Combining Exploration and Ad-Hoc Networking in RoboCup Rescue

Martijn N. Rooker and Andreas Birk

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

Abstract. In challenging environments where the risk of loss of a robot is high, robot teams are a natural choice. In many applications like for example rescue missions there are two crucial tasks for the robots. First, they have to efficiently and exhaustively explore the environment. Second, they must keep up a network connection to the base-station to transmit data to ensure timely arrival and secure storage of vital information. When using wireless media, it is necessary to use robots from the team as relay stations for this purpose. This paper deals with the problem to combine an efficient exploration of the environment with suited motions of the robots to keep data transmissions stable.

Final Version


```
@INCOLLECTION{commexlpore_rcup04,
AUTHOR = {Rooker, Martijn N. and Birk, Andreas},
TITLE = {Combining Exploration and Ad-Hoc Networking in RoboCup Rescue},
BOOKTITLE = {{RoboCup} 2004: Robot Soccer World Cup VIII},
PUBLISHER = {Springer},
SERIES = {Lecture Notes in Artificial Intelligence (LNAI)},
EDITOR = {Nardi, Daniele and Riedmiller, Martin and Sammut, Claude},
VOLUME = {3276},
PAGES = {pp.236-246},
YEAR = {2005}}
```


1 Introduction

At the International University Bremen (IUB), a team is working since 2001 in the domain of rescue robots [1–3](Figure1). Like in many other challenging domains for service robots, robot teams can be of huge benefit. As in many other applications, there is one basic chore that is of highest importance, namely to ensure the coverage of the entire environment by the robots. This task, commonly known as exploration, can obviously benefit from using multiple robots jointly working in a team. As the problem of exploration is a common one, there is already a significant amount of contributions using multiple robots as discussed in detail later on in section 2.

Another particular challenge is to ensure the transmission of data to an operators station. This is especially difficult when wireless media is used. Furthermore, it can 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[6], 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, i.e., to incorporate ad-hoc networking.

Fig. 1. *Two of the IUB rescue robots at the RoboCup 2003 competition.*

In this paper an approach is introduced that makes use of the Frontier-Based exploration algorithm [7, 8], which requires no explicit planning for the robots. This reactive approach for exploration has the disadvantage that all robots travel further and further into the unknown territory, hence loosing the contact to the base-station. 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 the base-station. 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. In our experiments, we show that the randomized algorithm yields results that are very close to the theoretical upper bound of coverage while maintaining communication.

The rest of this paper is structured as follows. Section 2 discusses related work. In section 3, the communicative exploration algorithm is introduced. Experiments and results are presented in section 4. Section 5 concludes the paper.

2 Related Work

Different approaches have been introduced in the field of exploration. Zelinsky [11] presents an algorithm where a quadtree data structure is used to model the environment and the distance transform methodology is used to calculate paths for the robots to execute. Makarenko et.al.[10] present an exploration strategy which balances coverage, accuracy and the speed of exploration. They also introduce a metric called localizability, that allows comparison of localization quality at different locations.

Another approach that is also used in this article is the Frontier-Based Exploration approach defined by Yamauchi [7, 8]. In this approach, 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 over the frontier. Burgard et.al.[9] define a similar approach, but the difference is that the robots in their approach coordinate their behaviors, so that multiple robots will not move to the same position.

The other aspect that is tackled in this paper is Ad-Hoc Networking [12]. Ad-hoc networks are defined by Perkins as wireless, mobile networks that can be set up anywhere and anytime. The concept of ad-hoc networks is applied in different approaches in exploration. Howard [13] implements a deployment algorithm whereby the robots are placed over the environment in a way that they are able to explore the whole environment, but through the network are still able to communicate with each other. Nguyen [14] describes a system exploring a complex environment. The system consist out of four Pioneer robots that are used as autonomous mobile relays, to maintain communication between a lead robot and a remote operator. Some other examples where ad-hoc networking is applied are habitat monitoring [15], medical sciences [16] and childcare [17].

3 Adding Communication to Exploration

The *communicative exploration* algorithm builds upon Frontier-Based approach on the exploration side. The crucial aspect that has been added is the maintenance of communication. Before our new algorithm is introduced, the basic Frontier-Based approach is shortly re-visited in the next subsection.

3.1 Frontier-Based Exploration and its Extension

Frontier-Based Exploration is introduced by Yamauchi in [7, 8]. He defines frontiers as regions on the boundary between explored and unexplored space. The idea behind this exploration approach is motivated as follows: *To gain as much new information about the world, move to the boundary between open space and uncharted territory.* When a robot moves to a frontier, it can look into the unexplored environment. By exploring the new environment, this data is added to a map that is maintained by the robot. Every time a robot explored new parts of the environment, the mapped region is expanded and the frontier moves over the environment. Through the moving of the frontiers, the robot increases its knowledge of the environment. The explored environment is represented by evidence grids [18]. On this grid a graph is created that is used to plan a path over the grid toward the nearest frontier.

Like the basic Frontier-based approach, the communicative exploration algorithm also avoids complex planning. We are interested in a reactive approach of exploration. To supplement the basic approach, a value is assigned to every movement of a robot, the so-called utility. The utility allows to penalize the loss of communication while rewarding the exploration of unknown space. Based on the utility, communicative exploration becomes an optimization problem.

3.2 The Utility of Robot Movements in Communicative Exploration

Communicative exploration proceeds in time-steps t. At every time-step t each robot has a position in the environment. The position of robot i at time t is denoted by $P[i](t) = (x_i(t), y_i(t))$. The set of positions of all n robots in the environment at time t is called a *configuration*, denoted as $cfg(t)$:

$$cfg(t) = \{(x_1(t), y_1(t)), (x_2(t), y_2(t)), \cdots, (x_n(t), y_n(t))\}$$

To calculate the Utility of a new configuration, the following algorithm is applied. For every robot a new configuration is calculated. In total there are 9 different possibilities for a new configuration for 1 robot, including that the robot stays at its position, as can be seen in figure 2.

So for n robots, there are 9^n different configurations possible. Formally, a configuration change at time t 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, defined as:

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

As mentioned, there are 9^n different configurations for n robots. Some of these configurations are not possible, like for example when multiple robots move to the same position or if a robot moves into an obstacle.

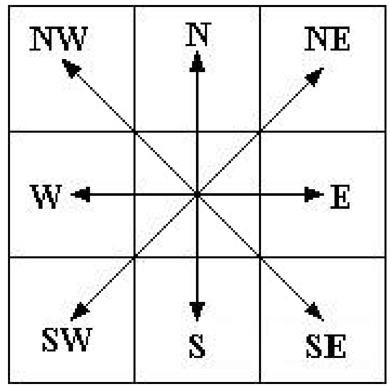


Fig. 2. *The 9 new possible new positions of a robot.*

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, giving an extremely fast evaluation of the possible configurations.

This creates a set $S(t)$ of k configuration changes:

$$S(t) = \{cfg\_c_1(t), cfg\_c_2(t), \cdots, cfg\_c_k(t)\}$$

wherein $cfg\_c_i(t)$ is a configuration change i for a set of robots. A configuration change causes a robot to move to a new position $P'[i](t)$.

For the calculation of the utility of a configuration change, the different options where a robot can move to have to be defined. When a robot wants to move to a new position, the following situations can occur:

- **Impossible position:** When one of the following situations occurs:
  - Two or more robots want to move to the same position.
  - A robot wants to move to a position that is occupied with an obstacle.

  These locations should be avoided, therefor a negative, repulsive value is assigned to those locations.
- **Loss of communication:** A robot wants to move to a location where there is no communication possible with the base-station, directly or indirectly. The process of checking if a robot is still in communication with the base-station is described in section 3.3. Also here it is the case that these locations need to be avoided, so once again a negative, repulsive value is assigned.
- **Frontier cell:** The location where the robot wants to move to is a location on the frontier. The frontier cells are the locations in the world that need to be explored, so a positive, attractive value is assigned to those locations.
- **Other:** The last option where a robot can move to is a location on the field that has already been explored. It could also mean that a robot maintains

its position. This option is not optimal, but definitely not wrong as it avoids obstacles, so therefor a “neutral” value is assigned to these locations.

The following return values are defined for the different situation:

$$U(P'[i](t)) = \begin{cases} -100 & \text{if infeasible} \\ -10 & \text{if loss of communication} \\ 1 & \text{if frontier cell} \\ 0 & \text{other} \end{cases}$$

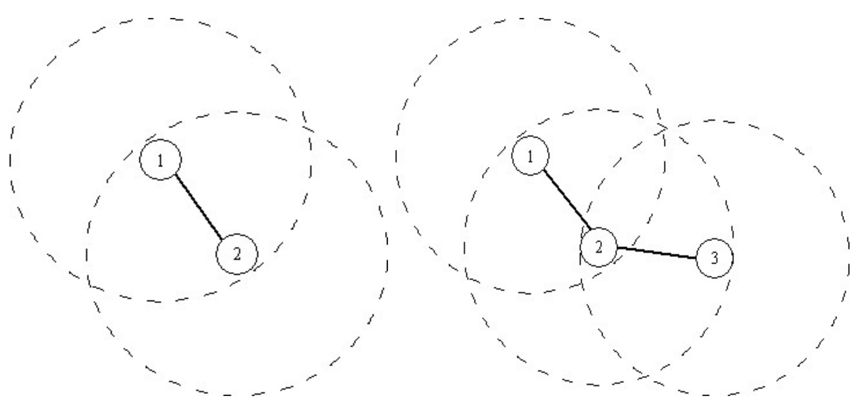
with $1 \leq i \leq n$.
The whole Utility of a new configuration is then calculated as follow:

$$U(cfg\_c_i(t)) = \sum_{i=1}^{n} U(P'[i](t))$$

The configuration change with the highest utility value is then selected. After all the robots have arrived at their new position, the whole process is repeated.

3.3 Detecting Communication

To properly determine the utility of a configuration, it is necessary that the robots check whether communication is maintained or not. A robot is able to directly communicate with another robot if it is within the communication range of that other robot. If a robot i is not within communication range of another robot j, the possibility still exist that robot i is able to communicate with robot j. This is illustrated in figure 3.



(a) Direct communication. (b) Robot 1 communicates indirect with robot 3 via robot 2 and vice versa.

Fig. 3. *Direct and indirect communication between robots.*

As the robots are moving around in the environment, connections between robots are broken and created. At a specific configuration cfg_i a certain amount of connections are possible. Through the locations of the robots and the connections between the robots a connection graph $CG = (V, E)$ is created, where $V = \#robots + \#base - station$ and E the set of communication connections between robots and base-station. If there is a path between robot i and robot j, than those robots can communicate with each other, i.e., we have a properly connected network.

If two robots are (indirectly) connected to each other, than there has to be a path between the two nodes in the graph. In our situation we are not only interested in connections between robots, but in connections between all the robots and the base-station. As the goal is to have to robots always in connection with the base-station, there always has to be a path between every robot and the base-station.

Of course, over a graph, different paths are possible. As it is ideal that the data is delivered as fast as possible to the base-station, the shortest communication path between a robot and the base-station has to be found. Shortest path in this situation is defined as the communication path with as few hops as possible.

This problem is now reduced to the well-known all-pairs shortest path problem [19]. To solve the problem, the *Floyd-Warshall Algorithm* [20] is used. This algorithm returns two matrices, one containing the amount of hops on a path and the other one containing the parent of a node. If the amount of hops between node i and node j is negative and node i does not have a parent, there is no path between these two nodes and thus there is no communication possible between these two nodes.

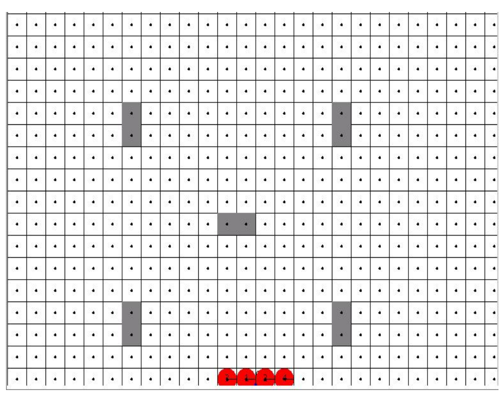
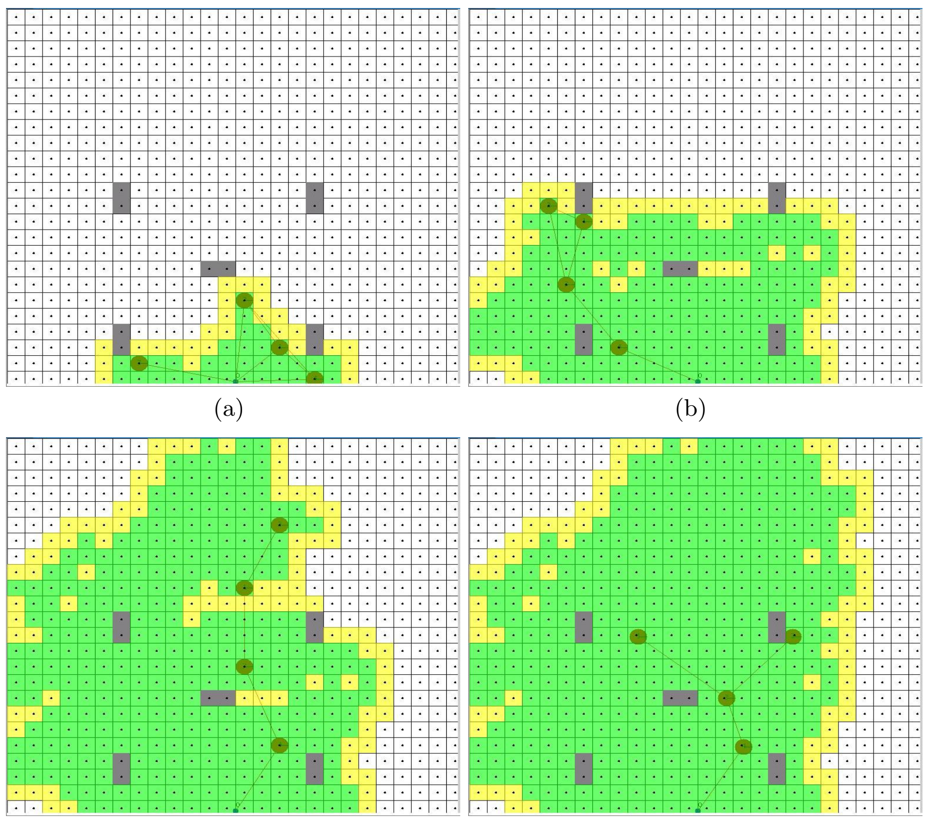


Fig. 4. *The start positions of the robots at the center bottom. The five grey areas are obstacles.*

4 Experiments and Results

The communicative exploration algorithm is tested in simulation. The robots start at a fixed position near the base-station. From here the robots are calculating new configurations and slowly start exploring the environment while maintaining communication with the base-station. A snapshot of the start situation can be seen in figure 4. The red circles in the bottom center of the figure are the robots. They are starting at the base-station. The grey squares are obstacles.

As can be seen in figure 5, the robots spread out nicely, while exploring the environment and maintaining communication with the base-station.



(a) (b) (c) (d)

Fig. 5. *Exploration of the environment by 4 robots.*

¿From figure 5 it can be seen that the frontier between explored and unexplored space is constantly connected. Only if an obstacle intersects the frontier it is not connected at that point. Through this connected frontier, a continuous explored space is created.

For the experiments the following measurements are taken. As one of the goals of the experiments is to remain communication with the base-station, there is a hard limit to the maximum explored space. The robots can move furthest away from the base-station by forming a chain over which the communication is relayed.

For n robots an upper limit m_n for the space they can explore with working communication can hence be calculated as follows:

$$m_n = (\pi(r \cdot n)^2)/2$$

with r being the communication range of a robot.

This upper limit can be directly expressed in grid-cells:

$$mg_n = m_n/S(gc)$$

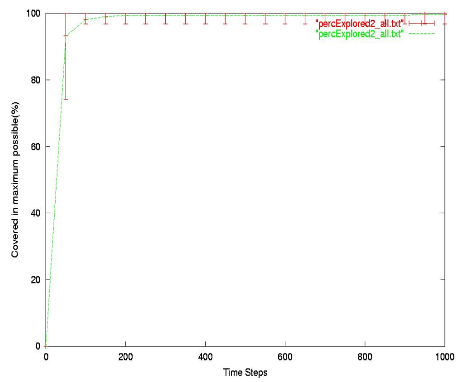
with mg_n being the maximum amount of grid-cells that can be explored with n robots and $S(gc)$ being the surface of a grid-cell.

This upper limit is roughly speaking the area of a half-cycle around the base-station with a radius of $r \cdot n$, i.e., the radius is achieved by chaining the robots at the limit of their communication ranges together. Note that this is an upper limit as there are several reasons why a smaller area is likely to be the real limit for the explorable area. Obstacles for example can easily limit the maximum range.

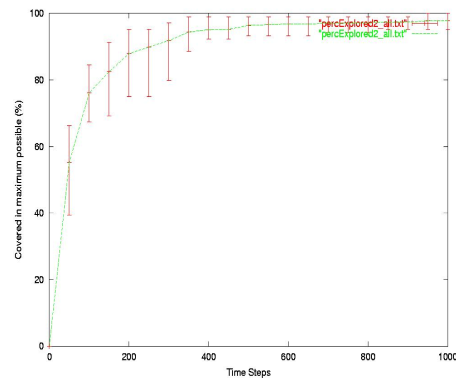
Based on this upper limit, the percentage of explored space while maintaining communication can be calculated in experiments (figure 6). ¿From the graphs it can be seen that in the case of 1 and 2 robots, almost constantly 100% of the upper limit is explored in all the runs. In the case of 3 or more robots, it appears that not the whole upper limit is achieved. A reason for this is the influence of the obstacles. 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. Our solution might "waste" a few cells that might have been reachable but that are not explored in the end. But this solution is highly efficient and yields large and continuous explored areas.

5 Conclusions

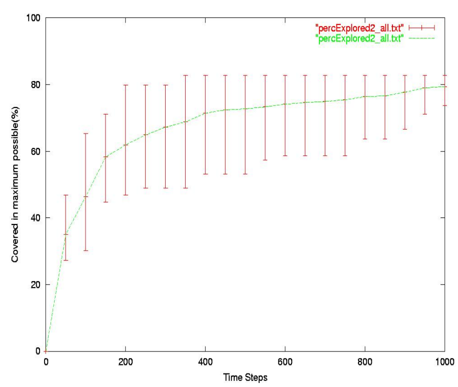
We presented the *communicative exploration* algorithm. It is an extension of the Frontier-Based approach for exploration [7][8]. Communicative exploration deals with the problem that in many real world applications, the robots have to be a part of a wireless network. In the original algorithm, all robots operate at the borderline to the unexplored space. They hence move further and further away from the point where they were deployed. But this point of deployment is typically an operator's station to which the robots should transmit their data. As they move away, all connections to the base-station get lost. Our extension of the algorithm ensures that each robot explored parts of the environment that are within the range of the robot's communication cell. Therefore, the transmission of data from all robots to the base-station can constantly be maintained.



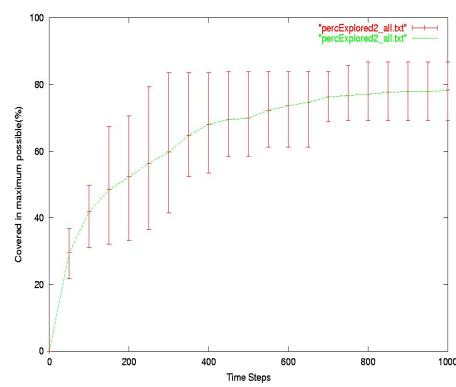
(a) 1 robot



(b) 2 robots



(c) 3 robots



(d) 4 robots

Fig. 6. *Area coverage. With 1 and 2 robots the whole maximum possible area is covered. With more robots it takes longer before the maximum possible area is covered.*

References

1. Andreas Birk, Holger Kenn and Max Pfingsthorn. The IUB Rescue Robots: From Webcams to Lifesavers. *1st International Workshop on Advances in Service Robotics (ASER)'03*, 2003
2. Andreas Birk, Stefano Carpin and Holger Kenn. The IUB 2003 Rescue Robot Team. *RoboCup 2003: Robot Soccer World Cup VII*, Springer, Lecture Notes in Artificial Intelligence (LNAI), 2004.
3. Andreas Birk, Holger Kenn, Martijn Rooker, Akhil Agrawal, Horia Vlad Balan, Nina Burger, Christoph Burger-Scheidlin, Vinod Devanathan, Dumitru Erhan, Ioan Hepes, Aakash Jain, Premvir Jain, Benjamin Liebald, Gediminas Luksys, Marisano James, Andreas Pfeil, Max Pfingsthorn, Kristina Sojakova, Jormquan Suwanketnikom and Julian Wucherpfennig. The IUB 2002 Rescue Robot Team. *RoboCup-02: Robot Soccer World Cup VI*, Gal Kaminka, Pedro U. Lima and Raul Rojas, Springer, LNAI, 2002.

4. M. Micire, R. Murphy, J. Casper and J. Hymas. Potential tasks and research issues for mobile robots in robocup rescue. In Tucker Balch and Gerhard Kraetszchmar, editors *RoboCup 2000: Robot Soccer World Cup IV*, Lectures keywords in Artificial Intelligence 2000. Springer Verlag, 2001.
5. Rosalyn Graham Snyder. Robots assist in search and rescue efforts at wtc. *IEEE Robotics and Automation Magazine*, 8(4):26–28, December 2001.
6. Stefano Carpin, Holger Kenn and Andreas Birk. Autonomous mapping in the real robot rescue league. In *RoboCup 2003: Robot Soccer World Cup VII*, Lecture Notes in Artificial Intelligence (LNAI). Springer, 2004.
7. Brian Yamauchi. A Frontier-Based Approach for Autonomous Exploration. In *Proceedings of the 1997 IEEE International Symposium on Computational Intelligence in Robotics and Automation*, Monterey, CA, July 1997, pp. 146–151.
8. Brian Yamauchi. Frontier-Based Exploration using multiple Robots. In *Proceedings of the Second International Conference on Autonomous Agents (Agents '98).* Minneapolis, MN, May 1998, pp. 46–53.
9. W. Burgard, M. Moors, D. Fox, R. Simmons and S. Thrun. Collaborative Multi-Robot Exploration. In *Proc. of the IEEE International Conference on Robotics & Automation (ICRA)*, 2000.
10. Alexei A. Makarenko, Stefan B. Williams, Frederic Bourgault and Hugh F. Durrant-Whyte. An Experiment in Integrated Exploration. Presented at *IEEE/RSJ Intl. Workshop on Intelligent Robots and Systems*, 2002.
11. Alexander Zelinsky. A Mobile Robot Exploration Algorithm. In *IEEE Transactions on Robotics and Automation*, Vol. 8, No. 6, December 92.
12. Charles E. Perkins. Ad Hoc Networking, Addison-Wesley, 2001
13. Andrew Howard, Maja J. Matarić and Gaurav S. Sukhatme. An Incremental Self-Deployment Algorithm for Mobile Sensor Networks. In *Autonomous Robots*, special issue on Intelligent Embedded System, G. Sukhatme, ed., 113-126, 2002
14. H. H. Nguyen, H. R. Everett, N. Manouk and A. Verma. Autonomous Mobile Communication Relays. In *SPIE Proc. 4715: Unmanned Ground Vehicle Technology IV*, Orlando, FL, April 1-5, 2002.
15. A. Cerpa, J. Elson, D. Estrin, L. Girod, M. Hamilton and J. Zhao. Habitat monitoring: Application driver for wireless communications technology. In *Proceedings of the 2001 ACM SIGCOMM Workshop on Data Communications in Latin America and the Caribbean, April 2001*, 2001.
16. Loren Schwiebert, Sandeep K. S. Gupta and Jennifer Weinmann. Research challenges in wireless networks of biomedical sensors. In *Mobile Computing and Networking*, pages 151-165, 2001.
17. Mani B. Srivastava, Richard R. Muntz and Miodrag Potkonjak. Smart kindergarten: sensor-based wireless networks for smart development problem-solving environments. In *Mobile Computing and Networking*, pages 132-138, 2001.
18. H. Moravec and A. Elfes. High resolution maps from wide angle sonar. In *Proceedings of the 1985 IEEE International Conference on Robotics and Automation*, (St. Louis, MO), pages 116-121, 1985.
19. http://www.nist.gov/dads/HTML/allpairsshrt.html
20. http://www.nist.gov/dads/HTML/floydWarshall.html