A Distributed, Probabilistic Method for Multi-Robot Exploration and Mapping

Paper 766

ABSTRACT

We consider the problem of multi-robot exploration of unknown environments, subject to various constraints. Specifically, we examine possible strategies for communication between agents, and experimentally determine which strategies yield more desirable results. Many existing exploration algorithms have operated under the assumption that agents are capable of maintaining constant communication while exploring, which is often not the case in practice. Additionally, those algorithms which do take some maximum communication range into account solve the problem by requiring the agents to remain within that maximum range at all times. We introduce a distributed, probabilistic method for multi-robot exploration which can be easily adapted to work under any number of constraints. We then use this adaptability to address which communication method yields the best results, by exploring disparate environments under various communication constraints. In doing this, we show that our exploration algorithm allows teams of multiple robots to explore efficiently without explicit coordination in environments where there are limitations on communication.

Categories and Subject Descriptors

I.2.9 [**Artificial Intelligence**]: Robotics

General Terms

Algorithms, Experimentation

Keywords

Mapping, Communication, Multi-robot Systems

1. INTRODUCTION

The exploration of unknown environments is an important problem in the field of autonomous robots. This problem requires one or more robot to fully cover an unexplored area while simultaneously building a map of said area. Robotic exploration has potential application in automated surveillance, search and rescue, and scientific survey of areas that are too hostile for human teams.

Appears in: *Proceedings of the 11th International Conference on Autonomous Agents and Multiagent Systems (AAMAS 2012)*, Conitzer, Winikoff, Padgham, and van der Hoek (eds.), June, 4–8, 2012, Valencia, Spain.

There are many benefits to using teams of robots to explore an area, rather than a single agent. By nature, using multiple robots makes the exploration process more robust to the failure of a single agent, which is very useful in hostile environments. In addition, the use of multiple robots makes it possible to explore an area more quickly. Finally, mapping can be made more accurate by combining any redundant information provided by multiple robots.

Although there are clear benefits to using a team of robots in exploration, the use of multiple agents also adds more complexity to the problem. First, there must be some form of communication between robots. This means that we must determine which information needs to be shared between robots, and how often the robots must communicate. Second, there must be a method for allocating tasks to each robot during the exploration process. This can either be done using a central command center to make all the decisions, a market-based system where robots bid amongst themselves for different tasks, or a fully distributed system in which the robots do not explicitly coordinate their actions.

This paper presents a probabilistic, fully distributed method for exploration with a team of homogeneous robots. The basic method we use for exploration is the Expansive Exploration Tree (EET), which is a method that constructs and stores a roadmap of the explored area. Our method for exploration is easily adaptable to different problem spaces, as the robots explore according to some probability density function (PDF) over the known space. This means that we can easily change the behavior of the exploring robots by changing the manner in which the PDF is computed. This formulation also allows us to easily introduce constraints on the exploring robots, and to explore the effects of those constraints. We use this adaptability to compare the effectiveness of different communication strategies in terms of exploration time, scalability, and usefulness of the final map. We also show that our algorithm allows teams of multiple robots to efficiently explore an environment without explicit coordination, even if there are limitations on the range in which the robots can communicate. This is possible because the structure of the EET-based roadmap implicitly encourages the robots to periodically come into communication range with each other.

The remainder of this paper is organized as follows. Section 2 gives an overview of related work in the area of multi-robot exploration. Section 3 discusses the EET method, and the manner in which it can be adapted to fit different constraints. Section 4 outlines the experiments that were performed to evaluate the effectiveness of various communication strategies, and discusses the results of those experiments. Section 5 presents conclusions and outlines some issues that shall be addressed in future work.

2. RELATED WORK

2.1 Coordination Under Communication Constraints

There have been many recent advances in the area of task allocation for multiple robots under communication constraints. Hung *et al.* [2] propose a distributed, market-based approach to task allocation, where robots bid on specific tasks, and each task gets assigned to whichever robot wins the bid. Communication between agents is handled in a novel manner in this work. Hung imposes a hard communication constraint on any robots that need to cooperate to achieve a specific task, and on all robots who are bidding on tasks, but does not require communication the rest of the time. This allows for more flexibility during task execution, as robots only need to remain in communication distance if they must coordinate to achieve a task. Another approach to task allocation is suggested by Pereira *et al.* [13]. In this work, robots must navigate from place to place on a map without colliding, and without blocking the path of another robot. [13] introduces a distributed navigation algorithm, where robots must move within a safe region, defined as a ring consisting of configurations which will not allow collision between robots, and which will keep the robots within communication distance. With these constraints enforced, the robots are able to use a reactive controller to navigate from place to place. Finally, Hollinger *et al.* introduce the idea of coordination with periodic connectivity [8]. In this work, robots are required to accomplish some task, but are not required to remain in constant communication. Instead, they are able to move about on their own, but must periodically seek out the other robots and regain communication so as to share information. This work shows marked improvement over works which require constant communication, as it allows robots more freedom of action, but still maintains good information sharing between agents.

2.2 Mapping

While the works mentioned above are good examples of task allocation under communication constraints, they do not extend well to the domain of multi-robot exploration. This is because many of the solutions to coordination require either that the environment be known, so that the robots can plan paths that keep them in communication [8], or that some communication infrastructure already be in place, which is usually not the case in unexplored environments. There have, however, been many recent advances in the area of multi-robot mapping. A seminal work in this area was presented by Yamauchi [17], which extends his previous work [16]. Yamauchi explores the idea of frontier-based exploration for multiple robots. In [17], he presents a distributed exploration algorithm, where each robot travels to the nearest boundary between explored and unexplored space (known as a frontier). The robots each maintain a local map, which contains data about the robot's current location and immediate surroundings, and a global map which contains the information gathered by all of the robots. While this algorithm works well for exploration, it assumes that there will be constant communication between agents, which is not always feasible. Additionally, the lack of explicit coordination means that there is often redundancy in the exploration, where two or more robots will select the same frontier to explore.

Simmons *et al.* present a centralized, frontier-based method for exploration with multiple robots in [15]. In this work, there is explicit coordination between the robots, which reduces redundancy and speeds up exploration time. However, [15] still maintains the assumption that all of the robots will remain in constant communication with the central controller. Burgard *et al.* also propose a centralized, frontier-based approach to mapping, where each robot is assigned to a frontier based on both the cost of moving to that frontier, and the expected utility of exploring that region [1]. In this work, although the algorithm is centralized by nature, Burgard also suggests an extension to cases in which communication is limited by allowing robots to break into subgroups, with a centralized control for each subgroup.

Franchi *et al.* move away from centralized controllers for exploration in [5, 6]. These papers present a probabilistic approach to exploration, where each robot explores according to points selected using the Rapidly-Exploring Random Tree (RRT) [11] algorithm. In [5], this results in each robot building a roadmap of the area it has explored, which represents safe paths through the explored space which the robot can use to navigate to new frontiers. [6] extends this work by allowing bridges between the roadmaps constructed by each individual robot, so that each robot is able to explore on the entire map, rather than being confined to a single area. In both of these papers, a hybrid control architecture is used, where robots behave in a distributed manner unless they are close enough to another robot to potentially come into conflict. In this case, one robot will take control of all conflicting robots, and will explicitly coordinate their tasks so as to prevent collisions.

While frontier-based methods are the most commonly used way of exploring an environment, there has also been some recent work that focuses on other approaches instead. Hoog *et al.* introduces the idea of role-based exploration in [3]. In this paper, robots are given the task of either exploring, or acting as a relay between the explorers and a central base. The exploring robots still use a greedy frontier-based algorithm to expand their map, but under the constraint that they must meet with a relay robot at some future time. Since the relay robots are traveling along the already-explored portions of the map, and have specific goal destinations, the exploring robots are able to calculate when the relay will arrive at the next rendezvous point, and minimize the time that the robots must sit idle while waiting to communicate. While this method does not perform as well as purely frontier-based exploration algorithms in terms of exploration time, it does explore the environment in a more balanced manner which could be useful in some applications. In [4], Hoog extends the work done in [3] by formalizing the manner in which rendezvous points are selected. Hoog uses Hilditch's algorithm [7] to compute good rendezvous points for the exploring robots and the relay robots. This allows for more efficient planning on the part of the exploring robots, which leads to considerably better exploration time.

Kovács *et al.* examine the problem of implementing a reasonable communication scheme on real robots as they explore [10]. They propose the idea of using a static Bluetooth communication chain between robots, where each robot communicates with two other robots down a chain, and the chain is ultimately connected to a base position. Their algorithm performed well in uncluttered environments, but maintaining the communication chain became less feasible when exploring more cluttered environments. Finally, Puig *et al.* propose an algorithm for balanced exploration using K-Means clustering to divide the environment into segments for each robot to explore. This algorithm performed well in terms of both exploration time and balanced exploration, but required a centralized planner for the robots and worked under the assumption that there would be constant communication between the exploring robots and the central planner.

While there is a large body of work devoted to both planning under communication constraints and to exploration with multiple robots, there is not a large amount of overlap between the two. While some approaches to multi-robot exploration do take communication constraints into account, they tend to do so by requiring the robots to remain in constant communication with each other or by imposing a periodic constraint, where the robots must rendezvous at pre-determined times in order to share information. Our work proposes a distributed method for multi-robot exploration. This work also allows for the behavior of the exploring robots to be easily changed, based on the constraints of the exploration problem. This allows us to explore the benefits of imposing varying constraints on the manner in which the robots explore when exploring an environment in which constant communication is not possible.

3. OUR APPROACH

3.1 Problem Setting

Our method for exploration takes inspiration from [6], where the robots cooperatively build a roadmap of the environment as they explore. Our algorithm builds a structure called an Expansive Exploration Tree (EET). The EET is a data structure which represents a roadmap of the explored area, and can be seen as an extension of the Expansive Spaces Tree (EST) proposed in [9]. In the case of the EET, each node in the tree contains an obstacle-free configuration q, and a Local Safe Region (LSR(q)) that the robot can perceive from said configuration. Edges between nodes in the tree represent obstacle-free paths in the environment along which the robots can travel. Each node in the tree is connected to at least one other node via an edge, and the tree is constructed incrementally by selecting a vertex based on some PDF, then expanding to a random section of the frontier associated with that node. The EET algorithm is presented under the following assumptions.

1: The workspace in which the robots move is $\mathbb{R}^2$ or a connected subset of it.

2: Each robot is a disk whose configuration q is the position of the disk center. (This allows the configuration space of the robot to be a copy of the workspace with the obstacles grown to allow for robot size.)

3: Each robot is equipped with sensors which provide LSR(q) or the *Local Safe Region* at q, which is a description of the free space surrounding the robot at q.

4: Each piece of frontier *must* be associated with at least one vertex in the EET.

5: Each robot knows its configuration.

6: Each robot can, at any time, broadcast its current location and a list of all nodes in its EET.

7: Each robot is always able to receive communication from any other robot.

While these assumptions are necessary for this approach to work, it is reasonable to expect that they will be easily enforceable in the real world. Most land-based robots will be moving on a connected subset of $\mathbb{R}^2$, or at least on terrain that can be approximated by $\mathbb{R}^2$. In addition, LSR(q) can be computed by any number of sensors that are in common use today, such as sonar, infrared laser, or RGBD camera. Finally, assumption 2 can be relaxed in the real world, by allowing for more complex configuration spaces for the exploring robots.

3.2 Algorithm

Algorithm 1 EET-based exploration

```
perceive LSR
add new node to EET
broadcast insertion of new node
check for updates from other robots
for each new node do
  if node is not root then
    add edge from current node to previous node
  end if
end for
if frontier exists for any node then
  computePDF()
else
  return to start location
  terminate program
end if
choose node to expand (based on results of computePDF())
plan path to node
travel to target node
choose target on local frontier (guaranteed to be in free space)
move to target
while not at target node do
  broadcast EET to other robots
  listen for updates to EET from other robots
end while
```

Algorithm 1 shows the steps of the algorithm used by each robot to expand the EET and explore the environment. At each iteration of this algorithm, each robot computes the Local Safe Region for its current configuration. This configuration and the associated LSR are then added to the EET as a node, and the map is updated with the new information. At this point, if there is no frontier visible in the LSR of any node, the environment can be assumed to be fully explored and the robot will return to its initial configuration. However, if there is remaining frontier, then each node in the EET will be assigned a weight by the *computePDF()* subroutine, which we discuss more fully below. Once the weights of the nodes have been updated, a node will be randomly selected from the EET, using the node weights as a probability density function. The robot will then travel along the EET until it reaches the selected node, broadcasting and receiving information as it travels. Once it arrives at its target, the robot chooses its next observation point along the local frontier. The algorithm then begins again with the addition of a new node to the EET, with an edge connecting the new node to the node most recently visited by the robot.

If our criterion for exploration termination is met, this means that the free space explored by the robots is now completely bounded by impassable terrain, and thus the space has been fully explored. In addition, the EET is guaranteed to possess the qualities needed to constitute a roadmap of the environment. First, it shall be traversable, since it is possible to reach the root of a tree from any node. Additionally, since every free configuration in the environment must be contained in the LSR of some node on the tree, it will be possible to to compute an obstacle-free path from any free configuration q in the environment to at least one node in the tree. Thus, the tree is also accessible from any free point in the environment. This means that, using our method for exploration, we can build a map that is already well-suited for solving motion planning problems in the environment.

3.2.1 Communication

One important aspect of multi-robot exploration is communication. If we want robots to communicate effectively, we must decide when and how the communication should happen, and what information needs to be communicated. In our work, we build our systems on the framework presented in [14], which provides a structured communications layer above the host operating systems of a heterogeneous compute cluster. [14] allows us to use a publish/subscribe communication scheme among our robots, where each robot publishes its information without knowing who will be receiving it, and subscribes to any information that it might find useful. By using a publish/subscribe communication scheme, we obviate the need for a coordination step among the robots, as each robot can publish new data as it is collected, and it becomes the responsibility of the individual robot to listen for important information.

To determine what information must be shared among the robots we first make some assumptions about the starting configuration of our robots. If we assume that the initial configuration of each robot is known to all other robots at the beginning of exploration, then each robot can initialize its EET with its own configuration as the root of the tree, and the configurations of the other robots as children of that root. This requires that the robots share a list of all of the nodes and their parents that are currently in the EET. This will allow each robot to integrate any new nodes it receives with its current EET, even if the respective EETs have changed considerably since the last communication. Beyond this list of nodes and parents, it is often useful for each robot to know the current location of all other robots. To these ends, each robot publishes two messages at each "broadcast" step in Algorithm 1. One of these messages is a vector containing each node and each node's parent in the EET. The other message is simply two floating-point values, giving the robot's current location. Each robot also subscribes to the messages being broadcast by all other robots, and uses the information from these messages to update the EET at each "listen" step in Algorithm 1. Constraints upon communication range shall be discussed in Section 4 of this paper.

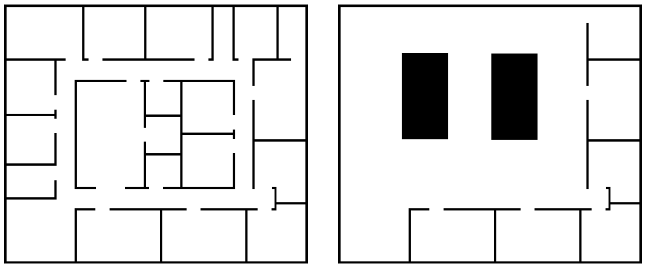

Figure 1: The two environments used for evaluating the EET method. *Left:* the office-like environment. *Right:* the conference room environment

3.2.2 The computePDF() Subroutine

The *computePDF()* subroutine is the part of our algorithm that is responsible for assigning weights to the nodes in the EET. The manner in which this is done drastically affects the behavior of the exploring robot. For instance, if the nodes were weighted with a Gaussian distribution centered at the robot's current location, the robot would be far more likely to pick nearby nodes to expand from. However, if an inverse Gaussian were used, the robot would be more likely to expand from nodes that were further away from its current configuration. In our paper, we explore the differences in performance when two very simple methods of weighting the tree are used. In the first of these, a weight of 1 is assigned to the nearest node that still retains a local frontier, while the rest of the nodes are assigned a weight of 0. This leads each robot to explore the environment in a depth-first manner, and replicates the approach to multi-robot exploration used in [17], where each robot explores the closest frontier until no frontier is left. We call this approach the Closest Frontier First method, or CFF, for the remainder of this paper. Our second approach to weighting the EET is similar to the CFF method, but also takes into account communication constraints. This approach assigns a weight of 1 to the nearest node that still has local frontier, *and* is within some communication range of at least one other robot. If there are no nodes that meet this criterion, the robot reverts to using the CFF method until the communication constraint can be met again. This method, while not imposing a hard communication constraint on the robots, encourages them to explore in similar directions so that there is more likelihood of maintaining communication over the course of the exploration process. We call this approach the Closest Communicable approach, or CC, for the remainder of this paper.

4. EXPERIMENTS

4.1 Simulations

Testing of the EET-based algorithm for exploration with multiple robots was performed using two simulated environments, in order to obtain repeatable quantitative results, and to be able to run large numbers of experiments in a relatively short amount of time. The environments used were a cluttered environment modeled after a floor of an of-

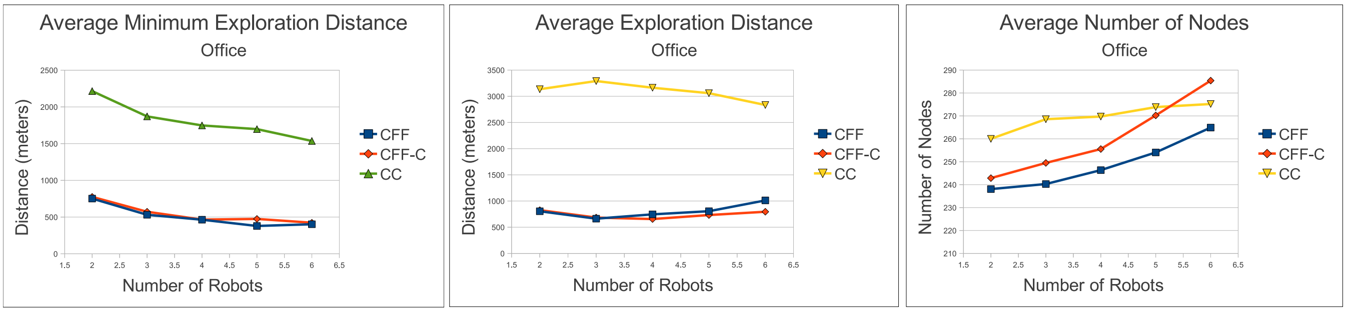

Figure 2: Performance of CFF, CFF-C, and CC in the office environment. *Left:* the average minimum exploration distance per team. *Center:* The overall average exploration distance per robot. *Right:* the average number of nodes generated during the exploration process.

fice building, and a more open environment modeled after a conference room with adjoining offices. These are pictured in Figure 1.

In each environment, we ran simulations for teams of 2, 3, 4, 5, and 6 robots, with varying constraints on communication. Our baseline method for comparison was to use the CFF method, but assume that the robots can always remain in communication. Ideally, this would give us our best results in terms of both exploration speed and redundancy, since the robots remain in constant communication. Against this baseline, we also compare the CFF method with a maximum communication range implemented, so robots can only communicate with one another when they are within a certain distance. We call this approach the CFF-Constrained method, or CFF-C. Finally, we ran simulations in each environment with the CC method. In both the CFF-C method and the CC method, we used a threshold of 15 meters for our communication range. For each team size and each method of exploration, we ran 50 experiments and averaged the results. The results of these experiments are discussed below.

4.2 Results

Figures 2 and 3 show the results of our experiments on the office environment and the conference room environment, respectively. We consider three different variables when evaluating the effectiveness of the CFF, CFF-C, and CC methods. First, we examine the minimum distance traveled by an exploring robot in each time. This value gives us an idea of the overall time taken to explore, since the robots explore in parallel, and all robots stop exploration as soon as the stopping criterion is met. Next, we examine the average distance traveled by each robot on the team, which helps to give us an idea of how well the robots cooperate while exploring. Ideally, the minimum exploration distance should be very similar to the average exploration distance. However, we can see that this is not always the case in practice, since tasks are not always allocated equally among the robots. Finally, we examine the average number of nodes in the EET for each team size. This gives us some idea of how much redundancy occurs during the exploration process. Ideally, we would want the fewest number of nodes possible while still completely covering the environment.

As can be seen in Figures 2 and 3, the average minimum exploration decreases steadily as the number of robots in a team increases. This is to be expected, since increasing the number of robots in a team reduces the amount of area that each robot needs to explore in order to fully cover the environment. We can also see that the CFF-C method vastly outperforms the CC method, coming very close to our baseline in terms of performance. This can be explained by the tree-like structure of the roadmap that the robots are required to move along while exploring. Since the CC method encourages robots to stay in communication range, if one robot exhausts new areas to explore on its current branch of the tree, it must traverse the tree until it reaches a new branch that is within communication range of another robot. This leads to inefficiencies in the exploration, as the robots end up traversing long distances on the tree without adding anything new to the map.

We might expect the average exploration distance to behave in a manner similar to the average minimum exploration distance. However, we can see from Figures 2 and 3 that this is clearly not the case. In fact, as the number of robots on the team increases, the average exploration distance tends to increase for the CFF and CFF-C methods. This occurs because once the number of robots increases beyond a certain point, the robots begin to get in each other's way as they explore. When this happens, one robot will have its exploration cut short by another, and will be required to backtrack along the tree until it reaches a branch that has yet to be explored. In the CC method, however, we see a different behavior. With this method, the average exploration distance tends to decrease as the number of robots increases. this can be explained by the fact that since the robots are encouraged to remain in communication distance, they have a tendency to explore each segment of the environment as a group. This means that as the number of robots increases, segments of the environment will be explored more quickly with less back-tracking necessary on the part of individual robots.

When examining how much redundancy occurs during exploration, we can see from Figures 2 and 3 that while the CFF method outperforms both the CFF-C and the CC methods in terms of average number of nodes, the CFF-C method still tends to outperform the CC method. This can also be attributed to the tendency of the CC method to encourage robots to explore as a group. Since the robots do not explicitly coordinate, they will often attempt to simultaneously explore the same section of frontier, which leads to a large amount of redundancy in the exploration process. We can also see that as the team size increases beyond 5, the CC method actually begins to outperform the CFF-C

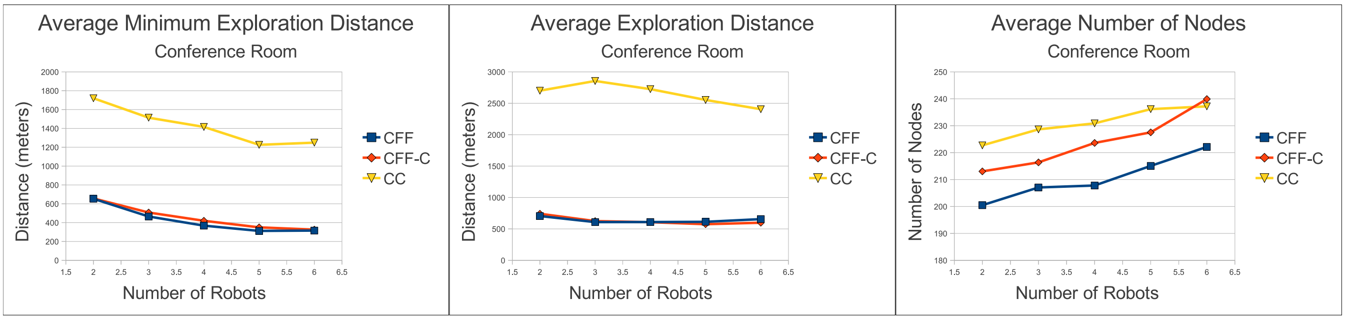

Figure 3: Performance of CFF, CFF-C, and CC in the conference environment. *Left:* the average minimum exploration distance per team. *Center:* The overall average exploration distance per robot. *Right:* the average number of nodes generated during the exploration process.

method. This is because as the group size increases, the completely uncoordinated robots are not able to spread out through the environment as much, leading to greater effort duplication.

When taken as a whole, our results make some interesting implications about the problem of exploring under communication constraints. We see that in the average case, a distributed, uncoordinated method for exploration can significantly outperform methods which encourage constant communication between the robots. Although the worst-case performance of the uncoordinated method could theoretically lead to each robot individually exploring the entire environment without ever communicating, this does not often happen in practice. Instead, the uncoordinated algorithm will generally lead to scenarios such as the one depicted in Figure 4, where each robot explores a separate segment of the environment, and end up within communication range without any explicit coordination. This is aided even further by our EET-based roadmap of the environment, since robots must travel towards the root of the tree once the have fully explored the branch that they are currently on. This tendency to travel towards the root of the tree acutely increases the probability that the robots will naturally come within communication range over the course of their exploration, even without explicit coordination.

5. CONCLUSIONS AND FUTURE WORK

5.1 Conclusions

In this paper, we present a distributed, probabilistic method for exploration with multiple robots. We show that this method allows for easily changing the behavior of the exploring team of robots, and we use this adaptability to evaluate two techniques for exploring with limited communication between robots. The CFF-C method, each robot is free to explore without attempting to maintain communication, and only shares information when it is within range of another robot, while the CC method encourages continuous communication between agents. Our results demonstrate that on average, it is more efficient to let the robots explore without requiring communication, as this allows for more efficient coverage of the terrain. Additionally, the roadmap constructed by our exploration algorithm implicitly encourages robots to periodically come within communication range of each other as they finish exploring sections of the environment and move on to different areas.

5.2 Future Work

Nieuwenhuisen *et al.* introduce the idea of adding useful cycles to a roadmap graph in order to gain higher quality paths [12]. This technique could be applied to our EET data structure to decrease the distance that robots must travel to reach new areas of the tree. This will also allow us to explore the question of which communication technique yields better results when cycles are allowed in the roadmap.

Another extension to our present work would be to create an adaptive *computePDF()* subroutine, which takes known information about the movement of other robots and uses that information to adjust the weights of the nodes in the EET. For instance, if it becomes apparent that a robot is focusing on a specific subsection of the map, other robots would adapt their PDF so as to reduce the probability of coming into conflict and duplicating efforts. This could potentially be done without any additional communication needed, as long as each robot keeps some history of recent actions made by every robot.

6. REFERENCES

[1] W. Burgard, M. Moors, C. Stachniss, and F. Schneider. Coordinated multi-robot exploration. *IEEE Transactions on Robotics*, 21:376–386, 2005.

[2] H. Cao, S. Lacroix, F. Ingrand, and R. Alami. Complex tasks allocation for multi robot teams under communication constraints abstract, 2010.

[3] J. de Hoog, S. Cameron, and A. Visser. Role-based autonomous multi-robot exploration. *Future Computing, Service Computation, Cognitive, Adaptive, Content, Patterns, Computation World*, 0:482–487, 2009.

[4] J. de Hoog, S. Cameron, and A. Visser. Selection of rendezvous points for multi-robot exploration in dynamic environments. In *Workshop on Agents in Realtime and Dynamic Environments, International Conference on Autonomous Agents and Multi-Agent Systems (AAMAS)*, May 2010.

[5] A. Franchi, L. Freda, G. Oriolo, and M. Vendittelli. A randomized strategy for cooperative robot exploration. Technical report, 2006.

[6] A. Franchi, L. Freda, G. Oriolo, and M. Vendittelli. A decentralized strategy for cooperative robot exploration. In *Proceedings of the 1st international conference on Robot communication and coordination*,

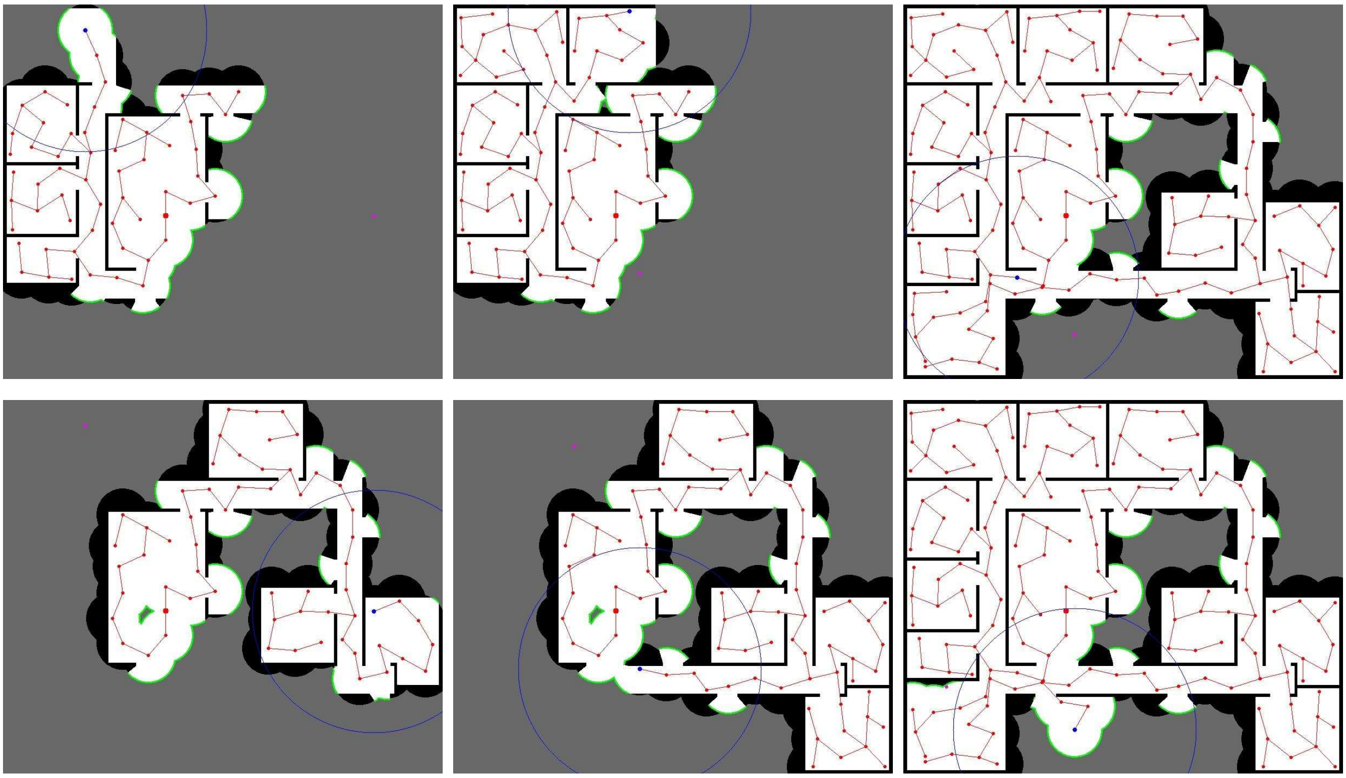

Figure 4: A typical exploration scenario for a two-robot team in the office environment. White represents free space, red represents nodes and edges in the EET, green represents frontier and blue represents the robot and its communication range. *Top:* the local map built by robot 0. *Bottom:* the local map built by robot 1. *Right:* the merging of local maps which occurs when the robots come within communication range.

RoboComm '07, pages 7:1–7:8, Piscataway, NJ, USA, 2007. IEEE Press.

[7] C. Hilditch. Linear skeletons from square cupboards. 4:403–420, 1969.

[8] G. Hollinger and S. Singh. Multi-robot coordination with periodic connectivity. In *Robotics and Automation (ICRA), 2010 IEEE International Conference on*, pages 4457 – 4462, May. 2010.

[9] D. Hsu, J. C. Latombe, and R. Motwani. Path planning in expansive configuration spaces. *Proceedings of International Conference on Robotics and Automation*, 3(April):2719–2726, 1997.

[10] T. Kovács, A. Pásztor, and Z. Istenes. A multi-robot exploration algorithm based on a static bluetooth communication chain. *Robot. Auton. Syst.*, 59:530–542, July 2011.

[11] S. M. LaValle, J. J. Kuffner, and Jr. Rapidly-exploring random trees: Progress and prospects, 2000.

[12] D. Nieuwenhuisen and M. Overmars. Useful cycles in probabilistic roadmap graphs. In *Robotics and Automation, 2004. Proceedings. ICRA'04. 2004 IEEE International Conference on*, volume 1, pages 446–452. IEEE, 2005.

[13] G. A. S. Pereira, G. A. S. Pereira, A. K. Das, V. Kumar, and M. F. M. Campos. Decentralized motion planning for multiple robots subject to sensing and communication constraints. In *in Proceedings of the Second MultiRobot Systems Workshop*, pages 267–278. Kluwer Academic Press, 2003.

[14] M. Quigley, B. Gerkey, K. Conley, J. Faust, T. Foote, J. Leibs, E. Berger, R. Wheeler, and A. Ng. Ros: an open-source robot operating system. In *IEEE international conference on robotics and automation (ICRA)*.

[15] 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*, pages 852–858. AAAI Press, 2000.

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

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