in robocup rescue. In Tucker Balch Peter Stone and Gerhard Kraetszchmar, editors, *RoboCup-2000: Robot Soccer World Cup IV*, Lecture keywordss in Artificial Intelligence 2019. Springer Verlag, 2001.

[RT99] E. M. Royer and C.-K. Toh. A review of current routing protocols for ad-hoc mobile wireless networks. *IEEE Personal Communications Magazine*, pages 46–55, April 1999.

[Sny01] Rosalyn Graham Snyder. Robots assist in search and rescue efforts at wtc. *IEEE Robotics and Automation Magazine*, 8(4):26–28, December 2001.

[Yam97] B. Yamauchi. A frontier-based approach for autonomous exploration. In *Proceedings of the 1997 IEEE International Symposium on Computational Intelligence in Robotics and Automoiaton*, pages 146–151, July 1997.

[Yam98] B. Yamauchi. Frontier-based exploration using multiple robots. In *Proceedings of the Second International Conference on Autonomous Agents(Agents '98)*, pages 47–53, May 1998.

[Zel92] Alexander Zelinsky. A mobile robot exploration algorithm. *IEEE Transactions on Robotics and Automation*, 8(6), 1992.