Coordination for Multi-robot Exploration Using Topological Maps*

Tiago Pereira[1,2,3], António Paulo Moreira[1,2], and Manuela Veloso[3]

[1] INESC TEC - INESC Technology and Science (formerly INESC Porto), Porto, Portugal
[2] FEUP - Faculty of Engineering, University of Porto, Portugal
{tiago.raul,amoreira}@fe.up.pt
[3] Carnegie Mellon University, Pittsburgh, PA 15213, USA
tpereira@andrew.cmu.edu, mmv@cs.cmu.edu

Abstract. This paper addresses the problem of decentralized exploration and mapping of unknown environment by a multiple robot team. The exploration methodology relies on individual decision rules and communication of topological maps to achieve efficient and fast mapping, minimizing overlap of explored space. This distributed solution allows scalability of the proposed methods.

Each robot broadcasts a graph representing the topological map, with information of exploration status of each region. Therefore, this kind of information can be transmitted to robots that are not in the communication range, through other robots in a multi-hop network.

This work has been tested in simulation, and the results demonstrate the performance improvements and robustness that arise from our multi-robot approach to exploration.

Keywords: multi-robot, exploration, coordination, topological maps.

1 Introduction

One of the central tasks in robotics is the localization and mapping. Without those, it is impossible to execute most tasks, such as navigation, path generation and motion control.

When doing SLAM, robots have also to explore the environment: plan and decide where to go in order to get the whole map. Particularly in SLAM with occupancy grid maps, the frontier-based exploration and information gain maximization has been used as exploration methodologies.

Being an established solution for single robots scenarios, it is still a research challenge for multi-robot teams when using distributed systems. The solution involves exchanging messages between robots, in order to coordinate the team

* This work is financed by the ERDF – European Regional Development Fund through the COMPETE Programme (operational programme for competitiveness), by National Funds through the FCT – Fundação para a Ciência e a Tecnologia (Portuguese Foundation for Science and Technology) within project FCOMP-01-0124-FEDER-022701, and under Carnegie Mellon-Portugal Program grant SFRH/BD/52158/2013.

A.P. Moreira et al. (eds.), *CONTROLO'2014 - Proc. of the 11th Port. Conf. on Autom. Control*, Lecture Notes in Electrical Engineering 321, DOI: 10.1007/978-3-319-10380-8_49

for the mapping task, merging measurements to produce more precise maps. This has great interest, because it allows more robust and fastest mapping [2].

In most literature, however, this is done in a centralized approach. There is a single agent (single point of failure) responsible for merging the individual maps, and sending orders to the robots telling where they should go. This has the disadvantage of not working in environments with limited networking conditions. For that purpose topological maps can be used, because of the limited amount of data that needs to be transmitted. Furthermore, this kind of representation is a high-level description of the world, allowing to use heterogeneous robot teams, and the inclusion of information from humans.

Therefore, the aim of this work is to use a decentralized method, which also allows intermittent and limited communication bandwidth, and scalability in terms of the number of robots that can be used. In this work the objective is only to create a communication strategy using topological maps in order to solve the distributed coordination in the multi-robot mapping tasks.

This kind of solution allows the appearance of global intelligent and efficient behaviors through local decisions and optimization.

2 Related Work

Various map representations can be used for mapping, such as occupancy grids, geometric features, or topological. This last one is used in [4]. Other approach is the concept of manifolds for the map representation, as presented in [8], so as the map is always locally consistent. In [13], an hybrid SLAM solution is proposed, combining occupancy grids and topological maps in a structure inspired in the manifold concept of [8], producing detailed maps without sacrificing scalability.

For unknown environments mapping using a team of robots, SLAM has been the ordinary solution [3]. In multi-robot systems specific problems arise, such as creating a global map from the independent information of each robot, and also the coordination difficulties. In order to coordinate exploration, a good approach makes use of the expected information gain and the movement costs [16,2].

However, these common approaches use a central agent to coordinate the others, which can be a problem. Regarding distributed computation, it facilitates the scalability of the adopted methods (in [9] a team of 80 robots is used for exploration). Besides, this is also advantageous when communication is limited [3], not allowing the transmission of entire individual maps.

Other approaches use information sharing among teammates to improve the robot beliefs, complementing the limited perceptions (due to, for example, uncertainty and hardware limitations) [5]. Nevertheless, the information merging task can be quite challenging, and a promising approach is to use two separate world models, one with the own robot's state (e.g. an individual topological map), and a shared one storing the overall team's state.

Regarding heterogeneous multi-robot systems, in [10] they use not only terrestrial mobile robots of different sizes and characteristic, but also aerial robots, for the purpose of rescuing missions. In this example, they use the different sensing capabilities of each one to attribute and coordinate the tasks given to the robots. In [12] a novel approach is presented for the formation control of heterogeneous robot teams, so as they navigate without colliding with each other.

The use of topological maps is shown to be effective when interacting with humans [15], and it is also important for heterogeneous teams, because their interface can be done using high-level representations of the environment.

A possibility to model humans' availability and costs of interruption to determine when to query them during navigation is presented in [14]. The human knowledge might be difficult to use for mapping, but it is certainly easier when using more abstract features, such as topological maps. So, the human can help robots localize themselves in these high-level maps.

Finally, there are other distributed exploration strategies using topological information. For example, in [6] swarm robots learn topological information about an unknown environment.This method does not consider the possibility of using humans in the systems, so the topological map is not extracted with that in consideration. Besides, this is especially designed for a large number of agents.

In [11], they address the problem of arranging a meeting between two or more robots in a topological environment, creating the possibility of loop-closure for better mapping, under limited communication. In our work, we consider the possibility of circumventing the limited communication problem making use of a multi-hop network of robots, not restricting their behavior so as to move to a predetermined spot.

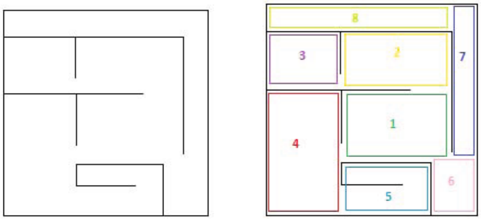
3 Exploration with Topological Maps

In our solution, robots use a wireless ad-hoc network. In exploration, there is no need to build extremely accurate maps because the global map can be estimated in the end when all robots finish exploring and join themselves again. For example, for coordination a robot only needs to know what others have already explored, not what the map is in those regions. So an algorithm that uses minimum communication to efficiently coordinate the robots is needed.

The solution we propose consists in using topological maps, which will not be used for the global map building and map fusion, but only during exploration. Each individual agent must create its own local map using SLAM (allowing its autonomy and independence), extract topological map from grid map, and then use frontiers to decide where to go. Robots must share only high-level information (topology) among the other robots to generate consensus on what to explore. An hybrid approach (grid and topological maps) allows to create detailed sub-maps until they reach its complexity limit, then becoming a node of the topological map, in a process resembling that of the *manifold* concept.

The topological map is then represented as a graph. The different regions are nodes, that can have information for identification (e.g., area and perimeter), and labels of exploration status (*explored* or *not explored*). Moreover, the regions connectivity is represented by the edges of the graph, as shown in Figure 2.

This message passing protocol contains only the topological information, but for the local exploration problem (inside same region) the frontiers of exploration



(a) Original map (b) Regions of map

Fig. 1. Example of a topological map (nodes correspond to the colored regions)

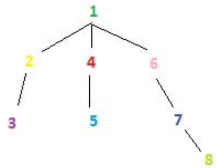


Fig. 2. The graph that would result from world of Figure 1

and local map might also need to be communicated. However, in this work we just rely on obstacle avoidance for local coordination.

4 Coordination Mechanisms

In this section we detail the kind of information that needs to be shared in order to achieve cooperation, and what heuristic rules each robot uses locally when deciding what region needs to be explored.

4.1 Message Structure

So, in this work, each robot stores a graph, with information in each node that is valuable for exploration. Each region, node i of graph G, is characterized as follows on each individual data structure m. This data structure is sent in messages to other robots.

- e^r_{mi}, the local exploration status, boolean variable that is true when region is *explored*, and false for *not explored*;
- e^o_{mi}, the known global exploration status, which is the information received by messages from other robots;
- r^o_{mi}, robot ID that explored region i, when e^o_{mi} is true;
- v_{mi}, true if any known robot has been or is currently in region i;

Furthermore, in each message sent the data structure contains not only the graph information, but also other useful details

- r_m, ID of robot sending message in data structure m;
- t_m, timestamp of message;
- G_m, graph with topological information of exploration status;
- h_m, region which robot is headed to explore;
- p^r_m, the current robot pose;
- p^g_m, the pose of the goal position the robot is trying to reach, in region h_m.

From now on, we consider the graph of own robot as G_A, and the received in messages as graph G_B.

Regarding the graphs received in messages, every robot will use it to update its own graph, which means for each region it will substitute its e^o_{Ai} by the maximum (status is maximum when it is explored) of three possibilities: keep the same, local exploration status, or global exploration status in message.

$$e^o_{Ai} \leftarrow \max\{e^o_{Ai}, e^r_{Bi}, e^o_{Bi}\} \tag{1}$$

Update Graph from Messages. When comparing with e^r_{Bi}, it will always update its own e^o_{Ai} with received information if it is from the same robot ($r_B = r^o_{Ai}$) and timestamp is more recent. This means if the robot is the same, the exploration status can decrease. This is a reasonable behavior because the robot could have said previously some region was explored by mistake, and then find out it was not truly explored.

The same reasoning can be done with e^o_{Bi}, with one exception. In this case it can be information from the robot that is receiving the message ($r^o_{Bi} = r_A$), which is a loop in the information exchange. In this case, it should never keep the information, because the robot itself has the most recent data.

Moreover, it is also easy to find out the importance of e^o_{mi}. With this, it is possible for a robot to receive information from another robot that it could not directly communicate with, through a chain of other agents, like multi-hop routing in mesh networks.

It is interesting to note that timestamps are compared only when belonging to the same robot, to decide when to update information received from others at different times. Thus, there is no need for clock synchronization among robots.

Finally, regarding v_{Ai}, in order to update this variable we only need to apply the OR operation and change its own v_{Ai} to *true* whenever entering a new region.

$$v_{Ai} \leftarrow v_{Ai} \vee v_{Bi} \tag{2}$$

4.2 Coordination Rules

First of all, the frontiers of exploration are sorted according with their information gain, that accounts for the size of frontier, and the cost to get there (distance). After that, the robot checks where each frontier is in the topological map. This could be problematic in the boundary of two regions. Therefore, in order to solve this problem, we introduce the concept of *margin* in the current node of the topological map, which makes frontiers between regions x and y belong to region x when robot is in that one. The implication of this is that robot changes region with confidence, adding hysteresis to the robot localization in the topological domain.

Then, the frontiers are separated in four categories, presented here in decreasing priority order:

- f^{fu}, frontiers in following unexplored regions;
- f^{c}, frontiers in current region;
- f^{fs}, frontier in following semi-explored regions;
- f^{o}, the other frontiers that do not fit in any of the previous categories.

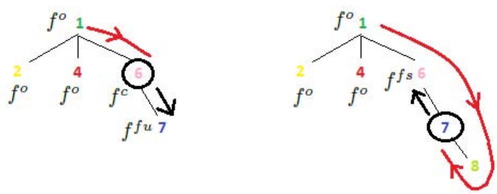


Fig. 3. When in a certain region, robot gives preference to the following regions that it still has not explored. In (a) the robot is in regions 6, coming from region 1, and is headed to the following region, 7. In (b) the robot has already reached the leaf of a branch of the graph (region 8), and has come back to region 7, finished exploring that region, and is headed to explore the following region, this time region 6.

So, the robot will choose first frontiers from the first type, f^{fu}, then the second, f^{c}, etc. Inside each type, they are ordered by information gain, as previously said in the beginning of this section.

The f^{fu} and f^{fs} frontiers belong to regions connected to the current one, but that are not the previous node. For example, as shown in Figure 3, when going from region 1 to 6, the following will be region 7. And when coming back to region 7 from 8, the following is region number 6.

In order to know if regions are unexplored, v_{mi} is used. When this variable is true but region is not completely explored, it is semi-explored, and its frontiers will belong to category f^{fs}.

The idea of using this type of classification is to generate a behavior that makes robots move as far away as possible from the initial region, allowing teams with a large number of robots to explore efficiently. Furthermore, when returning to previously visited regions, the behavior will be to explore those ones completely after moving to another region.

Our aim is to use simple rules in each robot, that result in a complex and organized behavior when considering the whole distributed system.

When there is only the possibility of exploring the same region as other robots, they still do it, with no need for mutual exclusion resolution, because there is no point in stopping when exploration is still going on. Furthermore, it is good to sometimes explore the same region, because that gives common areas in the map to be used later for a more robust map matching between them.

Nevertheless, the individual list of frontiers by type is just a preference list, and a specific goal has still to be chosen from that list.

Finally, a last rule is applied to choose a specific frontier as goal position, which takes in consideration the previously presented parameter h_m. From the list of frontiers ordered by types (individual preferences), it starts checking from the first if there is any robot going there to explore, now giving preference to the ones that are not an exploration goal of other robots.

This strategy might result in choosing one from the bottom of the list, which still is a reasonable choice, even if it is not the preferable region. If someone is already exploring the nearest frontiers, the robot should go to another in order to have efficient exploration. If two or more robots go to the same region, if the distance to the goal of one is less then the other, it will still go there. Therefore, when choosing a frontier to explore in the same region other robot is also headed to explore, each one will use the following rule to decide if it should go there:

$$\|p_A^g - p_A^r\| < \|p_B^g - p_B^r\| \tag{3}$$

One important consideration is that the frontier is not chosen taking into account current regions of other robots, but where they are headed. This means that a robot will probably decide to go to region x even if there is another in there, if that one is going to another region y and the first region is not completely explored.

This exploration strategy could generate temporary incorrect decisions. However, considering the time constant of mechanical systems compared to the communication module that controls them, the incoherent decisions would be negligible, because the time in transitory states of the decision space will still be smaller than the mechanical time constants. The overall decision process is presented in Algorithm 1.

5 Simulation Testbed

For the implementation the Robot Operating System (ROS) and the simulator Stage were used, because it allows the use of other algorithms such as mapping and localization, creating a path to reach a goal position, and follow that path. Another advantage of ROS is the possibility to transparently go from simulation to real scenarios, with little modifications.

The local maps of each robot are occupancy grids, created using an already available SLAM algorithm, Gmapping (a occupancy grid variation of FastSLAM with improved odometry) [7]. In this testbed, frontiers are the central points of clusters of free cells near unexplored cells. The topological map creation and matching are not executed in this application, but simulated using the known ground truth (robot position and known map structure). This, however, could be the result of an “oracle” giving information with no cost, as explained in [1].

```
Order frontiers by information gain and separate by category;
while goal undefined do
    Select next frontier from current category;
    if No more frontiers in current category then
        Change category and select first frontier;
    end
    if Current frontier's region being explored by other robot then
        if Distance to frontier smaller than other then
            Select frontier as goal position;
            Break;
        end
    else
        Select frontier as goal position;
        Break;
    end
end
if goal undefined then
    Select first frontier from first category;
end
```

Algorithm 1. How to choose a goal position from frontiers list

Finally, it is also possible to model network inefficiencies, such as limited communication range and packet losses. The first is done using the relative position between robots in the simulation to limit the range of communication. Regarding packet losses, in our applications we can simply accomplish it using a high period for the exploration package (but still enough to achieve coordination).

It is also important to notice that lack of communication does not result in inconsistencies, but mainly in inefficient exploration by the team.

6 Results

For this project we did not focus on the mapping quality, only the efficiency of exploration. So, we analyze the effect of using N robots on the total time, and how the increasing number of robots affects the exploration time.

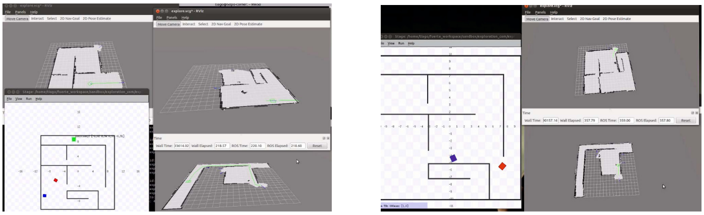
We used the environment from Figure 1(a), with simple structured environments, because the graph extraction had to be done manually, which would be impractical for real building maps. So, in this scenario the current position of the robot and frontiers in the topological map is given manually.

With that setup, four different scenarios were tested:

- Exploration with one robot and unlimited communication;
- Exploration with two robots and unlimited communication;
- Exploration with three robots and unlimited communication;
- Exploration with three robots and limited communication with range of 5 meters.

The first test took 450 seconds, the exploration with 2 robots took 75% of the that time, and with 3 robots around 50%.

When using limited communication, two of the three robots started exploring already explored regions, namely regions 6 and 7 of Figure 1(b). They decided to re-explore them because those regions were the last to be explored, so both robots do it until exploration is finished.



(a) (b)

Fig. 4. In (a), efficient exploration with 3 robots. Green robot partially explores region 1, and fully explores regions 6, 7 and 8. The red one explores regions 1, 2, 3 and partially region 4. The blue robot explored partially 1, and completely regions 4 and 5. In (b), the final state when exploring with 2 robots. The red robot explored regions 1, 2, 3, 4 and 5. The blue one explored partially region 1, and completely regions 6, 7 and 8.

Finally, it is possible to see the result of exploration in Figure 4, with little overlap of mapping. This demonstrates the efficiency of this coordination strategy, that can later result in a full global map although each robot explored only specific regions of the environment.

7 Conclusions and Future work

The abstraction of world structure through topological maps helps solving the coordination problem, allowing to use it in the future with heterogeneous teams.

With this paper we contributed with a communication framework for coordination, and a heuristic to decide what regions to explore, creating rules that result in a global intelligent behavior of the robots. So, using frontier-based exploration coupled with topological maps and message passing, we achieved a coordination strategy that can be used by robots in exploration and mapping.

As future work, it is important to create a local exploration strategy that can coordinate robots when they are in the same region, and to implement autonomous topological map extraction and graph matching procedures in order to be able to use it with real robots for autonomous exploration.

References

1. Armstrong-Crews, N., Veloso, M.: Oracular partially observable markov decision processes: A very special case. In: 2007 IEEE International Conference on Robotics and Automation, pp. 2477–2482. IEEE (2007)

2. Burgard, W., Moors, M., Stachniss, C., Schneider, F.: Coordinated multi-robot exploration. IEEE Transactions on Robotics 21(3), 376–386 (2005)
3. Carlone, L., Ng, M., Du, J., Bona, B., Indri, M.: Rao-blackwellized particle filters multi robot slam with unknown initial correspondences and limited communication. In: 2010 IEEE International Conference on Robotics and Automation (ICRA), pp. 243–249. IEEE (2010)
4. Chang, H., Lee, C., Hu, Y., Lu, Y.: Multi-robot slam with topological/metric maps. In: IEEE/RSJ International Conference on Intelligent Robots and Systems, IROS 2007, pp. 1467–1472. IEEE (2007)
5. Coltin, B., Liemhetcharat, S., Meriçli, C., Tay, J., Veloso, M.: Multi-humanoid world modeling in standard platform robot soccer. In: 2010 10th IEEE-RAS International Conference on Humanoid Robots (Humanoids), pp. 424–429. IEEE (2010)
6. Dirafzoon, A., Lobaton, E.: Topological mapping of unknown environments using an unlocalized robotic swarm. In: 2013 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), pp. 5545–5551 (2013)
7. Grisetti, G., Stachniss, C., Burgard, W.: Improved techniques for grid mapping with rao-blackwellized particle filters. IEEE Transactions on Robotics 23(1), 34–46 (2007)
8. Howard, A.: Multi-robot mapping using manifold representations. In: Proceedings of the 2004 IEEE International Conference on Robotics and Automation, ICRA 2004, vol. 4, pp. 4198–4203. IEEE (2004)
9. Howard, A., Parker, L., Sukhatme, G.: Experiments with a large heterogeneous mobile robot team: Exploration, mapping, deployment and detection. The International Journal of Robotics Research 25(5-6), 431–447 (2006)
10. Luo, C., Espinosa, A., Pranantha, D., De Gloria, A.: Multi-robot search and rescue team. In: 2011 IEEE International Symposium on Safety, Security, and Rescue Robotics (SSRR), pp. 296–301. IEEE (2011)
11. Meghjani, M., Dudek, G.: Multi-robot exploration and rendezvous on graphs. In: 2012 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), pp. 5270–5276 (2012)
12. Nascimento, T.P., Conceição, A.G.S., Alves, H.P., Fontes, F.A., Moreira, A.P.: A generic framework for multi-robot formation control. In: Röfer, T., Mayer, N.M., Savage, J., Saranlı, U. (eds.) RoboCup 2011. LNCS, vol. 7416, pp. 294–305. Springer, Heidelberg (2012)
13. Pfingsthorn, M., Slamet, B., Visser, A.: A scalable hybrid multi-robot SLAM method for highly detailed maps. In: Visser, U., Ribeiro, F., Ohashi, T., Dellaert, F. (eds.) RoboCup 2007. LNCS (LNAI), vol. 5001, pp. 457–464. Springer, Heidelberg (2008)
14. Rosenthal, S., Veloso, M.: Modeling humans as observation providers using pomdps. In: 2011 IEEE RO-MAN, pp. 53–58. IEEE (2011)
15. Santos, F.N., Moreira, A.P., Costa, P.C.: Towards extraction of topological maps from 2D and 3D occupancy grids. In: Correia, L., Reis, L.P., Cascalho, J. (eds.) EPIA 2013. LNCS, vol. 8154, pp. 307–318. Springer, Heidelberg (2013)
16. Simmons, R., Apfelbaum, D., Burgard, W., Fox, D., Moors, M., Thrun, S., Younes, H.: Coordination for multi-robot exploration and mapping. In: Proceedings of the National conference on Artificial Intelligence, pp. 852–858. AAAI Press, MIT Press, Menlo Park, Cambridge (1999, 2000)